Cumulonimbus: An Incentive Mechanism for Crypto Capital Commitment in Payment Channel Networks *

Yuhui Zhang Dejun Yang Guoliang Xue

***Abstract*—Payment channel networks (PCNs) are proposed to improve the cryptocurrency scalability by settling off-chain transactions. However, a significant barrier is that a PCN user must solicit sufficient capital owned by the counterparty on its channel (*i.e.*, inbound liquidity) to receive payments. To alleviate this inbound liquidity problem, Channel Liquidity Marketplaces (CLMs), *e.g.*, Bitcoin's Lightning Pool, have been introduced, such that users can buy and sell inbound liquidity by trading crypto capital commitment in PCNs. Existing CLMs lack good incentive mechanisms that can attract more user participation. To fulfill this void, we design Cumulonimbus, an incentive mechanism for trading crypto capital commitment, which satisfies truthfulness, individual rationality, budget balance, and computational efficiency. Particularly, Cumulonimbus considers two unique features of crypto capital commitment, referred to as demand indivisibility and supply divisibility. Extensive simulations demonstrate that Cumulonimbus achieves higher satisfaction ratio, liquidity utilization, and social welfare compared with a state-of-the-art CLM mechanism Lightning Pool [18].**

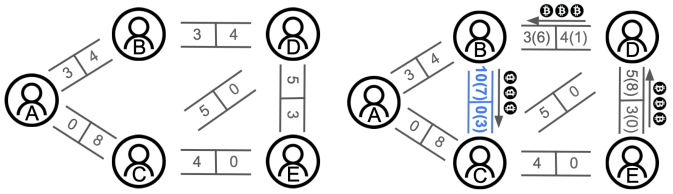
***Index Terms*—Cryptocurrency, payment channel network, incentive mechanism, blockchain**

I. Introduction

The past decade has seen a blooming of cryptocurrencies [20], *e.g.*, Bitcoin [17] and Ethereum [4]. However, cryptocurrencies cannot scale for wide-spread use, due to high overhead and storage requirement [14]. Payment channel networks (PCNs), *e.g.*, Bitcoin's Lightning Network [19] and Ethereum's Raiden Network [7], have been proposed to tackle the scalability issues [19]. However, a significant barrier in PCNs is that a user can receive payments only if there is sufficient capital owned by the counterparty (*i.e.,* the other user on its channel), which is referred to as the *inbound liquidity* [18]. Therefore, inbound liquidity is essential as it directly determines the success of payments, which is the ultimate purpose of PCNs. Fig. 1 illustrates the importance of inbound liquidity.

On the one hand, the *liquidity takers* who desire inbound liquidity need to convince others to open channels to them and to deposit crypto capital on their channels. On the other hand, the *liquidity makers* who own crypto capital seek to provide inbound liquidity for profit. To enable the liquidity takers and makers to publish their liquidity demand and supply information, researchers have developed various applications and systems for trading crypto capital commitment [15, 16, 18, 22, 23]. This new paradigm is commonly referred to as *channel liquidity marketplace* (CLM).

Zhang and Yang are affiliated with Colorado School of Mines, Golden, CO 80401. Xue is affiliated with Arizona State University, Tempe, AZ 85287. Email:{yuhzhang, djyang}@mines.edu, xue@asu.edu. This research was supported in part by NSF grants 2007083 and 2008935. The information reported here does not reflect the position or the policy of the federal government.



(a) Without inbound liquidity (b) With inbound liquidity

Fig. 1. Illustration for inbound liquidity. In the left, two values between two nodes represent the capital distribution. Although C owns $8฿+5฿+4฿ = 17฿$, C's inbound liquidity from the counterparties (*i.e.,* A, C, and D) is 0. Thus, C cannot receive payments from any other user. In the right, C has an inbound liquidity of $10฿$ from B by creating a payment channel. Now C can receive payments from A, B, D, and E. For example, C can receive $3฿$ along $E \to D \to B \to C$. The values in the brackets represent the capital distribution after C receives $3฿$ from E.

Unfortunately, the existing works only scratch the surface by either requiring voluntary participation without considering the design of incentive mechanisms or neglecting some critical properties of incentive mechanisms. When participating in a CLM, the liquidity makers must lock their capital in the channels for a certain time period. Since the value of cryptocurrencies fluctuates dramatically, the time value of money (TVM) concept in financial management also applies to cryptocurrencies, or might contribute more according to the historic volatility. Therefore, a liquidity maker would not be interested in participating in a CLM, unless it receives a satisfying reward to compensate its TVM consumption. Without adequate liquidity maker participation, it is impossible for the liquidity takers to solicit sufficient inbound liquidity, which is essential to the success of PCNs.

At first glance, crypto capital commitment seems to resemble the conventional capital commitment, *e.g.*, bond. However, there are two important differentiating features, which make the design of incentive mechanisms for crypto capital commitment more complex. The first feature is the **demand indivisibility**, due to the fact that a liquidity taker's demand should be either fully provided by one liquidity maker or none. This is because inbound liquidity separated on multiple channels does not help a liquidity taker receive sufficient large payments. The second feature is the **supply divisibility**, because a liquidity maker can split and deposit its crypto capital on multiple channels to fulfill multiple liquidity takers. Due to these reasons, incentive mechanisms for conventional goods or financial resources cannot be applied to crypto capital commitment.

In this paper, we design an incentive mechanisms to motivate both the liquidity takers and makers to participate in CLMs, which allows them to buy and sell inbound liquidity by trading

crypto capital commitment in PCNs. **The main contributions of this paper are**:

- To the best of our knowledge, we are the first to design an incentive mechanism for trading crypto capital commitment with the consideration of demand indivisibility and supply divisibility.
- We design an incentive mechanism Cumulonimbus, which guarantees that a liquidity taker's demand is either fully supplied by a liquidity maker or none, but a liquidity maker's supply can fulfill multiple liquidity takers.
- We rigorously prove that Cumulonimbus satisfies truthfulness, individual rationality, budget balance, and computational efficiency.

The remainder of the paper is organized as follows. In Section II, we provide a brief literature review of the related work. In Section III, we present the background in PCNs, formally describe the system model and incentive mechanism model, and give the necessary assumptions. In Section IV, we design an incentive mechanism Cumulonimbus and conduct detailed theoretical analysis. In Section V, we evaluate the performance of Cumulonimbus by comparing it with a state-of-art CLM. In Section VI, we conclude this paper.

II. Related Work

Up to now, there are only limited efforts on the design of incentive mechanisms for crypto capital commitment in PCNs. Realizing the great potential benefit, some PCN power users have provided channel liquidity services, *e.g.*, Bitrefill's Thor [10], Y'alls [12], LNBig [6], and ln2me [5]. These services promise to provide inbound liquidity by opening channels to the liquidity takers and deposit crypto capital on the channels. Nevertheless, there is only a single liquidity maker (the liquidity service provider itself) who determines and charges a posted price. The Celar Network [15] has designed a liquidity backing auction with a single liquidity maker, which utilizes Vickrey–Clarke–Groves (VCG) to determine the winning liquidity takers and payments. However, all these works do not involve adequate liquidity maker participation, which makes it impossible for the liquidity takers to solicit sufficient inbound liquidity,

ZmnSCPxj [23] has proposed a channel liquidity marketplace (CLM), where both the liquidity makers and takers can participate in trading crypto capital commitment. However, it focuses only on the protocol and smart contract design, instead of the incentive mechanism design. Recently, Lightning Lab [16] has released a CLM called Lightning Pool [18], which is implemented as a sealed-bid frequent batched uniform price double auction. Lightning Pool adopts a greedy algorithm to match the liquidity takers and makers, but does not consider truthfulness, individual rationality, or budget balance. Therefore, all the existing works either do not consider the design of incentive mechanisms or have neglected some critical properties of incentive mechanisms. In this paper, we design an incentive mechanism to motivate both the liquidity takers and makers to participate in CLMs to buy and sell inbound liquidity, while considering the unique features of crypto capital commitment, *i.e.,* the demand indivisibility and supply divisibility.

III. System Model and Problem Formulation

In this section, we present an overview of the channel liquidity marketplace system, describe the incentive mechanism model, and give the desired properties.

A. Background and System Overview

Cryptocurrencies have suffered from the large overhead of global consensus and security assurance, which largely limits their applications in real-world scenarios. To tackle the scalability issues, PCNs (*e.g.*, Bitcoin Lightning [8] and Ethereum Raiden [7]) have been proposed to offer off-chain settlement of transactions with minimal involvement of expensive blockchain operations. To send and receive payments in a PCN, a user must open a payment channel to another PCN user. Two users establish a payment channel by each depositing a certain amount of capital into a joint account and adding this transaction to the blockchain. Once the payment channel has been opened, a user is able to send payments, if it owns sufficient capital on its channel.

Inbound Liquidity. Unfortunately, it is not guaranteed that a user can receive payments by opening a payment channel. There has to be sufficient capital owned by the counterparty (*i.e.,* the other user on its channel). Such crypto capital commitment allocated by the counterparty is typically referred to as *inbound liquidity*. Because a user must convince other users to open channels to it and to allocate crypto capital commitment towards it, the *inbound liquidity problem* remains a significant barrier to the success of PCNs.

B. Channel Liquidity Marketplace

The adoption of *channel liquidity marketplace* (CLM) is one way to mitigate the inbound liquidity problem. A CLM, *e,g.*, Bitcoin Lightning Pool [16], is a marketplace that enables participants to buy and sell inbound liquidity by trading crypto capital commitment. In general, there are two roles of participants defined as follows.

Definition 1 (Liquidity Taker)**.** *A Liquidity Taker is a participant who wants to receive payments and is willing to pay for inbound liquidity.*

Definition 2 (Liquidity Maker)**.** *A Liquidity Maker is a participant who owns capital and is willing to sell inbound liquidity for profit.*

We consider a CLM consisting of an auctioneer, a set $\mathcal{T} = \{T_1, T_2, \ldots, T_i, \ldots, T_n\}$ of n liquidity takers and a set $\mathcal{M} = \{M_1, M_2, \ldots, M_j, \ldots, M_m\}$ of m liquidity makers. The non-trusted auctioneer designs incentive mechanisms to achieve desired economic properties, determines the winning takers and makers, and calculate their payments. The inbound liquidity is assumed to be *homogeneous*, which means that there is no preference for the takers using inbound liquidity from different makers. Note that this model can be easily extended to the *heterogeneous* case by rating the inbound liquidity quality of

TABLE I
MAIN NOTATIONS

Notation	Meaning
$\mathcal{T}$	set of liquidity takers (buyers)
$\mathcal{M}$	set of liquidity makers (sellers)
$\mathcal{T}_w$	winning set of buyers
$\mathcal{M}_w$	winning set of sellers
d_i	buyer T_i's liquidity demand amount
b_i	buyer T_i's bid for d_i units of liquidity
v_i^b	buyer T_i's valuation for d_i units of liquidity
p_i^b	payment for buyer T_i for d_i units of liquidity
u_i^b	utility for buyer T_i
s_j	seller M_j's liquidity supply amount
a_j	seller M_j's ask for one unit of liquidity
v_j^s	seller M_j's valuation for one unit of liquidity
p_j^s	payment to seller M_j for one unit of liquidity
u_j^s	utility for seller M_j
x_{ij}	binary variable indicating if M_j sells d_i units to T_i

each maker, such that a taker can specify its preference to solicit inbound liquidity from certain makers. Since a liquidity taker needs inbound liquidity to receive payments, it cannot receive sufficient large payments, if the inbound liquidity is separated on multiple channels. Therefore, a liquidity taker's demand should be either fully provided by one liquidity maker or none, referred to as *demand indivisibility*. Meanwhile, a liquidity maker's supply can be provided to multiple liquidity takers, since it can split and deposit its capital on multiple channels, referred to as *supply divisibility*.

C. Incentive Mechanism Model

We aim to design an incentive mechanism that allows both liquidity takers and makers to buy and sell inbound liquidity by trading capital commitment, while considering demand indivisibility and supply divisibility. In the incentive mechanism, the makers are sellers, and the takers are buyers. Throughout the rest of this paper, we use the terminology of maker and seller, taker and buyer interchangeably. Each buyer T_i requests an inbound liquidity demand of d_i units and holds a private valuation $v_i^b \geq 0$ for buying d_i units and a bid $b_i \geq 0$ as the maximum amount that it would pay for d_i units. Each seller M_j provides an inbound liquidity supply of s_j units and holds a private valuation $v_j^s \geq 0$ for selling one unit and an ask $a_j \geq 0$ as the minimum amount that it would sell one unit.

The incentive mechanism works as follows: after collecting the bids and asks privately from all buyers and sellers, the incentive mechanism decides the allocation for each buyer and seller. We use a binary variable x_{ij} to represent whether a buyer T_i's demand is fulfilled by a seller M_j, defined as:

$$x_{ij} = \begin{cases} 1, & \text{if } M_j \text{ sells } d_i \text{ units of liquidity to } T_i, \\ 0, & \text{otherwise.} \end{cases} \tag{1}$$

The incentive mechanism also computes the payment for each buyer and seller. A winning buyer T_i pays p_i^b for buying d_i units of inbound liquidity, and a winning seller M_j receives p_j^s for selling one unit of inbound liquidity. The total payment for seller M_j is $p_j^s \sum_{T_i \in \mathcal{T}} x_{ij} d_i$. Because a seller's sold liquidity cannot exceed its supply, we have $\sum_{T_i \in \mathcal{T}} x_{ij} d_i \leq s_j, \forall M_j \in \mathcal{M}$. Due to demand indivisibility, we have $\sum_{T_i \in \mathcal{T}} x_{ij} \leq 1, \forall T_i \in \mathcal{T}$. The utility of a buyer T_i is defined as follows:

$$u_i^b = \begin{cases} v_i^b - p_i^b, & \text{if } T_i \text{ wins,} \\ 0, & \text{otherwise.} \end{cases} \tag{2}$$

The utility of a seller M_j is defined as follows:

$$u_j^s = (p_j^s - v_j^s) \sum_{T_i \in \mathcal{T}} x_{ij} d_i, \tag{3}$$

D. Desired Properties

There are several desired properties for an incentive mechanism to satisfy:

- *Truthfulness*: an incentive mechanism is truthful if each buyer or seller obtains the highest utility by biding its true valuation of the resource.
- *Individual Rationality*: an incentive mechanism is individually rational if all buyers and sellers have non-negative utilities by revealing their true valuations.
- *Budget Balance*: an incentive mechanism is budget balanced if the auctioneer's profit is nonnegative, *i.e.*, the difference between the payments charged from buyers and paid to sellers is nonnegative.
- *Computational Efficiency*: an incentive mechanism is computationally efficient if it can be conducted within polynomial time.

IV. AN INCENTIVE MECHANISM FOR CRYPTO CAPITAL COMMITMENT IN CHANNEL LIQUIDITY MARKETPLACE

In this section, we design and analyze Cumulonimbus, an incentive mechanism for crypto capital commitment in payment channel networks. We first provide the high-level overview and intuition behind Cumulonimbus, and then follow the design goals that are outlined in Section III-D to provide a detailed incentive mechanism description.

A. Overview

Cumulonimbus consists of two stages: the winner selection stage and the pricing stage. The winner selection stage applies a linear-program-based mechanism, which introduces a virtual buyer to intensify the competition on the buyer side and guarantee budget balance. It first determines the winning buyers, and then determines the winning sellers. In the pricing stage, the incentive mechanism finds the Vickrey–Clarke–Groves (VCG) [9] payment for each winning buyer and seller to guarantee truthfulness. We present the detailed incentive mechanism in the following subsections.

B. Social Welfare Maximization

To motivate the participation of both buyers and sellers, we focus on maximizing the social welfare, *i.e.,* the summation of the payoff of the auctioneer and the utilities of all the buyers and sellers. If all the buyers and sellers bid truthfully,

Algorithm 1: Cumulonimbus-Buyer Selection

Input: a buyer set $\mathcal{T}$, a seller set $\mathcal{M}$, and a virtual buyer T_q
Output: a winning buyer set $\mathcal{T}_w$

1 $b_q \leftarrow \infty$; $d_q \leftarrow \max\{s_j | M_j \in \mathcal{M}\}$; $\mathcal{T}_w \leftarrow \emptyset$;
2 $T_q \leftarrow$ the virtual buyer associated with b_q and d_q;
3 Solve linear program $\hat{P}(\mathcal{T} \cup \{T_q\}, \mathcal{M})$;
4 **for** $(T_i, M_j) \in \mathcal{T} \times \mathcal{M}$ **do**
5 | **if** $\hat{x}_{ij} = 1$ **then** $\mathcal{T}_w \leftarrow \mathcal{T}_w \cup \{T_i\}$;
6 **end**
7 **return** $\mathcal{T}_w$

the maximal social welfare $W(\mathcal{T}, \mathcal{M})$ can be solved by the following integer program $P(\mathcal{T}, \mathcal{M})$:

$$\text{maximize } W(\mathcal{T}, \mathcal{M}) = \sum_{T_i \in \mathcal{T}} \sum_{M_j \in \mathcal{M}} (b_i - a_j d_i) x_{ij} \quad (4)$$

$$\text{s.t. } x_{ij} \in \{0, 1\}, \forall T_i \in \mathcal{T}, \forall M_j \in \mathcal{M}, \quad (5)$$

$$0 \leq \sum_{M_j \in \mathcal{M}} x_{ij} \leq 1, \forall T_i \in \mathcal{T}, \quad (6)$$

$$0 \leq \sum_{T_i \in \mathcal{T}} x_{ij} d_i \leq s_j, \forall M_j \in \mathcal{M}. \quad (7)$$

The first constraint represents that T_i either buys d_i units of liquidity from M_j or none. The second constraint represents demand indivisibility, such that T_i can buy liquidity from at most one seller. The third constraint represents supply divisibility, such that a seller's sold liquidity cannot exceed its supply. The social welfare maximization problem $P(\mathcal{T}, \mathcal{M})$ can be proven NP-hard by reducing the demand matching problem, which has been proved NP-hard [21].

Theorem 1. *The social welfare maximization problem $P(\mathcal{T}, \mathcal{M})$ is NP-hard.*

In order to design a computationally efficient mechanism, we can only resort $P(\mathcal{T}, \mathcal{M})$ to its linear relaxation formulation $\hat{P}(\mathcal{T}, \mathcal{M})$, where $\hat{x}_{i,j}$ is a fractional variable that represents the percentage of T_i's demand fulfilled by M_j:

$$\text{maximize } \hat{W}(\mathcal{T}, \mathcal{M}) = \sum_{T_i \in \mathcal{T}} \sum_{M_j \in \mathcal{M}} (b_i - a_j d_i) \hat{x}_{ij} \quad (8)$$

$$\text{s.t. } 0 \leq \hat{x}_{ij} \leq 1, \forall T_i \in \mathcal{T}, \forall M_j \in \mathcal{M}, \quad (9)$$

$$0 \leq \sum_{M_j \in \mathcal{M}} \hat{x}_{ij} \leq 1, \forall T_i \in \mathcal{T}, \quad (10)$$

$$0 \leq \sum_{T_i \in \mathcal{T}} \hat{x}_{ij} d_i \leq s_j, \forall M_j \in \mathcal{M}. \quad (11)$$

C. Incentive Mechanism Design

In this section, we describe the details of Cumulonimbus, which are illustrated in Algorithms 1, 2, 3, and 4.

Cumulonimbus-Buyer Selection (Algorithm 1) determines the winning buyers. In order to guarantee budget balance, Algorithm 1 introduces a virtual buyer to intensify the competition on the buyer side. The intuition behind the virtual buyer is to create imbalances between the supply availability and demand requirement [13]. Therefore, a virtual buyer T_q should have the largest demand and an unlimited bid, such that it can always occupy the largest liquidity supply. Specifically, we set T_q's bid as ∞ and set T_q's demand as $d_q = \max\{s_j | M_j \in \mathcal{M}\}$, which is the largest seller supply (Lines 1 to 2). Then, Algorithm 1 solves the linear program $\hat{P}(\mathcal{T} \cup \{T_q\}, \mathcal{M})$ with the virtual buyer T_q, the buyer set $\mathcal{T}$, and the seller set $\mathcal{M}$ (Line 3). In the solution of $\hat{P}(\mathcal{T} \cup \{T_q\}, \mathcal{M})$, if a buyer T_i's liquidity demand is fulfilled by exactly one seller, *i.e.*, $\exists M_j \in \mathcal{M}, \hat{x}_{ij} = 1$, then T_i wins (Line 5). Algorithm 1 generates the winning buyer set $\mathcal{T}_w$ by checking $\hat{x}_{i,j}$ for each $T_i \in \mathcal{T}$ and $M_j \in \mathcal{M}$.

Algorithm 2: Cumulonimbus-Seller Selection

Input: a winning buyer set $\mathcal{T}_w$, and a seller set $\mathcal{M}$
Output: a winning seller set $\mathcal{M}_w$

1 Solve linear program $\hat{P}(\mathcal{T}_w, \mathcal{M})$;
2 $\mathcal{M}_w \leftarrow \emptyset$; $\mathcal{T}_w \leftarrow \emptyset$;
3 **for** $(T_i, M_j) \in \mathcal{T} \times \mathcal{M}$ **do**
4 | **if** $\hat{x}_{ij} = 1$ **then** $\mathcal{M}_w \leftarrow \mathcal{M}_w \cup \{M_j\}$; $\mathcal{T}_w \leftarrow \mathcal{T}_w \cup \{T_i\}$;
5 **end**
6 **return** $\mathcal{M}_w$

Algorithm 3: Cumulonimbus-Buyer Pricing

Input: a buyer set $\mathcal{T}$, a seller set $\mathcal{M}$, and a winning buyer set $\mathcal{T}_w$
Output: the payments for all buyers

1 **for** $T_i \in \mathcal{T}$ **do** $p_i^b \leftarrow 0$;
2 **for** $T_i \in \mathcal{T}_w$ **do**
3 | Solve linear program $\hat{P}(\mathcal{T} \cup \{T_q\} \backslash \{T_i\}, \mathcal{M})$;
4 | $p_i^b \leftarrow b_i - \hat{W}^*(\mathcal{T} \cup \{T_q\}, \mathcal{M}) + \hat{W}^*(\mathcal{T} \cup \{T_q\} \backslash \{T_i\}, \mathcal{M})$;
5 **end**
6 **return** $(p_1^b, \ldots, p_i^b, \ldots, p_n^b)$

Cumulonimbus-Seller Selection (Algorithm 2) determines the winning sellers, similarly as Algorithm 1. The main difference is that Algorithm 2 determines the winning sellers by solving the linear program $\hat{P}(\mathcal{T}_w, \mathcal{M})$ with the winning buyer set $\mathcal{T}_w$ and the seller set $\mathcal{M}$ (Line 1). In the solution of $\hat{P}(\mathcal{T}_w, \mathcal{M})$, if a seller M_j's liquidity supply can fully satisfy any buyer, *i.e.*, $\exists T_i \in \mathcal{T}, \hat{x}_{ij} = 1$, then M_j wins.

Cumulonimbus-Buyer Pricing (Algorithm 3) determines the payments for all the buyers. In order to guarantee truthfulness, Algorithm 3 computes the VCG [9] payment for each buyer. The intuition behind VCG is to charge each winning buyer how much its participation hurts the others. Thus, for each winning buyer $T_i \in \mathcal{T}_w$, Algorithm 3 solves the linear program $\hat{P}(\mathcal{T} \cup \{T_q\} \backslash \{T_i\}, \mathcal{M})$ with the virtual buyer T_q, the buyer set $\mathcal{T}$ excluding T_i, and the seller set $\mathcal{M}$ (Line 3). The difference between $\hat{W}(\mathcal{T} \cup \{T_q\} \backslash \{T_i\}, \mathcal{M})$ and $\hat{W}(\mathcal{T} \cup \{T_q\}, \mathcal{M}) - b_i$ is the payment for a winning buyer T_i (Line 4). A losing buyer's payment is 0 (Line 1).

Cumulonimbus-Seller Pricing (Algorithm 4) computes the sellers' payments, similarly as Algorithm 3. The main difference is that, for each winning buyer $M_j \in \mathcal{M}_w$, it solves the linear program $\hat{P}(\mathcal{T}_w, \mathcal{M} \backslash \{M_j\})$ with the winning buyer

Algorithm 4: Cumulonimbus-Seller Pricing

Input: a winning buyer set $\mathcal{T}_w$, a seller set $\mathcal{M}$, and a winning buyer set $\mathcal{T}_w$

Output: the payments for all sellers

1 **for** $M_j \in \mathcal{M}$ **do** $p_i^s \leftarrow 0$;

2 **for** $M_j \in \mathcal{M}_w$ **do**

3 Solve linear program $\hat{P}(\mathcal{T}_w, \mathcal{M}\backslash\{M_j\})$;

4 $p_j^s \leftarrow \frac{a_j \sum_{T_i \in \mathcal{T}} d_i \hat{x}_{ij} + \hat{W}^*(\mathcal{T}_w, \mathcal{M}) - \hat{W}^*(\mathcal{T}_w, \mathcal{M}\backslash\{M_j\})}{\sum_{T_i \in \mathcal{T}} d_i \hat{x}_{ij}}$;

5 **end**

6 **return** $(p_1^s, \ldots, p_j^s, \ldots, p_n^m)$

set $\mathcal{T}_w$ and the seller set $\mathcal{M}$ excluding M_j (Line 3).

D. Analysis

We prove that Cumulonimbus satisfies the desired properties introduced in Section III-D.

Theorem 2. Cumulonimbus *satisfies truthfulness, individual rationality, budget balance, and computational efficiency for both buyers and sellers.*

We prove Theorem 2 by the following lemmas.

Lemma 1. Cumulonimbus *is truthful for buyers.*

Proof. We prove that each buyer obtains the highest utility by biding its true valuation of the liquidity. Assume there exists a buyer T_i, whose utility is higher when bidding $b_i' \neq v_i^b$. Let $u^{*b}{}_i$ and $u'^b{}_i$ denote T_i's utilities, and let $\hat{W}^*(\mathcal{T}\cup\{T_q\}, \mathcal{M})$ and $\hat{W}'(\mathcal{T}\cup\{T_q\}, \mathcal{M})$ denote the solutions of $\hat{P}(\mathcal{T}\cup\{T_q\}, \mathcal{M})$, when T_i bids v_i^b and b_i', respectively. Since $p_i^b = v_i^b - \hat{W}(\mathcal{T}\cup\{T_q\}, \mathcal{M}) + \hat{W}(\mathcal{T}\cup\{T_q\}\backslash\{T_i\}, \mathcal{M})$, we have $u_i^b = v_i^b - p_i^b = \hat{W}(\mathcal{T}\cup\{T_q\}, \mathcal{M}) - \hat{W}(\mathcal{T}\cup\{T_q\}\backslash\{T_i\}, \mathcal{M})$ according to Equation (2). Since $u^{*b}{}_i < u'^b{}_i$, we have $\hat{W}^*(\mathcal{T}\cup\{T_q\}, \mathcal{M}) - \hat{W}^*(\mathcal{T}\cup\{T_q\}\backslash\{T_i\}, \mathcal{M}) < \hat{W}'(\mathcal{T}\cup\{T_q\}, \mathcal{M}) - \hat{W}^*(\mathcal{T}\cup\{T_q\}\backslash\{T_i\}, \mathcal{M})$. Thus, $\hat{W}^*(\mathcal{T}\cup\{T_q\}, \mathcal{M}) < \hat{W}'(\mathcal{T}\cup\{T_q\}, \mathcal{M})$. This contradicts that $\hat{W}^*(\mathcal{T}\cup\{T_q\}, \mathcal{M})$ is the optimal solution that maximizes $\hat{W}(\mathcal{T}\cup\{T_q\}, \mathcal{M})$. Therefore, if a buyer bids the true valuation of the liquidity, its utility will not be less than that when it lies. ■

Lemma 2. Cumulonimbus *is individually rational for buyers.*

Proof. Assume that each buyer T_i bids truthfully, *i.e.*, $b_i = v_i^b$. For each winning buyer T_i, its payment is $p_i^b = v_i^b - \hat{W}(\mathcal{T}\cup\{T_q\}, \mathcal{M}) + \hat{W}(\mathcal{T}\cup\{T_q\}\backslash\{T_i\}, \mathcal{M})$. According to Equation (2), we have $u_i^b = v_i^b - p_i^b = \hat{W}(\mathcal{T}\cup\{T_q\}, \mathcal{M}) \geq 0 - \hat{W}(\mathcal{T}\cup\{T_q\}\backslash\{T_i\}, \mathcal{M})$. Therefore, $u_i^b \geq 0$, for all winning buyers. For a losing buyer, $u_i^b = 0$. Thus, $u_i^b \geq 0$. Cumulonimbus is individually rational for all buyers. ■

Lemma 3. Cumulonimbus *is truthful for sellers.*

Lemma 4. Cumulonimbus *is individually rational for sellers.*

Lemmas 3 and 4 can be proved similarly as Lemmas 1 and 2. It is trivial to prove the computational efficiency. We focus on proving budget balance in the following.

Lemma 5. Cumulonimbus *is budget balanced.*

Proof. We first consider the buyer side and calculate a lower bound on the payments charged from the buyers. Let $a_{[k]}$ denote the seller's ask bid for the k-th lowest liquidity unit. Thus, the lower bound of all buyer payments is $a_{[\sum_{T_i\in\mathcal{T}_w} d_i + d_q]} \sum_{T_i\in\mathcal{T}_w} d_i$. Then, we consider the seller side and calculate an upper bound on the payments paid to the sellers. Because the highest price is no more than $a_{[\sum_{T_i\in\mathcal{T}_w} d_i]}$, the upper bound is $a_{[\sum_{T_i\in\mathcal{T}_w} d_i]} \sum_{T_i\in\mathcal{T}_w} d_i$. Since $a_{[\sum_{T_i\in\mathcal{T}_w} d_i + d_q]} \geq a_{[\sum_{T_i\in\mathcal{T}_w} d_i]}$, the difference between the payments charged from buyers and paid to sellers is nonnegative. Thus, Cumulonimbus is budget balanced. ■

V. Performance Evaluation

In this section, we evaluate the performance of Cumulonimbus. As we surveyed in Section II, there is no existing truthful incentive mechanism designed for crypto capital commitment. Therefore, we demonstrate the effectiveness of Cumulonimbus by comparing it to Lightning Pool [18], a state-of-the-art channel liquidity marketplace (CLM) mechanism. Lightning Pool adopts a greedy matching algorithm to select liquidity takers and makers and adopts a uniform pricing rule.

A. Environment Setup

We use the liquidity demand information from the Bitcoin Lightning Network [8], which is the most widely used real-world PCN. In particular, we crawled a snapshot topology of the Lightning Network on September 6, 2021. To crawl the Lightning Network, we ran the Bitcoin Core daemon (bitcoind) [1], built a c-lightning [3] node on mainnet, and connected it to an existing Lightning node, which is the Bitstamp's Lightning Network node [2]. The network consists of $15,413$ nodes and $65,505$ channels. We use the liquidity supply information from the Bitcoin [11], which is the most widely used real-world blockchain-based cryptocurrency.

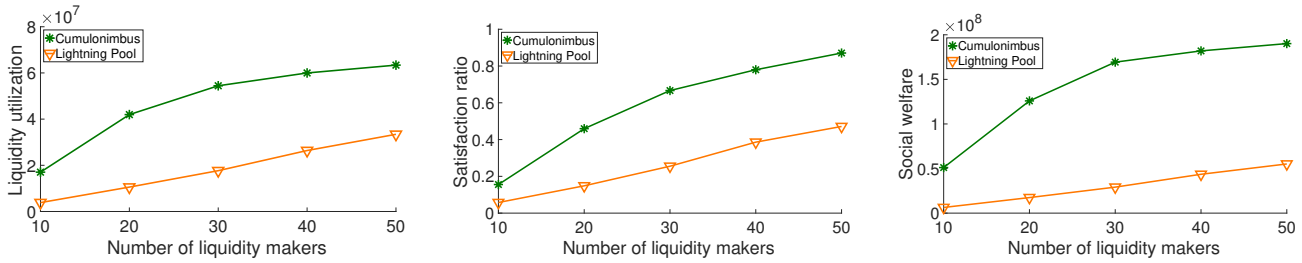
B. Performance Metrics

We use the following metrics for performance evaluation:

- **Liquidity utilization:** The total of liquidity allocated from liquidity makers to liquidity takers.
- **Satisfaction ratio:** The percentage of liquidity takers whose inbound liquidity demands are fully satisfied.
- **Social welfare:** The summation of the auctioneer's payoff and all the participants' utilities.

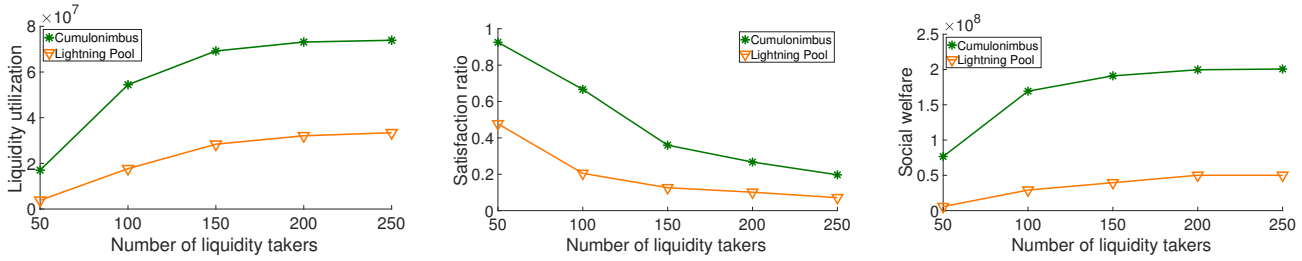
C. Evaluation of Cumulonimbus

Fig. 2 shows the impact of the number of liquidity makers on the liquidity utilizations, satisfaction ratios, and social welfares of Cumulonimbus and Lightning Pool. The number of liquidity takers is 100 and the number of liquidity makers varies from 10 to 50 with an increment of 10. It can be observed that Cumulonimbus outperforms Lightning Pool due to its demand indivisibility and supply divisibility guarantees. We can witness the growing gap between Cumulonimbus and Lightning Pool, which indicates that the greedy allocation



(a) Liquidity utilization (b) Satisfaction ratio (c) Social welfare

Fig. 2. Impact of number of liquidity makers on Cumulonimbus and Lightning Pool.



(a) Liquidity utilization (b) Satisfaction ratio (c) Social welfare

Fig. 3. Impact of number of liquidity takers on Cumulonimbus and Lightning Pool.

without considering the demand indivisibility and supply divisibility decreases the satisfaction ratio and liquidity utilization significantly. Cumulonimbus outperforms Lightning Pool due to its maximization on social welfare. Lightning Pool gives a lower social welfare, because it applies a uniform pricing rule.

Fig. 3 shows the impact of the number of liquidity takers on the liquidity utilizations, satisfaction ratios, and social welfares of Cumulonimbus and Lightning Pool. The number of liquidity makers is 50 and the number of liquidity takers varies from 50 to 250 with an increment of 50. In Fig. 3(b), we can observe that both mechanisms have dropping satisfaction ratios with more liquidity takers, because the liquidity makers cannot provide sufficient supply to satisfy all the liquidity takers. Fig. 3(a) shows that Lightning Pool gives a lower liquidity utilization, because it adopts greedy allocation without considering demand indivisibility and supply divisibility. Fig. 3(c) indicates that Cumulonimbus outperforms Lightning Pool, because Cumulonimbus aims to maximize social welfare.

VI. Conclusion

In this paper, we designed an incentive mechanism Cumulonimbus for crypto capital commitment in PCNs, which motivates participants to sell and buy inbound liquidity, while guaranteeing demand indivisibility and supply divisibility. We analyzed Cumulonimbus and proved that it satisfies truthfulness, individual rationality, budget balance, and computational efficiency. Extensive simulations demonstrated that Cumulonimbus achieved outstanding satisfaction ratio, liquidity utilization, and social welfare compared to a state-of-the-art mechanism Lightning Pool.

References

[1] "Bitcoin core daemon (bitcoind)." [Online]. Available: https://github.com/bitcoin/bitcoin/

[2] "Bitstamp's lightning network node." [Online]. Available: https://www.bitstamp.net/lightning-network-node/

[3] "c-lightning daemon." [Online]. Available: https://github.com/ElementsProject/lightning/tree/master/lightningd/

[4] "Ethereum project." [Online]. Available: https://www.ethereum.org/

[5] "Lightningto.me." [Online]. Available: https://lightningto.me/

[6] "lnbig." [Online]. Available: https://lnbig.com/

[7] "Raiden network." [Online]. Available: https://raiden.network/

[8] "The Lightning Network." [Online]. Available: https://lightning.network/

[9] M. Babaioff and N. Nisan, "Concurrent auctions across the supply chain," *Journal of Artificial Intelligence Research*, vol. 21, pp. 595–629, 2004.

[10] Bitrefill, "Thor: Lightning channel-opening service." [Online]. Available: https://www.bitrefill.com/thor-lightning-network-channels/?hl=en

[11] Blockchair, "Bitcoin addresses and balances." [Online]. Available: https://gz.blockchair.com/bitcoin/addresses/

[12] A. Bosworth, "Y'alls." [Online]. Available: https://yalls.org/about/

[13] L. Y. Chu, "Truthful bundle/multiunit double auctions," *Management Science*, vol. 55, no. 7, pp. 1184–1198, 2009.

[14] C. Decker and R. Wattenhofer, "A fast and scalable payment network with bitcoin duplex micropayment channels," in *SSS*. Springer, 2015, pp. 3–18.

[15] M. Dong, Q. Liang, X. Li, and J. Liu, "Celer network: Bring internet scale to every blockchain," *arXiv preprint arXiv:1810.00037*, 2018.

[16] L. Labs, "Lightning pool." [Online]. Available: {https://lightning.engineering/pool/}

[17] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system." Working Paper, 2008.

[18] O. Osuntokun, C. Fromknecht, W. Paulino, O. Gugger, and J. Halseth, "Lightning pool: A non-custodial channel lease marketplace," 2020.

[19] J. Poon and T. Dryja, "The bitcoin lightning network: Scalable off-chain instant payments," 2016. [Online]. Available: lightning.network/lightning-network-paper.pdf

[20] F. Reid and M. Harrigan, "An analysis of anonymity in the bitcoin system," in *Security and privacy in social networks*. Springer, 2013, pp. 197–223.

[21] F. B. Shepherd and A. Vetta, "The demand-matching problem," *Mathematics of Operations Research*, vol. 32, no. 3, pp. 563–578, 2007.

[22] T. Walther, "An optimization model for multi-asset batch auctions with uniform clearing prices," in *Operations Research Proceedings 2018*. Springer, 2019, pp. 225–231.

[23] ZmnSCPxj, "Towards a market for liquidity providers – enforcing minimum channel lifetime." [Online]. Available: https://lists.linuxfoundation.org/pipermail/lightning-dev/2018-November/001555.html