Study and Implementation of Split Multi-Channel Rebalancing Strategy for Off-Chain Payments

Wei-Jun Gao, Jia-Ming Guo, and Cheng-ying Jiao
(Corresponding author: Jia-Ming Guo)

School of Computer and communication, Lanzhou University of Technology
No. 36, Peng-Jia-Ping Road, Lanzhou 730050, China
Email:15101321931@163.com

(Received May 18, 2024; Revised and Accepted Oct. 22, 2024; 1st & 2nd Online Mar. 14 & June 22, 2025)

Abstract

Payment channel networks are a promising solution to improve the scalability of blockchains. However, as the frequency of transactions increases, the channel funds in one direction may be exhausted, thereby preventing further transactions. Therefore, this paper proposes a split multi-channel rebalancing strategy based on off-chain payments, utilizing channels with higher traffic load weights and lower loss probabilities to increase the balance of depleted channels. it uses algorithms to evaluate channel payment demands and balance capacities, providing a feasible and efficient rebalancing platform for users in need of balance. Simulation results demonstrate that compared to existing rebalancing strategies, this paper excels in improving channel imbalance and transaction efficiency.

Keywords: Blockchain; Channel Imbalance; Off-chain Payment Network; Rebalance Planning

1 Introduction

Blockchain has attracted widespread attention from scholars in various disciplines such as law [18], finance [36,37], and IoT security [6, 7] due to its features including distributed structure [1], immutability [20], security [5, 19], and privacy [38]. However, due to the consensus mechanism requiring a consistent transaction view among all nodes, blockchain faces challenges such as low throughput and poor scalability. Specifically, popular cryptocurrencies like Bitcoin and Ethereum achieve approximately 7 and 30 transactions per second respectively, while traditional centralized transaction systems like Visa confirm thousands of transactions per second [29]. Payment channel networks are one of the mainstream solutions [11, 21, 23] to enhance blockchain scalability, it allow participants to conduct multiple off-chain peer-to-peer transactions without recording every transaction on the blockchain, thus avoiding the inefficiencies and network congestion associated with on-chain transactions. This technology has been deployed in many blockchains, including the Lightning Network built on the Bitcoin system [27] and the Raiden Network built on the Ethereum system [15]. Although payment channels benefit blockchain scalability, there are some unavoidable limitations compared to traditional networks [4]. In traditional networks [16,17], link bandwidth is generally fixed. However, in payment channels, due to the payment habits of nodes, funds gradually accumulate at the high-traffic end, resulting in the inability to initiate transaction requests in that direction, which poses a threat to the usability of the payment channel network [10].

Currently, there are three types of solutions for funding depleted channels. The first simple method is to close and reopen the channel, but this results in two expensive and time-consuming on-chain transactions. The second method is to facilitate balanced fund flow through routing selection, using channels with higher balances to transmit transactions, alleviating further deterioration due to fund depletion, though there is still room for improvement in enhancing the balance of unbalanced channels. The third method is currently the most effective solution for mitigating fund depletion. By proactively rebalancing channels through unbalanced nodes, it extends the lifespan of depleted channels without involving the blockchain, aiming to repay weak channels by redistributing deposits from adjacent channels. However, this approach is limited by network topology and adjacent channels, and thus lacks general applicability.

Addressing the current shortcomings of channel rebalancing strategies, We propose Split, a split multichannel rebalancing scheme based on off-chain payment channel networks, which monitors fund flows and transaction activities to automatically trigger fund splitting and transfer operations. Split protocol breaks through the limitations of existing rebalancing strategies constrained by network topology, overcoming constraints on fund balancing im-

posed by adjacent channels. The contributions of this paper are summarized as follows:

1) We introduce the concept of payment task queues to predict the possible future payment demands of the channel. We filter out reliable balanced channels with low demand and substantial balances. Additionally, We also consider the imbalance degree of the channel in the traffic load weights.

2) To ensure prolonged use of channels after rebalancing operations, we introduce the concept of minimum channel loss probability to split balanced amounts, thereby reducing the impact on subsequent transactions of adjacent channels to imbalanced nodes.

3) We propose Split, a new approach that splits a rebalancing amount among multiple channels for individual rebalancing. Through experiments in a simulation environment, compared with existing Revive and Shaduf rebalancing strategies, it was verified that the Split rebalancing scheme improves success rates and balances channel funds more effectively.

The organization of this paper is as follows: Section 2 introduces the relevant foundational background. Section 3 reviews related research both domestically and internationally. Section 4 constructs the framework of the proposed solution and describes the design and implementation of each module. Section 5 discusses the implementation and evaluation of experiments. Section 6 identifies the shortcomings of this paper and discusses future work.

2 Background

2.1 Bidirectional Payment Channel

The bidirectional payment channel, based on the micropayment channel [39], breaks the limitation that funds flowing only in one direction, allowing both parties to conduct multiple small transactions off-chain. In the bidirectional payment channel channel, both users can send and receive payments.

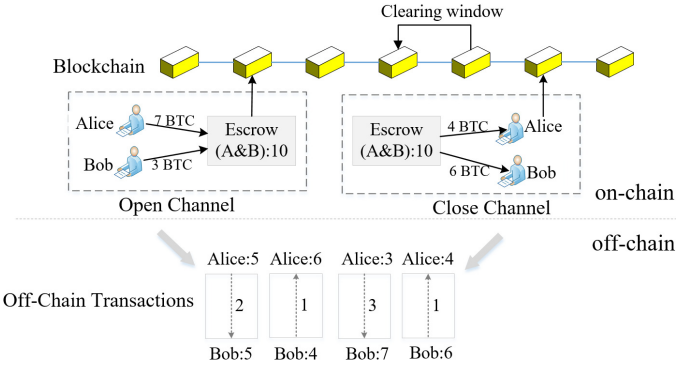


Figure 1: Bidirectional payment channel life cycle

As Figure 1 describes the lifecycle of the bidirectional payment channel, participants Alice and Bob inject 7 BTC and 3 BTC respectively into the jointly controlled multi-signature address account to construct the payment channel when the channel is opened, and these funds can be flexibly used for off-chain payments and receipts by the two users to realize the function of instant settlement. Ultimately, by the time the channel is closed or in the dispute phase, Alice and Bob have completed multiple off-chain transactions, the balance in the channel becomes 4 BTC for Alice and 6 BTC for Bob, and both parties get their coins back by posting the latest status to the chain to liquidate their assets. To ensure the security and correctness of the payment, cryptographic methods such as Hash Time Locks [27] or Anonymous Multi-Hop Lock [34] are usually enforced, thus preventing any fraudulent behavior of the parties during the payment process. When one party attempts to gain illicit profit by publishing invalid historical transactions, the payment channel will deprive the dishonest party's entire balance and transfer these funds to the honest party's account. This approach aims to eliminate the risk of malicious behavior by either party, while protecting the interests of legitimate participants in the transaction.

2.2 Payment Channel Network

As the number of participants and payment channels increases, a transaction network consisting of countless nodes and edges gradually forms, known as the Payment Channel Network (PCN). In a PCN, even if there is no directly connected channel between two nodes, payments can still be completed through paths formed by multiple end-to-end payment channels. The Lightning Network [27], as the most well-known payment channel network, employs core technologies such as Revocable Sequence Maturity Contracts (RSMC) and Hash Time Lock Contracts (HTLC). RSMC ensures the security and revocability of payment transactions based on micropayment channels, providing users with a safe and efficient bidirectional payment mechanism. HTLC combines hash functions and time lock features, enabling cross-channel fund transactions between participants who do not have directly connected channels.

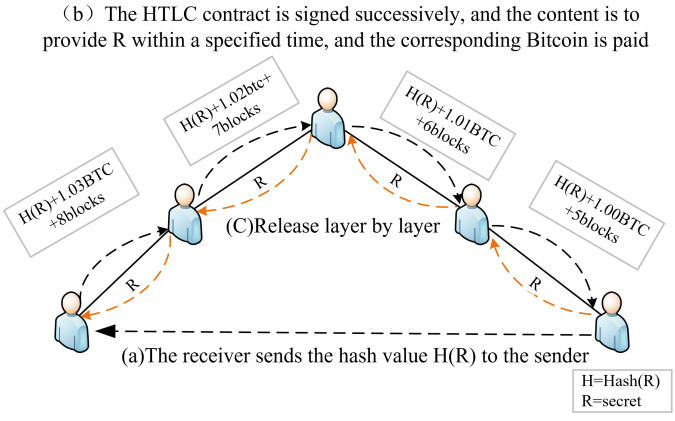


Figure 2: Core idea of HTLC

Figure 2 illustrates the core concept of HTLC. As a conditional contract enforced by the blockchain, it does not require trust from any participant in the network. This contract locks a portion of tokens, which can only be released to the recipient when certain conditions are met, or returned to the owner upon contract expiration.

2.3 Channel Rebalancing

When payment amounts accumulate towards the end with higher traffic, channels may deplete, making routing payments in that direction impossible.Users need to close the depleted payment channel (PC) and open a new one,requiring two costly on-chain transactions. Rebalancing mechanisms in Payment Channel Networks (PCN) search for imbalanced nodes with depleted channels where replenishment is needed, restoring exhausted channels by transferring tokens from other channels of the node through off-chain transfers. Currently, rebalancing mechanisms can be categorized into periodic and non-periodic types. Revive [14] is the first periodic protocol that helps multiple users rebalance channels simultaneously. We illustrate how Revive works using the directed cycle B→C→D→B in Figure 3. Nodes first submit requests to the selected leader, detailing the channels they wish to rebalance and their token requirements. Subsequently, the leader generates a periodic cycle-based rebalancing solution where the total tokens sent by each node equal those received. In essence, Revive executes fund flows along directed cycles to achieve self-payment. We illustrate how Shaduf [9] works using C↔E↔F in Figure 3. Unlike Revive, Shaduf can recover channels from depletion in general situations without relying on loop topologies. It allows users to perform unlimited off-chain token transfers between channels after a single on-chain binding, secured through multi-party secure computation protocols to ensure transaction privacy. Figure 3(C) shows the network after rebalancing,demonstrating significant improvement in highly skewed fund allocation issues.

3 Related Work

Our current research on PCN performance can be categorized into three areas: routing algorithms, rebalancing strategies, and security issues. Flare [28] is the first hybrid routing algorithm proposed for the Lightning Network, which uses beacon nodes to enhance network visibility for other nodes. The sender uses a combination of it and the receiver's routing table to find possible routes in the network from the sender to the receiver. LEAF [25] explores the issue of path overlap in PCNs and proposes a decentralized payment routing scheme to improve network throughput and reduce redundant traffic overhead in PCNs. Sharma *et al.* [31] proposed Swift, a decentralized routing algorithm focused on fee optimization, which minimizes path length and total transaction fees. The aforementioned routing protocols focus on transaction throughput and cost overhead, overlooking the issue of balance depletion. Recently, some routing work has considered avoiding fund depletion while ensuring network vitality. For example, Cai *et al.* [2] proposed a novel routing protocol for concurrent PCNs that reduces transaction failures caused by channel balance fluctuations during routing by reserving sufficient balance during path probing. Luo *et al.* [24] proposed a priority-aware transaction allocation mechanism to balance transaction rates and forwarding costs, preventing periodic fund depletion in channels facing skewed payment flows. Wang *et al.* [35] propose an online balance-aware fee-setting algorithm that sets transaction fees based on the current balance and congestion level of each channel to incentivize payers to use more balanced and less congested paths.

Rebalancing approaches address the problem from a different angle compared to routing schemes. Khalil and Gervais introduced the first rebalancing scheme — Revive [14]. This scheme relies on an untrusted third party, which creates a transaction block to rebalance the deposits in the channels. It allows nodes to rebalance their connected channels through a loop formed by channels of other nodes, deploying smart contracts on these loops to lock funds and redistribute them according to predetermined rules. Building on Revive, Camilo *et al.* [3] propose a node positioning strategy that encourages creating cycles in PCNs to counteract centralization trends and achieve cost-effective off-chain rebalancing. Ge *et al.* [9] introduce a novel non-cyclic off-chain rebalancing protocol that allows direct fund transfers between channels, eliminating the Revive protocol's dependency on cyclic topologies. Sangram *et al.* [30] proposes a heuristic-based distributed rebalancing solution that enhances the fund transfer capabilities of channels while preserving the privacy and anonymity of the parties involved in the rebalancing process.

Moreover, much research indicates issues such as privacy, security, and robustness in off-chain payments [8,12]. Malavolta *et al.* [26] address wormhole attacks in cross-channel payments by proposing the cryptographic principle of anonymous multi-hop locks. Khalil introduce FAKEY [13], a type of attack based on fake hash keys that can block an entire set of transaction channels for a certain period, depending on the hashed timelock contract attacking payment paths. Wang *et al.* [33] present a new type of attack that allows malicious service nodes to abandon valid paths for profit and design a feedback mechanism to mitigate this attack. They introduce a new identity information transmission scheme called encrypted identity chain to conceal the identities of senders/receivers of payment paths' intermediate nodes.Zhang *et al.* [40] propose the AMHL+ scheme using the general structure of AMHL, eliminating the assumption of anonymous channels between payers and relay nodes. To reduce communication overhead, they propose the EAMHL+ scheme based on bilinear pairings, although it significantly increases computational costs.

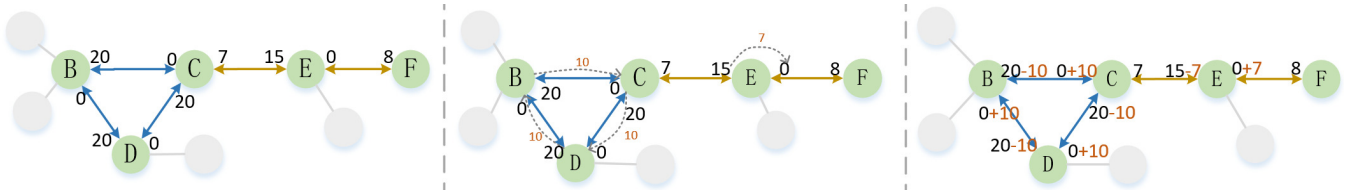


Figure 3: (a)Before Rebalancing (b)Rebalancing (c)After Rebalancing

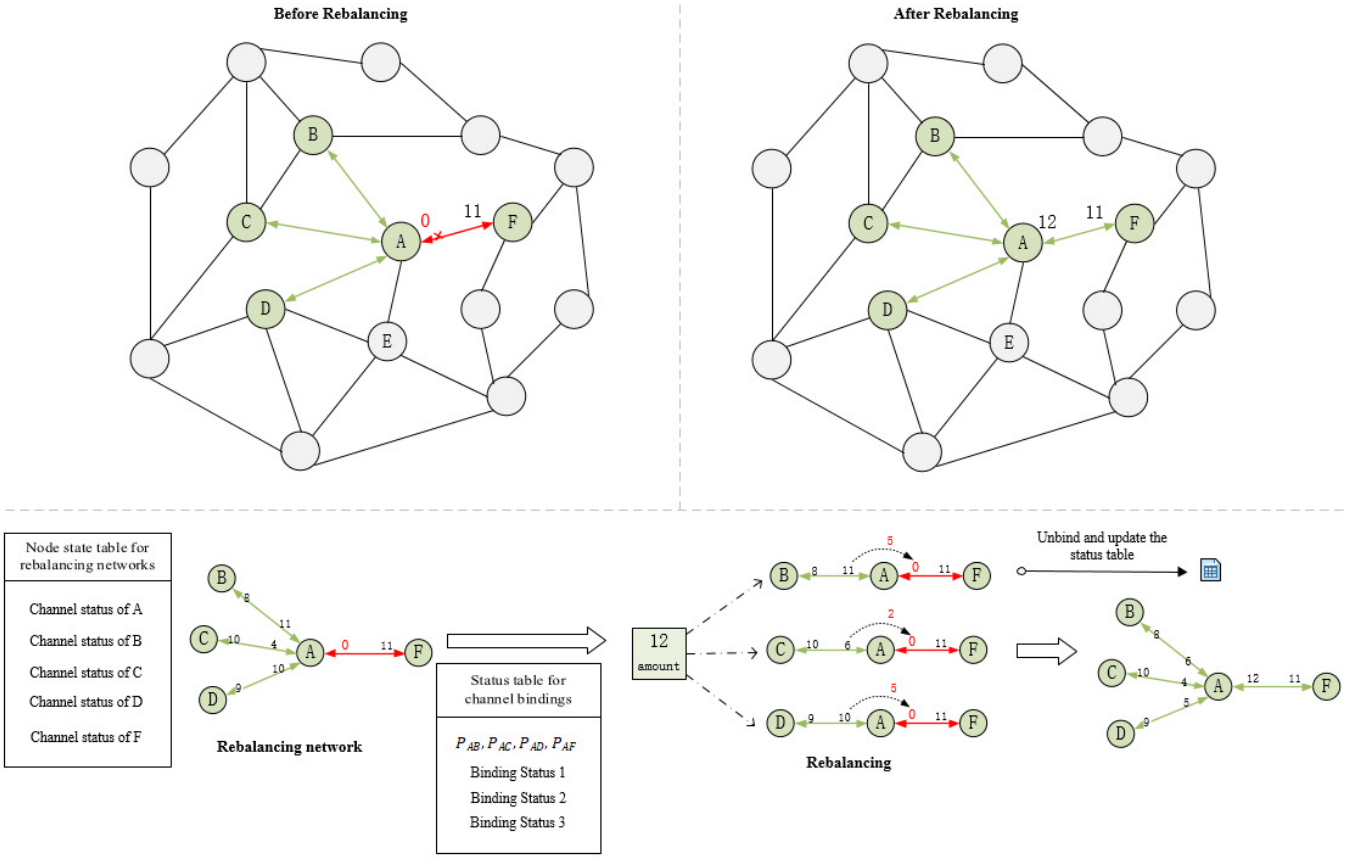


Figure 4: A block diagram of Split

4 Scheme Design

4.1 Scheme Idea Overview

The split multi-channel rebalancing scheme (Split) constructed in this paper aims to distribute the pressure of balancing amounts across multiple channels to mitigate potential biases in the payment network. Inspired by the balance planning service PNP [22], we monitor real-time transactions of each node in the network. The imbalanced node calculates the Traffic Load Weight (TLW) of connected channels based on the historical asset transfer within the channels, thereby determining the transaction activity gradient to identify channels for participation in balancing. Based on the balance capacities of each payment channel, the rebalancing amount is divided into multiple parts for fund allocation among channels, aiming to minimize channel losses.

The Architecture of the split rebalancing process is shown in the Figure 4. A needs to send 12 tokens to F, but there are no funds in the direction of the A→F channel to initiate any transaction request. At this time, the maximum balance on any adjacent channel is 11 tokens, which is insufficient to meet A's transaction requirement with a single balancing fund transfer. Therefore, in the Split scheme,it is necessary to distribute the pressure of the balancing amount across multiple channels, each contributing a portion of tokens to the A→F channel. Compared to other rebalancing strategies, this approach not only overcomes the limitations of network topology but also alleviates to some extent the impact of rebalancing amounts being constrained by adjacent channels. To prevent double-spending attacks when multiple balancing networks are operational simultaneously, the scheme mandates that channels participating in the same balanc-

ing network must be bound together. This binding can be released once balancing is completed to proceed with the next balancing operation. The key issue addressed in this paper is to find the optimal balancing partitioning scheme to minimize the impact of rebalancing on subsequent transactions in other channels.

4.2 Model Construction

4.2.1 Network Model

This paper constructs a payment channel network as a directed graph $G = (V, E)$, where $V = (v_1, v_2, v_3, ..., v_n)$ is the set of nodes in the network and E is the set of currently open off-chain payment channels.Each bidirectional channel is considered as two directed edges, with each directed edge $e = (v_i, v_j)|v_i, v_j \in E$ containing $b_{i,j}$, where $b_{i,j}$ defines the maximum balance that node v_i can send to v_j. Therefore, the total capacity $C_{i,j}$ of each bidirectional channel is subject to the following constraints:

$$C_{i,j} = b_{i,j} + b_{j,i} \tag{1}$$

As the dynamic balance of funds within the channel changes, the total deposits $C_{i,j}$ will vary over time, along with the balance $b_{i,j} \geq 0$ of each directed channel in the network. For each channel $e = (v_i, v_j)|v_i, v_j \in E$, we define the balance deviation and imbalance degree as:

$$\Delta_{i,j} = b_{i,j} - b_{j,i} \tag{2}$$

$$D_{ij}^{im} = \frac{\Delta_{i,j}}{C_{i,j}} \tag{3}$$

4.2.2 Channel Screening

Due to the limited funds in the payment channels, the concept of a payment task queue is introduced to predict potential future payment demands within the channels. The channels connecting the imbalanced nodes v_i are constructed as a directed graph $G' = (V', E')$, with time discretized into fixed-length slots t. During each time slot t, node v_i receives new transaction requests from outside the network, requesting the transfer of a total token amount $N_i(t)$ to any node within the network.$\sigma_{ij}(t)$ represents the number of tokens sent to the target node through the intermediary channel $e = (v_i, v_j)$ in a transaction task. Each node v_i needs to maintain the payment task queue $Q_{ij}(2t)$ for each connected channel. We define the queue length $Q_{ij}(t)$ to reflect the usage of each channel over a fixed period. The longer the queue length, the more funds are needed in the channel; the shorter the queue length, the fewer funds are needed in the channel. The queue length relationship is as follows:

$$Q_i(2t) = [Q_i(t) + N_i(t) + \sum_{j \in M_i}^{i \to j} \sigma_{ij}(t) - \sum_{j \in M_i}^{j \to i} \sigma_{ji}(t)] \tag{4}$$

In Equation (4), where $Q_i(2t)>0$, because each transferred token represents an actual non-negative amount. M_i represents the set of neighboring nodes of node v_i. The traffic load weight of node v_i in each connected channel $e = (v_i, v_j)|j \in M_i$ is:

$$W_{ij}(t) = \mu D_{ij}^{im}(t) - \tau Q_i(t) \tag{5}$$

In Equation (5), μ and τ are adjustable positive parameters, and D_{ij}^{im} represents the imbalance degree of the channel connecting node v_i and its neighboring nodes. The longer the task queue, the higher the activity level of the channel where the node is located, which implies a greater amount of funds needed for this channel in the future. During the rebalancing process, in order to minimize the impact on subsequent transactions of the balanced channel, balance funds should be taken from this channel as little as possible. Conversely, channels with low activity and excess retained funds should be prioritized for fund extraction.

$$\gamma = maxW_{ij}(t) \tag{6}$$

Within a fixed time period t, the imbalanced node v_i calculates the traffic load weight $W_{ij}(t)$ based on the payment task queue length and imbalance degree of the channels connected to each neighboring node, and selects the optimal balancing channel γ.

4.2.3 Rebalancing Amount Segmentation

After obtaining the optimal set of balancing channels, an indicator that can measure the balance capacity of the payment channels is needed to divide the balancing amounts. Considering that this paper uses the Lightning Network as an example to explain the algorithm, we assume that the balance distribution of the payment channel network model follows a discrete uniform distribution. Based on the discrete uniform distribution of channel balances, the success probability of a rebalancing amount a flowing out from a channel $e = (v_m, v_n)$ with capacity $C_{m,n}$ is defined as:

$$S(C_{m,n}, a) = P(X \geq a) = \frac{C_{m,n} - a}{C_{m,n} + 1} \tag{7}$$

After performing the split rebalancing, the balance amount will be transferred from multiple neighboring balancing channels to the depleted channel, thereby reducing the available funds in these balancing channels. This paper defines the channel loss probability $\mu_{m,n}$ as the difference in the success rate of a token payment before and after executing the rebalancing amount bal.

$$\mu_{m,n}(bal_{i,j}) = S(C_{m,n}, bal) - S(C_{m,n}, bal - 1) \tag{8}$$

The division of the balancing amount aims to find a split scheme that minimizes the total channel losses by measuring the balance capacities of each neighboring balancing channel, thereby making the channels more durable and robust.When rebalancing is complete, the sum of losses of

all balancing channels should be minimized, and in this paper we define the splitting objective as solving the following problem:

$$\begin{aligned} \min \quad & \sum_{j \in N_i} \mu_{i,j}(bal_{i,j}) \\ \text{s.t.} \quad & \text{totalBal}_i = \sum_{j \in N_i} bal_{i,j} \\ & bal_{i,j} < C_{m,n} \end{aligned} \tag{9}$$

Here, N_i represents the neighboring nodes selected by node v_i to participate in the rebalancing. $totalBal_i$ is the total balancing amount, and $bal_{i,j}$ represents the sub-amount of rebalancing that the imbalanced node v_i needs to transfer out from the selected channel set.

4.3 Model Implementation

Split's channel selection algorithm integrates the length of transaction task queues and imbalance degree in the weight function, and it allows adjusting the proportion of these two indicators in the weight function.For the depleted channel, Algorithm 1 is prioritized to find the optimal channel connected to the imbalanced node, then the weight corresponding to this channel is multiplied by a reduction factor. Algorithm 1 is executed again until a set of M channels is filtered out.

Algorithm 1 Channel Selection Algorithm

Input: Node_i, Node_j, number of channels m
Output: Channel_assemble

bestchannels ← empty list
maxWeight ← 0
bestchannels.length ← 0
while bestchannels.length < m **do**
  for nextNode **in** Node$_i$.neighbors **do**
    if nextNode ≠ Node$_j$ **then**
      nextNode.Weight ← Calculate the channel weight according to equation (5)
      if maxWeight < nextNode.Weight **then**
        maxWeight ← nextNode.Weight
        S ← nextNode
      end if
    end if
  end for
  best_channel ← (Node$_i$, S)
  if best_channel not in Channel_assemble **then**
    Add best_channel to Channel_assemble
  end if
  Reduce the maxWeight weight
end while
return Channel_assemble

The parameter M is a crucial parameter in this algorithm, and its value needs careful consideration. If M is too small, the amount of balancing received by the fund-depleted channel is very limited, and if M is too large, it will increase the algorithm's time overhead. Assuming that the number of channels involved in balancing is M, requiring P rounds of selection. For a network with nodes E and edges V, the time complexity of Shaduf's algorithm is $O(E\log(V))$. Therefore, the time complexity of the split rebalancing strategy in this paper is denoted by $O(M \cdot E\log(V) + P)$. The actual algorithm runtime will be analyzed in Section 5.

Next, it is necessary to allocate appropriate rebalancing amounts to the channels selected by Algorithm 1. To improve computational efficiency, in Algorithm 2, we divide the rebalance amount evenly into x units of amount s. We iterate through each channel in the set and calculate the loss probability for each channel under the condition of unit outflow, and select the channel with the smallest loss after transferring the unit. Repeat the above process until all units of division are allocated, ensuring that the allocation results of Algorithm 2 achieve the minimum channel loss sum.

Algorithm 2 Balance Amount Split

Input: Number of split units x, amount of units s, number of channels m, Channel_assemble
Output: *amounts*

$loss$ ← 0
amounts[m] ← 0
i ← 0
while $i < x$ **do**
  for channel **in** Channel_assemble **do**
    if the channel balance is sufficient to cover the unit amount **then**
      $loss$ ←Calculate the channel loss according to formula (8)
    end if
  end for
  amounts[channel] ← amounts[channel] + s
  i ← i + 1
end while
return amounts

Algorithm 3 Balance Execution

Input: Channel_assemble,*amounts*

Channels within Channel_assemble bind to each other
for channel **in** Channel_assemble **do**
  TransferTokens(channel, accounts)
end for
Unbind a channel

Algorithm 3 is based on the confirmed set of channels participating in rebalancing and the division results, executing each sub-balancing Re_s in a loop to transfer partial funds from each rebalancing channel into the imbalanced channel. It is crucial to ensure balance security during this process, meaning that the funds of each participating node remain consistent before and after rebalancing.

Therefore, channels entering the rebalancing channel set must be mutually bound to prevent certain channels from participating in multiple balances simultaneously, thereby avoiding double-spending attacks. After ensuring the successful transfer of rebalancing funds, the binding of channels within this set can be released.

5 Implementation and Evaluation

5.1 Implementation

We used the programming language Python to create the rebalancing network and implement the division of balanced amounts (Algorithm 1, Algorithm 2). We implemented Algorithm 3 using Solidity on Remix, an official Ethereum open-source online integrated development environment. We used the Python Simpy simulator to evaluate the performance of network transactions under different rebalancing scenarios. The transactions in the simulation were implemented as message transfers between nodes with simulated real-time delays, representing a real network.

We deployed the rebalance execution contract on the Ethereum test network, focusing on the on-chain costs incurred by running the Split protocol, which are the fees paid to miners during contract interaction. On Ethereum, this is computed in gas. We set the gas price at 20 Wei (as of June 2022). Due to Ethereum's price fluctuations, we use the exchange rate of 600 USD per Bitcoin as of June 2022. The cost of the contract depends on the amount of data and the complexity of contract calls, categorized by each program and differentiated based on initiators and responders. The execution costs of Split are shown in Table 1. Registering participants in their initial state requires 26k gas for the initiator and 16k gas for the responder. The initiator of a binding, which is the imbalanced node, needs to pay a binding cost of 326k gas. The Split.rebalancing program needs to execute multiple Re_s, thus consuming more gas. Unbinding can be initiated by any user in the rebalancing collection. When all user behaviors are honest, the cost of Split is acceptable, with the most expensive operations for both the initiator and the responder costing less than 330k gas.

5.2 Experimental Design

Topology. To simulate the payment channel network as realistically as possible, we base our topology on the Lightning Network,which is one of the deployed payment channel networks in the real world. We crawled complete network channel data for March 2023 from the Lightning Network data website (Amboss.space), including network topology and channel capacities. Considering the complex situation of multiple channels between nodes, we simplified by merging these channels and aggregating their balances. Ultimately, we obtained a network containing 14,102 nodes and 38,633 channels. Similar to reference [32], we used a snowball sampling method starting from the highest degree nodes, filtering out 761 nodes and 1,158 channels. Figure 5 displays the distribution of total balances across 1158 channels. It is evident that the majority of channels have total balances ranging from 0 to 3 BTC, with only a very few channels having balances exceeding 3 BTC. Due to the privacy protection design of the Lightning Network, we cannot access the fund allocation at each end of the channel. Therefore, this study simulated the network state under real conditions by generating random numbers within the $(0, C_i, j)$ range to satisfy Equation (1). The randomly generated channel fund distribution is shown in Figure 6.

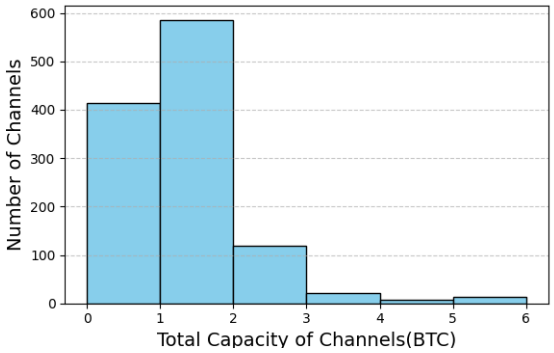


Figure 5: Channel capacity statistics

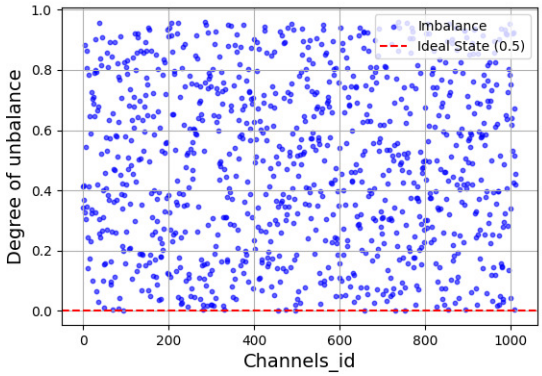


Figure 6: Scatterplot of channel funding distribution

The scatter plot distribution shown in Figure 6 reveals that only a very small number of channels exhibit funds distributed in the most ideal state (imbalance degree= 0). Most channels are in an unbalanced state, with many channels' imbalance degrees approaching the scale of 1.

Off-chain payments simulation. For receiving and sending, we considered both uniform and skewed distributions as typical scenarios. However, users paying with equal probability does not apply to real scenarios and fails to highlight the advantages of on-demand allocation in this scheme. Therefore, inspired by [32], we randomly sampled senders and receivers from independent exponential distributions. A greater skewness in the distribution indicates a higher probability of payment.

Transaction. Since off-chain transactions are privately maintained between nodes, it is not possible to collect

Table 1: The gas cost for executing the Split

Transaction Load	Split.setup	Split.Bind	Split.rebalancing	Split.Unbind
Initiator	26893	326893	562734	264095
Responderr	16893	0	135482	115843

historical transaction data from the Lightning Network. Therefore, we use the transaction dataset provided in the literature [9], which randomly samples based on Bitcoin trajectories from 2021-03-01 to 2021-03-31.This work has already filtered out large transactions unsuitable for off-chain, so we only need to randomly sample from it. Finally, the average, maximum, and minimum values of the transaction load distribution (about 1.8 million) were 285,085,868sat, 1,244,088,945sat, and 375,120sat.

Routing. We use the single-path shortest path first algorithm, Dijkstra, as the routing algorithm, meaning that each transaction uses the path with the fewest hops between the sender and the receiver.

5.3 Evaluation Indicators

When evaluating the performance of different rebalancing strategies, we selected six key metrics to ensure comprehensive and accurate assessment.

1) Success rate: The ratio of the number of successful transactions to the total number of generated transactions. A transaction is considered successful only when all sub-transactions reach their recipients.

2) Success Volume: The total amount of all successful transactions.

3) Shifted tokens: The total number of shifted tokens in one rebalancing operation. A higher number of shifted tokens indicates a better rebalancing effect.

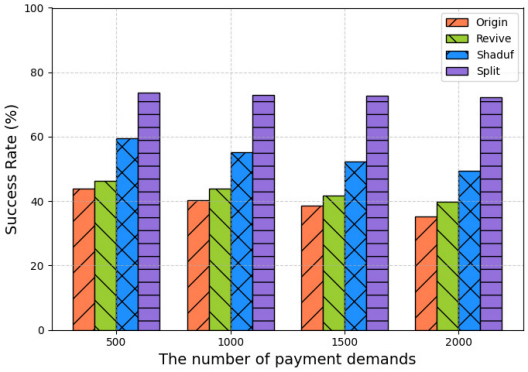
4) Running times: Assesses the time required for an effective rebalancing operation. Shorter execution times indicate higher method efficiency.

5) Skew Level: An important indicator for assessing network health, it calculates the percentage of unbalanced channels in the network. (A channel is considered imbalanced when the imbalance degree exceeds 70%.)

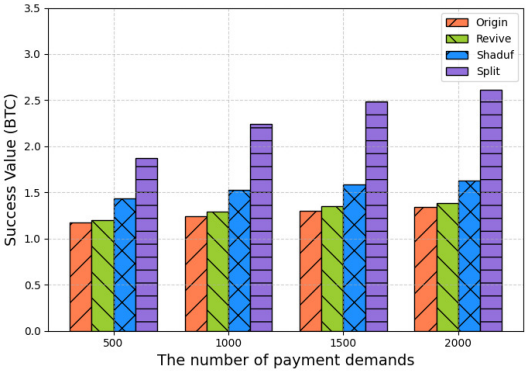
5.4 Analysis of Results

5.4.1 Transaction Load

Figure 7 compares the performance of three rebalancing strategies, Split, Revive [14], and Shaduf [9], in improving success rate and total successful transactions under different transaction load levels. Here, Orgin represents the original network state without any rebalancing operations.



(a)The number of payment demands vs. Transaction success rates



(b)The number of payment demands vs. Transaction success values(BTC)

Figure 7: Changes in success rate and total successful transactions under different transaction loads

It can be seen that all three strategies to some extent improve the success rate of transactions, but there are significant differences in their performance in enhancing success rates. As transaction volume increases, Revive and Shaduf show a declining trend in success rate, whereas Split maintains a certain stability. Specifically, under identical payment demands, Split achieves a success rate 26.3% to 33.4% higher than Revive and 14.7% to 21.9% higher than Shaduf. Additionally, the total amount of successful transactions is much higher than other methods, significantly enhancing Split's transaction success rate and payment capability.

Table 2 shows the changes in channel skew levels with increasing transaction demand after the three rebalancing operations. As the number of transactions increases, the flow of funds in the channels accelerates, leading to more

Table 2: Changes in network skew levels under different transaction loads

Transaction Load	500	1000	1500	2000	**Average Variation**
Revive	24.67%	27.23%	30.38%	35.2%	+3.25%
Shaduf	22.31%	23.43%	25.64%	28.53%	+1.85%
Split	15.25%	15.8%	15.75%	16.46%	+0.41%

unbalanced channels and a higher probability of channel depletion. The average skew levels for each group increased by 3.25% and 1.85% for Revive and Shaduf, respectively. In contrast, Split only increased the average channel change by 0.41%, demonstrating that this algorithm more evenly distributes channel funds, ensuring channel stability and durability under higher loads.

5.4.2 Network Size

We randomly remove some nodes from the network, each removing 50 as a group, and each group removes 50, 100, 150, 200, 250 nodes in turn, the last of which results in five network topologies of different sizes. Observe the changes in the success rate and the total number of successful transactions of the three rebalancing schemes, Split, Revive, and Shaduf, under different network sizes.

Figure 8 demonstrates the effect of network size, where the success rate decreases slightly as the network size increases. This is due to the fact that in a larger network, the payment path between two random nodes becomes longer, increasing the probability that the intermediate channel will be depleted of funds. It is particularly noteworthy that, compared to the other two schemes, Revive is more significantly affected by network size, with a decrease of about 17.6%. This is because Revive can only perform rebalancing within loops. Although the likelihood of loops increases with the network size, this is far from sufficient to support payments over long paths in large-scale networks. In contrast, Split is not restricted by network topology, and also overcomes the condition of limited balancing funds of weak channels. As a result, its success rate and total successful transactions remain relatively stable with changes in network size, with a decrease of only 2.14%.

When the maximum channel budget d increases, Revive takes a considerable amount of time to run. To accurately estimate the time consumption of the three algorithms, we chose to run them on a small-scale network. Figure 10 shows that the average running time of three rebalancing algorithms increases with the network size. Revive exhibits an exponential growth trend, with the average time to find the optimal solution approaching 7 seconds, making it impractical for real-world applications due to the immense computational cost. Split grows linearly and has a higher time loss than Shaduf, and this gap increases gradually as the network size increases. This is because Split divides the balance amount into multiple parts, increasing the computational load during the decision-making process, which inevitably extends its time complexity.

5.4.3 Frequency of Rebalancing Implementation

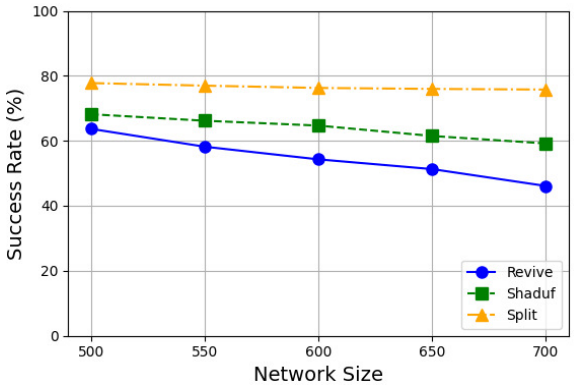
This paper sets five groups with different rebalancing execution frequencies within ten minutes, namely 5, 10, 15, 20, and 25. The variations in success rates and network skew levels of the three schemes under different rebalancing frequencies were analyzed.

Figure 9(a) illustrates the fluctuations in network skew levels under different rebalancing frequencies. The horizontal axis represents the number of rebalancing executions within ten minutes. It can be observed that the more frequent the rebalancing executions, the lower the channel imbalance levels. With frequent rebalancing, Split gradually stabilizes the channel imbalance levels at a relatively low value without further significant declines, indicating that the network balance reaches saturation. However, more rebalancing executions are not always better. Figure 9(b) indicates that when the execution frequency reaches 20 times within ten minutes, the success rates of all three schemes decline to varying degrees. This is because executing rebalancing requires freezing the channels for a certain period, interrupting their operation.

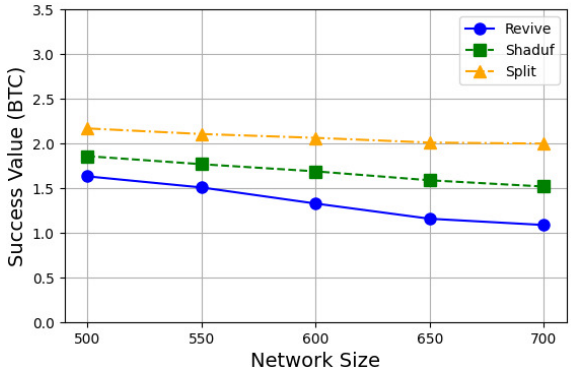
Table 3: Comparison of the number of tokens transferred in networks with different skew levels

Skew Level	25%	20%	15%	10%
Revive	0.733	0.573	0.383	0.212
Shaduf	0.964	0.863	0.734	0.674
Split	1.342	1.213	1.162	1.081

In Table 3, the experiment sets up five networks with different skew levels (with unchanged network topology) to observe the number of shifted tokens during one round of execution for each method. As the skew level decreases, the number of shifted tokens also decreases. The reason is that lower skew levels result in fewer unbalanced channels, and thus fewer tokens are needed for rebalancing. Specifically, Split transfers more tokens than Revive and Shaduf at the same skew level, indicating better rebalancing effect. As the rebalancing demand decreases, the fluctuations of the shifted tokens for Revive range from 0.154 to 0.196, for Shaduf from 0.058 to 0.131, and for Split from 0.029 to 0.075, showing that Split is far more stable than the other two strategies.

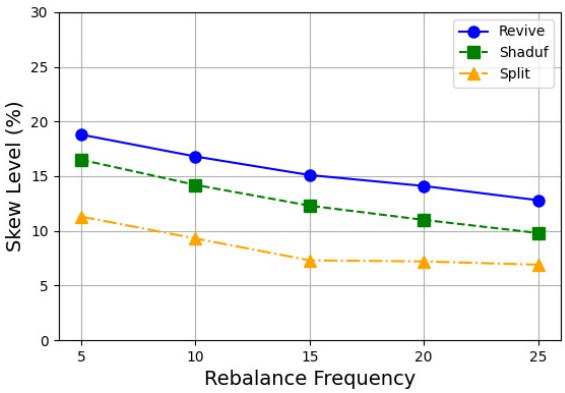


(a)Network size vs. Transaction success rates

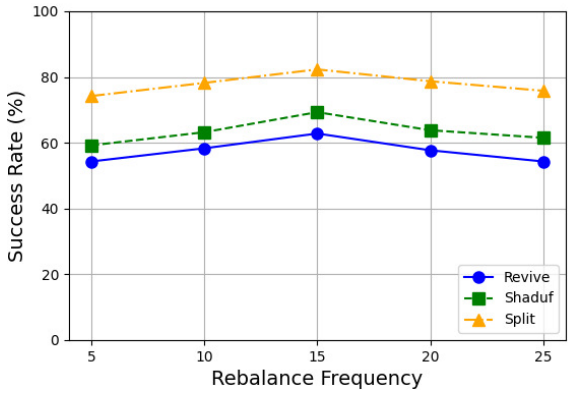


(b)Network size vs. Transaction success values(BTC)

Figure 8: Changes in success rates and total successful transactions at different network sizes



(a)Rebalance frequency vs. Skew level



(b)Rebalance frequency vs. Transaction success rates

Figure 9: Changes in success rate and skew level at different rebalancing execution frequencies

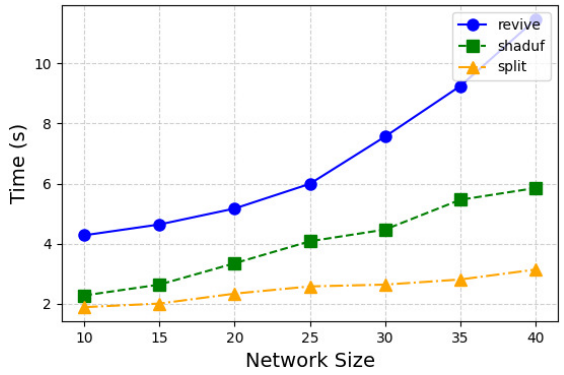


Figure 10: Runtime of the algorithm

6 Conclusions

This paper proposes Split, a new split multi-channel rebalancing scheme. We introduce the concept of payment task queues, which consider both queue length and imbalance degree to select adjacent channels with lower utilization and larger balance deviations. Additionally, to make rebalanced channels more durable, we use minimal channel probability loss to split rebalancing amounts, ensuring each channel is used reasonably. We conducted simulation experiments on a real Lightning Network topology, and the results indicate that compared to two existing rebalancing schemes, Split demonstrates superior performance in success rate, total successful transactions, and skew level. We have built a reliable and efficient rebalancing platform for imbalanced nodes in the network, enhancing the transaction efficiency of off-chain payments and improving the scalability of the PCN to support more users and transaction volume.

However, while Split significantly enhances the payment capability of payment channels, there are still some unresolved issues. Dividing a complex rebalancing process into several simpler sub-balances in this paper increases computational complexity in the decision-making process, which is not ideal in terms of time complexity performance. Moreover, frequent channel freezing for rebalancing may interrupt the normal operation of the network. Furthermore, in the current work, we only consider the performance of Split in single-path payments. Therefore, future work is summarized as follows:

- Optimize Split's Channel Selection and Balance Amount Split algorithms to reduce time consumption, and evaluate Split's runtime and transaction

success rate under different parameter values M and S to find an optimal parameter range.

- How to use current and future transaction loads as parameters to determine when to trigger rebalancing, thereby making better triggering decisions to ensure a low probability of interruption.
- Due to the atomicity of multipath routing, where each transaction atom must follow a specified path to reach the target node to be considered successful, the performance of Split combined with multipath routing needs further research.

Acknowledgments

This study was supported by the National Science Council of Taiwan under grant NSC 95-2416-H-159-003. The authors gratefully acknowledge the anonymous reviewers for their valuable comments.

References

[1] J. Cai, W. Liang, X. Li, K. Li, Z. Gui, and M. K. Khan, "Gtxchain: A secure iot smart blockchain architecture based on graph neural network," *IEEE Internet of Things Journal*, 2023.

[2] Q. Cai, J. Chen, D. Luo, G. Sun, H. Yu, and M. Guizani, "Deter-pay: A deterministic routing protocol in concurrent payment channel network," *IEEE Internet of Things Journal*, 2024.

[3] G. F. Camilo, G. A. F. Rebello, L. A. C. de Souza, M. E. M. Campista, and L. H. M. Costa, "Profitpilot: Enabling rebalancing in payment channel networks through profitable cycle creation," *IEEE Transactions on Network and Service Management*, 2024.

[4] P. Y. Chang, M.S. Hwang, C.C. Yang, "A blockchain-based traceable certification system", in *Security with Intelligent Computing and Big-data Services*, pp. 363-369, 2018.

[5] L. Chen, Q. Fu, Y. Mu, L. Zeng, F. Rezaeibagha, M. S. Hwang, "Blockchain-based random auditor committee for integrity verification", *Future Generation Computer Systems*, vol. 131, pp. 183-193, 2022.

[6] J. Dong, G. Xu, C. Ma, J. Liu, and U. G. O. Cliff, "Blockchain-based certificate-free cross-domain authentication mechanism for industrial internet," *IEEE Internet of Things Journal*, 2023.

[7] J. Dong, G. Xu, C. Ma, J. Liu, and U. G. O. Cliff, "Certificate-free cross-domain fine-grained access control mechanism for industrial internet," *HUMAN-CENTRIC COMPUTING AND INFORMATION SCIENCES*, vol. 14, 2024.

[8] L. Duan, Y. Sun, W. Ni, W. Ding, J. Liu, and W. Wang, "Attacks against cross-chain systems and defense approaches: A contemporary survey," *IEEE/CAA Journal of Automatica Sinica*, vol. 10, no. 8, pp. 1647–1667, 2023.

[9] Z. Ge, Y. Zhang, Y. Long, and D. Gu, "Shaduf++: Non-cycle and privacy-preserving payment channel rebalancing," *Cryptology ePrint Archive*, 2022.

[10] M. S. Hwang and P. C. Sung, "A study of micropayment based on one-way hash chain", *International Journal of Network Security*, vol. 2, no. 2, pp. 81-90, 2006.

[11] P. Jiang, J. Zhu, and L. Zhu, "Balancing privacy and regulation of cross-chain transaction systems via sok-assisted policy enhancement," *IEEE Transactions on Information Forensics and Security*, 2024.

[12] W. Jie, W. Qiu, A. S. V. Koe, J. Li, Y. Wang, Y. Wu, and J. Li, "A secure and flexible blockchain-based offline payment protocol," *IEEE Transactions on Computers*, 2023.

[13] A. A. Khalil, M. A. Rahman, and H. A. Kholidy, "Fakey: Fake hashed key attack on payment channel networks," in *2023 IEEE Conference on Communications and Network Security (CNS)*. IEEE, pp. 1–9, 2023.

[14] R. Khalil and A. Gervais, "Revive: Rebalancing off-blockchain payment networks," in *Proceedings of the 2017 acm sigsac conference on computer and communications security*, 2017, pp. 439–453.

[15] J. Lee and C. Park, "Analysis and evaluation of the raiden network in ethereum blockchain," *Proceeding of KIISE*, pp. 1484–1486.

[16] C.-T. Li and M.-S. Hwang, "A lightweight anonymous routing protocol without public key en/decryptions for wireless ad hoc networks," *Information Sciences*, vol. 181, no. 23, pp. 5333–5347, 2011.

[17] C. T. Li, C. C. Yang, and M. S. Hwang, "A secure routing protocol with node selfishness resistance in manets," *International Journal of Mobile Communications*, vol. 10, no. 1, pp. 103–118, 2012.

[18] D. Li, "Research on blockchain technology for security protection of network information data in the legal system background," *International Journal of Network Security*, vol. 26, no. 2, pp. 173–179, 2024.

[19] F.-P. Li, Q.-Y. Zhang, Y.-J. Hu, and Y.-B. Huang, "Verifiable encrypted speech retrieval method based on blockchain and c-bigru," *International Journal of Network Security*, vol. 26, no. 3, pp. 486–500, 2024.

[20] K. Li, T. Chen, and W. Yang, "Blockchain technology: The prevention of corporate financial statement forgery and frau," *International Journal of Network Security*, vol. 25, no. 6, pp. 964–969, 2023.

[21] M. Li, X. Luo, K. Xue, Y. Xue, W. Sun, and J. Li, "A secure and efficient blockchain sharding scheme via hybrid consensus and dynamic management," *IEEE Transactions on Information Forensics and Security*, 2024.

[22] P. Li, T. Miyazaki, and W. Zhou, "Secure balance planning of off-blockchain payment channel networks," in *IEEE INFOCOM 2020-IEEE conference on computer communications*. IEEE, 2020, pp. 1728–1737.

[23] Z.-C. Li, J.-H. Huang, D.-Q. Gao, Y.-H. Jiang, and L. Fan, "Iscp: An improved blockchain consensus protocol." *Int. J. Netw. Secur.*, vol. 21, no. 3, pp. 359–367, 2019.

[24] X. Luo and P. Li, "Learning-based off-chain transaction scheduling in prioritized payment channel networks," *IEEE Journal on Selected Areas in Communications*, vol. 40, no. 12, pp. 3589–3599, 2022.

[25] X. Luo and P. Li, "Leaf: Let's efficiently make adaptive forwarding in payment channel networks," *IEEE Access*, vol. 11, pp. 4194–4206, 2023.

[26] G. Malavolta, P. Moreno-Sanchez, C. Schneidewind, A. Kate, and M. Maffei, "Anonymous multi-hop locks for blockchain scalability and interoperability," *Cryptology ePrint Archive*, 2018.

[27] S. Martinazzi and A. Flori, "The evolving topology of the lightning network: Centralization, efficiency, robustness, synchronization, and anonymity," *Plos one*, vol. 15, no. 1, p. e0225966, 2020.

[28] P. Prihodko, S. Zhigulin, M. Sahno, A. Ostrovskiy, and O. Osuntokun, "Flare: An approach to routing in lightning network," *White Paper*, vol. 144, 2016.

[29] S. S. Sahoo and V. K. Chaurasiya, "Proof of location based delivery system using multi-party virtual state channel: a blockchain model," *The Journal of Supercomputing*, vol. 80, no. 1, pp. 703–733, 2024.

[30] S. S. Sahoo, M. M. Hosmane, and V. K. Chaurasiya, "A secure payment channel rebalancing model for layer-2 blockchain," *Internet of Things*, vol. 22, p. 100822, 2023.

[31] N. Sharma, K. Kapoor, and V. Anirudh, "Design and evaluation of swift routing for payment channel network," *Blockchain: Research and Applications*, vol. 5, no. 2, p. 100179, 2024.

[32] V. Sivaraman, S. B. Venkatakrishnan, K. Ruan, P. Negi, L. Yang, R. Mittal, G. Fanti, and M. Alizadeh, "High throughput cryptocurrency routing in payment channel networks," in *17th USENIX Symposium on Networked Systems Design and Implementation (NSDI 20)*, pp. 777–796, 2020.

[33] Q. Wang, Y. Zhang, Z. Bao, W. Shi, H. Lei, H. Liu, and B. Chen, "Sortee: Service-oriented routing for payment channel networks with scalability and privacy protection," *IEEE Transactions on Network and Service Management*, vol. 19, no. 4, pp. 3764–3780, 2022.

[34] X. Wang, C. Lin, X. Huang, and D. He, "Anonymity-enhancing multi-hop locks for monero-enabled payment channel networks," *IEEE Transactions on Information Forensics and Security*, 2023.

[35] X. Wang, R. Yu, D. Yang, G. Xue, H. Gu, Z. Li, and F. Zhou, "Fence: Fee-based online balance-aware routing in payment channel networks," *IEEE/ACM Transactions on Networking*, 2023.

[36] C.-C. Wu, C.-T. Chang, I.-C. Lin, and M.-S. Hwang, "Research on blockchain secret key sharing and its digital asset applications," *International Journal of Network Security*, vol. 26, no. 1, pp. 160–166, 2024.

[37] S. Wu, Q. Liu, and J. Li, "Research on the detection of illegal transactions in currency transactions based on blockchain technology," *International Journal of Network Security*, vol. 26, no. 1, pp. 19–24, 2024.

[38] J. Zhao and J. Zhu, "Data privacy protection based on unsupervised learning and blockchain technology," *International Journal of Network Security*, vol. 26, no. 2, pp. 312–320, 2024.

[39] D. Zhang, J. Le, N. Mu, and X. Liao, "An anonymous off-blockchain micropayments scheme for cryptocurrencies in the real world," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 50, no. 1, pp. 32–42, 2018.

[40] Y. Zhang, X. Jia, B. Pan, J. Shao, L. Fang, R. Lu, and G. Wei, "Anonymous multi-hop payment for payment channel networks," *IEEE Transactions on Dependable and Secure Computing*, vol. 21, no. 1, pp. 476–485, 2023.

Biography

Wei-Jun Gao received his Bachelor of Science degree from Lanzhou University in 1997, and has been engaged in teaching and research at Lanzhou University of Science and Technology since 2000. He is mainly engaged in software engineering, distributed and cloud computing, big data processing, and graphic image processing.

Jia-Ming Guo required by the She received her B.S. degree in Computer Science and Technology from Lanzhou City College in 2021, and is now studying for her M.S. degree at the School of Computer and Communication, Lanzhou University of Technology. His main research interests are network and information security, blockchain and off-chain payment channel.

Cheng-ying Jiao received his B.S. degree in Computer Science and Technology from Bohai University in 2021, and is now studying for his M.S. degree in the School of Computer and Communication, Lanzhou University of Technology. His main research interests are blockchain, IoT security and big data security and privacy protection.