Characterizing Performance Limits in Payment Channel Networks

Yuechen Tao, Bo Li, *Fellow, IEEE*, Baochun Li, *Fellow, IEEE*, and Lei Chen, *Fellow, IEEE*

Abstract—With their instant transaction confirmation and high scalability, payment channel networks (PCNs), running off-chain and in parallel with blockchain systems, have recently attracted a substantial amount of research attention. It has been shown that there exists a significant gap between the theoretically optimal performance and the performance achievable given the stringent privacy requirements in practice. However, it remains unclear what the fundamental performance limits and key factors involved are, which turns out to be a challenging problem due to the unique characteristics in PCNs. In this paper, we, for the first time, develop a mathematical model capturing the PCN performance, and examine the impact from a number of factors including channel capacity and transactions. We are articularly interested in obtaining the gap between the theoretically optimal performance and the performance achievable in practice, which characterizes the design space in PCNs for scheduling transactions. Specifically, we derive how different transactions and channel capacities affect the PCN performance and the performance gap. Our analytical characterization of PCNs offers an in-depth understanding on their fundamental trade-off, and provides important insights on the design of PCNs.

Index Terms—Blockchain, payment channel network, off-chain transaction, channel capacity, performance gap, privacy

1 Introduction

Payment channel networks (PCNs) [1], [2], [3], [4], [5], [6] are off-chain networks, whose transactions run in parallel to on-chain transactions in a blockchain system. In general, a payer and a payee in a PCN can collaboratively open a bidirectional channel with an initial *capacity* in the form of coins through an on-chain transaction. After that, the payer and the payee can directly conduct confirmed transactions with each other in an off-chain manner, as long as the capacity of this channel is no *less* than the number of coins requested by those transactions. After each successful transaction, the channel capacity is reduced by the amount requested by the transaction. The most appealing feature in PCNs is that the updated capacity after each transaction does not need to be updated in the on-chain blockchain system, instead the update will be done when the channel is closed. This results in an instant transaction confirmation in off-chain. In addition, two users without opening a direct channel in a PCN can also conduct transactions through intermediate channels. In other words, transactions between these two users requiring instant confirmations can be relayed through multiple intermediate channels, whose capacities are consumed and deducted by the amount required by such transactions.

There are several distinctive features in PCNs that are different from conventional blockchain systems, such as Bitcoin [7], Ethereum [8] and Hyperledger [9]. *First*, it is secure and private [4], [5], [10], [11], [12], [13], [14], [15], [16]. A transaction is oblivious to all the users except its sender and receiver to ensure the security, which can be achieved with HTLC contracts [17] and onion routing protocols [18]. *Second*, transactions in PCNs are cheaper and faster. Once channels are established, users can perform as many off-chain transactions as needed as long as the capacities satisfy the transaction requests. *Third*, it only needs to update the blockchain twice when a channel opens and closes. *Finally*, PCNs are highly scalable with respect to the number of users involved, since transactions are confirmed without requiring validation from all the users.

However, there exist fundamental performance limits in a PCN due to its stringent privacy requirements. Suppose intermediate channels are requested by multiple transactions and there are insufficient capacities to relay all of them. Transaction scheduling at intermediate channels, where transactions are selected for relay, is critical for performance in term of the number of successful transaction confirmations. The optimal performance in theory can be obtained if the information on all the channel capacities and transactions are known in advance, which is not practically feasible in PCNs with privacy requirements. Thus, transaction scheduling in practice is conducted in a highly distributed manner with limited local knowledge on channel capacities and transactions [19], [20], [21], [22], [23].

Evidently, in a PCN, different sets of transactions that are selected to relay during the scheduling process must result in different performances, i.e., transactions can affect the performance. Further, if transactions are fixed, the performance must change with different channel capacities, which further affect the gap between the theoretically optimal performance and the performance achievable in practice. In

- *Yuechen Tao, Bo Li, and Lei Chen are with the Department of Computer Science and Engineering, The Hong Kong University of Science and Technology, Hong Kong. E-mail: {ytaoaf, bli, leichen}@cse.ust.hk.*
- *Baochun Li is with the Electrical and Computer Engineering, University of Toronto, Toronto, ON M5S 1A1, Canada. E-mail: bli@ece.toronto.edu.*

Manuscript received 15 May 2021; revised 23 September 2021; accepted 1 October 2021. Date of publication 19 October 2021; date of current version 3 February 2023.
This work was support in part by RGC RIF under Grant R6021-20 and in part by RGC GRF under Grants 16207818 and 16209120.
(Corresponding author: Bo Li.)
Recommended for acceptance by Y. Zhang.
Digital Object Identifier no. 10.1109/TKDE.2021.3120842

some existing works, to reduce such a gap as much as possible, channel capacities are initialized and adjusted based on estimated transactions amounts [21], [24].

Despite some success of the real world deployment about PCNs, there are several fundamental questions that have not been properly addressed. How do we quantitatively obtain the gap between the theoretically optimal performance and the performance achievable in practice? How do we characterize such a gap by capturing effects of channel capacities and transactions? Do there exist some insights for PCNs from these analyses?

Given the unique characteristics of PCNs, these problems turn out to be particularly challenging. To start with, transaction or capacity characterization is drastically different from their counterpart in conventional networks. Channel capacities in PCNs are discrete in nature, and are reduced by the amount required by the transaction upon each successful transaction. The residual capacity may not be sufficient to accommodate future transactions. Consequently, existing analytical techniques such as queueing theory cannot be applied.

In this paper, we first develop a new analytical model to theoretically compute performances and the performance gap over scheduling transactions in a PCN. We then proceed to study the effects of transactions by exploring the computation results of performances during the scheduling process. Further, by computing the performance gap in a PCN with different channel capacities, we can learn the impact of channel capacities on the performance gap.

Our main contributions are as follows.

- To our best knowledge, this is the first systematic study on the performance of payment channel networks. This provides a theoretical framework for characterizing the gap between the theoretically optimal performance and the performance achievable in practice of PCNs, where the effects of channel capacities and transactions are captured.
- We distinguish transactions into two categories, with distinct effects on the performance in a PCN. Based on their impact on the performance, we obtain conditions on selecting transactions to relay at channels during the scheduling process to approximate the optimal performance.
- We examine the effects on the performance gap from channel capacities. Specifically, the performance gap can first increase with higher channel capacities, then decrease after channel capacities are larger than a certain value. We believe this can provide insights on adjusting channel capacities to reduce the performance gap in a PCN.

2 Background

In popular payment channel networks (PCNs) [1], [2], [3], [4], [5], [6], two users can open a bidirectional channel, where a specific number of coins, denoted as initial *channel capacities*, are jointly charged from these two users through an on-chain transaction. Through that channel, transactions between these two users can be confirmed immediately in an off-chain fashion by consuming the capacity of that channel without being validated by others in the network, as long as the capacity of that channel is larger than the number of coins requested by that transaction. After all of those transactions are confirmed, the capacity of a channel decreases by the same amount.

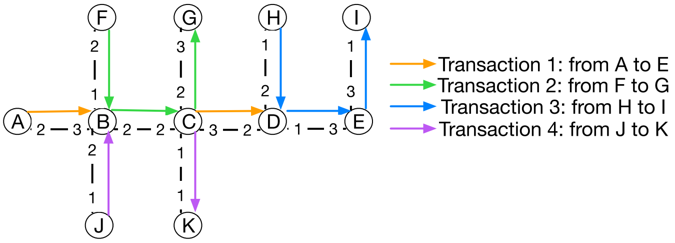

F
G
H
I
2
1
3
2
1
2
1
3
A
2
3
B
2
2
C
3
2
D
1
3
E
2
1
1
1
J
K
Transaction 1: from A to E
Transaction 2: from F to G
Transaction 3: from H to I
Transaction 4: from J to K

Fig. 1. A PCN example to illustrate the gap between the theoretical optimality and the performance achievable in practice.

Further, a transaction between two users who have not opened a channel can also be confirmed immediately as follows. Those two users establish a connection through a set of intermediate channels, which then relay this transaction to the destination. Namely, by consuming capacities of these intermediate channels, this transaction is confirmed shortly.

As illustrated in Fig. 1, suppose A needs to send 2 coins to C through two channels: (A, B) and (B, C). A will pay B first through the channel (A, B), then B pays C through the channel (B, C). After these payments, the capacities in both of these two channels decrease from 2 to 0. Without subsequent charging operations, which needs to be validated by all miners through on-chain transactions, these two channels cannot relay any further transactions.

For the favor of the security, the capacities of channels and the arrival of transactions shall be kept private in PCNs, with the help of the HTLC contract [17] and the onion routing protocol [18]. Each intermediate channel only knows the amount of capacities that it is requested for. Further, an intermediate channel does not know the sources, destinations, and the routing information of its transactions, and does not know capacities of other intermediate channels.

With such privacy requirements, the performance achievable in practice of a PCN, in terms of *the number of coins that are transferred*, can hardly be the theoretically optimal performance. If capacities of multiple intermediate channels are not sufficient to support all the transactions, scheduling is performed to select a subset of those transactions to be satisfied by those intermediate channels. To obtain the theoretically optimal performance, such a scheduling process must be conducted by combining the global channel capacities and transactions together. Unfortunately, such global information is kept private in PCNs due to their privacy requirements. Thus, in real-world implementations without such information, it is unlikely for the theoretically optimal subset of transactions to be obtained.

In Fig. 1, there are four transactions, and transaction 1 requests 2 coins, while each of the remaining three transactions requests one coin. Specifically, two different channels, i.e., (B, C) and (D, E), receive requests from these four transactions, while not having sufficient capacities to satisfy all of them. Scheduling is performed on (B, C) and (D, E). If all channel capacities and transactions are known, to obtain the theoretically optimal performance, channel (B, C) selects

TABLE 1
Key Notations

c_{uv}	Capacity of channel (u, v).
DTs or RTs	Dominant transactions or regular transactions
d_i or r_j	Dominant transaction i or regular transaction j
a_i	The number of coins requested by d_i
a_j	The number of coins requested by r_j
m_i	The number of channels that d_i competes for
SRT_{uv} orSDT_{uv}	The number of coins requested by successful DTs or RTs from (u, v)
SRT	The set of successful RTs
SDT	The set of successful DTs
n_{uv}	The number of transactions requesting (u, v)
nRT_{uv} or nDT_{uv}	The number of coins requested by DTs or RTs from (u, v)
ΔP	The difference between any two performances
P^*	the theoretically optimal performance
R	The upper bound of the performance gap

transaction 2 and 4, and channel (D, E) selects transaction 3, and 3 coins can be transferred. However, without such information, if both of those two channels (B, C) and (D, E) select transaction 1, only transaction 1 can succeed.

Obviously, transactions affect the performance achievable in practice. In our example, if transaction 1 succeeds, the theoretically optimal performance cannot be achieved. Further, the gap between the theoretically optimal performance and the performance achievable in practice is affected by channel capacities. For example, if the capacity of (B, C) is 1, transaction 1 cannot succeed. All performances achievable in practice and the theoretically optimal performance are 3, i.e., the upper bound of the performance gap is 0.

However, existing works only focus on designing new algorithms on scheduling transactions or adjusting channel capacities, without thoroughly analyzing the gap between the theoretically optimal performance and the performance achievable in practice. It remains unclear what such a gap and the theoretically optimal performance are in a PCN. Further, they are not characterized without examining channel capacities and transactions.

We believe such a theoretical analysis is important. First, the gap between theoretical optimality and the performance achievable in practice can characterize the design space in a PCN for scheduling transactions. For example, if the maximum gap between the theoretically optimal performance and the performance achievable in practice is small, improvements on the performance are limited by optimizing the scheduling algorithm. Second, the impact of key network parameters provides insights on how to design a PCN, such as dynamically adjusting channel capacities, so that the performance gap can be reduced.

3 Assumptions and Modeling

In this section, we first present key assumptions used in the full course of our analyses, and classify transactions into two categories based on the number of channels they compete for. We then introduce some mathematical notations, and model all probable performances using these notations. Important notations are summarized in Table 1.

3.1 Assumptions and Notations

Following the convention of bursty arrivals in typical networks [25], [26], [27], we consider the case in which transactions arrive at their intermediate channels in a PCN at the same time. Capacities requested by these concurrent transactions are, of course, not the same. A channel may receive relaying requests from multiple transactions. Namely, it is competed for by these transactions. However, it may not have sufficient capacities to relay all of these transactions. Thus, it must select a subset of these transactions to relay. The performance achievable in practice depends on the subsets of transactions selected by all intermediate channels. As discussed before, the performance achievable in practice can hardly achieve theoretical optimality with privacy requirements. Based on this assumption, in this paper, we analyze the fundamental performance limits and key factors involved, i.e., channel capacities and transactions in a PCN.

A PCN is modeled as a directed graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$. $\mathcal{V}$ is the set of nodes, and $\mathcal{E}$ is the set of edges. Each edge $(u, v) \in \mathcal{E}$ is a channel from u to v, and c_{uv} is the capacity of (u, v).

To capture the impact on the fundamental performance limits from these transactions, we classify them into two categories. A transaction competing for more than one channel with others is called a *dominant transaction* (DT). Otherwise, it is referred to as a *regular transaction* (RT). In Fig. 1, transaction 1 competes for channel (B, C) and (D, E) with the remaining three transactions. Transaction 2 only competes for channel (B, C) with transaction 1. Transaction 3 only competes for (D, E), and transaction 4 only competes for (B, C). Thus, transaction 1 is a DT, while transaction 2, 3 and 4 are RTs. This classification is useful, as we will show later that DTs and RTs have different impacts on the performance.

Consider two arbitrary sets of DTs and RTs, $\{d_1, \ldots, d_i, \ldots\}$ and $\{r_1, \ldots, r_j, \ldots\}$, requesting coins from channels in $\mathcal{G}$. Specifically, a DT d_i requests a_i coins from each intermediate channel, and competes for m_i intermediate channels with others. A RT d_j requests for a_j coins from each of its intermediate channels, and only competes for *one* intermediate channel at most.

We now denote the performances and related parameters in a PCN. P is the performance that can be achieved in practice, i.e., the number of coins requested by successful transactions. The sets of successful DTs and RTs are SDT and SRT. The numbers of coins requested by transactions in SDT and SRT from (u, v) are SDT_{uv} and SRT_{uv}. The amounts of capacities requested by DTs and RTs from channel (u, v) are nRT_{uv} and nDT_{uv}. The number of transactions requesting for the relaying service from the channel (u, v) is n_{uv}. The difference between any two actual performances is ΔP. To put into one extreme, the maximum value of P is the theoretically optimal performance, denoted as P^*. On another extreme, the minimum value of P is the worst performance, denoted as $\bar{P}$. And we denote the upper bound of the gap between the theoretically optimal performance and the performance achievable in practice as R.

3.2 Modeling Performance

We now model the difference between any two actual performances, ΔP, with regard to channel capacities and transactions. This is essential in the understanding of the gap between the theoretically optimal performance and the performance achievable in practice, and we will analyze such a gap based on this model in later sections.

Lemma 1. *The difference between any two actual performances, ΔP, can be formulated according to how* DT*s and* RT*s compete for channels as follows:*

$$\begin{aligned}\Delta P = P_2 - P_1 = &\sum_{n_{uv}\geq 2}(\Delta\mathrm{SDT}_{uv} + \Delta\mathrm{SRT}_{uv})\\ &+ \sum_{d_i\in\mathrm{SDT}_1 d_i \notin \mathrm{SDT}_2}(m_i-1)a_i - \sum_{d_i\in\mathrm{SDT}_2 d_i \notin \mathrm{SDT}_1}(m_i-1)a_i.\\ \Delta\mathrm{SRT}_{uv} &= \mathrm{SRT}_{uv2} - \mathrm{SRT}_{uv1}.\\ \Delta\mathrm{SDT}_{uv} &= \mathrm{SDT}_{uv2} - \mathrm{SDT}_{uv1},\end{aligned} \tag{1}$$

where P_1 and P_2 are any two performances achievable in real implementations of a PCN. Specifically, SDT_1 *and* SDT_2 *are sets of successful DTs corresponding to P_1 and P_2. The numbers of coins requested by successful DTs from channel (u, v) are* SDT_{uv1} *and* SDT_{uv2}*. Similarly,* SRT_{uv1} *and* SRT_{uv2} *represent the numbers of coins requested by successful RTs from (u, v).*

Proof. As discussed before, the performance varies during the scheduling of transactions at channels that receive multiple relaying requests. For each channel (u, v) that is requested by multiple transactions, the difference in the number of coins it successfully transfers, is

$$\Delta T_{uv} = \Delta\mathrm{SDT}_{uv} + \Delta\mathrm{SRT}_{uv}.$$

For channels that are competed for by multiple transactions

$$\Delta T = \sum_{n_{uv}\geq 2}\Delta T_{uv} = \sum_{n_{uv}\geq 2}\Delta\mathrm{SDT}_{uv} + \Delta\mathrm{SRT}_{uv}. \tag{2}$$

However, ΔT repeatedly accumulate numbers of coins requested by DTs for X times. For each d_i belonging to SDT_1 or SDT_2, its contribution to the difference in performances must have been calculated for m_i times. Thus, we have

$$X = \sum_{d_i\in\mathrm{SDT}_2 d_i \notin \mathrm{SDT}_1}(m_i-1)a_i - \sum_{d_i\in\mathrm{SDT}_1 d_i \notin \mathrm{SDT}_2}(m_i-1)a_i. \tag{3}$$

Combine (2) and (3), we have

$$\begin{aligned}\Delta P = \Delta T - X = &\sum_{n_{uv}\geq 2}(\Delta\mathrm{SDT}_{uv} + \Delta\mathrm{SRT}_{uv})\\ &+ \sum_{d_i\in\mathrm{SDT}_1 d_i \notin \mathrm{SDT}_2}(m_i-1)a_i - \sum_{d_i\in\mathrm{SDT}_2 d_i \notin \mathrm{SDT}_1}(m_i-1)a_i.\\ \Delta\mathrm{SRT}_{uv} &= \mathrm{SRT}_{uv2} - \mathrm{SRT}_{uv1}\\ \Delta\mathrm{SDT}_{uv} &= \mathrm{SDT}_{uv2} - \mathrm{SDT}_{uv1}.\end{aligned} \tag{4}$$

□

Example 1. In Fig. 1, transaction 1 is a DT competing for (B, C) and (D, E), i.e., $m_1 = 2$. When transaction 1 is successful, and none of the remaining three transactions succeed, the performance equals 2, i.e., $P_1 = 2$,

$$\begin{aligned}&\mathrm{SRT}_1 = \emptyset, \mathrm{SDT}_1 = \{T_1\},\\ &\mathrm{SRT}_{\mathrm{BC1}} = |\emptyset| = 0,\ \mathrm{SDT}_{\mathrm{BC1}} = |\{T_1\}|a_1 = 2\\ &\mathrm{SRT}_{\mathrm{DE1}} = |\emptyset| = 0,\ \mathrm{SDT}_{\mathrm{DE1}} = |\{T_1\}|a_1 = 2.\end{aligned}$$

When only transaction 2, 3 and 4 are successful, the performance equals 3, i.e., $P_2 = 3$

$$\begin{aligned}&\mathrm{SRT}_2 = \{T_2, T_3, T_4\}, \mathrm{SDT}_2 = \emptyset\\ &\mathrm{SRT}_{\mathrm{BC2}} = |\{T_2, T_4\}| = 2, \mathrm{SDT}_{\mathrm{BC2}} = |\emptyset| = 0\\ &\mathrm{SRT}_{\mathrm{DE2}} = |\{T_3\}| = 1, \mathrm{SDT}_{\mathrm{DE2}} = |\emptyset| = 0.\end{aligned}$$

We have

$$\begin{aligned}&\Delta\mathrm{SDT}_{\mathrm{BC}} = -2, \Delta\mathrm{SRT}_{\mathrm{BC}} = 2,\\ &\Delta\mathrm{SDT}_{\mathrm{DE}} = -2, \Delta\mathrm{SRT}_{\mathrm{DE}} = 1.\\ &\quad \Delta T = -1, X = -2, \Delta P = \Delta T - X = 1.\end{aligned}$$

We explore the relationship between $\Delta\mathrm{SRT}_{uv}$ and $\Delta\mathrm{SDT}_{uv}$ in Lemma 2 as follows:

Lemma 2. $\Delta\mathrm{SRT}_{uv}$ *is formulated with respect to* SDT_1 *and* SDT_2

$$\begin{cases}\Delta\mathrm{SRT}_{uv} & = 0, \text{if } \{\mathrm{SDT}_{uv1}, \mathrm{SDT}_{uv2}\} \leq c_{uv} - \mathrm{nRT}_{uv}\\ \Delta\mathrm{SRT}_{uv} & \geq \mathrm{nRT}_{uv} - c_{uv} + \mathrm{SDT}_{uv1},\\ & \text{if } \mathrm{SDT}_{uv1} \geq c_{uv} - \mathrm{nRT}_{uv} \geq \mathrm{SDT}_{uv2}\\ \Delta\mathrm{SRT}_{uv} & \leq c_{uv} - \mathrm{SDT}_{uv2} - \mathrm{nRT}_{uv},\\ & \ \text{if } \mathrm{SDT}_{uv2} \geq c_{uv} - \mathrm{nRT}_{uv1} \geq \mathrm{SDT}_{uv1}\\ |\Delta\mathrm{SRT}_{uv}| & \leq |\Delta\mathrm{SDT}_{uv}|,\ \text{else.}\end{cases}$$

Proof. There are two circumstances on SRT_{uv}. *First,* (u, v) is able to support all the RTs and transactions in SDT: $\mathrm{SDT}_{uv} \leq c_{uv} - \mathrm{nRT}_{uv}$. In another word: $\mathrm{SRT}_{uv} = \mathrm{nRT}_{uv}$. *Second,* (u, v) can only support part of RTs and transactions in SDT: $c_{uv} - \mathrm{SDT}_{uv} \leq \mathrm{nRT}_{uv}$. In another word: $\mathrm{SRT}_{uv} \leq c_{uv} - \mathrm{SDT}_{uv}$. Thus, we have

$$\begin{cases}\Delta\mathrm{SRT}_{uv} = & 0, \text{if } \{\mathrm{SDT}_{uv}, \mathrm{SDT}'_{uv}\} \leq c_{uv} - \mathrm{nRT}_{uv},\\ \Delta\mathrm{SRT}_{uv} \geq & \mathrm{nRT}_{uv} - c_{uv} + \mathrm{SDT}_{uv},\\ & \text{if } \mathrm{SDT}_{uv} \geq c_{uv} - \mathrm{nRT}_{uv} \geq \mathrm{SDT}'_{uv},\\ \Delta\mathrm{SRT}_{uv} \leq & c_{uv} - \mathrm{SDT}'_{uv} - \mathrm{nRT}_{uv},\\ & \ \text{if } \mathrm{SDT}'_{uv} \geq c_{uv} - \mathrm{nRT}_{uv} \geq \mathrm{SDT}_{uv}.\end{cases}$$

Then we formulate $\Delta\mathrm{SRT}_{uv}$ when

$$\{\mathrm{SDT}_{uv}, \mathrm{SDT}'_{uv}\} \geq c_{uv} - \mathrm{nRT}_{uv}. \tag{5}$$

Suppose d_i requests (u, v), $\mathrm{SDT}' \setminus \mathrm{SDT} = \{d_i\}$. Obviously, SRT_{uv} exceeds SRT'_{uv} by $|\Delta\mathrm{SDT}_{uv}| = a_i$ coins at most: $|\Delta\mathrm{SRT}_{uv}| \leq |\Delta\mathrm{SDT}_{uv}|$. Combine (5), we have (??). □

4 Understanding the Theoretically Optimal Performance

We now proceed to analyze the theoretically optimal performance P^* in several steps. *First,* what is the maximum number of coins that can be transferred successfully in a PCN? *Second,* based on this formulation, is it probable to achieve the theoretically optimal performance in practice but obviating the need for full knowledge on capacities and transactions of all channels? *Third*, how does the theoretically optimal performance correlate with channel capacities and transactions, i.e., DTs and RTs?

We formulate on the theoretically optimal performance and conditions on achieving the theoretically optimal performance as follows. L is the set of channels, whose capacities are larger than the number of coins requested by RTs.

Theorem 1. *To achieve the theoretically optimal performance, RTs can be relayed by all channels, while DTs can only be relayed by channels belonging to L. In this condition, the theoretically optimal performance P^* of a PCN can be computed as follows:*

$$P^* = \sum_{n_{uv} \geq 2 (u,v) \notin L} c_{uv} + \sum_{n_{u'v'} \geq 2\$ u',v') \in L} \mathrm{nRT}_{u'v'} + \sum_{d_i \in \mathrm{SDT}^*} a_i.$$

Proof. We first prove that P^* is achieved when DTs are only relayed by channels belonging to L, i.e., SDT^* consists of DTs that only request channels belonging to L. Consider another set of successful DTs, denoted as SDT. If the performance P corresponding to SDT is not larger than P^*, the theorem holds. Trivially, only channels competed for by multiple transactions are needed to be considered. Based on Lemma 2, the gap ΔP between the performances corresponding to SDT^* and SDT is

$$\begin{aligned}\Delta P = & \sum_{d_i \in \mathrm{SDT} d_i \notin \mathrm{SDT}^*} (m_i - 1) a_i - \sum_{d_i \in \mathrm{SDT}^* d_i \notin \mathrm{SDT}} (m_i - 1) a_i \\ & + \sum_{(u,v) \in L n_{uv} \geq 2} (\mathrm{sDT}^*_{uv} + \mathrm{nRT}_{uv} - c_{uv}).\end{aligned} \tag{6}$$

Based on Lemma 2, we have

$$\sum_{(u,v) \in L n_{uv} \geq 2} \mathrm{sDT}^*_{uv} - \sum_{d_i \in \mathrm{SDT}^* d_i \notin \mathrm{SDT}} (m_i - 1) a_i \geq \sum_{d_i \in \mathrm{SDT}^*} a_i. \tag{7}$$

$$\mathrm{nRT}_{u'v'} - c_{u'v'} \geq -\mathrm{sDT}_{u'v'}, \forall (u', v') \in L. \tag{8}$$

$$\forall d_i \in \mathrm{SDT}^*, d_i \in X. \tag{9}$$

Let X be the set of DTs requesting channels in L, i.e., $\mathrm{SDT}^* \subseteq X$. Let $X' = \mathrm{SDT} \cap X$. Combine (6) (7), and (8)

$$\begin{aligned}\Delta P = & \sum_{d_i \in \mathrm{SDT} d_i \notin \mathrm{SDT}^*} (m_i - 1) * a_i - \sum_{d_i \in \mathrm{SDT}^* d_i \notin \mathrm{SDT}} (m_i - 1) * a_i \\ & + \sum_{(u,v) \in L n_{uv} \geq 2} (\mathrm{sDT}^*_{uv} + \mathrm{nRT}_{uv} - c_{uv}) \geq \sum_{d_i \in \mathrm{SDT} d_i \notin \mathrm{SDT}^*} (m_i - 1) a_i \\ & + \sum_{(u,v) \in L n_{uv} \geq 2} (\mathrm{nRT}_{uv} - c_{uv}) + \sum_{d_i \in \mathrm{SDT}^*} a_i \\ \geq & \sum_{d_i \in \mathrm{SDT} d_i \notin \mathrm{SDT}^*, X'} (m_i - 1) a_i + \sum_{d_i \in \mathrm{SDT}, X' d_i \notin \mathrm{SDT}^*} (m_i - 1) a_i \\ & + \sum_{(u,v) \in L n_{uv} \geq 2} (\mathrm{nRT}_{uv} - c_{uv}) + \sum_{d_i \in \mathrm{SDT}^*} a_i \\ \geq & \sum_{d_i \in \mathrm{SDT} d_i \notin \mathrm{SDT}^*, X'} (m_i - 1) a_i + \sum_{d_i \in \mathrm{SDT}, X' d_i \notin \mathrm{SDT}^*} 1 + \sum_{d_i \in \mathrm{SDT}^*} a_i > 0.\end{aligned}$$

Then we prove the formulation on P^*. The number of successful RTs in the theoretically optimal case, noted as N

$$N = \sum_{(u,v) \notin L n_{uv} \geq 2} \mathrm{sRT}_{uv} + \sum_{(u',v') \in L n_{u'v'} \geq 2} \mathrm{sRT}_{u'v'}. \tag{10}$$

Based on (8), we have

$$\mathrm{sRT}_{u'v'} = \min\{\mathrm{nRT}_{u'v'}, c_{u'v'} - \mathrm{sDT}_{u'v'}\} = \mathrm{nRT}_{u'v'}, \tag{11}$$

Obviously, we have: $\mathrm{sRT}_{uv} = c_{uv}$. With (8), (11), (10) and (11), the theoretically optimal performance P^* is: $\sum_{(u,v) \notin L n_{uv} \geq 2} c_{uv} + \sum_{(u',v') \in l n_{u'v'} \geq 2} \mathrm{nRT}_{u'v'} + \sum_{d_i \in \mathrm{SDT}^*} a_i$. Let $\mathrm{SDT} \setminus \mathrm{SDT}^* = Y$, $\bar{P} = \min(P^* - \Delta P) = \min\ P$ is

$$P^* - \Delta P = \sum_{n_{uv} \geq 2} c_{uv} - \sum_{d_i \in X'} a_i (m_i - 2) - \sum_{d_i \in SDT \setminus \mathrm{SDT}^*} a_i m_i. \tag{12}$$

Based on our formulation and conditions on the theoretically optimal performance, we can now present Algorithm 1 on scheduling transactions for maximizing the performance, with the expense of exposing limited information on intermediate relaying channels. In Algorithm 1, each channel without sufficient capacities schedules transactions independently without a centralized server. According to Theorem 1, SDT^* can be derived by enumerating channels in L as in Line 11 to 13 of Algorithm 1. Then, the remaining capacity in each channel is occupied by SRT^*. The algorithm needs to select a subset of RTs to maximize the number of coins requested by RTs without exceeding the remaining capacity of each channel as in Line 14 to Line 20. □

Algorithm 1. Scheduling Transactions to Achieve the Theoretically Optimal Performance

```
1:  procedure TransScheduling
2:      Input: The set of intermediate channels $E_t$ competed for by each transaction $t$, the set of transactions $F$, the set of all intermediate channels $\mathcal{E}$, the capacity of each channel $c_{uv}$, the number of coins transferred by each transaction $a_t$. the sets of RTs requesting $(u,v)$: $\mathrm{RT}_{uv}$.
3:      Output: The set of transactions to be relayed $S$.
4:      for each $(u,v)$ of $\mathcal{E}$ do                ▷Get $L$.
5:        $\mathrm{nRT}_{uv} \leftarrow 0$
6:        if $n_{uv} \geq 2$ then
7:          for each $t$ in $\mathrm{SRT}_{uv}$ do
8:            $\mathrm{nRT}_{uv} \leftarrow \mathrm{nRT}_{uv} + a_t$
9:          if $c_{uv} \geq \mathrm{nRT}_{uv}$ then
10:           $L \leftarrow L \cup (u,v)$,
11:     for each DT $t$ in $F$ do ▷Get $\mathrm{SDT}^*$ based on $L$.
12:       if $\forall (u,v) \in E_t, (u,v) \in L$ and $c_{uv} \geq a_t$ then
13:         $\mathrm{SDT}^* \leftarrow \mathrm{SDT}^* \cup t, c_{uv} \leftarrow c_{uv} - a_t$
14:     for each $(u,v)$ of $\mathcal{E}$ do                ▷Get $\mathrm{SRT}^*$.
15:       if $n_{uv} \geq 2$ then
16:         if $c_{uv} \leq \mathrm{nRT}_{uv}$ then
17:               Select a subset of RTs, denoted as $s_{uv}$, from those requesting $(u,v)$, so that the sum of coins requested by RTs belonging to $s_{uv}$ is maximized without exceeding $c_{uv}$.   ▷A Dynamic Programming Problem.
18:             $\mathrm{SRT}^*_{uv} \leftarrow \mathrm{SRT}^*_{uv} \cup s_{uv}$
19:         else
20:             $\mathrm{SRT}^*_{uv} \leftarrow \mathrm{SRT}^*_{uv} \cup \mathrm{RT}_{uv}$   ▷All the RTs in $(u,v)$ can be supported.
    $\mathrm{SRT}^* \leftarrow \mathrm{SRT}^* \cup \mathrm{SRT}^*_{uv} S \leftarrow \mathrm{SRT}^* \cup \mathrm{SDT}^*$
```

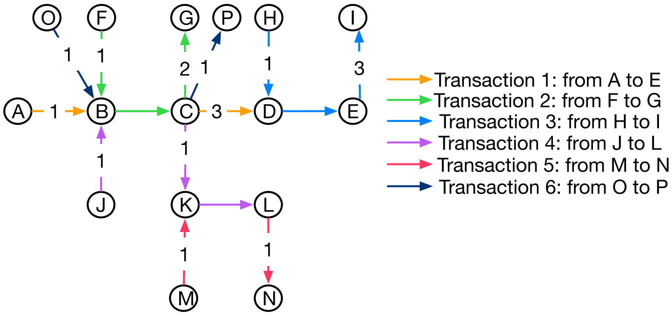

O
F
G
P
H
I
A
B
C
D
E
J
K
L
M
N
1
1
2
1
1
3
1
3
1
1
1
1
Transaction 1: from A to E
Transaction 2: from F to G
Transaction 3: from H to I
Transaction 4: from J to L
Transaction 5: from M to N
Transaction 6: from O to P

Fig. 2. An example illustrating the maximum gap between the theoretically optimal performance and the performance in practice.

Example 2. The PCN in Fig. 2 can be used as an example to illustrate Theorem 1. There are two DTs and four RTs requesting one coin at the same time. Table 2shows five sets of capacities of channels $(\mathrm{B},\mathrm{C}),(\mathrm{D},\mathrm{E})$ and (K,L), being competed for by multiple transactions. We record all the four probable performances in Table 2, together with corresponding SDTs. The circled numbers are the theoretically optimal performances under different channel capacities.

Due to space constraints, we only take a set of channel capacities, $c_{\mathrm{D}E}=1, c_{\mathrm{B}C}=4, c_{\mathrm{K}L}=2$ to illustrate how the algorithm works. Initially, we have $L=\emptyset$. In the iteration from Line 4 to 10, the channels (B,C) and (K,L) are added into L in Line 9, i.e., $L=\{(\mathrm{B},\mathrm{C}),(\mathrm{K},\mathrm{L})\}$. Then the iteration from Line 11 to 13 will select the theoretically optimal set of successful DTs, where transaction 4 is added to SDT^*, i.e., $\mathrm{SDT}^*=\{t_4\}$. Then, the iteration from Line 14 to 20 selects successful RTs in each channel. Combining SDT^* and channel capacities, the result in this iteration is, $\mathrm{SRT}^*_{\mathrm{BC}}=\{t_2,t_6\}, \mathrm{SRT}^*_{\mathrm{DE}}=\{t_3\}$, and $\mathrm{SRT}^*_{\mathrm{KL}}=\{t_5\}$. Thus, the theoretically optimal performance is 5.

We derive effects of channel capacities on the theoretically optimal performance in Corollary 1 and 2. Corollary 3 further shows how the optimality relates with transactions.

Corollary 1. *For any channel $(s,t)\notin L$ increasing its capacity from c_{st} to c'_{st}, which still cannot relay all RTs, the theoretically optimal performance increases linearly*

$$\text{If } c'_{st} > c_{st} > \mathrm{nRT}_{st} \text{ and } (s,t)\notin L,\ P^*(c'_{st}) > P^*(c_{st}).$$

Proof. Since $(s,t)\notin L$, we have

$$\begin{aligned} P^*(c'_{st})-P^*(c_{st}) = & \sum_{n_{uv}\geq 2(u,v)\notin L(u,v)\neq(s,t)} c_{uv}+c'_{st} \\ & -\sum_{n_{uv}\geq 2(u,v)\notin L(u,v)\neq(s,t)} c_{uv} \\ & -c_{st} = c'_{st}-c_{st} > 0. \end{aligned}$$

□

Corollary 2. *The increase of $c_{u'v'}$ will not lead to a higher theoretically optimal performance, if there is only (u',v') whose capacity is already larger than the number of coins requested by RTs from (u',v'), i.e., $|L|=|\{u',v'\}|$.*

TABLE 2
Performances Under Different Channel Capacities in Fig. 2

$\mathrm{nRT}_{\mathrm{BC}}=2,\mathrm{nRT}_{\mathrm{DE}},\mathrm{nRT}_{\mathrm{KL}}=1$				
	SDT			
	$\emptyset$	$\{t_1\}$	$\{t_4\}$	$\{t_1,t_4\}$
$c_{\mathrm{B}C}=4,c_{\mathrm{D}E}=1,c_{\mathrm{K}L}=2$	$\emptyset$	$\emptyset$	⑤	5
$c_{\mathrm{B}C}=4,c_{\mathrm{D}E}=2,c_{\mathrm{K}L}=2$	$\emptyset$	$\emptyset$	$\emptyset$	⑥
$c_{\mathrm{B}C}=4,c_{\mathrm{D}E},c_{\mathrm{K}L}=1$	④	4	4	4
$c_{\mathrm{BC}}=3,c_{\mathrm{D}E},c_{\mathrm{K}L}=1$	④	4	4	3
$c_{\mathrm{B}C}=2,c_{\mathrm{D}E},c_{\mathrm{K}L}=1$	④	3	3	2
$c_{\mathrm{B}C},c_{\mathrm{D}E},c_{\mathrm{K}L}=1$	③	2	2	$\emptyset$

Proof. In this situation, we have $L=\{(u',v')\}$. Since DTs compete for more than one channel, we have: $\mathrm{SDT}^*=\emptyset$. Thus, the theoretically optimal performance is: $P^*=\sum_{n_{uv}\geq 2(u,v)\neq(u',v')} c_{uv}+\mathrm{nRT}_{u'v'}$. Hence, the theoretically optimal performance is not related to the capacity of $c_{u'v'}$. □

Example 3. We now use Fig. 2 to briefly explain Corollary 2. There are three channels (B, C), (D, E), (K, L) that are competed for by multiple transactions: $\mathrm{nRT}_{\mathrm{BC}}=2,\ \mathrm{nRT}_{\mathrm{DE}}=1,\ \mathrm{nRT}_{\mathrm{KL}}=1$. Consider the following sets of channel capacities in the third, fourth, and fifth lines in Table 2. The capacity of (B, C) ranges from $[2,4]$, while the capacities of others remain the same, i.e., $L=\{(\mathrm{B},\mathrm{C})\}$. It is easy to find that, the theoretically optimal performance does not change.

Corollary 3. *Given fixed channel capacities, the theoretically optimal performance is positively related to the amount of capacities requested by RTs*

$$\begin{aligned} P_1^*(\ldots,\mathrm{nRT}'_{uv},\mathrm{nDT}'_{uv}) &> P_2^*(\ldots,\mathrm{nRT}'_{uv},\mathrm{nDT}'_{uv}), \\ \mathrm{nRT}'_{uv} &> \mathrm{nRT}_{uv} \text{ s.t.} n_{uv} > 2, \\ \mathrm{nRT}'_{uv}+\mathrm{nDT}'_{uv} &= \mathrm{nRT}_{uv}+\mathrm{nDT}_{uv} \text{ s.t.} n_{uv} > 2. \end{aligned}$$

Proof. Consider theoretically optimal performances P_1^* and P_2^* corresponding to different sets of RTs or DTs. There are two circumstances with respect to the number of coins requested by RTs. *First*, the capacity of each channel cannot support RTs requesting for it: $\forall(u,v)\notin L$ and $n_{uv}>2$, we have $\mathrm{nRT}'_{uv}>c_{uv}$. Based on Theorem 1, the theoretically optimal set of DTs in this case SDT_1^* is $\emptyset$, and the theoretically optimal performance is P_1^*. *Second*, there is a set of channels L whose capacities can relay all the RTs

$$\begin{aligned} &\exists(u,v),(u,v)\notin L. \\ &\mathrm{nRT}_{uv}\leq c_{uv} < \mathrm{nRT}'_{uv} \text{ s.t. } n_{uv} > 2. \end{aligned} \tag{13}$$

Since the total channel capacities in these two circumstances are the same, RTs request fewer coins in this scenario. Denote the set of successful DTs as SDT_2^*, and the theoretically optimal performance is P_2^*, we have

$$\begin{aligned} P_2^* &= \sum_{(u,v)\notin L n_{uv}\geq 2} c_{uv} + \sum_{(u',v')\in L n_{u'v'}\geq 2} \mathrm{nRT}_{u'v'} + \sum_{d_i\in \mathrm{SDT}_2^*} a_i \\ &= \sum_{(u,v)\notin L n_{uv}\geq 2} \mathrm{sRT}_{uv} + \sum_{(u',v')\in L n_{u'v'}\geq 2} \mathrm{sRT}_{u'v'} + \sum_{d_i\in \mathrm{SDT}_2^*} a_i \\ &\leq \sum_{(u,v)\notin L n_{uv}\geq 2} \mathrm{sRT}_{uv} + \sum_{(u',v')\in L n_{u'v'}\geq 2} (\mathrm{sRT}_{u'v'} + \mathrm{sDT}_{u'v'}) \\ &\leq \sum_{(u,v)\notin L n_{uv}\geq 2} \mathrm{sRT}_{uv} + \sum_{(u',v')\in L n_{u'v'}\geq 2} c_{u'v'} \\ &\leq \sum_{(u,v)\notin L n_{uv}\geq 2} c_{uv} + \sum_{(u',v')\in L n_{u'v'}\geq 2} c_{u'v'} + \sum_{d_i\in \mathrm{SDT}_1^*} a_i \\ &\leq \sum_{(u,v)\notin L n_{uv}\geq 2} c_{uv} + \sum_{(u',v')\in L n_{u'v'}\geq 2} \mathrm{nRT}'_{u'v'} + \sum_{d_i\in \mathrm{SDT}_1^*} a_i = P_1^*. \end{aligned}$$

PCNs with the topological and transaction workload characteristics that RTs request a great number of coins are preferred for the theoretically optimal performance. □

In Fig. 2, suppose transaction 4 now starts from K and ends at L, i.e., there is one more RT. Consider the set of channel capacities in the first line of Table 2, the theoretically optimal performance increases to 6 now, where all transactions can succeed.

5 A Study on the Performance Gap

We now analyze the maximum gap between the theoretically optimal performance and the performance achievable in practice, R, with regard to channel capacities and transactions. Based on Lemma 1, R is formulated as follows:

$$\begin{aligned} R = \max\Delta P = \max\{ &\sum_{n_{uv}\geq 2} (\Delta\mathrm{sDT}_{uv} + \Delta\mathrm{sRT}_{uv}) \\ &+ \sum_{d_i\in \mathrm{SDT} d_i\notin \mathrm{SDT}'} (m_i - 1)a_i - \sum_{d_i\in \mathrm{SDT}' d_i\notin \mathrm{SDT}} (m_i-1)a_i. \end{aligned}$$

Corollary 4 *With fixed transactions, R is positively correlated with the capacity of channel (u, v), if (u, v) satisfies $c_{uv} \leq \mathrm{nRT}_{uv}$, i.e.,*

$$R(c'_{uv}) \geq R(c_{uv}), \text{if } \mathrm{nRT}_{uv} \geq c'_{uv} \geq c_{uv}. \quad (14)$$

Proof. We first formulate R between the theoretically optimal and the performance achievable in practice. SDT is the set of successful DTs corresponding to $\bar{P}$. Note that the capacity of each channel is not larger than the number of coins requested by RTs from that channel in this circumstance. With Theorem 1, $L = \emptyset$. Thus, the theoretically optimal set of successful DTs is empty, i.e., $\mathrm{SDT}^* = \emptyset$. R is

$$\begin{aligned} R = &\sum_{d_i\in \mathrm{SDT} d_i\notin \mathrm{SDT}^*} (m_i-1)a_i - \sum_{d_i\in \mathrm{SDT}^* d_i\notin \mathrm{SDT}} (m_i-1)a_i \\ &+ \sum_{n_{uv}\geq 2} (\Delta\mathrm{sDT}_{uv} + \Delta\mathrm{sRT}_{uv}) \\ = &\sum_{d_i\in \mathrm{SDT}} (m_i-1)a_i + \sum_{n_{uv}\geq 2} (\Delta\mathrm{sDT}_{uv} + \Delta\mathrm{sRT}_{uv}). \end{aligned}$$

Combine Lemma 2, we have

$$R = P^* - \bar{P} = \sum_{d_i\in \mathrm{SDT}} (m_i-1)a_i. \quad (15)$$

We then show that when the capacity of a channel increases, SDT is still a probable set of successful DTs. New successful RTs relayed with more capacities are added into SRT.

The maximum gap between the theoretically optimal performance and the performance achievable in practice is only related to SDT based on (15). Denote the set of successful DTs corresponding to $\bar{P}$ when the capacity of channel is c_{uv} as $\mathrm{SDT}_{c_{uv}}$. With a higher capacity of channel (u', v'), denoted as c'_{uv}, new sets of successful DTs may appear, resulting in new gap, denoted as $R(c'_{uv})$. Since $\mathrm{SDT}(c_{uv})$ still exists, we have $R(c'_{uv}) = \max\{R(c_{uv}), R(c'_{uv})\} \geq R(c_{uv})$. □

In Fig. 2, the channel (B, C) is competed for by transaction 2 and 6, which are all RTs, i.e., $\mathrm{nRT}_{(\mathrm{B,C})} = 2$. In the last two lines of Table 2, the channel capacity of (B, C) increases from 1 to 2, i.e., $\mathrm{nRT}_{\mathrm{BC}} \geq c'_{\mathrm{BC}} > c_{\mathrm{BC}}$, satisfying the condition in (14). Obviously, the maximum gap between the theoretically optimal performance and the performance achievable in practice also grows by 1.

Corollary 5. *With fixed transactions, R is reversely proportional to the capacity of channel (u, v), if (u, v) satisfies $c_{uv} \geq \max\{\mathrm{nRT}_{uv}, \mathrm{nDT}_{uv}\}$, i.e.,*

$$R(c'_{uv}) \leq R(c_{uv}), \text{if } c'_{uv} \geq c_{uv} \geq \max\{\mathrm{nRT}_{uv}, \mathrm{nDT}_{uv}\}.$$

Proof. Let L be the set of channels whose capacities can relay all DTs or RTs. Consider only channels in L change capacities. The sets of channel capacities are C and C' before and after changing. We first formulate R. Let the theoretically optimal set of successful DTs be SDT^*. SDT is the set of successful DTs corresponding to $\bar{P}$. We have the maximum gap between the theoretically optimal performance and the performance achievable in practice $R(C)$

$$\begin{aligned} R(C) = &\sum_{(u,v)\in L n_{uv}\geq 2} (\Delta\mathrm{sDT}_{uv} + \Delta\mathrm{sRT}_{uv}) \\ &+ \sum_{d_i\in \mathrm{SDT} d_i\notin \mathrm{SDT}^*} (m_i-1)a_i \\ &- \sum_{d_i\in \mathrm{SDT}^* d_i\notin \mathrm{SDT}} (m_i-1)a_i. \end{aligned}$$

Let $X' = \mathrm{SDT}^* \setminus \mathrm{SDT}$, and $Y' = \mathrm{SDT} \setminus \mathrm{SDT}^*$. Based on Lemma 2, $\forall (u, v) \in L$

$$\begin{aligned} R(C) = &\sum_{(u,v)\in L n_{uv}\geq 2} (\Delta\mathrm{sDT}_{uv} + \Delta\mathrm{sRT}_{uv}) \\ &+ \sum_{d_i\in Y'} (m_i-1)a_i - \sum_{d_i\in X'} (m_i-1)a_i. \end{aligned}$$

$$\Delta\mathrm{sDT}_{uv} + \Delta\mathrm{sRT}_{uv} = \begin{cases} \Delta\mathrm{sDT}_{uv}, \text{if } c_{uv} - \mathrm{sDT}_{uv} \geq \mathrm{nRT}_{uv}, \\ \mathrm{sDT}^*_{uv} - c_{uv} + \mathrm{nRT}_{uv}, \text{else}. \end{cases} \quad (16)$$

Different from Theorem 4, SDT may not be a probable set of successful DTs when channel capacities increase in this circumstance. If all the RTs have been relayed before the capacity rising, there do not exist new successful RTs in SRT. Thus, SDT may include new DTs, i.e., it cannot represent a set of successful DTs under a new set of channel capacities.

Thus, we discuss whether P decreases if SDT changes. *First*, SDT does not change. Based on the above discussion, we have $c_{uv} - \mathrm{sDT}_{uv} \geq \mathrm{nRT}_{uv}$. We have: $\sum_{(u,v)\in L, n_{uv}\geq 2}(\Delta \mathrm{sDT}^*_{uv} - \Delta c_{uv}) \leq 0$. Thus, the first item of (16) decreases. Further, Y' will not change, while X' includes new DTs. In other words, the third item increases, causing a decrease in $R(C)$. *Second*, SDT changes. From the discussions above, $c_{uv} - \mathrm{sDT}_{uv} \geq \mathrm{nRT}_{uv}$. Combining (7)

$$\begin{aligned} R(C) = & \sum_{(u,v)\in L n_{uv}\geq 2} (\mathrm{sDT}^*_{uv} - \mathrm{sDT}_{uv}) \\ & + \sum_{d_i\in Y'}(m_i-1)a_i - \sum_{d_i \in X'}(m_i-1)a_i \\ = & \sum_{(u,v)\in L n_{uv}\geq 2} \mathrm{sDT}^*_{uv} - \sum_{d_i\in X'}(m_i-1)a_i \\ & - \left(\sum_{(u,v)\in L n_{uv}\geq 2} \mathrm{sDT}_{uv} - \sum_{d_i\in Y'}(m_i-1)a_i \right) = |X'| - |Y'|. \end{aligned}$$

Obviously, $|\Delta X'| \leq |\Delta Y'| \leq |\Delta \sum_{(u,v)\in L} c_{uv}|$, since each DT belonging to X' requests coins from multiple channels in L. Therefore, $\Delta R(C) \leq 0$ with higher channel capacities. In other words, $R(C)$ decreases with higher capacities. □

In Fig. 2, transaction 2 and 6 are RTs, and both of them request channel (B, C), i.e., $\mathrm{nRT}_{(\mathrm{B,C})} = 2$. In the second and third lines of Table 2, the capacity of channel (B, C) increases from 3 to 4, i.e., $\mathrm{nRT}_{(\mathrm{B,C})} < c_{\mathrm{BC}} < c'_{\mathrm{BC}}$, satisfying the condition in Lemma 5. Obviously, the maximum gap between the theoretically optimal performance and the performance achievable in practice decreases from 1 to 0.

Corollary 6. *With fixed transactions, when the capacities of channels are at two extremes, either very large or very small, R is minimized.*

Proof. Suppose a channel (u,v), whose initial capacity is zero, now increases its capacity. Based on Corollary 4, at first, R rises with higher capacities of (u,v) until all the RTs requesting (u,v) can be relayed. According to Corollary 5, if the capacity of (u,v) is also larger than the number of coins requested by DTs from (u,v), R will no longer increase with higher c_{uv}. Thus, channel capacities shall be either very large or very small to minimize R. □

According to Corollary 4 and 5, Corollary 7 formulates the largest R in a PCN and the conditions on channel capacities given fixed transactions.

Corollary 7. *Consider a PCN with distinct sets of channel capacities. The largest gap, denoted as $R^*(C^*)$, is achieved under the set of channel capacities C^**

$$R^*(C^*) = \sum_{d_i}(m_i-1)a_i, \forall c_{uv} \in C^*, c_{uv} = \mathrm{nDT}_{uv}. \tag{17}$$

Proof. According to Theorems 4 and 5, when $\mathrm{nRT}_{uv} \geq \mathrm{nDT}_{uv}$, it is obvious that Corollary 7 holds. Otherwise, we need to compare $R(C)$ and $R(C')$, corresponding to the following two circumstances. First, $\forall(u,v), c_{uv} = \mathrm{nRT}_{uv}$. Second, $\forall(u,v), c_{uv} = \mathrm{nDT}_{uv}$

$$R^*(C^*) = \max\{R(C), R(C')\}. \tag{18}$$

When $\forall(u,v), c_{u,v} = \mathrm{nRT}_{uv}$, based on (15)

$$R(C) = \max \sum_{d_i\in \mathrm{SDT}} (m_i-1)a_i \leq \sum_{d_i}(m_i-1)a_i. \tag{19}$$

When $\forall(u,v), c_{u,v} = \mathrm{nDT}_{uv}$, from (12),$\bar{P}'$ appears when all the DTs succeed

$$\begin{aligned} \bar{P}' = \max\sum_{d_i} a_i, P'^* \leq \sum c_{uv}, R(C') = P'^* - \bar{P}' \\ \leq \sum_{d_i}(m_i-1)a_i. \end{aligned} \tag{20}$$

Combine (18), (19) and (20), we have

$$R^*(C^*) = \max\{R(C), R(C')\} \leq \sum_{d_i}(m_i-1)a_i.$$

□

There exist proposals in designing PCNs for adjusting channel capacities dynamically with an estimate on payment demands [21], [24]. We believe our theoretical analyses can provide insights on such designs. Specifically, Corollary 7 gives us two hints. First, the initial channel capacities shall not satisfy (17) to minimize R. Second, consider a PCN with the characteristic that DTs requests a large number of coins. The maximum gap between the theoretically optimal performance and the performance achievable in practice, i.e., $R^*(C^*)$, may be very large if the condition on channel capacities in (17) is satisfied. Thus, PCNs with such a characteristic are not preferred to minimize R.

6 Evaluation

We now present a rigorous evaluation of our theoretical formulations and analyses. Numerical results are obtained by simulating the process that channels with different capacities select transactions to relay running on two kinds of representative topologies in C++. Scheduling algorithms are not specified. Each channel (u,v) selects a set of transactions arbitrarily, leading to one probable performance level. By iterating all the N_{uv} probable sets of selected transactions for each channel (u,v), we obtain all the $\prod_{(u,v)} N_{uv}$ probable performances achieved in practice.

Performance Metrics. In each PCN, our objective is to derive both the theoretically optimal performance P^* and the maximum performance gap R by iterating all the $\prod_{(u,v)} N_{uv}$ probable performances. We compute P^* and R repeatedly given miscellaneous circumstances on channel capacities and numbers of coins requested by transactions.

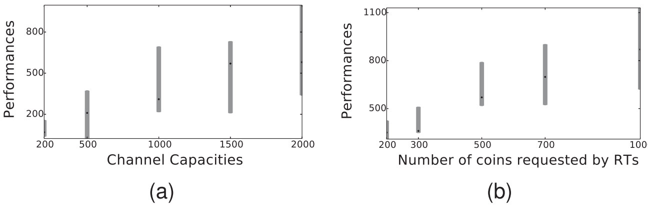


Fig. 3. There exist a gap between the performance achievable in practice and the theoretically optimal performance in a PCN.

Topologies. In our simulation, we use two kinds of topologies, and both of them have 360 nodes. The first is a scale-free topology generated randomly by the *networkx* package in *Python*. In a scale-free topology network, there are only a few nodes connecting to a great number of other nodes. For a real PCN implementation, i.e., Lightning network, it has been shown that nodes and channels have formed a scale-free topology [28], [29]. The second is a star-like network generated by a project in *Python* [30], where each user is attached to some specific nodes, called servers, with a probability proportional to the degree of those servers. This is reasonable consider the scenario that a user prefers to establish a channel with a node whose network connectivity is larger than others so that its transactions can be easily relayed to destinations.

Capacities Requested by Transactions. Considering the privacy requirements in PCNs, we do not have existing data or traces to simulate. Unless otherwise specified, the number of coins requested by transactions is heavy-tailed fitted by a Pareto distribution with the following cumulative density function:

$$F_X(x) = 1 - \left(\frac{x_m(\alpha-1)}{\alpha x}\right)^{\alpha}, x \geq \frac{x_m(\alpha-1)}{\alpha}, \tag{21}$$

where α is the *shape parameter*, and x_m is the *scale parameter* of such a Pareto distribution. In our evaluation, we have a random variable $X \sim \psi(\alpha = 3, x_m = 15)$ representing the number of coins requested by transactions, and the expected value of X satisfies $\mathbb{E}[X] = x_m = 15$.

Regular Transactions and Dominant Transactions. The sets of regular and dominant transactions are determined by payers, payees, and intermediate channels of all transactions. For each transaction, since there does not exist any available canonical datasets due to the privacy property in PCNs, we randomly select a set of edges that are continuous in a PCN for this transaction as intermediate channels, whose starting and ending nodes are the payer and payee. By repeatedly conducting such a random process, we obtain multiple sets of regular transactions and dominant transactions that are needed in the following evaluations.

6.1 Performances Achievable in Practice

In this section, we aim to give a straightforward evidence on the existence of varying performances by simulation. We use the scale-free topology which is a minimal version of the Lightning network. With the methods for simulating RTs and DTs as introduced above, we get 83 dominant transactions and 47 regular transactions competing for 52 channels in total. Performance dynamics with different total channel capacities and number of coins requested by regular transactions are shown in Fig. 3.

In Fig. 3b, each of those 52 channels has a capacity of 20 coins. In Fig. 3a, regular transactions request 500 coins in total. The y-axis is performances achieved in our simulation, and the x-axis are the total capacities requested by regular transactions and channel capacities. It is easy to find that, given fixed channel capacities and transactions, there exist multiple performances achieved in our simulations, i.e., the fundamental performance limits exist in real implementations. We then study the effects of channel capacities and transactions on the theoretically optimal performance and R.

6.2 The Theoretically Optimal Performance

6.2.1 Formulation on the Theoretically Optimal Performance.

We first prove that our formulation on the optimal performance is valid. Based on the methods above, we get 104 regular transactions requesting 1,387 coins and 26 dominant transactions requesting 563 coins in total under the scale-free topology. Specifically, they compete for 84 channels. During the simulation, we randomly select a channel and increase its capacity each time, then record the corresponding maximum number of coins that can be transferred. We also calculate the formulated optimal performance under each set of channel capacities to be compared with the simulation.

The results are shown in Fig. 4b. The $x-$axis is total channel capacities, and the $y-$ axis is the optimal performance P^*. The two lines represent the simulated and formulated ones. It is obvious that, our formulated optimal performance always exceeds the simulated one.

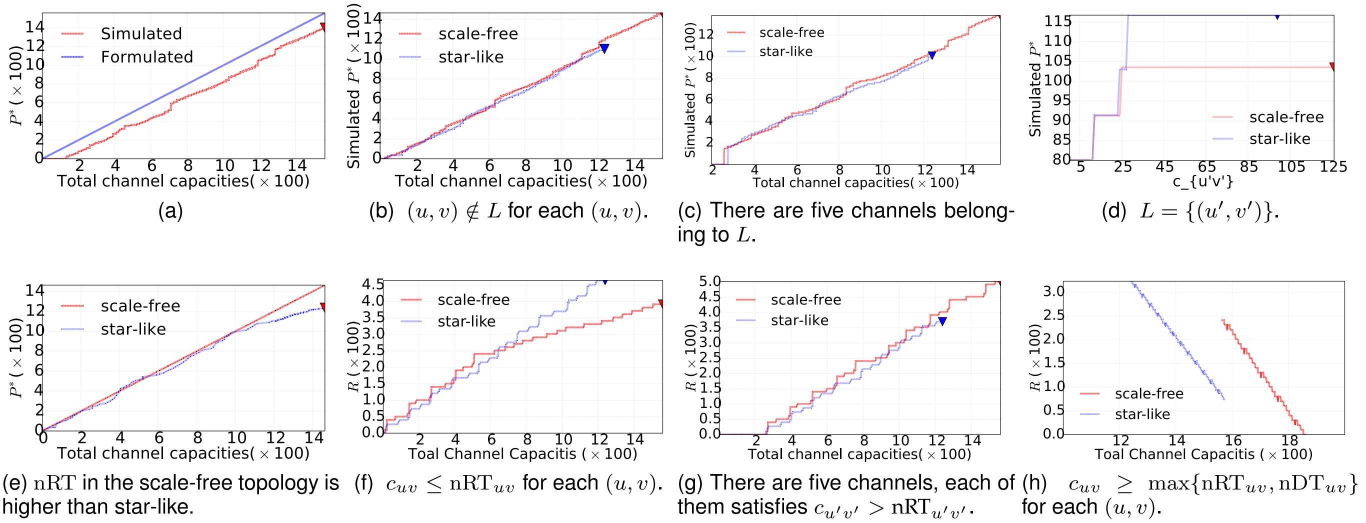
6.2.2 Impact From Channel Capacities.

We fix regular transactions, dominant transactions, and the amount of capacities requested by them as follows. Both of the scale-free topology and the star-like topology are incorporated. Transactions in the scale-free topology are the same to the one in the last evaluation on the formulated optimal performance. For the star-like topology, our regular transactions request 1,163 coins in total, while the remaining dominant transactions request 792 coins together. And the star-like topology has 61 channels that are competed for by multiple transactions.

We evaluate this impact in three moves with respect to different settings of channel capacities.

First, the capacity of each channel is not larger than the number of coins requested by RTs from that channel, i.e., $\forall(u,v), (u,v) \notin L$. We repeatedly increase the capacity of a randomly selected channel by 20 until $\forall(u,v), (u,v) \in L$. Therefore, the sum of channel capacities will increase from zero to the one that can exactly relay all the RTs, i.e $[0, \sum_{(u,v)} \mathrm{nRT}_{uv}]$.

The simulated optimal performances with regard to total channel capacities in the first circumstance is recorded in Fig. 4b. The x-axis represents the total channel capacities, and the y-axis represents the simulated optimal performance. Obviously, in both of these topologies, the optimal performance increases almost linearly with total channel capacities. Note that we only increase the capacity of one



(a)

(b) $(u, v) \notin L$ for each (u, v).

(c) There are five channels belonging to L.

(d) $L = \{(u', v')\}$.

(e) nRT in the scale-free topology is higher than star-like.

(f) $c_{uv} \leq \mathrm{nRT}_{uv}$ for each (u, v).

(g) There are five channels, each of them satisfies $c_{u'v'} > \mathrm{nRT}_{u'v'}$.

(h) $c_{uv} \geq \max\{\mathrm{nRT}_{uv}, \mathrm{nDT}_{uv}\}$ for each (u, v).

Fig. 4. (a): Our formulation on the optimal performance is able to describe the optimal case in PCNs. (b)(c)(d): P^* is positively correlated to channel capacities if $|L| \neq 1$. (e): The amount of capacities requested by RTs has positive effects on the optimal performance. (f)(g): R increases when the capacities of channels who cannot support all the RTs increase. (h): R decreases when the capacities of channels who can support all the RTs or DTs increase.

channel each time. In another word, the optimal performance is linearly and positively related to the capacity of each channel $(u, v) \notin L$.

Second, there are five channels, whose capacities are larger than the number of coins requested by RTs from those channels, i.e., $\exists (u', v'), (u', v') \in L$. The capacity of a randomly selected channel rises by 20 until $\forall (u, v), (u, v) \in L$. Fig. 4a records the simulated optimal performances with regard to total channel capacities in the second circumstance. It is easy to find that the optimal performance increases with higher channel capacities.

Combing findings in the first and second circumstances, the result in Corollary 1 can be verified.

Third, Corollary 2 is verified as follows. We randomly select a channel (u', v'), and let its initial capacity be 0. The capacities of other channels support one regular transaction and do not change in this evaluation. Then we increase the capacity of (u', v') by 10 repeatedly, and record the corresponding optimal performances.

In Fig. 4d, the optimal performances with regard to different capacities of channel (u', v') are illustrated. The x-axis represents the capacity of channel (u', v'), and the y-axis represents the simulated optimal performance. Obviously, when the capacity of (u', v') is the number of the coins requested by RTs, i.e., $c_{u'v'} = \mathrm{nRT}_{u'v'}$, the optimal performance peaks, and does not increase with higher $c_{u'v'}$.

6.2.3 *Impact From Transactions.*

Using the same simulation settings of the last experiment, RTs request for 20% fewer coins under our star-like topology than the scale-free topology under all channel capacities. Thus, we compare P^* between these two topologies.

Fig. 4e records the optimal performances under these two topologies. The $y-$axis is the optimal performance, and $x-$axis is total channel capacities. The two lines represent the scale-free and star-like topologies. Obviously, the optimal performance under the scale-free topology is always higher than the star-like topology for all sets of channel capacities. This verifies our conclusion in Corollary 3.

Insight. After the channel capacity can relay all the RTs in the star-like topology, the optimal performance increases much slower with higher channel capacities. This is reasonable in reality. After all RTs have succeeded, channels must relay dominant transactions. As dominant transactions competed for more than one channel, it costs more total capacities to relay a dominant transaction than a regular transaction when transferring the same amount of coins. For example, in Fig. 1, to support transaction 1, both channels (B, C) and (D, E) shall be equipped with a capacity higher than 2. However, to relay any other regular transactions, either (B, C) or (D, E) needs to increase its capacity.

6.3 *The Performance Gap*

We evaluate R in three circumstances. Regular transactions, dominant transactions and the amount of requests from them are the same to the last evaluation.

First, we evaluate R when the capacity of each channel cannot relay all the RTs requesting that channel. The total channel capacities will increase from zero to the one that can exactly relay all the RTs, i.e., $[0, \sum_{(u,v)} \mathrm{nRT}_{uv}]$. Capacities are assigned to channels iteratively as follows. Consider a process that we randomly select a channel and increase its capacity so that one more transaction can be supported by that channel. The initial capacity of each channel is zero. When the total channel capacities increase, we repeat such a process until all the capacities have been assigned.

R with respect to channel capacities in this circumstance is recorded in Fig. 4f. The x-axis represents total channel capacities, and the y-axis denotes the simulated R. Based on our capacity assigning process, there does not exist a decrease in the capacities of any channels when the total channel capacities increase.

The findings in Fig. 4f are two-fold: *First*, when total channel capacities increase, R does not decrease. *Second*, R is higher in a star-like network than in a scale-free network.

Second, we testify R when there are five channels, whose capacities are sufficient to relay all the transactions, i.e.,

$\forall (u', v') \in L, c_{u'v'} = \mathrm{nRT}_{u'v'} + \mathrm{nDT}_{u'v'}$. We randomly select five channels being added to L. Capacities of other channels are the same to the last experiment. Fig. 4g records the results in this circumstance. Obviously, R does not decrease when the total channel capacities increase.

Results in these two circumstances show that R is positively correlated with the capacities of channels who can only support part of RTs, verifying Corollary 4.

Third, to manifest the consistency of Corollary 5, we examine R when the capacity of each channel (u, v) before increasing can already support all the RTs or DTs. Then we repeatedly increase the capacities of all the channels by 20 until all the transactions can be relayed. In other words, the total capacities range in $[\sum_{(u,v)} \max\{\mathrm{nRT}_{uv}, \mathrm{nDT}_{uv}\}, \sum_{(u,v)}(\mathrm{nRT}_{uv} + \mathrm{nDT}_{uv})]$. The capacity assigning process is the same to the last evaluation. R with regard to total channel capacities following these parameter settings is shown in Fig. 4h. Obviously, R never increases with higher channel capacities.

Note that, Fig. 4g seems to contradict Figs. 4f and 4h on the maximal R. In Figs. 4f and 4h, the maximal R in the star-like topology, whose DTs request more capacities, is higher, verifying Corollary 7. However, Fig. 4f does not give the same result. Since the set of channel capacities in Fig. 4g do not satisfy the condition in Corollary 7, thus cannot be applied as a verification.

7 Related Work

A payment channel network (PCN) [3], [4], [22] speeds up transaction confirmation processes and improves the scalability by processing transactions in parallel, while using the main chain only as dispute arbitrators. Yet, its performances, in terms of the number of coins that are transferred, are heavily constrained, due to inherent privacy requirements [23], [31], [32], [33]. Therefore, how to improve the performance while preserving privacy becomes a major research issue. There has been a number of protocols proposed recently [19], [20], [20], [21], [22], [34], [35], [36], and with different assumptions. For instance, transactions are scheduled with global priorities of all transactions in [23]. In Speedymurmur [20] and SlientWhispe [19], transactions are accomplished in a centralized manner. In Revive [21], a centralized algorithm is supplemented using a recharging scheme for channels with insufficient capacities. In Pripay [22], a server schedules transactions through trusted hardware, suffering from single node failures. In [24], the performance is enhanced by dynamically setting channel capacities given estimated payment demands.

Rather than designing new privacy-preserving algorithms or performance-enhancing algorithms, this paper focuses on a thorough analytical understanding on the fundamental performance limits over scheduling transactions in a PCN. We believe this is one of the fundamental questions unique in PCNs, which is of particular relevance in designing algorithms with the objective of performance optimization. Rather than maximizing the performance by introducing elaborate proposals, this paper concentrates on the maximum gap between the theoretically optimal performance and the performance achievable in practice. We believe such a gap characterizes the design space for new scheduling algorithms, and provides insights on dynamically adjusting channel capacities.

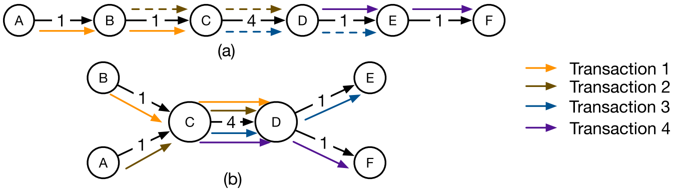


Fig. 5. In (a), only transaction 1 and 4 are RTs, and the theoretical optimal result is achieved when these 2 RTs succeed. In (b), all of these 4 transactions are RTs, the optimal result is 4.

Recently, Tang *et al.* [32] discussed the relationship between a utility and the extent of privacy exposure under a specific attack model. Their focus is different from ours in that we derive the performance gap on a generic setting without any attack models. Further, their utility is a function proportional to the number of coins that are transferred, which instead is the objective function in our work, where the impact from channel capacities and incoming transactions is examined at the same time.

8 Discussions

Topological Impact. Apparently, the selection of intermediate channels for forwarding each transaction must take the entire topology into consideration. This indicates that the PCN topology dictates the set of transactions requesting for capacities at each intermediate channel. Noticing that our model distinguishes transactions at each intermediate channel into two categories, i.e., dominant and regular transactions (DTs and RTs), and has computed the PCN performance given DTs and RTs. As shown in Corollary 3, more coins requested by RTs indicates better performance. Therefore, a topology where it is more likely for a transaction to become a RT is appreciated.

There is a popular belief, originated from computer networking research, that hub-and-spoke network topologies lead to poor performance in general. In contrast, PCN topologies with an extremely long route contain many intermediate channels, and may lead to even worse performance. For example, in Fig. 5, there are two topologies and 4 transactions. The total channel capacities are 8 in these two topologies. Each transaction requests for 1 coin. To minimize the number of channels a transaction may compete for, each transaction only goes through 2 intermediate channels. The first topology in Fig. 5a has 6 nodes connecting with each other linearly, forming a long route. There will be 2 DTs at least, whose theoretically optimal performance is 2. The second topology in Fig. 5b has a hub between node 2 and 3, and all of these 4 transactions go through this hub. Apparently, none of them are DTs, and the theoretical optimality is 4, where all of these 4 transactions succeed.

To further quantitatively understand this issue, a mathematical characterization on how a topology affects the sets of DTs and RTs is needed, which can be conducted in our future work. Intuitively, which parameters shall be incorporated to model a topology shall be taken into account at first. Based on this, the formulation on the probability that a transaction becomes a RT or a DT with respect to the topology may be derived.

Privacy Implications. It is not feasible to achieve optimality while satisfying all the privacy requirements, which has been formally discussed in [23]. However, our analysis indicates that there must exist scheduling algorithms that can achieve optimality only in violation with a part of the privacy requirements. Algorithm 1 in Section 4 can approach the theoretically optimal performance, where the sets of transactions pending at all intermediate channels and the initial capacity of each channel before scheduling are provided as algorithm inputs. Apparently, only this initial channel capacity when the scheduling is about to start is required, instead of all capacities upon each transaction confirmation, i.e., not all the privacy requirements are violated.

Another issue is related to the relationship between the extent of privacy infringement and the performance achieved in practice during a scheduling process. For example, if the set of transactions to be relayed in a PCN is exposed, by giving each transaction a global priority, deadlock can be avoided, indicating a higher performance through this scheduler [23]. However, with more privacy exposure, e.g., channel capacities are further exposed, without knowing how many coins each transaction requested, a scheduling algorithm may not be able to improve the performance. To quantitatively formulate how probable the performance can be improved with different extents of privacy exposure and different scheduling algorithms, a model mathematically describing the extent of privacy infringement is necessary. This is, however, not directly related to our current model, and can be conducted in future work.

9 Conclusion

Payment channel networks (PCNs) have the potential to significantly enhance the throughput of blockchain systems, while its performance, in terms of the number of coins that are transferred successfully, can suffer with inherent privacy requirements. There has been no systemic study in capturing the fundamental performance properties in PCNs. This turns to be a challenging problem with the unique characteristics in PCNs, in particular the link capacity is of a discrete number with a residual value. In this paper, we, for the first time, investigate the fundamental performance limits of PCNs and compute the maximum gap between the theoretically optimal performance and the performance achievable in practice. Through rigorous analysis, we derive conditions on scheduling transactions to achieve the theoretically optimal performance. Finally, we show that such a gap between the theoretically optimal performance and the performance achievable in practice in a PCN can be minimized when the channel capacities are at two extremes: either very large or very small. We also give a specific set of channel capacities in a PCN that results in the maximum performance gap.

References

[1] Bitcoin Wiki: Bitcoin contract. Accessed: May 2021. [Online]. Available: https://en.bitcoin.it/wiki/Contract

[2] S. Team, "Stellar network," Accessed: May 2021. [Online]. Available: https://www.stellar.org

[3] F. Armknecht, G. O. Karame, A. Mandal, F. Youssef, and E. Zenner, "Ripple: Overview and outlook," in *Proc. Springer Int. Conf. Trust Trustworthy Comput.*, 2015, pp 163–180.

[4] J. Poon and T. Dryja, "The bitcoin lightning network: Scalable off-chain instant payments," 2016. [Online]. Available: https://www.bitcoinlightning.com/wp-content/uploads/2018/03/lightning-network-paper.pdf

[5] V. Buterin *et al.*, "Raiden network," Accessed: May 2021. [Online]. Available: https://raiden.network/

[6] A. Miller, M. Möser, K. Lee, and A. Narayanan, "An empirical analysis of linkability in the Monero blockchain," 2017, *arXiv: 1704.04299*.

[7] S. Nakamoto *et al.*, "Bitcoin: A peer-to-peer electronic cash system," 2018. [Online]. Available: https://bitcoin.org/bitcoin.pdf

[8] G. Wood *et al.*, "Ethereum: A secure decentralised generalised transaction ledger," Ethereum Project, London, U.K., Yellow Paper, 2014.

[9] E. Androulaki *et al.*, "Hyperledger Fabric: A distributed operating system for permissioned blockchains," in *Proc. ACM Eur. Conf. Comput. Syst.*, 2018, pp. 1–15.

[10] J. Herrera-Joancomartí and C. Pérez-Solà, "Privacy in bitcoin transactions: New challenges from blockchain scalability solutions," in *Proc. Springer Int. Conf. Model. Decis. Artif. Intell.*, 2016, pp. 26–44.

[11] K. Torpey, "Does the lightning network threaten bitcoin's censorship resistance?," 2015. [Online]. Available: https://bitcoinmagazine.com/articles/does-the-lightning- network-threaten-bitcoin-s-censorship-https://lists.linuxfoundation.org/pipermail/lightning-dev/2015-November/000309.htmlresistance-1461953131/

[12] M. Green and I. Miers, "Bolt: Anonymous payment channels for decentralized currencies," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, 2017, pp. 473–489.

[13] E. Heilman, L. Alshenibr, F. Baldimtsi, A. Scafuro, and S. Goldberg, "TumBlebit: An untrusted bitcoin-compatible anonymous payment hub," in *Proc. Netw. Distrib. Syst. Secur. Symp. (NDSS)*, 2017.

[14] J. Lind, I. Eyal, P. Pietzuch, and E. G. Sirer, "Teechan: Payment channels using trusted execution environments," 2016, *arXiv:1612.07766*.

[15] E. Team, "Eclair implementation of the lightning network," Accessed: Jul. 2020. [Online]. Available: https://github.com/ACINQ/eclair

[16] A. Towns, "Better privacy with mailing list," Accessed: Jun. 2020. [Online]. Available: https://lists.linuxfoundation.org/pipermail/lightning-dev/2015-November/000309.html

[17] E. Paul, "What is digital signature-how it works, benefits, objectives, concept," 2017. [Online]. Available: http://www. emptrust.com/blog/benefits-of

[18] L. N. Team, "Bolt #4: Onion routing protocol," 2020. [Online]. Available: https://github.com/lightningnetwork/lightning-rfc/blob/master/04-onion-routing.md

[19] G. Malavolta, P. Moreno-Sanchez, A. Kate, and M. Maffei, "SilentWhispers: Enforcing security and privacy in decentralized credit networks," in *Proc. Netw. Distrib. Syst. Secur. Symp.*, 2017.

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

[21] R. Khalil and A. Gervais, "Revive: Rebalancing off-blockchain payment networks," in *Proc. ACM Conf. Comput. Commun. Secur.*, 2017, pp. 439–453.

[22] P. Moreno-Sanchez, A. Kate, M. Maffei, and K. Pecina, "Privacy preserving payments in credit networks," in *Proc. Netw. Distrib. Syst. Secur. Symp.*, 2015.

[23] G. Malavolta, P. Moreno-Sanchez, A. Kate, M. Maffei, and S. Ravi, "Concurrency and privacy with payment-channel networks," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, 2017, pp. 455–471.

[24] P. Li, T. Miyazaki, and W. Zhou, "Secure balance planning of off-blockchain payment channel networks," in *Proc. IEEE Conf. Comput. Commun. (INFOCOM)*, 2020, pp. 1728–1737.

[25] S. Low and P. Varaiya, "Burstiness bounds for some burst reducing servers," in *Proc. IEEE Int. Conf. Comput. Commun.*, 1993, pp. 2–9.

[26] S. S. Lam and G. G. Xie, "Burst scheduling networks: Flow specification and performance guarantees," in *Proc. Springer Int. Workshop Netw. Oper. Syst. Support Digit. Audio Video*, 1995.

[27] I. Rubin and K. D. Lin, "A burst-level adaptive input-rate flow control scheme for ATM networks," in *Proc. IEEE Int. Conf. Comput. Commun.*, 1993, pp. 386–394.

[28] I. A. Seres, L. Gulyás, D. A. Nagy, and P. Burcsi, "Topological analysis of bitcoin's lightning network," 2019, *arXiv:1901.04972*.

[29] M. Conoscenti, A. Vetrò, J. De Martin, and F. Spini, "The CLoTH simulator for HTLC payment networks with introductory lightning network performance results," *Information*, vol. 9, no. 9, 2018, Art. no. 223.

[30] bitromortac, "lnregtest - Lightning networks on bitcoin regtest," 2020. [Online]. Available: https://github.com/bitromortac/lnregtest

[31] S. Meiklejohn *et al.*, "A fistful of bitcoins: Characterizing payments among men with no names," in *Proc. ACM Internet Meas. Conf.*, 2013.

[32] W. Tang, W. Wang, G. Fanti, and S. Oh, "Privacy-utility tradeoffs in routing cryptocurrency over payment channel networks," in *Proc. ACM SIGMETRICS/Perform. Joint Int. Conf. Meas. Model. Comput. Syst.*, 2020, pp. 81–82.

[33] R. Henry, A. Herzberg, and A. Kate, "Blockchain access privacy: Challenges and directions," *IEEE Symp. Secur. Privacy*, vol. 16, no. 4, pp. 38–45, Jul./Aug. 2018.

[34] P. Prihodko, S. Zhigulin, M. Sahno, A. Ostrovskiy, and O. Osuntokun, "Flare: An approach to routing in lightning network," Flare, Bitfury Group Ltd., London, U.K., White Paper, 2016.

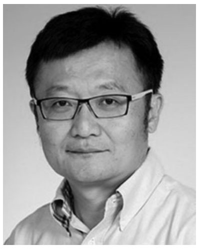
[35] S. Dziembowski, L. Eckey, S. Faust, and D. Malinowski, "Perun: Virtual payment hubs over cryptocurrencies," in *Proc. IEEE Symp. Secur. Privacy*, 2019, pp. 106–123.

[36] V. Sivaraman *et al.*, "High throughput cryptocurrency routing in payment channel networks," in *Proc. USENIX Symp. Netw. Syst. Des. Implementation*, 2020, pp. 777–796.

Yuechen Tao received the BEng degree from the Department of Computer Science, Shandong University, China, in 2016. She is currently working toward the PhD degree with the Department of Computer Science and Engineering, Hong Kong University of Science and Technology. Since 2016, she has been with the Department of Computer Science and Engineering, Hong Kong University of Science and Technology. Her current research interests include performance analyses and improvements in blockchains and atomicity assurance of cross-chain transactions.

Bo Li (Fellow, IEEE) received the BEng (summa cum laude) degree in the computer science from Tsinghua University, Beijing, China, and the PhD degree in electrical and computer engineering from the University of Massachusetts at Amherst. He is currently a chair professor with the Department of Computer Science and Engineering, Hong Kong University of Science and Technology. Between 2010 and 2016, he was the Cheung Kong visiting chair professor with Shanghai Jiao Tong University and an adjunct researcher with the Microsoft Research Asia (MSRA) from 1999 to 2006 and the Microsoft Advanced Technology Center from 2007 to 2009. He made pioneering contributions in multimedia communications and the Internet video broadcast, in particular the Coolstreaming system, which was credited as first large-scale Peer-to-Peer live video streaming system in the world. It attracted significant attention from both industry with substantial VC investment, and academia in receiving the Test-of-Time Best Paper Award from IEEE INFOCOM in 2015. He was the recipient of the six best paper awards from the IEEE, including INFOCOM in 2021. He is an editor or the guest editor of more than a two dozen of IEEE and ACM journals and magazines. He was the Co-TPC Chair for the IEEE INFOCOM 2004.

Baochun Li (Fellow, IEEE) received the BEng degree from the Department of Computer Science and Technology, Tsinghua University, China, in 1995 and the MS and PhD degrees from the Department of Computer Science, University of Illinois at Urbana-Champaign, Urbana, in 1997 and 2000. Since 2000, he has been with the Department of Electrical and Computer Engineering, University of Toronto, where he is currently a professor. Since August 2005, he has been the Bell Canada Endowed Chair of computer engineering. His research interests include cloud computing, distributed systems, datacenter networking, and wireless systems.



Lei Chen (Fellow, IEEE) received the BS degree in computer science and engineering from Tianjin University, China, in 1994, the MA degree from the Asian Institute of Technology, Bangkok, Thailand, in 1997, and the PhD degree in computer science from the University of Waterloo, Canada, in 2005. He is currently the chair professor with the Department of Computer Science and Engineering, Hong Kong University of Science and Technology. His research interests include data-driven AI, knowledge graphs, blockchains, data privacy, crowdsourcing, and spatial and temporal databases and query optimization on large graphs and probabilistic databases.