Gnocchi: A Multiplexed Payment Channel Scheme

Chen Pan, Shuyang Tang, Zhiqiang Liu*, Yu Long*, Zhen Liu*, and Dawu Gu*

Department of Computer Science and Engineering,
Shanghai Jiao Tong University, Shanghai, China

Abstract. As the underlying technology of mainstream cryptocurrencies such as Bitcoin and Ethereum, blockchain builds a decentralized payment system. However, the security and consistency of such system derive from the wide replication of transaction data and expensive distributed consensus mechanism, which makes current cryptocurrencies suffer great scalability gap from meeting commercial demand. To date, several solutions have been proposed to solve the scalability bottleneck of blockchain system such as Bitcoin-NG, sharding mechanism and off-chain payment network. Off-chain payment network is one of the most promising approaches since it could significantly extend the throughput of the system and provide cheap instant micro-payment. In this work, we introduce Gnocchi, a multiplexed off-channel payment channel scheme that offers a novel method to construct an off-channel payment system to allow multi-party payment in one channel. We formally model our payment network and analyze network connectivity of our scheme to show that this newly proposed scheme significantly reduces the routing cost and transfer fee for cross-channel payment in contrast to existing off-chain payment scheme such as Lightning Network. This improves the efficiency of off-chain payment system further. We introduce transaction supervisor in a channel, while maintaining the decentralization feature of the system by elaborately designing a fraud proof contract to restrain any adversary in the channel.

Keywords: Blockchain, Scalability, Off-chain, Bidirectional Payment Channel, Multiplexed Payment Channel

1 Introduction

Since Bitcoin [17] was first introduced in 2008, blockchain, the fundamental technology of Bitcoin, has shown great promise of decentralized payment infrastructure or other commercial application scenarios. Blochchain provides a novel trust model in open network without relying on any trusted third party. In general, blockchains achieve their decentralization and security by wide replication of transaction data. All nodes in the system share the ledger and participate in the distributed consensus to modify the ledger database.

* Corresponding author

Despite its great potential, one of the main costs of the decentralization is the performance. For instance, Bitcoin can only process up to 7 transactions per second [9], while the practical payment systems such as VISA are reported to process 2000 transactions/sec on average, with peak rate of 56000 transactions/sec [4]. Simple approaches to improve the throughput of system include adjusting key blockchain parameters such as block interval and block size. However, it has been shown that in this way the system cannot deal with more than 100 transaction per second [3]. Currently, effective blockchain scaling solutions can be roughly divided into three categories: (*i*) designing alternative underlying consensus mechanisms to support more transactions [6,22], (*ii*) introducing off-chain payment channels which process most transactions off chain and reduce the blockchain load [19,18,1] or (*iii*) developing sharding mechanism to divide all nodes into several shards and process the transactions in parallel [2,13,8].

Off-chain payment network allows theoretically unlimited instant payments conducted simultaneously, providing almost infinite throughput for the system. Existing payment channel solutions consist of two components: bidirectional payment channel protocol and cross-channel payment protocol. In the bidirectional channel, to establish an off-chain channel, a pair of users generate a funding transaction and publish it to the blockchain to lock their fund as deposit. The off-chain payment can be viewed as local update of the new distribution of the balance in the channel. Then, the user wishing to withdraw his deposit needs to publish the final state of deposit balance to the blockchain. Importantly, to prevent users from publishing the previous state, the withdraw request would open a dispute time window for the counterparty to ratify the balance distribution and cause a penalty to user's misbehavior if necessary. Since the opening of bidirectional payment channel needs deposit to lock in the channel, it's expensive to maintain excessive channels at the same time for common users. By the cross-channel payment protocol, channel users could borrow others' channels to complete indirect off-chain payments. The cross-channel payment relies on the hash pre-image to synchronize the payment result in the payment path to ensure the fund security of the channel provider.

However, existing two-party payment channel schemes have following drawbacks:

- A high cost to maintain enough channels to ensure the success ratio of cross-channel payment.
- A complex routing problem to search a payment path.
- Expensive transaction fees during cross-channel payments when meeting long payment routes.

1.1 Our Contribution

In this work, we present Gnocchi, a multiplexed payment channel that supports multiplexed payment in one channel under a decentralized setting and outperforms the existing payment channel scheme such as Lightning Network

and Raiden Network, in terms of cost to maintain the payment network connectivity and expense on cross-channel payments including transaction fees and payment route searching. In Gnocchi, several users could request to establish a multiplexed channel with the cooperation of a transaction supervisor. The supervisor is introduced to improve the efficiency of updating the state in the channel during off-chain delivery. However, the transaction supervisor is not able to modify the balance of any user without the user's authentication. We demonstrate how Gnocchi achieves decentralization by using a fraud proof contract to regulate the behavior of supervisor.

More specifically, our contributions are listed as follows.

- We propose Gnocchi for off-chain payments which supports multi-party off-chain payments by introducing a novel fixed-member channel state consensus mechanism. All users could transfer their deposit to any other user in the off-chain payment network, which significantly improve the throughput of the blockchain.
- We employ an elaborately designed fraud proof contract in withdraw step to monitor the supervisor's potential malicious behavior.
- We define the security goal of off-chain payment scheme and analyze Ginocchi to show that the distribution of balance in the channel cannot be modified without the authentication of related users, while we only require supervisor to be online.
- We formalize the model of payment network to demonstrate that Ginocchi significantly improves the efficiency of off-chain payment and reduces the cost of payment in terms of transfer fee and routing expense.

In summary, Gnocchi offers an efficient, decentralized and secure solution for multiplexed off-chain payment channel to scale the current blockchain system.

1.2 Paper Organization

The remainder of the paper is organized as follows. Sec.2 overviews the related work on payment channel. In Sec.3, we present our construction of Gnocchi, a multiplexed payment channel scheme. Then we analyze the security and connectivity of this scheme in Sec.4. Finally, we conclude our work in Sec.5.

2 Related Work

2.1 Two-Party Payment Channel

Payment channel was first proposed for instant micro-payment because of the expensive transaction fee and long confirmation period in Bitcoin. The protocol can be divided into two steps: *Establishment* and *Payment* [15]. In the establishment step, customer needs to deposit fund to a multi-signature contract address. To complete an off-chain payment, customer needs to sign a newly update transaction to modify the current state of balance in the channel and send it to the

merchant. Since only the merchant is allowed to close the channel by publishing the update transaction to the blockchain, it could only satisfy the scenario that customer send incremental payments to the merchant.

Payment channel shows great potential to scale the blockchain since it could support payment without the on-chain consensus. Generally, a payment network scheme includes two components: bidirectional payment protocol and cross-channel payment protocol. In the bidirectional payment channel, the payment protocol needs to prevent the involved user from publishing the previous state such as penalty mechanism in Lightning Network and invalidation tree structure in Duplex Channel [5]. Two party bidirectional payment channel can be seen as the process to maintain a private ledger. The security of the payment channel is guaranteed by a dispute mechanism to affirm the current channel state during a fixed time window on blockchain.

Such dispute mechanism can be divided into two categories. The first category relies on penalty mechanism. One instance of such payment channel is Lightning Network. The malicious attacker would lose all his deposit in the channel once publishing the previous balance distribution. However, it requires user to store all the remedy transaction as the proof for malicious action. Raiden Network also adopts such punishment mechanism by introducing a round number i in update transactions to sort the update transactions. Since users only need to store the newest update the transaction with the largest round number, it reduces the local storage of user. The second category relies on invalidation tree structure to invalidate the previous state. The invalidation tree is based on descending time lock on update transaction to ensure the update transaction with less time lock to be written in blockchain.

Since it's expensive to maintain payment channel to every individual that wishes to make off-chain payment, linked payment is also part of the off-chain scaling solution. Through cross-channel payment protocol, channel user could find indirect payment path to complete off-chain payments. The cross-channel payment uses hash pre-image to interconnect payment channels to build a payment path and synchronize the payment result sequentially in the path.

Other payment channel schemes were also proposed to improve the efficiency [11,21,20] or privacy [14,10] on payment channel. Sprites [16] introduced a global pre-image for parallel synchronization of the payment result to reduce the lock time of deposit in hashed timelock contract. Multi-hop HTLC [14] was introduced to keep the sender and receiver privacy during cross-channel payment, since the hash locks are different in each hop.

2.2 N-Party Payment Channel

NOCUST [12] introduces a commitment scheme to store the private ledger of channel to provide N-party off-chain payment. Compared with above off-chain solutions, it reduces the cost of channel setup and routing. However, the introduction of financial intermediary without any punishment for its misbehavior may cause complex malicious acts. In NOCUST, intermediary needs to publish the commitment of balance state periodically, which would expand the online

ledger. Additionally, to prevent malicious withdrawal, customers could arise a challenge to ask the intermediary to open the current on-chain commitment. Once the challenge fails, to keep fund security, all transactions after the generation of fail-opened commitment would be reverted, which means the off-chain payment in NOCUST is not confirmed immediately in contrast to other off-chain scheme.

3 The Gnocchi Scheme

In this section, we present our construction of multiplexed payment channel scheme, Gnocchi, which addresses the drawbacks of current mainstream off-chain payment technology, such as expensive routing in cross-channel payment, high cost on maintaining widely-connected payment network, etc.

3.1 Overview

Our construction consists of two components: multiplexed payment channel and cross-channel payment technology. Multiplexed payment channel provides direct instant off-chain payment among all users in the channel, while cross-channel payment technology allows indirect payment in the payment network as other existing payment network scheme. The cross-channel payment technology in our scheme could reuse the existing technology, which utilizes hash pre-image to interconnect several multiplexed payment channels to establish a synchronized payment path. The detailed description of our multiplexed payment channel is presented in the following section.

In our scheme, we consider blockchain as the existing platform serving as the trust root of the system, which stores and maintains the global information of the transaction data and balance of each address(account) in the system. All parties in the system could manipulate their possessions with their respective private keys as certificates on the global blockchain. Besides, our construction requires a Turing-complete smart contract platform which could store the state and result of the contract execution, such that provided by Ethereum [7]. We assume that once a smart contract is deployed on the blockchain, it cannot be modified or obstructed by adversary during execution. We also assume that all participants in the multiplexed hub need to communicate by an extra underlying network, such as TCP connections.

To support a multiplexed off-chain payment function, a trivial solution is to simply adjust the two-party (the existing lightning network) payment channel scheme by introducing a mechanism that all the participants in the hub involve the update process of the state (balance of each party) during each off-chain payment delivery. However, such mechanism causes high cost of interaction between all participants in one update and the online requirement for all the parties in the channel. Once some party in channel is offline, any off-chain delivery cannot utilize this channel, which will reduce the usability of the off-chain payment channel.

Due to this, we consider the update process of the off-chain payment as fix-member consensus. All the off-chain payment happened in a multi-party channel form a sequential subchain as the transaction history while a transaction supervisor S is introduced to take charge for the transaction sort and verification. In this way, we don't require all member being online except the S and counterparty of payment.

To avoid S's being corrupted by adversary, we elaborately design a fraud proof function to monitor the misbehavior of S. We introduced the supervisor role only for the efficiency and no offline requirement for our scheme. Since then, we need to disqualify the S from being a centralized node which builds the trust of protocol. Our scheme should hold following properties:

- Channel participants must be able to withdraw their balance in off-chain channel to their on-chain address at any time.
- The account balance in off-chain channel cannot be changed without owner's authorization.

These properties maintain the decentralization of our scheme, which do not rely on the honest assumption of supervisor. We illustrate these properties in Section 4.

3.2 Description of Multiplexed Payment Channel

The off-chain payment delivery in our multiplexed channel can be divided into following three steps: (1) Channel Setup; (2) Off-chain payment; (3) Withdraw. Tab.1 presents some of the notations used in this section.

Notation	Description
P_i	the identity of participant i
D_i	the original deposit of P_i in the channel
r	the round of update transaction in the channel
$A_i(r)$	the current balance of P_i in the channel in r round
n	the number of participants in the channel
G	the guarantee deposit for supervisor S

Table 1. Table of Notations

3.2.1 Channel Setup

In this step, n participants and one supervisor construct a multi-party payment channel corporately. All parties need to deposit their fund in the channel while the supervisor S is forced to lock guarantee deposit. The guarantee deposit cannot be transferred until the some P_i requests to close the channel on the

blockchain. The guarantee deposit is used to prevent the misbehavior of supervisor. Once the on-chain verifier notice the fraud proof of relative supervisor, its deposit will become the penalty and transfer to every participant in the channel to cover the potential loss of each P_i. Since then, we set the amount of guarantee as following:

$$G = n \times \sum_{i=1}^{n} D_i$$

By this way, each party will get $\sum_{i=1}^{n} D_i$ as their penalty reward as possible loss, which is the most fund they can receive in the off-chain channel.

The channel setup may only be done once per off-chain channel. After each party deposit their fund in the channel, the initial balance in the channel of all accounts is also generated.

$$A_i(0) = D_i$$

After the validation of on-chain setup transaction, the off-chain complete construction.

Input: On-chain transfer from all P_i and S to contract V_f

1. Validate G
2. Set $A_i(0) = D_i$ for all P_i

3.2.2 Off-chain Payment

In Off-chain Payment, the counterparty and supervisor need to generate a update transaction $\mathsf{Update}(i, j, r, m)$ (or simplified as $\mathsf{Update}(i, j, r)$ without considering the specific amount) with signatures of counterparty and supervisor, where m denotes the amount of this off-chain payment. A P_i wishing to transfer its fund to a P_j need to interact with supervisor S as follows:

1. P_i sends a new signed update transaction $\mathsf{Update}(i, j, r, m)$ to supervisor.
2. P_j sends a signed update transaction $\mathsf{Update}(i, j, r, m)$ to supervisor.
3. Supervisor S check the remaining balance of P_i.
4. After validating the balance of P_i, S set $A_i(r) = A_i(r-1)$ for P_i and P_j.
5. S sign update transaction $\mathsf{Update}(i, j, r, m)$ and latest state of P_i and P_j $\mathsf{State}(r+1)$ to ratify this off-chain transfer and broadcast to all channel participants.

To prevent double spend attack, the round number r needs progressive increase after each update of channel state.

3.2.3 Withdraw

A P_i wishing to withdraw its fund from payment channel need to publish its latest signed update transaction $\mathsf{Update}(i, j, r, m)$ as its current balance in the channel. To avoid P_i to publish its previous balance and monitor the supervisor,

the withdrawing balance need to be locked for a time window(a fixed number of blocks) before transferred to P_i's address on chain. In this disputing time, every party in the channel can submit fraud proof to V_f if necessary to challenge P_i's withdraw.

The fraud proof verifier V_f process the request and proof as followed:

Input: On-chain withdraw request $\mathsf{Update}(i, j, r_1, m_1)$ and $\mathsf{State}(r_1+1)$ from P_i and any fraud proof $\mathsf{Update}(i, k, r_2, m_2)$, $\mathsf{State}(r_2+1)$ to contract V_f

1. validate the signature of withdraw request $\mathsf{Update}(i, j, r_1, m_1)$ and $\mathsf{State}(r_1+1)$.
2. enter the dispute process, all fraud proof should be submitted during the time window.
3. validate the signature of fraud proof $\mathsf{Update}(i, k, r_2, m_2)$ and $\mathsf{State}(r_2+1)$.
4. verify $r_1 = r_2$, set guarantee deposit to all parties in the channel and close the channel (supervisor misbehavior).
5. verify $r_2 > r_1$, set $A_i(r) = 0$ and divide its fund equally to all other parties in the channel(P_i misbehavior)
6. transfer m_1 to P_i on chain address and set $A_i = 0$.

4 Security and Efficiency Analysis

In this section we will analyze the security and efficiency of Gnocchi.

4.1 Security Analysis

Our model is robust in face of invalid transactions and states, like transactions deducting excessive money from an address and states with invalid signatures. As long as an invalid state or transaction with the signature of the supervisor is broadcast to channel participants, a proof of misbehavior is formed and then other participants of the channel can submit this proof to the blockchain and redeem their deposits (from the guarantee deposit of the supervisor).

Also, the basic liveness of the network is guaranteed since any participant can terminate the channel and redeem their deposits in case of the breaking down of channel nodes. Besides, this seldom happens since no participant gains anything from an artificial malfunction unless a natural malfunction happens to the receiver or the supervisor which happens with a narrow possibility.

However, more considerations are required to evaluate a payment system besides the very basic guarantee of avoiding invalid transactions and states. Specifically, we consider the *consistency* property, which is defined as follows.

Consistency. We say that an update transactions $\mathsf{Update}(i, j, r)$ is *complete* if it is signed by its sender, receiver and the supervisor, and broadcast already by the supervisor to all channel participants. A *complete* state is defined recursively: an empty state is complete and a state updated by a complete transaction from a complete state is also complete. The consistency consists of two parts.

- (a) Each state $\mathsf{State}(r)$ is updated from its precedent state $\mathsf{State}(r-1)$ with a complete transaction $\mathsf{tx} = \mathsf{Update}(i, j, r-1)$ if the supervisor is rational. We denote such a relation with

$$\mathsf{State}(r-1) \overset{\mathsf{tx}}{\triangleright} \mathsf{State}(r)$$

  or simplify the notation '$\overset{\mathsf{tx}}{\triangleright}$' to '$\triangleright$' if there is no ambiguity.
- (b) For each state $\mathsf{State}(r-1)$, a rational supervisor would not broadcast two complete states $\mathsf{State}(r)$ and $\mathsf{State}(r')$ ($\mathsf{State}(r) \neq \mathsf{State}(r')$) such that

$$(\mathsf{State}(r-1) \triangleright \mathsf{State}(r)) \wedge (\mathsf{State}(r-1) \triangleright \mathsf{State}(r')).$$

Participants can withdraw their deposit in case that the supervisor is irrational and the rule is broken (as we have discussed).

Demonstrating our defined consistency of our system is equivalent to showing that any participant breaking rule (a) or rule (b) is irrational. This is easy to show since participants can withdraw their deposit if the supervisor is irrational and rule (a) or rule (b) is broken by the supervisor. Specifically, if the supervisor breaks rule (a), it should have signed on both $\mathsf{State}(r-1)$ and $\mathsf{State}(r)$ that $\neg(\mathsf{State}(r-1) \triangleright \mathsf{State}(r))$. Other participants can withdraw their money with a withdraw transaction with the witness of these signatures and states (from the guarantee deposit of the supervisor). This also works if rule (b) is broken in which case the witness is states and supervisor signatures of $\mathsf{State}(r-1)$, $\mathsf{State}(r)$ and $\mathsf{State}(r')$.

4.2 Efficiency Analysis

We first define the connectivity property of off-chain payment network, with which we discuss the efficiency of Gnocchi in the following.

Connectivity. A fine connectivity of nodes can be reached by any model of payment channels by covering the network with a sufficient amount of channels. However, in this way, a large cost of ledger storage (since payment channels are essentially transactions with specialized scripts) is required to reach a fine connectivity of the payment network. Therefore, it is crucial to estimate the cost to reach the same network connectivity of the lightning network and our model. In fact, our model demands a smaller cost.

Since it is too ideal to ask for the full connectivity of all network nodes, we consider a network is well-connected if more than 90% of them are reachable to each other through existing channels. However, having nodes connected is far from claiming a fine connectivity since the shortest path between two nodes is probably hundreds of tunnels in which case it is hard to build a "rapid" payment channel. Therefore, we define and estimate the *diameter*s (the maximum number of least channels required to build a path between two nodes) of two well-connected networks to form a comparison on their connectivity.

We refer to the amount of information required to be recorded on the ledger to form the network of payment channels as the *script cost*. With the same script cost, our network has a significantly better connectivity measured in graph diameter compared with the existing lightning network. Namely, less intermediate tunnels are required for one payment between two nodes in distant. To show this, we firstly provide a theoretical model based on which formal analyses are provided. Afterwards, we provide experimental simulations to show that our model matches the practice and that the difference in the connectivity of two networks is considerable.

4.2.1 Formal Model. Generally, our newly proposed multiplexed network is topologically equivalent to a *hypergraph* in the combinatorics literature where each edge is a set of nodes instead of two represents for the a multiplexed channel of our protocol. In contrast, the lightning network is topologically a classic graph connected by lightning channels.

We firstly show a certain script cost is sufficient to bring about a diameter of 3 of the multiplexed network. Afterwards, we show the lightning network with the same amount of vertices and script cost has a greater diameter.

To describe a lightning network, we use a graph $G_{n',m'} = (V', E')$ (G for short) where V' of size $|V'| = n'$ is the set of all vertices where each one represents for a network node and E' of size m' in expectation is the set of all edges where each stands for a lightning channel. There is an edges between each two nodes with a probability of p (the density). Similarly, we use the notation $\mathcal{G}_{n,m,\ell} = (V, E)$ to describe a multiplexed network where V of size $|V| = n$ is the set of all vertices, E of size $|E| = m$ is the set of all "hyper-edge"s, each of which is a group of vertices $\{u_1, u_2, \ldots, u_{\ell'}\} \in E$ standing for a channel of our model and that the average size of all hyper-edges is ℓ. To each node, $\frac{m\ell}{n}$ hyper-edges it participates are randomly chosen from E. For a graph $G = (V, E)$ its diameter is defined as $\max_{u,v \in V}\{\mathsf{D}(u, v)\}$ if well-connected where $\mathsf{D}(u, v)$ is the minimum edges required to establish a path from u to v. If it is not well-connected, its diameter is defined as ∞. The diameter of hypergraph $\mathcal{G}$ is defined as same. Before arguing about the connectivity, we formalize the script cost.

Definition 1 (Script Cost). *The script cost is $\ell \times m$ to a hypergraph $\mathcal{G}_{n,m,\ell}$, or $2 \times m'$ to a classical graph $G_{n',m'}$.*

In practice, the exact amount of information required to be store on the ledger is $(\ell + \delta)m$ to the multiplexed network while $(2 + \delta)m'$ to the lightning network for some marginal cost δ for auxiliary information. However, this only strengthens our result since that given a multiplexed network of diameter 3 with script cost ℓm, if the lightning network with $m' = \frac{\ell m}{2}$ channels has a diameter greater than 3, it furthermore has a greater diameter with $\frac{(\ell+\delta)m}{2+\delta} < \frac{\ell m}{2}$ edges.

Theorem 1 (Connectivity of Multiplexed Network). *Our multiplexed network $\mathcal{G}_{n,m,\ell}$ almost surely has a diameter of 3 if*

$$m = \frac{(1+\epsilon) \cdot n(2n \ln n)^{\frac{1}{3}}}{\ell^2},$$

in which case its script cost is

$$m = \frac{(1+\epsilon)\cdot n(2n\ln n)^{\frac{1}{3}}}{\ell}. \tag{1}$$

Proof. When the graph is sparse and $\ell \ll n$, the event of two vertex pairs having two common hyper-edges are nearly independent. Due to this, we approximate the graph as a classical graph of n vertices and $m' = \binom{\ell}{2}m$ edges. Borrowing lemma. 1, this graph (hence $\mathcal{G}$) almost surely has a diameter of 3 if its density of edge

$$\frac{m'}{\binom{n}{2}} \approx \frac{\ell^2 m}{n^2} = \frac{(1+\epsilon)(2n\ln n)^{\frac{1}{3}}}{n},$$

which leads to $m = \frac{(1+\epsilon)\cdot n(2n\ln n)^{\frac{1}{3}}}{\ell^2}$.

Theorem 2 (Connectivity of Lightning Network). *With the same vertex set $|V'| = |V| = n$ and the same script cost of formula (1), the existing lightning network $G = (V', E')$ can only reach a diameter of*

$$\lceil \frac{3\ln(2n\ln n)}{\ln(2n\ln n) - 3\ln \ell} \rceil > 3.$$

Proof. With the same complexity of formula (1), we can build a graph of edges

$$m' = \frac{(1+\epsilon)\cdot n(2n\ln n)^{\frac{1}{3}}}{2\ell}.$$

This leads to a density

$$p' = \frac{m}{\binom{n}{2}} \approx \frac{(1+\epsilon)(2n\ln n)^{\frac{1}{3}}}{n\ell}.$$

The threshold of the least diameter k it meets satisfies

$$\frac{(1+\epsilon')(2n\ln n)^{\frac{1}{k}}}{n} < p' \approx \frac{(1+\epsilon)(2n\ln n)^{\frac{1}{3}}}{n\ell}.$$

Rearrange the formula above, we have its diameter

$$k = \lceil \frac{3\ln(2n\ln n)}{\ln(2n\ln n) - 3\ln \ell} \rceil > 3.$$

Lemma 1 (Threshold by Diameter). *The property of a connected graph $G_{n,m}$ of diameter k ($\ell \ll n$ and k is small) has a threshold at $p \gtrsim 2^{\frac{1}{k}} \cdot \frac{(\ln n)^{\frac{1}{k}}}{n^{\frac{k-1}{k}}}$. Namely, $G_{n,m}$ has a diameter no greater than k if $p \gtrsim 2^{\frac{1}{k}} \cdot \frac{(\ln n)^{\frac{1}{k}}}{n^{\frac{k-1}{k}}}$ except for a probability trends towards zero as n grows to infinity.*

$$\lim_{n\to\infty} \Pr\left[\mathit{diameter}(G_{n,m}) > k\right] = 0$$

Proof. We call a movement from a vertex to an adjacent vertex as a hop. If G has a diameter greater than k, then there is a pair of nonadjacent vertices i, j such that i could not reach j in k hops. We call such a pair of vertices *bad*.

We Introduce a set of indicator random variables $I_{i,j}$ for each $i < j$. $I_{i,j}$ is 1 iff (i, j) is a bad pair and 0 otherwise. Let

$$X = \sum_{i<j} I_{i,j}$$

be the number of bad pairs. A graph has diameter at most k iff $X = 0$. Thus,

$$\lim_{n\to\infty} \Pr\left[\mathsf{diameter}(G) > k\right] = \lim_{n\to\infty} \Pr[X \neq 0].$$

$$E[X] = \binom{n}{2} \cdot (1 - \mathsf{fes}_1^{n-2}) \cdot (1 - \mathsf{fes}_2^{n-2}) \cdots (1 - \mathsf{fes}_k^{n-2}).$$

where $\mathsf{fes}_k^{\bar{n}}$ is the probability that two given vertices are reachable in k hops passing through $\bar{n}$ other nodes. Thus,

$$\begin{cases} \mathsf{fes}_1^{\bar{n}} = p \\ \mathsf{fes}_k^{\bar{n}} = 1 - (1 - p + p(1 - \mathsf{fes}_{k-1}^{\bar{n}-1}))^{\bar{n}} \\ \qquad = 1 - (1 - p \cdot \mathsf{fes}_{k-1}^{\bar{n}-1})^{\bar{n}} \end{cases}$$

Rearranging the formula, we find

$$1 - \mathsf{fes}_k^{\bar{n}} \cong (1 - \mathsf{fes}_{i,p,k-1}^{\bar{n}-1})^{p\bar{n}}$$

where formulas on two sides of a "$\cong$" are almost equivalent for a sufficiently large n. By a simple induction,

$$\begin{aligned} 1 - \mathsf{fes}_k^{\bar{n}} &\cong (1-p)^{p^{k-1} \cdot \bar{n} \cdot (\bar{n}-1) \cdots (\bar{n}-k+2)} \\ &\cong (1-p)^{(p\bar{n})^{k-1}} \\ &\cong e^{-p^k \bar{n}^{k-1}} \end{aligned}$$

Taking $\bar{n} = n - 2$,

$$\begin{aligned} -\ln E[X] &\cong -2\ln n + p \cdot \sum_{i=0}^{k-1} p^i (n-2)^i \\ &= -2\ln n + p^k \cdot n^{k-1}, \end{aligned}$$

Setting $p = c \cdot \frac{(\ln n)^{\frac{1}{k}}}{n^{\frac{k-1}{k}}}$,

$$-\ln E[X] \cong -2\ln n + c^k \ln n = (c^k - 2)\ln n.$$

To satisfy $\lim_{n\to\infty} -\ln E[X] = \infty$, we assign $c = (1+\epsilon) \cdot 2^{\frac{1}{k}}$ with a small positive margin value ϵ.

Therefore, with a density

$$p = \frac{(1+\epsilon)(2n\ln n)^{\frac{1}{k}}}{n} \gtrsim 2^{\frac{1}{k}} \cdot \frac{(\ln n)^{\frac{1}{k}}}{n^{\frac{k-1}{k}}},$$

$E[X]$ goes to zero and thereby

$$\lim_{n\to\infty} \Pr\left[\mathsf{diameter}(G) > k\right] = \lim_{n\to\infty} \Pr[X \neq 0] = 0.$$

4.2.2 Simulations. To support our theoretical conclusion of a better connectivity with the same script cost. We consider a lightning network G of $n' = 10^5$ total nodes and a multiplexed network $\mathcal{G}$ of expected group size $\ell = 25$ and total nodes of $n = n' = 10^5$. We randomly sample edges (hyper-edges) for G ($\mathcal{G}$) from the family of random graphs (hypergraphs) of 10^5 nodes with script cost varying from 0 to 2.0×10^6 (by 2500). Our results are shown in Fig. 1. Each triangle shows the diameter of a randomly sampled lightning network, while its corresponding dot on the same column shows the diameter of a randomly sampled multiplexed network. Obviously, hypergraph has a significantly better connectivity especially when channels are distributed sparsely (manifesting as a low script cost).

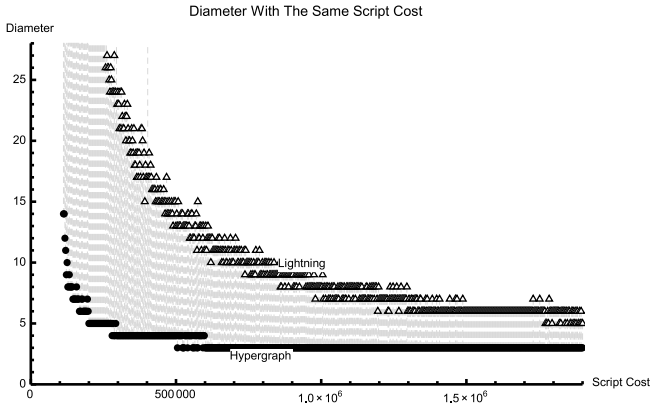

Fig. 1. Comparison on Diameter

5 Conclusion

With the advent of off-chain payment channel, the throughput of blockchain could significantly increase. However, existing payment channel schemes face the challenges of complicated and inefficient routing for cross-channel payment and high expense to maintain strong connectivity among the off-chain payment network. In this work, we presented Gnocchi, a multiplexed payment scheme, to

address the current drawbacks of off-chain payment scheme. Gnocchi supports instant off-chain payment by introducing a fixed-member channel state consensus mechanism. While introducing a transaction supervisor in each channel, we maintained the decentralization of our scheme by elaborately designing the fraud proof contract to resist malicious behavior well. Security and efficiency analysis of our scheme showed that it is consistent, and significantly improves the efficiency of cross-channel routing and reduces the cost of cross-channel payment in terms of transfer fee under certain off-chain network connectivity.

References

1. Raiden network. http://raiden.networkwork/.
2. Zilliqa. https://docs.zilliqa.com/whitepaper.pdf.
3. G. Andresen. Bip 101: Increase maximum block size. https://github.com/bitcoin/bips/blob/master/bip-0101.mediawiki, 2016.
4. K. Croman, C. Decker, I. Eyal, A. E. Gencer, A. Juels, A. E. Kosba, A. Miller, P. Saxena, E. Shi, E. G. Sirer, D. Song, and R. Wattenhofer. On scaling decentralized blockchains - (A position paper). In *Financial Cryptography and Data Security - FC 2016 International Workshops, BITCOIN, VOTING, and WAHC, Christ Church, Barbados, February 26, 2016, Revised Selected Papers*, pages 106–125, 2016.
5. C. Decker and R. Wattenhofer. A fast and scalable payment network with bitcoin duplex micropayment channels. In *Stabilization, Safety, and Security of Distributed Systems - 17th International Symposium, SSS 2015, Edmonton, AB, Canada, August 18-21, 2015, Proceedings*, pages 3–18, 2015.
6. I. Eyal, A. E. Gencer, E. G. Sirer, and R. van Renesse. Bitcoin-NG: A scalable blockchain protocol. *13th USENIX Symposium on Networked Systems Design and Implementation, NSDI 2016, Santa Clara, CA, USA, March 16-18, 2016*, pages 45–59, 2016.
7. GavinWood. Ethereum:asecuredecentralisedgeneralisedtransactionledger. ethereum project yellow paper, 2014.
8. A. E. Gencer, R. van Renesse, and E. G. Sirer. Short paper: Service-oriented sharding for blockchains. In *Financial Cryptography and Data Security - 21st International Conference, FC 2017, Sliema, Malta, April 3-7, 2017, Revised Selected Papers*, pages 393–401, 2017.
9. A. Gervais, G. O. Karame, K. Wüst, V. Glykantzis, H. Ritzdorf, and S. Capkun. On the security and performance of proof of work blockchains. *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, Vienna, Austria, October 24-28, 2016*, pages 3–16, 2016.
10. M. Green and I. Miers. Bolt: Anonymous payment channels for decentralized currencies. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security, CCS 2017, Dallas, TX, USA, October 30 - November 03, 2017*, pages 473–489, 2017.
11. R. Khalil and A. Gervais. Revive: Rebalancing off-blockchain payment networks. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security, CCS 2017, Dallas, TX, USA, October 30 - November 03, 2017*, pages 439–453, 2017.
12. R. Khalil and A. Gervais. NOCUST - A non-custodial 2nd-layer financial intermediary. *IACR Cryptology ePrint Archive*, 2018:642, 2018.

13. L. Luu, V. Narayanan, C. Zheng, K. Baweja, S. Gilbert, and P. Saxena. A secure sharding protocol for open blockchains. *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, Vienna, Austria, October 24-28, 2016*, pages 17–30, 2016.
14. G. Malavolta, P. Moreno-Sanchez, A. Kate, M. Maffei, and S. Ravi. Concurrency and privacy with payment-channel networks. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security, CCS 2017, Dallas, TX, USA, October 30 - November 03, 2017*, pages 455–471, 2017.
15. P. McCorry, M. Möser, S. F. Shahandashti, and F. Hao. Towards bitcoin payment networks. In *Information Security and Privacy - 21st Australasian Conference, ACISP 2016, Melbourne, VIC, Australia, July 4-6, 2016, Proceedings, Part I*, pages 57–76, 2016.
16. A. Miller, I. Bentov, R. Kumaresan, and P. McCorry. Sprites: Payment channels that go faster than lightning. *CoRR*, abs/1702.05812, 2017.
17. S. Nakamoto. Bitcoin: A peer-to-peer electronic cash system, 2008.
18. J. Poon and V. Buterin. Plasma: Scalable autonomous smart contracts. https://www.plasma.io/plasma.pdf, 2017.
19. J. Poon and T. Dryja. The bitcoin lightning network: Scalable off-chain instant payments. https://lightning.network/lightning-network-paper.pdf, 2016.
20. P. Prihodko, S. Zhigulin, M. Sahno, A. Ostrovskiy, and O. Osuntukun. Flare: An approach to routing in lightning network. https://bitfury.com/content/downloads/whitepaper_flare_an_approach_to_routing_in_lightning_network_7_7_2016.pdf, 2016.
21. S. Roos, P. Moreno-Sanchez, A. Kate, and I. Goldberg. Settling payments fast and private: Efficient decentralized routing for path-based transactions. In *25th Annual Network and Distributed System Security Symposium, NDSS 2018, San Diego, California, USA, February 18-21, 2018*, 2018.
22. C. Wan, Y. Zhang, C. Pan, Z. Liu, Y. Long, Z. Liu, Y. Yu, and S. Tang. Goshawk: A novel efficient, robust and flexible blockchain protocol. *IACR Cryptology ePrint Archive*, 2018:407, 2018.