Engineering Economics in the Conflux Network

Yuxi Cai
Dept. of Electrical and Computer Engineering
University of Toronto
Toronto, ON, Canada
caiyuxi@ece.utoronto.ca

Fan Long
Dept. of Computer Science
University of Toronto
Toronto, ON, Canada
fanl@cs.toronto.edu

Andreas Park
Rotman School of Management
University of Toronto Mississauga
Toronto, ON, Canada
andreas.park@rotman.utoronto.ca

Andreas Veneris
Dept. of Electrical and Computer Engineering
Dept. of Computer Science
University of Toronto
Toronto, ON, Canada
veneris@eecg.toronto.edu

***Abstract*—Proof-of-work blockchains need to be carefully designed so as to create the proper incentives for miners to faithfully maintain the network in a sustainable way. This paper describes how the economic engineering of the Conflux Network, a high throughput proof-of-work blockchain, leads to sound economic incentives that support desirable and sustainable mining behavior. In detail, this paper parameterizes the level of income, and thus network security, that Conflux can generate, and it describes how this depends on user behavior and "policy variables" such as block and interest inflation. It also discusses how the underlying economic engineering design makes the Conflux Network resilient against double spending and selfish mining attacks.**

***Index Terms*—Token Economy, economic design, miner incentives.**

I. Introduction

Blockchain technology allows peer-to-peer electronic value transfers without the involvement of trusted third parties. Trust about the completion of financial transactions in the traditional world of finance rests on the economic principle that the trusted (third) party has too much to lose from negligence or cheating (e.g., regulation penalties, loss of reputation, reduced future income revenue streams, etc). Blockchain networks like Bitcoin [1] and Ethereum [2] use a different mechanism that decentralizes the financial ledger among all participants of the network.

Both Bitcoin and Ethereum employ Proof-of-Work (PoW) schemes to secure the networks and to defend against Sybil attacks [3]. In PoW, miners compete to solve a cryptographic puzzle that requires excessive computational random guessing (aka "work"). The winner has the right to generate a new block and receives a reward for generating the block in the native crypto-currency. The PoW mechanism accomplishes several things simultaneously: it creates consensus as to who proposes a new block, and it introduces sufficient uncertainty as to who gets to propose one next. This implicit randomness is subject to resource expenditures, *i.e.,* the more one spends/works, the more likely that person wins. Since PoW involves expenditure of resources, the rewards that miners receive are directly related to the security of the network: the more miners earn, the more they computationally compete to secure the network.

In PoW, participants in the network agree on the "longest chain" as the transaction history of the blockchain ledger. To make the transaction history secure and irreversible, new blocks are expected to be appended at the end of the longest chain to make it even harder and hence economically costly to revert [1]. Notably, users have to wait for a sufficient number of blocks after the transaction so that the state change is irreversible. This serial processing of blocks puts severe constraints on network throughput, which limits the usability of the platforms in day-to-day real-life monetary transactions.

Aside from the performance challenge of the limited throughput when compared to traditional financial networks (VISA, SWIFT, etc), blockchain networks face two economic challenges. First, the long-term economic sustainability of blockchain networks like Bitcoin and Ethereum remains unclear. Bitcoin is currently secure because miners receive substantial block rewards. Several studies argue, however, that as the block rewards phase out, Bitcoin will be much more vulnerable to double spending attacks [4]. Second, the cost of maintaining a blockchain network grows as the network adds users and transactions. For example, Ethereum supports the deployment of decentralized applications through the execution of "smart contracts." Users pay only a one-time inclusion payment for their smart contract code, but following this, the smart contract occupies state storage without further costs. As such, inactive smart contracts (that is, the majority of smart contracts in Ethereum today) lead to inefficient usage of space and drive up the cost of maintaining the network.

Notably, blockchain networks with sequential ledgers are also vulnerable to fairness attacks. A network participant with more than 23.21% computation power can employ a special block mining strategy to launch selfish mining attacks to obtain block rewards that are disproportional to its computation power [5]. Because PoW mining is a winner-take-all game for miners to compete on including blocks into the longest chain, a malicious participant can strategically withhold some of her mined blocks to gain the advantage of exclusively mining on the longest chain [6]. Such fairness attack strategies put small miners into a disadvantage and may cause the blockchain network to become increasingly centralized, therefore exploiting fairness and undermining the fidelity of the blockchain ledger.

Conflux [7] is a new PoW network with a Turing-complete smart contract language similar to this of Ethereum. Similar to GHOST [8] and PHANTOM [9] protocols, Conflux network provides significant performance improvements with its processing of parallel blocks in a directed acrylic graph (DAG)

978-1-7281-7091-6/20/$31.00 ©2020 IEEE

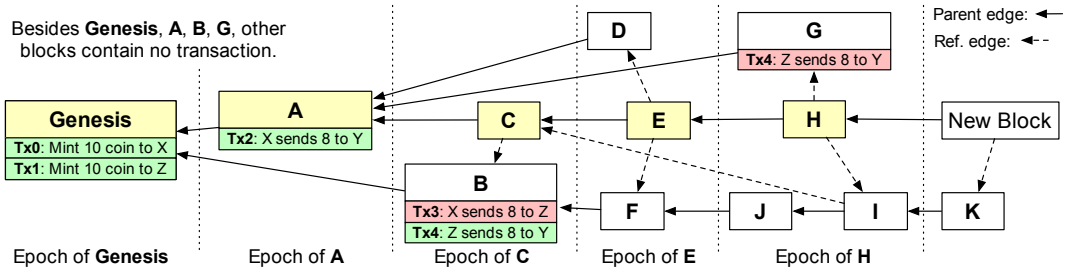


Fig. 1. TreeGraph structure example in Conflux. Yellow blocks corresponds to the pivot chain.

structure, which lowers confirmation times and increases transaction throughput substantially.

This paper focuses on the economic engineering and the incentive mechanism design of the Conflux network. To address the space congestion challenge, Conflux requires users to stake native tokens to bonded storage to occupy space, which implicitly creates a disincentive to occupy space unnecessarily. The disincentive stems from the payment of interest on existing tokens in the system. The interest on the bonded storage is payed to miners instead of the users to create a long term income to the miners. To address the fairness attack challenge, Conflux assigns the block reward in a way that eliminates the winner-take-all characteristic of mining. Instead of competing for the longest chain, miners in Conflux receive block rewards for all the blocks that they generate, albeit with some penalty mechanisms that encourage following the consensus protocol. Competing blocks are jointly penalized so that selfish mining is not profitable and different miners are incentivized to cooperate along the protocol to keep the network stable and secure.

This paper makes the following contributions: 1) we analyze the economic impact of the proposed token rules for Conflux; 2) we show that an optimal selfish mining strategy is not profitable on Conflux; 3) we show that a double-spending attack on Conflux is more difficult compared to legacy blockchain networks with sequential ledgers.

The remainder of the paper is organized as follows. Section II presents Conflux with the focus on its economic and incentive mechanisms. Section III derives a calibrated economic model for miner income to analyze the long term sustainability of Conflux. Section IV shows that Conflux has a stricter requirement for potential attacker than sequential systems. Section V concludes.

II. An Overview of the Conflux Network

The section presents an overview of the Conflux network [7], [10]. Similarly to Ethereum, Conflux operates with an account-based model that every normal account associates with a balance and each smart contract account contains the corresponding byte codes as well as an internal state. Conflux supports a modified version of Solidity (the main contract language in Ethereum) and Ethereum Virtual Machine (EVM) for its smart contracts, so that smart contracts from Ethereum can migrate to Conflux easily.

A transaction in Conflux refers to a message that initiates a payment transaction, or deploys/executes smart contract code. Each block consists of a list of transactions that are verified by the proposing miner. Each node maintains a pool of verified, received transactions that have not yet been included in a block. Miners compete with one another by solving PoW puzzles to include transactions into blocks. Similar to Bitcoin and Ethereum, Conflux adjusts the PoW difficulty so as to maintain a stable block generation rate. Each node also maintains a local state constructed from the received blocks.

A. Consensus with TreeGraph

The Conflux consensus algorithm operates with a special directed acyclic graph (DAG) structure called TreeGraph. Figure 1 presents an example of the TreeGraph structure that the Conflux consensus algorithm uses to organize blocks. Unlike Ethereum which only accepts transactions on a single chain into its ledger, the Conflux consensus algorithm safely incorporates and processes transactions in all concurrent blocks [7], [10]. There are two kinds of edges between blocks, *parent* edges and *reference* edges. Each block (except the genesis) in the TreeGraph has exactly one parent edge to its chosen parent block (*i.e.,* solid edges in Figure 1). Each block can also have multiple reference edges to refer previous blocks (*i.e.,* dotted edges in Figure 1). All parent edges form a tree embedded inside a directed acyclic graph (DAG) of all edges.

At a high level, Conflux uses the novel Greedy Heaviest Adaptive SubTree (GHAST) [10] algorithm, which assigns a weight to each block according to the topologies in the TreeGraph. Under this weight assignment, there is a deterministically heaviest chain within the graph called *pivot chain*, which corresponds to the relatively most stable chain from the genesis to the tip of the parental tree. For example, in Figure 1 the pivot chain contains blocks Genesis, A, C, E, and H. To generate a new block, a miner will choose the last block of the pivot chain as the parent of the new block. The new block will also reference all blocks that have no incoming edge (parent or reference edges) as shown in Figure 1. This is similar to the idea of extending the longest chain. The goal is to make the pivot chain even more stable so that everyone in the network can converge and agree on the same pivot chain.

Parent edges, reference edges, and the pivot chain together enable Conflux to split all the DAG blocks into *epochs*. As shown in Figure 1, every block in the pivot chain corresponds to one epoch. Each epoch contains all blocks that are reachable from the corresponding block in the pivot chain via the combination of parent edges and reference edges and that are not included in previous epochs. Conflux then derives a deterministic total order of blocks as follows: 1) first sort blocks based on epochs (e.g., A is ahead of F); 2) for blocks in the same epoch, sort them based on the topological order (e.g., J is ahead of H); 3) use block id to break ties. Because all

participants will converge and agree on the same pivot chain over time, they will also derive and agree on the same total order of blocks. Participants therefore process all transactions based on the derived block total order. For duplicate and conflicting transactions, Conflux will only process the first occurrence and discard the remaining as no-ops.

Experimental results have shown that Conflux is capable of processing $3,200$ tps for simple payment transactions [7], at least two orders of magnitude higher throughput than Ethereum and Bitcoin. The improvement in throughput is a result of the DAG structure and the consensus algorithm, so that the network can operate with a much faster block generation rate, no forks are discarded, and with a higher utilization of block space. According to the technical specification [10], the main net of Conflux (expected in the second quarter of 2020) will run under a fixed block generation rate at two blocks per second. The daily block generation rate is therefore $60 \cdot 60 \cdot 24 \times 2 = 172,800$ blocks per day.

B. Conflux Token and Interest

There is a unique native token on the Conflux network, hereafter referred to as *CFX*. Each CFX contains 10^{18} *Drip*, the minimum unit of the native token. CFX plays a similar role as the native tokens in the Ethereum networks. Users submit a contract with a gas limit and a gas price where the latter is denominated in CFX.

The issued CFXs exist in two forms: *liquid* and *illiquid*. In the liquid form, they can be immediately transferred/used on the Conflux network while the user does not receive any interest payment. Illiquid tokens cannot be transferred unless they are "unlocked". There are two forms of illiquid tokens:

1) **Locked tokens:** Tokens can be locked up so as to earn the user interest, and
2) **Bonded storage:** Tokens can be put into bonded storage to rent space on the network (*e.g.,* for running smart contracts). The required amount is proportional to the amount of space that the contract occupies.

All illiquid CFXs generate interest in the Conflux network. Users receive tokens from locked tokens. Miners receive the interest payment from bonded storage as maintenance fees for storing contract data.

In this paper, we use r_c for the system base interest rate, expressed in annual terms, and interest is compounded per block. Therefore, a user that stakes for b blocks receives an interest payment of

$$\left(1 + \frac{r_c}{63,072,000}\right)^b - 1$$

per staked token. For instance, if the annual interest is $r_c = 2\%$, a user that stakes for 15,768,000 blocks (around a fiscal quarter) will receive interest of around $.5\%$ per staked token. In calculations, interest payments are rounded *down* to the nearest one (1) drip.

The economic mechanism is straightforward: paying interest leads to an increase in the number of tokens (the "monetary base"). Since users only receive payments from illiquid tokens, the interest payments implicitly shift value from those who do not stake to those who stake.

C. Mining Rewards

Network maintainers (miners) of the Conflux network receive income from three sources: transaction fees, block rewards, and interest income that arises from users "renting" space on the blockchain, as follows.

1) **Transaction Fees:** In the long run, transaction fees will make up the major portion of rewards because of the higher transaction throughput of the network. With many transactions, even very small fees add up to a substantial income.
2) **Block Rewards:** As the common practice in PoW networks, the miner of a block receives a coin-base reward. Over time, these rewards increase the monetary base and lead to inflation. Ignoring any market-driven price changes, economically coin-based rewards are a transfer of wealth from existing CFX holders collectively to the winning miner.
3) **Storage interest:** As mentioned in Section II-B, when tokens are used as bonds for storage, the interest paid on those tokens is passed on to miners. Similar to the block reward, the total amount of interest from bonded storage tokens will be distributed according to the block weights for each miner.

D. Anti-cone Penalty Ratio

The final mining reward of a block is modified by an anti-cone penalty ratio in Conflux. Suppose the base reward of a block b combining transaction fees, block reward, and storage interest payment is B_0. In this paper, we define the final reward of b as:

$$B_0 \cdot \max\left\{0, 1 - \left(\frac{|\mathrm{Anticone}(b)|}{100}\right)^2\right\}$$

In the above, $\mathrm{Anticone}(b)$ denotes the set of blocks that are not in the past sub-graph of b (*i.e.,* reachable via parent and/or reference edges from b) nor in the future sub-graph of b (*i.e.,* reachable via parent and/or reference edges to b). For example, $\mathrm{Anticone}(F) = \{D, G\}$ in Figure 1. Because the anti-cone of a block may keep growing, $\mathrm{Anticone}(b)$ here only includes blocks that are within 10 epochs after the epoch where b resides in. Note that for simplicity, we exclude difficulty adjustment from the consideration of the formula and assume the difficulty remains constant. We refer the interested reader to [10] for a comprehensive description of difficulty adjustment.

For a new block, the base reward is the maximum block reward the generator can possibly receive. For every anti-cone block of the new block, a portion of the block reward will be deducted till zero. Intuitively, this block reward formula encourages the generator to conform with the honest behavior as defined by the consensus protocol. It encourages the generator to refer as many blocks as possible to avoid unreferenced anti-cone blocks. It also encourages the generator to propagate the new block as soon as possible to avoid anti-cone blocks due to network delay. Unlike the winner-take-all mining game for the longest chain in Bitcoin, all blocks in Conflux receive block rewards and miners who cooperate with one another minimize the anti-cone. This makes Conflux secure against selfish mining attacks which exploit the winner-take-all nature of Bitcoin mining [6].

TABLE I
LIST OF SYMBOLS

Symbol	Meaning
G	genesis tokens
D	number of seconds in a day, $60 \times 60 \times 24$
d	days since main-net launch
B	block reward
$b(d)$	block rewards per day
r_b	annual inflation rate from block rewards
$u(d)$	user uptake rate $\in (0, 1)$)
u^{ETH}	estimated user uptake rate based on Ethereum
$u^{\text{fast}}(d),\ u^{\text{slow}}(d)$	u^{ETH} advanced/delayed by 180 days respectively
$T(d)$	number of transactions on day d
f	average transaction fee
$F(d)$	total transaction fees paid to miners on day d
α	fraction of tokens that are locked
r_c	annual rate of inflation due to interest payments
R	daily interest rate for compound transactions
$\gamma(d)$	fraction of gas used by computations
β	required fraction of tokens in bonded storage
$I(d)$	interest income from bonded storage for miners
$p(d)$	inflation adjusted price on day d
$G(d)$	number of coins outstanding on day d
$m(d)$	total revenue for miners on day d
$\bar{m}(d)$	total miner revenue averaged over 1 year

III. CALIBRATED ECONOMIC MODELS

Miners are essential to the security of the network, and the computing power they contribute to secure the network is (empirically) increasing in the revenue that they can earn. In this section, we develop an economic approach to determine the expected revenues that miners gain from participating in Conflux. We calibrate this model based on our technical specification as well as historical data from Ethereum given the similarity in network features. As a reference, Table I outlines the symbols used in this section. Each of the following first three subsection discusses one component of the miner reward and the last subsection presents the overall expectation.

A. Block Rewards

Assuming a constant mining rate of 2 blocks per second [10], there are $2D \cdot 365$ blocks mined per year by Conflux. As such, if B denotes the number of newly minted tokens created as block reward to the miners, the system needs to issue $B \cdot 2D \cdot 365$ new tokens annually as block rewards. Blocks rewards increase the monetary base and create inflation. Specifically, Conflux's objective is to set the block reward based on an annual inflation target rate of $r_b \in (0, 1)$. Therefore, for target value r_b, the block reward must solve

$$B \cdot 2D \cdot 365 \equiv G \cdot r_b \quad \Leftrightarrow \quad B = \frac{Gr_b}{730D}.$$

Overall, on any given day d, total block reward $b(d)$ is:

$$b(d) = Gr_b/365. \tag{1}$$

B. User Uptake and Transaction Fees

To model the expected transaction fees, we first develop a model for the user uptake rate modeling after Ethereum. Network user adoption directly relates to the demand for transactions and smart contract computation, the fees paid by users, and the storage interest distributed to miners.

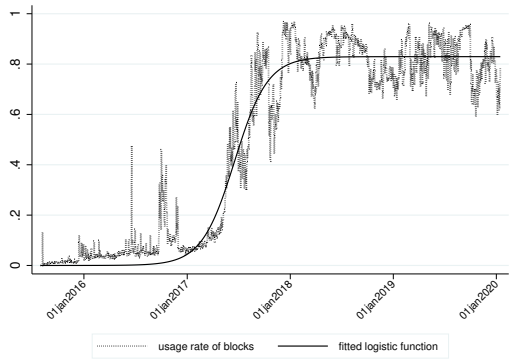


Fig. 2. Adoption of Ethereum: a fitted logistic function

A common feature of new technologies is that their adoption follows a S-shaped pattern with slowly increasing usage early on and then a sudden jump of activity [11]. The user adoption rate in Ethereum, as depicted in Figure 2, indeed shows such a feature. We plot the average fraction of space (or gas) occupied in a block as a function of time, based on data from [12].

We characterize the Ethereum's user uptake sigmoid curve with a logistic function which has the form:

$$y = \frac{\xi_0}{1 + e^{-\xi_1 \cdot (x - x_0)}}, \tag{2}$$

In the equation above, y is the uptake rate at time x, ξ_0 is the maximum value for uptake, ξ_1 is the growth rate, and x_0 is the time-value of when the curve reaches 50% of its maximum value (formally, the value of the horizontal axis at the sigmoid's midpoint). Following the results for the non-linear least squared regression of (2) we obtain a user uptake rate function $u^{\text{ETH}}(d)$ as follows:

$$u^{\text{ETH}}(d) = \frac{0.83}{1 + e^{-0.017 \cdot (d - 690)}}. \tag{3}$$

It is notable that blocks can theoretically be filled up to 100% of the gas limit, yet the estimate for ξ_0 indicates that the Ethereum blockchain's usage rate currently maxes out at 83%. There could be three explanations for this. First, miners may collude so to not include transactions that offer low transaction fees. Next, the 83% usage rate is the "technological" upper bound of what miners can actually include accounting for validation and transaction submission latency. Third, it is possible that once the network becomes congested, users no longer send new transactions to the network because of the long delay; this would create an endogenous upper bound on the demand for transaction processing. Under this estimated model, it will take 718 days until Conflux reaches a network capacity of 50% and 793 days to reach capacity of 70%.

In calibrating our model, we provide an analysis under two different adjustments to the estimated model as the uptake of Conflux may vary from the model described above in terms of the time needed to reach a specific adoption rate. First, we shift the adoption curve 180 days to the right, meaning that adoption is delayed by a quarter. Second, we shift the adoption curve 180 days to the left, meaning that adoption is sped up

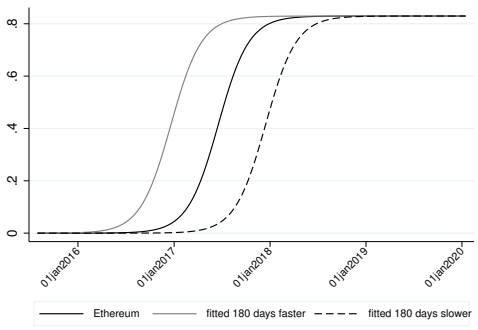


Fig. 3. The Three Calibrated Adoption Rates

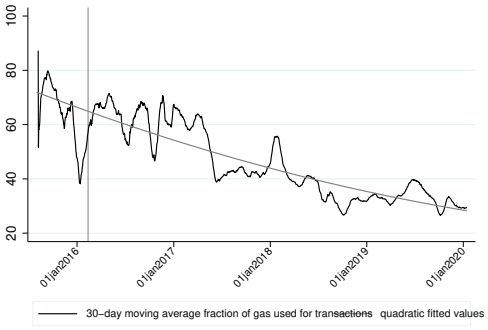


Fig. 4. Transactions vs. Computations

by a quarter. Formally, this shift is an increase/decrease in parameter x_0 to 870 and 510 calendar days, respectively:

$$u^{\text{fast}}(d) = \frac{0.83}{1+e^{-0.017\cdot(d-510)}}, \quad (4)$$

$$u^{\text{slow}}(d) = \frac{0.83}{1+e^{-0.017\cdot(d-870)}}. \quad (5)$$

Figure 3 illustrates the three adoption rate models, labelled as *fast* ($u^{\text{fast}}(d)$), *Ethereum* ($u^{\text{ETH}}(d)$), and *slow* ($u^{\text{slow}}(d)$).

With an uptake rate of $u(d)$, the average daily number of transactions is as follows:

$$T(d) = u(d) \cdot 3,200D.$$

At capacity, Conflux has a throughput of $3,200$ tps. With a long-run adoption rate of $u(d) = 80\%$, this amounts to an expected $2,560$ tps utilization. One can also argue that the adoption rate in Conflux may exceed the above estimates. Ethereum is arguably at capacity most of the time (see Figure 2 and its mem-pool of unsettled transactions is non-empty). Since Ethereum is at capacity, there is a limited incentive for developers to introduce new DApps, especially for enterprise-scale use-cases. Conflux's higher throughput mitigates the concerns that the transactions do not get confirmed timely, and since it is compatible with Solidity, developers face a flat learning curve. Together this should contribute to a fast adoption of Conflux.

To simplify, we denominate the capacity by the number of native token tps. We assume that users on average pay a transaction fee of value f. Therefore, average daily fees, as a function of day d, $F(d)$, are as follows:

$$F(d) := f \times T(d) = f \cdot u(d) \cdot 3,200 \cdot D. \quad (6)$$

For Ethereum, at its current block reward and hash rate, total rewards are on the order of \$2.3M daily or \$840M annually, including both block rewards and transaction fees [13]. As a result, transaction fees account for less than 3% of the rewards. In Conflux, with a similar block-usage rate, transaction fees would provide the same total fee income as the *total* revenue (fees plus block rewards) in Ethereum as long as user are willing to pay on average \$0.01 per transaction, a desirable feat. Even for a moderate willingness of users to pay fees, annual income can be substantial. In comparison, the median transaction fee on the Ethereum blockchain for January–February 2020 has been between \$0.08 and \$0.15 [14].

C. Bonded Storage and Interest payment

To characterize the size of bonded storage, we start with the modeling of transaction fees split by token ownership transfers vs. computation. Figure 4 shows the fraction of gas attributable to address-to-address transfers in percentiles in Ethereum.[1] As the figure shows over time simple ownership transfers account for a decreasing proportion of transactions.

We characterize the computation-rate with an OLS regression for a quadratic fit in the following form:

$$\%\ \textit{computation gas} = \alpha + \beta_1 \cdot d + \beta_2 \cdot d^2 + \epsilon, \quad (7)$$

where d are the number of days since main-net launch. The goal here is to measure the *% computation gas* as a quantity $\in [0, 100]$.

Specifically, let $\gamma(d)$ denote the fraction of gas usage for computation. Following the quadratic fit of data in Figure 4, we obtain:

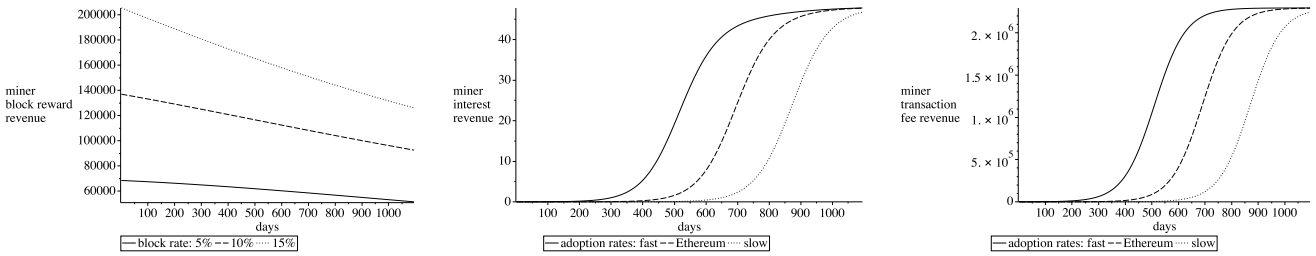
$$\begin{aligned}\gamma(d) &:= 1 - \left(72 - 0.04 \cdot d + 7.05 \cdot 10^{-6} \cdot d^2\right)/100 \\ &= -0.0000000007(d - 2,837)^2 + .85. \end{aligned} \quad (8)$$

We note that the parameter estimated for the quadratic term, β_2, is very small, around 7.05×10^{-6}, owing to the size of the associated variable. Therefore, when there are $T(d)$ transactions on day d, we say that $(1 - \gamma(d)) \cdot T(d)$ of these are token ownership transfers and $\gamma(d) \cdot T(d)$ involve smart contract executions that require data storage on the chain.

To simplify the interest payment estimation, we make the assumption that users make the decision of whether to put tokens into bonded storage each day and, therefore, that the total transactions fully reflect the extent of interest payments. This rules out a possible scenario that a user buys storage (*i.e.,* put tokens into bonded storage) but never executes the contract hereafter. In other words, we account for only "new" bonding of tokens. As such, the calibration model likely *conservatively underestimates* the interest income to miners.

The required bonded storage is proportional to the size of the contract code. We assume that this amount is proportional to the gas usage of the contract or, as one may argue, the number of actual transactions since each of them requires gas.

[1] We derive this line as follows. We obtain from [12] the data series for daily transactions and daily Gas used. A simple transfer of ETH transaction requires 21,000 Gas, and we therefore obtain the computation-driven Gas amount by subtracting the number of transactions times 21,000 from the total gas.



Panel A: Block Rewards *Panel B: Interest Income* *Panel C: Transaction Fee Revenue*

Fig. 5. Miner Revenues over time as a Function of the Adoption Rate

For x transactions, users need to put $\beta \cdot x$ tokens into bonded storage and on day d it is $\gamma(d) \cdot T(d)$ transactions that require bonded storage. In total, the required amount is $\beta \cdot \gamma(d) \cdot T(d)$. We conclude that each day the miners receiving interest paid on the bonded storage is:

$$I(d) := \beta \cdot \gamma(d) \cdot T(d) \cdot R, \tag{9}$$

where R represents the daily interest rate for compound transactions.

D. Total Miner Revenue

To summarize, total miner revenue, denoted by $m(d)$, consists of *(a)* the block reward from equation (1), *(b)* transaction fees expressed by equation (6), and *(c)* interest income from bonded tokens as shown by equation (9):

$$m(d) \quad = \quad p(d) \cdot b(d) + F(d) + p(d) \cdot I(d). \tag{10}$$

Absent exogenous forces that affect the market price, the price of CFX token on day d, $p(d)$, is determined simply by the total number of tokens outstanding, $p(d)$ =initial price $\times$ genesis tokens/(genesis tokens + block rewards + interest payments).

Before we present our calibration results for mining revenue on Conflux, as a benchmark we set how much Ethereum miners earn. There are around 6,500 blocks created per day, paying around 13,500 ETH so that the total average daily block rewards is around \$3M USD (at current ETH/USD prices) [13].

We use four values for average transaction fees, $f \in \{.005, .01, .02, .08\}$, where the highest number \$.08 corresponds to the low-end median fee paid on Ethereum in early 2020, as we discussed earlier. For the uptake rate, we consider the three benchmark rates $u^{\text{fast}}(d)$, $u^{\text{ETH}}(d)$, and $u^{\text{slow}}(d)$ from Subsection III-B. For the bonded storage requirement, we use $\beta = 1\%$ meaning that if the user occupies space on the blockchain for future computation that is equivalent to what one virtual-machine opcode transaction occupies, then this user has to put 1/100 of a CFX token into bonded storage. We also assume that there are no exogenous market-driven price changes except where explicitly stated.

Figure 5 plots the three components of miner revenue: block rewards, interest income, and transaction fees. These figures use an annual interest payments of $r_c = 2\%$, and average fees of \$.01. The \$-value of block rewards (Panel A) declines because the price declines due to inflation; note that we assume that the number of tokens given as a block reward is constant within the interval. For the remaining two panels, we set the annual block inflation rate to $r_b = 5\%$. Interest income (Panel B) rises with blockchain usage, but it is small in magnitude.

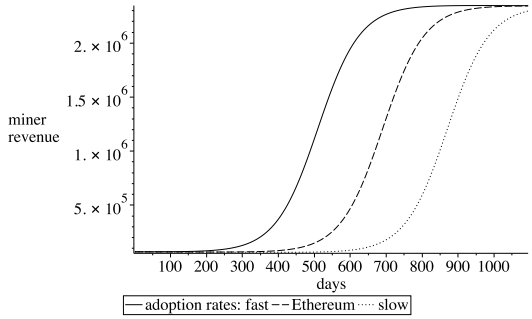


Fig. 6. Miner Revenues over time as a Function of the Adoption Rate

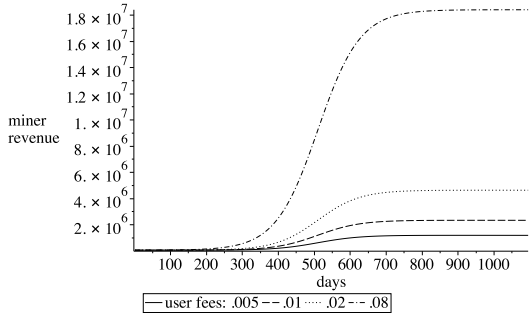


Fig. 7. Miner Revenues over time as a Function of Average Fees

Finally, transaction fee revenue (Panel C) plots fee income. The values recorded on the vertical axis indicates that these fees are expected to be an order of magnitude larger than interest income or block reward income, except immediately after the launch of the main-net.

Combining these three figures, Figure 6 plots expected daily miner revenue $m(d)$ over three years following the launch of the main-net for the three different user uptake speed scenarios. This figure uses an annual block inflation rate of $r_b = 5\%$, annual interest payments of $r_c = 2\%$, and average fees of \$.01.

Figure 7 shows the time series of expected miner revenues per day with the four different average transaction fees. When Conflux is at capacity, even for moderate fees of \$.02, miner revenue will be around \$4.6M. This figure uses block inflation rate of 5%, interest payments of 2%, and Ethereum-like adoption rates.

For the sake of the argument, we also consider a situation

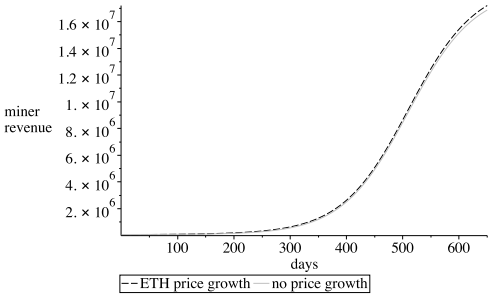


Fig. 8. Miner Revenues if prices would grow to ETH levels

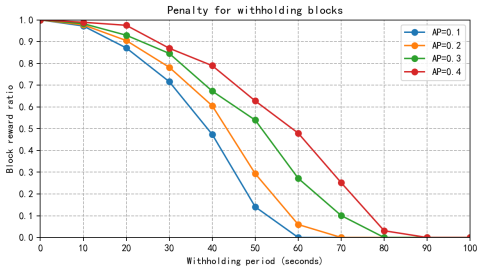


Fig. 9. Penalty of attackers on different attacker ratios of block generation power (AP)

when market forces lead to increases in the price of CFX tokens such that in three years Conflux has the same market valuation as Ethereum today, that is, roughly a $15B market-cap. Further assume that the price change follows linear growth at some rate g such that the price at time d is $p^{\text{ETH}}(d) = p(0) \cdot (1+g)^d$. The rate g that ensures that the market evaluation of Conflux three years after launch is the same as Ethereum at the beginning of 2020 is $g \approx 0.0031$. We compute total miner revenue for the "speculative" price $p^{\text{ETH}}(d)$, i.e., in (10), we substitute $p(d)$ with $p^{\text{ETH}}(d)$ so that:

$$m^{\text{ETH}}(d) \quad = \quad p^{\text{ETH}}(d) \cdot b(d) + p^{\text{ETH}}(d) \cdot I(d) + F(d) \quad (11)$$

Figure 8 shows the time series of expected miner revenues per day for this alternative price path, p^{ETH}, where we plot only the first 650 days. In this Figure, we use a block inflation rate of 5%, interest payments of 2%, Ethereum level adoption rates, and willingness to pay fees at current Ethereum rates ($0.08). We also include the revenue case when there is no price growth (it corresponds to the most "optimistic" case in Figure 7) as a point of reference. The *key* takeaway from this figure is that when we assume that prices rise significantly, miner income in the medium run is not affected, simply because transaction fees continue dominate.

We conclude that early on, block rewards play the most important role in miner income at the beginning, whereas, once a certain adoption rate is reached, transaction fees will be the most important source of income. We emphasize, however, that this is not to say that interest is irrelevant for user decisions. Instead, there will be many users who each have to pay a small but possibly for their case significant implicit fee for storing data on the network.

IV. Economic Limits against Attacks

In this section, we examine the limits of the Conflux network under two different attacks, the selfish mining attack and the double-spending attack.

A. Selfish Mining Attacks

If a participant in Bitcoin holds more than 23.21% of the network computation power, she can gain more mining profit by strategically withholding her mined block for a period of time before broadcasting them to the network [5]. This is because Bitcoin only gives reward to the blocks in the longest chain. When she withholds the newly mined block, she has the exclusive privilege to mine under her new block which is the current longest chain. Of course, withholding the block brings the risk that someone else may mine a new block concurrently to become the new longest chain, but the study shows that if the participant has more than 23.21% of the network computation power, the benefit of withholding will outweighs the risk [5]. Because Bitcoin mining is a winner-take-all game, honest miners expect to get less reward comparing to their computation power when the selfish participant launches such fairness attacks.

Conflux is more resilient against selfish mining attacks because withholding a block leads to less reward. Unlike Bitcoin, all blocks receive a reward in Conflux and the reward of a block is discounted by its anti-cone size. Withholding the block will prevent future blocks from referencing it. Therefore, it increases the anti-cone size of the block and consequently decreases the block reward. Given all network participants are rational, honest mining is incentive compatible.

Figure 9 presents our experimental results to illustrate the resilience of Conflux against selfish mining attacks. We run a Conflux network simulation with 10000 nodes. One of them is the attacker which will withhold her generated block for a certain period of time. In the simulation, normal nodes have the network delay (4.1 seconds in average). The attacker, however, has the capability of instantly receive and send its block to all other nodes. We run the simulation for 2000 blocks and measure the reward ratio the attacker receives comparing to the normal honest strategy for the last 1000 blocks under different the block generation power and the block withholding period. Our results show that the attacker consistently receives less reward than she would with the normal honest strategy (i.e., the reward ratio is less than 1). The longer she withholds the blocks, the less reward she will receive. More computation power will help the attacker to receive more reward, but even with 40% of the computation power of the whole network, the attacker would still get more reward if she just participates the network honestly.

B. Double Spending Attacks

Several works in the economics literature highlight that PoW networks face fundamental constraints in terms of the economic incentives that can sustain ongoing security of the network [15]. The Conflux network is no different but in what follows, we argue that the constraints of Conflux are "looser" when compared to existing networks. In this section, we make the reasonable assumption that an attacker is not capable of reversing cryptographic functions, therefore honest miners behave correctly even with the presence of an attacker. We focus on double-spending attacks with selfish mining through withholding of blocks.

We first repeat the arguments from [16] which apply to serial blockchains. We assume that the mining of each block involves a cost c (including physical equipment and electricity) and that there are N identical miners who compete. For the scenario with negligible user fees, the most significant revenue is the block reward B per block. The miners' participation constraint requires the expected gain to exceed the expected cost, that is:

$$\text{probability of winning the block} \times B \geq \text{cost} \quad \Leftrightarrow \quad B/N \geq c.$$

This condition holds for all identical miners, and in equilibrium it must hold that the aggregate cost of mining agrees with the aggregate benefit:

$$c \times N = B. \tag{12}$$

Now suppose an attacker wants to double-spend a transaction of value V. The attack proceeds in the sense that the attacker builds an alternative chain faster than all remaining miners. Assume that to gain 50% power, the attacker has to pay $c \times N$, and to gain a majority they have to pay in excess of this. If the attacker spends $A \times c \times N$ on equipment, with $A > 1$, they gain an advantage of $A/(A+1) > 50\%$; the larger A, the larger the advantage (and thus the faster they finish the attack). For a successful attack, they earn value V, which is the amount that they can double spend. Assume that, conditional on the equipment advantage A, it takes t blocks (in expectation) to complete the attack, that is creating a longer chain than the chain honest miners collaboratively generating. Then the cost of the attack is:

$$t \times A \times c \times N.$$

Once successful, however, the attacker earns not only the attack value V but also rewards for the t blocks. Therefore, for attacks to be *unattractive*, it must hold that:

$$t \times A \times c \times N > V + t \times B. \tag{13}$$

Using equation (12), we obtain the following:

$$t \times B(A-1) > V. \tag{14}$$

Therefore, for an expected attack time t, there exists a value $\mathcal{V}$ such that for all $V > t \times B(A-1) = \mathcal{V}$, and the transaction of value V cannot be secured. Inequality (14) is a firm constraint on the economics (and the security) of a serial chain such as Bitcoin.

Conflux subjects to a different lower bound for V. First, to be successful in an attack, the attacker's alternative chain must become the pivot chain. Since any epoch may contain multiple blocks, not only the attacker needs to create blocks faster, but also to generate a "heavy" chain, which will require relatively more time (and thus more resources). To simplify the argument, we abstract from this issue and assume, as before, that the honest chain contains a single block per epoch.

Next, when creating the alternative chain, an attacker does not receive the full reward because block rewards are assigned based on the block's anti-cone size. As before, suppose there is a single attacker in the system, who succeeds an attack in t blocks. Assume that the attacker references honest block as soon as one is seen, the attacker's first block in the alternate chain has an anti-cone of size of at least $t-1$, the second of $t-2$, and so forth. Therefore, the block reward for block a since the start of the attack is $B \times \left(1 - (\min\{t-a, 10\}/100)^2\right)$ assuming a fixed per block reward B. For the longest chain (now the pivot chain) of length t since the start of the attack, the attacker will therefore earn:

$$B \cdot \underbrace{\sum_{i=1}^{t} \left(1 - \left(\frac{\min\{t-i, 10\}}{100}\right)^2\right)}_{\Pi_t} < t \times B.$$

Using the same argument as above, and therefore, the economic constraint for Conflux becomes:

$$B(tA - \Pi_t) > V \tag{15}$$

In other words, there exists a value $\mathcal{V}'$ such that for all $V \in (\mathcal{V}, \mathcal{V}']$, the following holds:

$$B(tA - \Pi_t) > V > B(tA - t)$$

The implication of this relationship is that the set of transaction values V that can be secured on the Conflux network is strictly larger than in "traditional" serial blockchains such as Bitcoin under such an attack strategy.

V. Conclusion

The long term sustainability and economic resilience to attacks are critical to a decentralized, proof-of-work blockchain network. When basing our economic calibration on similar uptake and usage of Conflux as of Ethereum, we observe that as adoption increases, the significantly higher throughput of the network allows user fees and storage interest payment to make up the bulk of income for miners, making the mining activity sustainable in the long term. Our analysis results also show that Conflux with its novel incentive mechanism is more resilient when facing double-spending attacks and selfish mining attacks than sequential blockchains.

References

[1] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," 2009. [Online]. Available: http://www.bitcoin.org/bitcoin.pdf

[2] V. Buterin, "Ethereum: A next-generation smart contract and decentralized application platform," 2014. [Online]. Available: https://github.com/ethereum/wiki/wiki/White-Paper

[3] J. Douceur, "The Sybil Attack," in *IPTPS '01 Revised Papers from the First International Workshop on Peer-to-Peer Systems*, 2002, pp. 251–260.

[4] M. Carlsten, H. Kalodner, S. Weinberg, and A. Narayanan, "On the instability of bitcoin without the block reward," in *2016 ACM SIGSAC Conference*, 10 2016, pp. 154–167.

[5] A. Sapirshtein, Y. Sompolinsky, and A. Zohar, "Optimal selfish mining strategies in bitcoin," in *Financial Cryptography and Data Security: 20th International Conference*, 05 2017, pp. 515–532.

[6] I. Eyal and E. Sirer, "Majority is not enough: Bitcoin mining is vulnerable," in *Financial Cryptography and Data Security: 18th International Conference*, vol. 8437, 11 2013.

[7] C. Li, P. Li, D. Zhou, W. Xu, F. Long, and A. Yao, "Scaling nakamoto consensus to thousands of transactions per second," arXiv preprint 1805.03870, 2018.

[8] Y. Sompolinsky and A. Zohar, "Secure high-rate transaction processing in bitcoin," 01 2015.

[9] ——, "Phantom: A scalable blockdag protocol," *IACR Cryptol. ePrint Arch.*, vol. 2018, p. 104, 2018.

[10] C. Li and G. Yang, "Conflux protocol specification," 2020. [Online]. Available: https://confluxnetwork.org/static/Conflux_Protocol_Specification_20200327.pdf

[11] E. M. Rogers, *Diffusion of Innovations*. Simon and Schuster, 2003.

[12] Etherscan. Ethereum average block size chart. [Online]. Available: https://etherscan.io/charts

[13] BitInfoCharts. Ethereum (eth) price stats and information. [Online]. Available: https://bitinfocharts.com/ethereum/

[14] ycharts. Ethereum average transaction fee. [Online]. Available: https://ycharts.com/indicators/ethereum_average_transaction_fee

[15] R. Auer, "Beyond the doomsday economics of "proof-of-work" in cryptocurrencies," BIS Working Papers No 765, Tech. Rep., 2019.

[16] E. B. Budish, "The economic limits of bitcoin and the blockchain," University of Chicago, https://ssrn.com/abstract=3197300, Chicago Booth Research Paper No. 18-07, 2018.