Blockchain-based P2P File Sharing Incentive

Qingzhao Zhang[1], Yijun Leng[1], and Lei Fan[1]

Shanghai Jiao Tong University

Abstract. P2P file sharing systems require proper incentive mechanisms to encourage active data sharing. However, traditional incentives based on reputation, credit or tit-for-tat are still challenged by free riding and whitewashing. We explore solutions based on blockchain, which is the new emerged decentralized trustful public ledger, and propose a blockchain-based file sharing incentive mechanism leveraged by cryptocurrency and smart contracts. In the proposed scheme, a file is sliced into pieces. A user who downloads data will request pieces with randomized order and directly pay for each piece. With the analysis in game theoretic models, rational players intend to cooperate in the procedure. We also evaluate the approach with simulations and experiments.
We envision that our solution is not only promising for P2P file sharing, but also a stepping stone for general data sharing applications over the public blockchain.

1 Introduction

P2P file sharing is one of the most popular P2P applications. P2P file sharing schemes, for instance, BitTorrent [1], Napster and Gnutella [2], managed to attract millions of users. However, P2P network causes potential threats along with its advantages. There are two main challenges confronted by these systems. First, free riding causes unbalance between uploading and downloading, because free riders do not make contributions while downloading resources greedily. Another challenge is whitewashing which means peers can easily discard current identity and generate a series of new ones to continue misbehaving in the network.

To eliminate these two threats, there are lots of successful researches on P2P file sharing and its incentive mechanisms, for example, BitTorrent's tit-for-tat and choke policy [3], reputation managements [4, 5] or credit mechanisms [6]. Also, game theory and model analysis is an efficient approach to study p2p incentive. Papers [7] [8] and [9] used dynamic equation and learning model to analyze p2p incentive mechanisms. From the above approaches, we find that reputation-based and credit-based incentives rely on central servers for efficiency and trustfulness. In the real P2P system, it is hard to find a Third Trusted Party (TTP). This brings difficulties to implement efficient and reliable incentive mechanisms.

Blockchain technology, regarded as also a P2P application, offers potential solutions. In 2008, Nakamoto published his celebrated paper [10] in which introduce a practical blockchain consensus protocol and later was known as *Bitcoin* protocol. Generally, the protocol is a method to organize a trusted ledger which is safeguarded by all peers in the network [11]. Peers use transactions to interact with blockchain. The records confirmed by the ledger cannot be modified anymore so that a consensus is achieved without TTP. The first generation of blockchain is mainly designed for cryptocurrency until smart contract appears.

Smart contracts, which is a script language embedded in blockchain network, leverage more flexible applications. One of the most famous smart contract platforms is *Ethereum* [12]. *Ethereum* leverages Ethereum Virtual Machine (EVM) which executes stack-based Turing-complete language and manipulate blockchain states. But current smart contracts still cannot tolerate heavy computation tasks and storage load. For classic PoW (Proof of Work) blockchain like *Ethereum*, the throughput of transactions is quite slow and also affects the performance of its smart contracts.

Based on blockchain and smart contracts, we notice that cryptocurrencies, as the initial application of blockchain, can be an effective incentive for P2P systems. If uploading files can earn money and downloading costs the currency as well, it is the simplest prototype of blockchain-based pecuniary incentive. Since the cryptocurrency has direct financial value, in intuition, free riding and whitewashing can be eliminated because the expensive cost of downloading. In this way, the on-chain currency transfer and off-chain file data transfer can be combined together to incentive active uploading and punish misbehavior.

What is more, blockchain enhances decentralization and truthfulness of P2P file sharing system. Each operation, which is confirmed in blocks, is not modifiable unless blockchain system breaks down. As a result, any peer can simply parse the blockchain to find truthful history of file sharing behaviors or information about files and peers.

1.1 Related Work

Related works on blockchain-based incentive depict its potential usage in P2P data exchange. Cryptocurrency can be the straightforward and efficient incentive in P2P systems. In the fundamental idea of Bitcoin in 2008, miners, who consume computing power to maintain system functionality, is simply driven by profits. He et al. [13] proposed an incentive mechanism within P2P delivery service which makes use of Bitcoin script to reward related contributors. However, because of the limited efficiency of Turing-incomplete language, the design is limited in scenarios. He et al. [14] proposed a secure validation method and a pricing strategy, and integrated them into the incentive mechanism, through a game theoretical analysis. Dennis and Owen [15] designed a framework of novel reputation system based on blockchain which could be integrated into P2P sharing systems. Kishigami et al. [16] developed blockchain-based digital content distribution system to safeguard the copyrights holder. Shrestha and Vassileva [17] delivered a usable blockchain-based model for collecting and sharing researcher's data.

1.2 Our Contributions

We design a P2P file sharing protocol which leverages blockchain-based incentive. The protocol requires downloaders to pay cryptocurrency to uploaders when requesting data piece by piece. This will incentive rational players to cooperate in the downloading procedure.

1. We introduce the first file sharing system with blockchain-based incentive mechanism for active data sharing.
2. We introduce evolutionary game model to analyze the proposed incentive mechanism formally.
3. We introduce repeated game model and learning model to analyze the potential unfairness and cooperative behavior in the process of file data transfer.

In Section 2, we give an overview of the system and define the problem formulation. Section 3 elaborately depicts the system workflow and protocol design. In Section 4, we use game theoretical model to analyze our protocol. Related experiment result is displayed in Section 5.

2 Problem Formulation

In this section, we will present formulated model of blockchain-based P2P file sharing and its incentive. We will also specify our design goals.

2.1 System Overview

Figure 1 depicts the overview of blockchain-based file sharing system. First, we will explain some objects appeared in Figure 1.

Blockchain. Blockchain constructs a public trusted ledger [11]. Operations on the ledger are leveraged by transactions. First, we assume the blockchain module is always available and reliable to provide trusted storage and computation service. Blockchain can store static data, such as peer properties and file properties in our proposed system. As for computation service, blockchain can execute predefined logic and the execution is reliable and verifiable. Note that blockchain is limited in storage and computation capability. so we should keep on-chain operation light enough.

Peer. Each peer in the system has a unique identity (i.e. address). In blockchain system (e.g. A 160-bit hex string in Ethereum). Each peer in the system can act as a downloader, uploader or both.

Currency. In the system, currency objects are leveraged by cryptocurrency embedded in blockchain system, for example, BTC in Bitcoin and ETH in Ethereum. The currency can be transferred among blockchain addresses.

File piece. In the system, each file is divided into multiple pieces and a piece of data is the basic unit of data transfer.

Conditional payment and *payment token.* Conditional payment is an application of blockchain. One action (transfer of currency) and a condition is prespecified on blockchain. When blockchain is invoked by associated transactions,

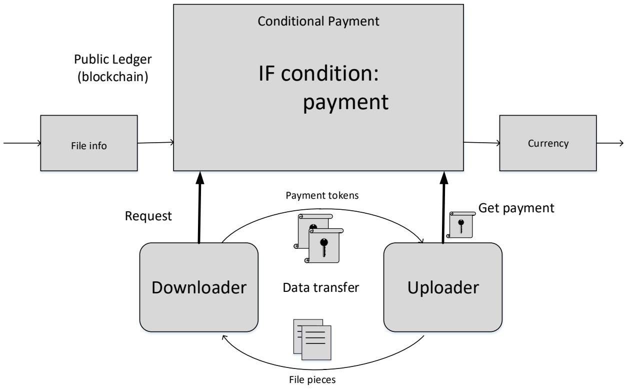


Fig. 1: Model of blockchain-based P2P data sharing. Downloaders and uploaders perform conditional payment on chain and transfer file pieces off-chain.

the action is triggered only if the condition is satisfied. We use *payment token* to notify a piece of data which satisfies the condition of conditional payment.

Downloader sends transactions to blockchain to request data and construct conditional payments. Also, downloaders accept off-chain file data directly from uploaders and give uploaders payment tokens back. Uploader possesses file data and provides downloading service. Uploader receives requests from downloaders and transfers file pieces to downloaders. After receiving downloaders' payment tokens, uploaders are able to invoke conditional payments on-chain to obtain downloaders' payments.

Therefore, when file pieces flow from uploaders to downloaders, currency flows from downloaders to uploaders as well. Note that the whole system doesn't have TTP as arbiter. The fairness of the exchange of file pieces and currency is leveraged by blockchain module, especially its conditional payments on the ledger. In the exchange between downloaders and uploaders, uploaders gain payments from downloaders as while downloaders cost currency for downloading service. Both sides make benefits through cooperation.

2.2 Design Goals

We declare design goals from three aspects: incentive, fairness, and scalability.

1. *Incentive*. Under the incentive mechanism, free riding and whitewashing are discouraged and active sharing is encouraged. The system should have a stable satisfying equilibrium in which a large fraction of system participants would like to share data and download data honestly.

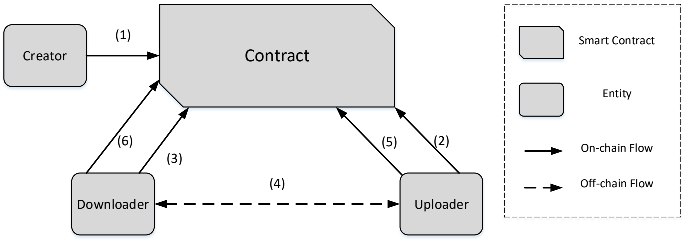


Fig. 2: Workflow of blockchain-based file sharing. (1) Creation (2) Announce (3) Request (4) Transfer (5) Payment (6) Withdraw

2. *Fairness.* It is hard to guarantee strict fairness of interactions between peers without TTP but the unfairness remained in the system should be limited. Also, the unfairness should't affect the normal execution of system protocol and incentive mechanism.
3. *Scalibility.* Because of the limitation of the current blockchain, operations on public ledger should be light enough. For example, no heavy computation and no operations on data of large size. The frequency of transactions required by system protocol should not exceed the limitation of blockchain system. The system should be feasible to implement and stable to use on large scale.

3 System Protocol

We will elaborately illustrate protocol of blockchain-based P2P file sharing systems. Blockchain module discussed in Section 2 is leveraged with smart contracts. Each peer can join the public blockchain network and interact with these contracts for file creation, announcement and download services.

There are two main components of smart contract. One is *Register* which is responsible for file announcement. It is unique in the system and organizes information of files, for example, fingerprints of file pieces and download history. Another is *Transfer*, which hosts conditional payments for uploaders and downloaders to perform the exchange of currency and data pieces. Next, we will present the workflow of file sharing and define data transfer protocol more precisely.

3.1 Workflow

The workflow of the system is depicted in Figure 2. There are three entities: file creator (the first uploader of a specific file), downloader and uploader. A smart contract is located on public blockchain system. Before the workflow starts, all entities have done preparation work: register the system and submit some amount of monetary tokens as a deposit. There are six steps in the workflow.

1. *Creation.* File creator announces a new file and uploads related file information to blockchain. For example filename and fingerprint of each piece, etc. This creator is the first peer who becomes uploader of this file.
2. *Announce.* Uploader announces his ownership of file pieces. He publishes the identity of a specific file in his possession to blockchain.
3. *Request.* Downloader requests uploader for data. Downloader sends a request transaction to contract and specify which file to download and which uploader to download from. Meanwhile, the contract constructs a conditional payment for both sides and a fund is transferred from downloader deposit to conditional payment's fund pool. Note the conditional payment sets a timeout property.
4. *Transfer.* Both sides set up TCP connection and perform the exchange of file pieces and payment tokens. Through the exchange process, downloader receives one file piece first and responds uploader with corresponding payment token. Then both sides repeat the procedure to transfer next file piece.
5. *Payment.* Uploader invokes conditional payment within its timeout using payment tokens he has. Payment is transferred from fund pool of conditional payment to uploader deposit if the condition is satisfied.
6. *Withdraw.* After conditional payment's timeout, downloader can cancel the request and regain the fund remained in the fund pool of conditional payment.

3.2 Data Transfer Process

In step 4 of workflow, downloader and uploader set up TCP connection. Later both sides should follow data transfer protocol (defined in Protocol 3.2) to complete the download. The protocol is quite straightforward: Downloader tells uploader which piece to download first. Uploader than transferring the piece to the downloader. Downloader responds with payment token and this process can be repeated until any side aborts, an error occurs or timeout.

To construct the payment token and secure the messages, each downloader or uploader should have a keypair for signature, in which public key is published on chain and private key is preserved confidentially. We can directly use original signature keypair in blockchain system.

We denotes following variables for convenience. P_d: Downloader; P_u: Uploader; (pk_d, sk_d) Downloader keypair; (pk_u, sk_u): Uploader keypair; sid: Fingerprint of a specific piece; m_{sid}: Binary data of piece sid. r_i:Random value. $(msg)_{sk}$ means message msg is signed by private key sk.

The payment token contains public key of the recipient of payment (uploader), fingerprint of file piece and a random number. The token is signed by the payer (downloader). The payment token can identify one specific transfer of file piece and can only be constructed by the downloader. Checking this token is simple: recipient of payment should accord with expected uploader; fingerprint of file piece should be valid; the signature can be verified using downloader's public key. Uploaders will check the token when receiving it and conditional payment also takes the check as one condition. When the uploader invokes conditional payment with satisfied payment token, the payment will be transferred to the uploader.

Protocol 1 Data Transfer

Inputs. P_d, P_u and their keypair (pk_d, sk_d), (pk_u, sk_u).

Goal. Downloader get a set of pieces while uploader obtain correct payment tokens.

The protocol:

1. **Setup.**
   (a) P_d sends a request to blockchain. Set up conditional payment.
   (b) Set up off-chain TCP connection between P_d and P_u.
2. **Transfer.**
   (a) P_d randomly choose sid and sends sid to P_u.
   (b) P_u checks whether he owns sid. If so, P_u sends back $data_{sid} \leftarrow (sid, m_{sid}, r_u)_{sk_u}$.
   (c) P_d receives $data_{sid}$ and check the file piece. If correct, P_d sends back a payment token $t_{sid} \leftarrow (pk_u, sid, r_d)_{sk_d}$.
   (d) P_u checks the integrity of t_{sid}. Repeat transfer process.
3. **Abort.** Either side can terminate the protocol by sending an abort message, especially when the opposite side misbehaves or P_d finishes his expected download.
4. **Payment** P_u invoke on-chain conditional payment with a set of t_{sid}.

4 Analysis

In this section, we use game theory models to analyze our incentive mechanism. First, we deploy evolutionary model to analyze the efficiency of blockchain-based incentive. The peers should have incentive to become active honest uploaders and downloaders.

However, in the process of data transfer, both sides have the possibility to cheat the other side, for instance terminating protocol in advance. The downloader, especially, has the chance to refuse to pay the uploader. For this potential unfairness, we use a repeated game model to demonstrate how cooperation can take place and whether the threat can affect overall system performance.

4.1 Evolutionary Model of Incentive

Evolutionary Game We consider an evolutionary game model in file sharing scenario. Each peer can act as downloader and uploader simultaneously, and we assume that peers are rational and strategic for the most profit.

Each uploader or downloader has two strategies: cooperate (C) and defect (D). Also, we assume that each data transfer process is homogeneous and produces benefit α for downloader and cost β for the uploader. Meanwhile, the downloader pays π to the uploader. Since the system is built on blockchain system, the network can be regarded full-connected. In other words, each downloader can directly find any available uploader and launch data transfer process.

Cost and Payoff In one single data transfer process, downloader invokes contract at least once (request) and bears communication cost to download and computing cost to find uploaders. The direct profit for downloader is the data itself. So α should be data value minus all these costs. In the same way, uploader invokes contract at least once (request payment) and bears bandwidth cost. The cost β should include these costs.

Meanwhile, if the opposite side unexpectedly aborts the data transfer process, both downloader and uploader have an extra cost. Downloader must cost more computing power to find another uploader while uploader may lose the last payment as described in Section 4.2. The extra costs for downloader and uploader are respectively denoted by t_d and t_u.

$$P = \left[\begin{array}{lcc} & C & D \\ C & \alpha - \pi, -\beta + \pi & -t_d, 0 \\ D & 0, -t_u & 0, 0 \end{array}\right] \tag{1}$$

Matrix P in Equation 1 shows payoff in one interaction between downloader and uploader (P_{ij} denotes payoff when downloader holds strategy i and uploader holds strategy j). In one generation of evolutionary model, each peer plays game with all other peers, so the distribution of strategy C and D has an important influence to average payoff in one generation. We use x_d denotes the fraction of strategy C among downloaders while x_u denotes the fraction of strategy C in uploaders. Equation 2 shows payoff in one generation for each role and each strategy, in which P_i^S denotes payoff for role i (downloader or uploader) with strategy S.

$$\begin{cases} P_d^C = x_u(\alpha - \pi + t_d) - t_d \\ P_d^D = 0 \\ P_u^C = x_d(-\beta + \pi + t_u) - t_u \\ P_u^D = 0 \end{cases} \tag{2}$$

Given payoff for each strategy and each entity, the total payoff in one generation with a strategy set $S = (S_d, S_u)$ is $P^S = P_d^{S_d} + P_u^{S_u}$:

Equilibrium Points From the payoff we list above, we found: 1) When x_u is small, which indicates restricted resource, peers trend to shift to strategy D as downloader since P_d^C may below 0. Otherwise, cooperation is a better choice. 2) When x_d is small, which indicates inactive downloader group and few profits for uploaders, peers trend to shift to strategy D as uploader since P_d^C may below 0. Otherwise, continuously providing download service earns more.

To further analyze this model, we use replicate dynamic equations [18]: $\dot{x_i} = x_i[f(x_i) - \Phi(x)]$, $\Phi(x) = \sum_{j=1}^{n} x_j f(x_j)$. x_i denotes distribution of each strategy and in our model there are two strategies (C, D) for downloader and also two strategies (C, D) for uploader. f is fitness of strategy, which equals to payoff analyzed in our model.

$$\begin{cases} \dot{x_d} = x_d(1-x_d)P_d^C \\ \dot{x_u} = x_u(1-x_u)P_u^C \end{cases} \tag{3}$$

To find Evolutionary Stable Strategy (ESS), the replicator dynamics equation should be equal to 0. Strategy C for uploader is stable only if $x_u = 0, 1$ or $P_u^C = 0$. In the same way, strategy C for downloader is stable when $x_d = 0, 1$ or $P_d^C = 0$. We can use Jacobian matrix 4 to investigate ESS in evolutionary game model. Possible equilibrium points are listed in Table 1.

$$J = \begin{bmatrix} (1-2x_d)[x_u(\alpha - \pi + t_d) - t_d] & x_d(1-x_d)(\alpha - \pi - t_d) \\ x_u(1-x_u)(-\beta + \lambda\pi + t_u) & (1-2x_u)[x_d(-\beta + \lambda\pi + t_u) - t_u] \end{bmatrix} \tag{4}$$

Table 1: Analysis of equilibrium points

Equilibrium point	$det(J)$	$tr(J)$	result
$x_d = 0$ $x_u = 0$	+	-	ESS
$x_d = 0$ $x_u = 1$	+	+	Not stable
$x_d = 1$ $x_u = 0$	+	+	Not stable
$x_d = 1$ $x_u = 1$	+	-	ESS
$x_d = \frac{t_u}{\pi-\beta+t_u}$ $x_u = \frac{t_d}{\alpha-\pi+t_d}$	+	0	Saddle point

We paint five equilibrium points $O(0,0)$, $A(0,1)$, $B(1,0)$, $C(1,1)$, $D(x_{d0}, x_{u0})$ in one coordinate plate (Figure 3). From above analysis, the evolutionary game model has two ESS point: (0,0) and (1,1). Point $(\frac{t_u}{\pi-\beta+t_u}, \frac{t_d}{\alpha-\pi+t_d})$, denoted by (x_{d0}, x_{u0}), is the saddle point. If initial state of system locates inside area OADB, system is more likely to converge to O. Otherwise, system has larger probability to evolve to C, which indicates cooperation equilibrium. We notice that free riding (point A) is unstable equilibrium point.

In conclusion, if parameters are properly set to make (x_{d0}, x_{u0}) closer to $(0, 0)$ and make sure there are enough proportion of cooperators at the beginning of the system, the whole system will converge to overall cooperation and keep stable in the end. Free riding is eliminated because of the high cost because they always pay currency for data they downloaded but never gain profit by uploading. Whitewashing is also not profitable. Though generating new blockchain accounts is very cheap, it doesn't make difference when downloading a file because the payment is determined by each single file piece rather than accounts.

4.2 Repeated Game Model for Data Transfer

Repeated Game Downloader requests a sequence of pieces from one uploader during one-time connection. The procedure can be regarded as a repeated game,

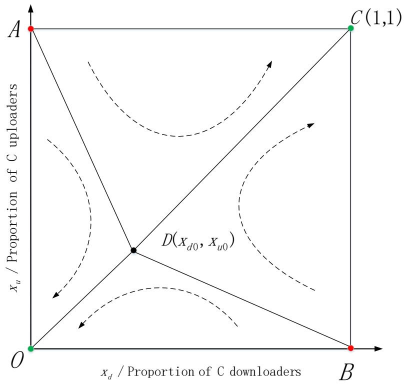

A
C (1,1)
x_u / Proportion of C uploaders
D(x_{d0}, x_{u0})
O
x_d / Proportion of C downloaders
B

Fig. 3: Diagram of equilibrium points in evolutionary model. Points O, A, B, C and D denotes various possible equilibriums while arrows denotes the path of revolution.

whose maximum number of stages is the number of pieces the uploader announces to possess.

Downloader bears a cost of transaction fee when requesting data. In general, downloader expects to download from multiple uploaders to maximum his download rate. However, the downloader cannot download one file from too many uploaders because he will cost much more to send request transactions and start connections. So downloaders prefer to find a balance to distribute their download to multiple uploaders. We use a probability model to present this situation. First, we define a downloader's *demand* in one data transfer procedure. The *demand* means the count of pieces which the downloader expects to download in one data transfer. If the downloader completes downloading demanded pieces, he will terminate the connection. Since the demand is affected by properties of files, downloader's characteristics and even current download status, we assume the distribution of the demand suits Gaussian distribution with expectation μ and variance σ.So the probability for downloader to demand ω pieces can be simply calculated as $p_\omega \sim N(\mu, \sigma)$.

The process of data transfer discussed in Section 3.2 consists of three steps: 1. Downloader sends a piece id to uploader; 2. Uploader sends back the correct data piece; 3. Downloader responds uploader with payment tokens. Both sides take trigger strategy: quit protocol in the next stage if the other side misbehaves. It accords with the reality that uploader won't be cheated twice. In intuition, a greedy downloader will quit protocol at step 3 so that he can get a free data piece from uploader while uploader can only quit at step 2. Note that uploaders actually cannot predict downloader's demand at the beginning of data transfer since the downloader just randomly choose pieces to download. However, the further data transfer procedure goes, the higher probability the downloader will quit. Therefore the repeated game will satisfy the following conditions:

1. In the first t periods, both sides cooperate. $t \in [0, n]$.
2. In the period $t + 1$, downloader exits at step 3 or uploader exits at step 2.
3. After period $t + 1$, protocol is terminated.

Cooperation Behavior In classic game theory, similar to finite repeated prisoner dilemma, this finite repeated game has Subgame Perfect Equilibrium (SPNE) that both sides won't cooperate from the beginning. However, participants are not completely rational. They usually have a belief of the cooperation from the other side and are greedy to take risk to cooperate. Both sides incline to deviate before the other side but intend to cooperate as much as they can. So the analysis of end behavior is important. There is a classical learning model which models cooperation behavior in finite repeated prisoner dilemma observed in experiments [19]. Above this classic model we define our learning model of repeated data transfer game. Downloaders and uploaders repeatedly play the finite repeated game (data transfer process). Downloader has a random demand ω for each round. Both sides respectively have an intended deviation period t_d and t_u. The execution of the model has the following various situations:

1. Downloader has a demand ω with probability p_ω. If $\omega < t_d$, downloader deviates in advance.
2. If one side observes that the opponent deviated before he intended to deviate, he has a probability p_1 to shift his intended deviation from t to $t - 1$.
3. If one side observes that the opponent deviated in the same period as he intended to, he has a probability p_2 to shift his intended deviation from t to $t - 1$.
4. If one side observes that the opponent hadn't deviate when he intended to deviate, he has a probability p_3 to shift his intended deviation from t to $t+1$.

From this model, we notice that if $t_d = t_u = n$, they all incline to deviate earlier, which represents that downloader wants to cheat for one free piece and uploader want to avoid this. When $t_d < t_u$ downloader terminates the protocol so early that misses more pieces to download. It is same for uploader when $t_d > t_u$ is observed. They have the belief that the other side intends to cooperate longer, so they intend to shift their deviation later.

According to our simulation (Section 5.2), though exchange protocol without TTP can hardly be definitely fair, the attractive rewards for cooperating makes the cooperation possible. The learning model shows an evolution of end behavior, and as a result, they can perform cooperation after a period of evolution. Besides, the damage of betraying and fraud is limited in only one piece. So the potential unfairness in data transfer process cannot affect system incentive and overall performance.

5 Experiments

We have four parts of experiments. The first experiment is the simulation of revolutionary model. We want to use simulations to prove that blockchain-based incentive is efficient to encourage active data sharing and eliminate free riding. The second experiment is a simulation of learning model introduced by Section 4.2 to see whether peers can achieve cooperation in data transfer process.

What is more, we implement smart contracts on *Ethereum* to discuss the feasibility of system substantiation and scalability of system protocol in Appendix A. Second, in Appendix B, we do a simple simulation about how a new file copies itself for the first time just after the creation.

5.1 Evaluation of System Incentive

Algorithm 1 Simulation process of evolutionary model

```
Initialize parameters like strategy distribution in $x_d$, $x_u$ and bandwidth distribution
  in Zipf distribution.
loop
    for peer $i = 1$ to $N$ do
        Let $j$ a random number unequal to $i$.
        Peer $i$ plays game (data transfer process) with peer $j$.
    end for
    for peer $i = 1$ to $N$ do
        randomly select another peer $j$
        Compute probability of learning process $p_{i \to j}$.
    end for
    Update strategy of peers with probability matrix $p_{i \to j}$.
end loop
```

Incentive Simulation Framework In our simulation framework, we first initialize parameters and the original state of the network. Especially the initial fraction of active downloaders x_d and uploaders x_u is important for the outcome of incentive. Then each peer plays game with other peers. Then calculate the payoff of each peer and run learning process.

When observing the real P2P file sharing, the distribution of data is heterogeneous and bandwidth of uploaders varies. Surveyed by paper [20], peer preference, popular file categories and bandwidth capabilities in P2P network can be modeled by Zipf distribution. We adapt Zipf distribution to simulate bandwidth of uploaders. In each generation of the simulation, when downloaders find uploaders with probability $f(j, N) = \frac{1/j}{\sum_{n=1}^{N} 1/n}$, in which j denotes the rank of uploader.So top rank uploaders have more opportunities to sell data.

In the evolutionary process, peers will learn another peer's strategy with a specific probability at the end of each round. Fermi update [7] [8] suits in our model as evolutionary updating rule. At the end of each round, peer i learns to follow another peer j's strategy with probability $p_{i \to j} = \frac{1}{1+e^{\omega(P_i - P_j)}}$. P_i denotes payoff of peer i. Parameter ω is a selection intensity factor. The larger the ω, the faster the system evolves.

Table 2: Parameters for simulation of system incentive

Parameter	Description	Value
N	Peer count	1000
ω	Learning coefficient	0.1
α	Benefit for downloader	1.6
β	Cost for uploader	1
π	Payment from downloader to uploader	1.4
t_d	Downloader cost	0.05
t_u	Uploader cost	0.10

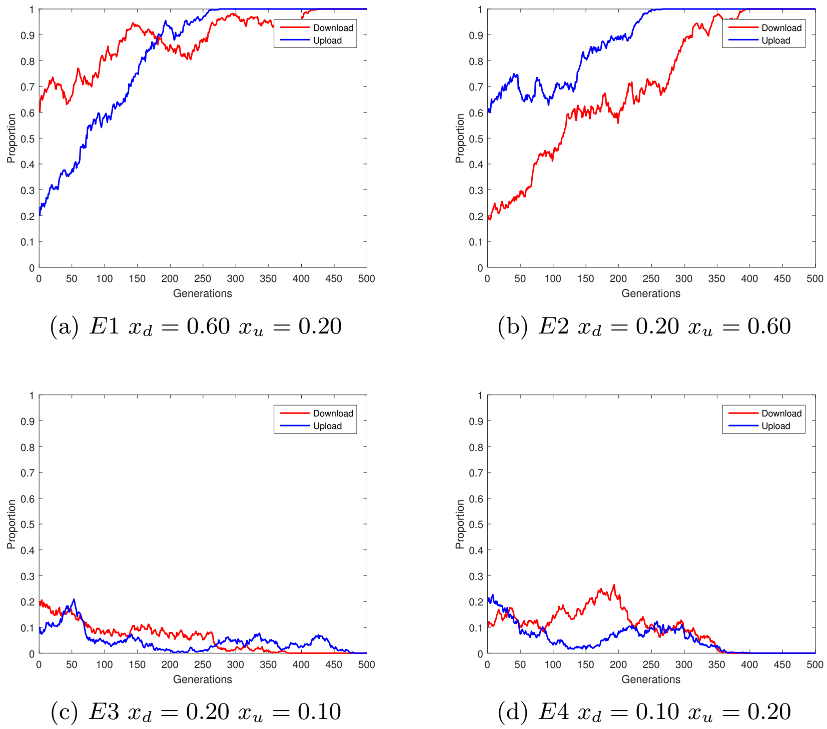
Simulation Parameters The values of parameter α, β, t_d and t_u depend on reality. First, we set $\beta = 1$ as a standard. β is the cost for uploader in a completed data transfer process, mainly include blockchain transaction fee and network bandwidth cost. α is the benefit for downloader in one exchange, including accessed data minus transaction fee and network cost. Compared with α and β, t_u is minor because it is limited in one piece and t_d is small because downloaders can simply request another uploader when uploaders are not so scarce.

According to above analysis, we estimated β much higher than t_u and t_d. Payment π should be larger than β and lower than α. we finally choose parameters in Table 2.

Simulation Result We distribute bandwidth of uploaders in Zipf distribution (Figure 4). Zipf bandwidth distribution, which can better depict real-world P2P file sharing, suits the evolutionary model and our blockchain-based incentive well. The estimated saddle point (Table 1) is $x_d = 0.2$, $x_u = 0.2$. We choose four typical initial values of x_d and x_u which locate in each area in Figure 3 respectively. Point $E1$ and $E2$ can converge to point C while point $E3$ and $E4$ fail to achieve a cooperative situation. If we properly set parameters of incentive mechanism and there is a satisfied fraction of active uploaders and downloaders at the beginning, the system will converge to cooperative equilibrium. And free riding will never become an equilibrium.

5.2 Simulation of Data Transfer Process

We use the parameters in table 3. In the simulation, we allocate 100 downloaders and 100 uploaders and each data transfer game is performed on a file with maximum piece count 100. The average initial demand for downloaders lies on 50 pieces. In each generation of simulation, each downloader randomly chooses one uploader, execute learning algorithm described in Section 4.2 and update expected deviation. As for probability parameters in learning process, It is plausible to assume $p_3 > p_1 > p_2$ because peers are greedy for more profits and last cooperation longer (analyzed in Section 4.2. Since peers have various



(a) $E1$ $x_d = 0.60$ $x_u = 0.20$

(b) $E2$ $x_d = 0.20$ $x_u = 0.60$

(c) $E3$ $x_d = 0.20$ $x_u = 0.10$

(d) $E4$ $x_d = 0.10$ $x_u = 0.20$

Fig. 4: Simulation results of different initial strategy distribution. Uploaders' bandwidth is in Zipf distribution. Predicted saddle point $(0.20, 0.20)$.

characteristics, their probability parameters should be different. We use random value within a range as the probability parameters.

From simulation result (Figure 5), if both sides are greedy enough (p_{3d} and p_{3u} are large enough), the finite repeated game will reach a dynamic balance point. After 50 loops of simulation, a significant proportion of peers have shifted their intended deviation closer to average demand. After 500 loops, expected deviation of both downloader and uploader locate very close to the point of average demand and keep a dynamical balance. This simulation represents the cooperation behavior of repeated games. Though the game doesn't reach a stable equilibrium, cooperation exists when peers are not completely rational. For example, their greed for profit and belief in others' cooperation encourage them to cooperate. Therefore, unfairness in data transfer process won't affect system incentive and won't damage overall download services.

Table 3: Parameters for simulation of data transfer process

Parameter	Description	Value
N	Peer count	100
n	Pieces count	100
μ	Expectation of Gaussian distribution of piece demand	50
σ	Variance of Gaussian distribution of piece demand	10
p_{1d}, p_{1u}	Probability parameter of learning model	0.3-0.5, 0.3-0.5
p_{2d}, p_{2u}	Probability parameter of learning model	0.2-0.4, 0.1-0.3
p_{3d}, p_{3u}	Probability parameter of learning model	0.6-0.8, 0.6-0.8

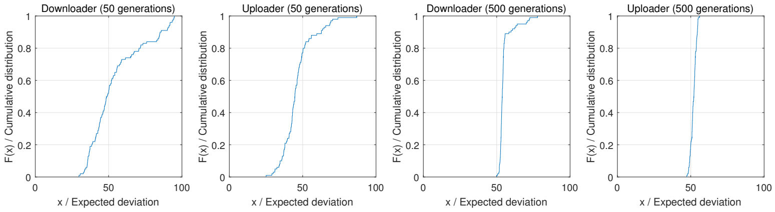


Fig. 5: Simulation of Data Transfer Game: cumulative distribution of expected deviation

6 Conclusion

In this paper, we proposed a blockchain-based file sharing protocol. We use evolutionary game model to analyze proposed incentive mechanism and use repeated game model and learning model to analyze potential unfairness in the proposed system. Then we carry out experiments to simulate the analysis. As the result shows, payment mechanism between downloaders and uploaders incentives active uploading and punishes free riders. Also, the data transfer protocol is almost fair since the potential unfairness is limited in only one piece of the file and this won't affect incentive and overall system performance.

Our scheme takes P2P file sharing application as an example to illustrate proposed blockchain-based incentive. The scheme is also a start to bring public blockchain into general P2P data exchange.

Bibliography

[1] Johan Pouwelse, Paweł Garbacki, Dick Epema, and Henk Sips. The bittorrent p2p file-sharing system: Measurements and analysis. In *International Conference on Peer-To-Peer Systems*, pages 205–216, 2005.

[2] Stefan Saroiu, Krishna P. Gummadi, and Steven D. Gribble. Measuring and analyzing the characteristics of napster and gnutella hosts. *Multimedia Systems*, 9(2):170–184, 2003.

[3] B Cohen. Incentives build robustness in bittorrent. In *The Workshop on Economics of Peer-To-Peer Systems*, pages 1–1, 2003.

[4] Yanchao Zhang and Yuguang Fang. *A Fine-Grained Reputation System for Reliable Service Selection in Peer-to-Peer Networks*. IEEE Press, 2007.

[5] Runfang Zhou, Kai Hwang, and Min Cai. Gossiptrust for fast reputation aggregation in peer-to-peer networks. *IEEE Transactions on Knowledge & Data Engineering*, 20(9):1282–1295, 2008.

[6] Tzu Ming Wang, Wei Tsong Lee, Tin Yu Wu, Hsin Wen Wei, and Yu San Lin. New p2p sharing incentive mechanism based on social network and game theory. In *International Conference on Advanced Information NETWORKING and Applications Workshops*, pages 915–919, 2012.

[7] Kun Lu, Junlong Wang, and Mingchu Li. An eigentrust dynamic evolutionary model in p2p file-sharing systems. *Peer-to-Peer Networking and Applications*, 9(3):599–612, 2016.

[8] Kun Lu, Shiyu Wang, Ling Xie, Zhen Wang, and Mingchu Li. A dynamic reward-based incentive mechanism: Reducing the cost of p2p systems. *Knowledge-Based Systems*, 112:105–113, 2016.

[9] Qiang Zhang, Hui Feng Xue, and Xiao Dong Kou. An evolutionary game model of resources-sharing mechanism in p2p networks. In *The Workshop on Intelligent Information Technology Application*, pages 282–285, 2007.

[10] Satoshi Nakamoto. Bitcoin: A peer-to-peer electronic cash system. *Consulted*, 2008.

[11] Marc Pilkington. Blockchain technology: Principles and applications. *Social Science Electronic Publishing*, 2015.

[12] Gavin Wood. Ethereum: A secure decentralised generalised transaction ledger. *Ethereum project yellow paper*, 151:1–32, 2014.

[13] Yunhua He, Hong Li, Xiuzhen Cheng, Yan Liu, and Limin Sun. *A Bitcoin Based Incentive Mechanism for Distributed P2P Applications*. 2017.

[14] Yunhua He, Hong Li, Xiuzhen Cheng, Yan Liu, Chao Yang, and Limin Sun. A blockchain based truthful incentive mechanism for distributed p2p applications. *IEEE Access*, 6:27324–27335, 2018.

[15] Richard Dennis and Gareth Owen. Rep on the block: A next generation reputation system based on the blockchain. In *Internet Technology and Secured Transactions*, pages 131–138, 2016.

[16] Junichi Kishigami, Shigeru Fujimura, Hiroki Watanabe, Atsushi Nakadaira, and Akihiko Akutsu. The blockchain-based digital content distribution sys-

tem. In *Big Data and Cloud Computing (BDCloud), 2015 IEEE Fifth International Conference on*, pages 187–190. IEEE, 2015.

[17] Ajay Kumar Shrestha and Julita Vassileva. Blockchain-based research data sharing framework for incentivizing the data owners. In *International Conference on Blockchain*, pages 259–266. Springer, 2018.

[18] Josef Hofbauer and Karl Sigmund. *Evolutionary games and population dynamics /.* Cambridge University Press,, 1998.

[19] Reinhard Selten and Rolf Stoecker. End behavior in sequences of finite prisoner's dilemma supergames a learning theory approach. *Journal of Economic Behavior & Organization*, 7(1):47–70, 1986.

[20] Mario T Schlosser, Tyson E Condie, and Sepandar D Kamvar. Simulating a file-sharing p2p network. *Workshop on Semantics in Grid Andp Networks*, 2003.

[21] Solidity — solidity 0.4.24 documentation. https://solidity.readthedocs.io/en/v0.4.24/. Accessed September 10, 2018.

[22] Go ethereum. https://geth.ethereum.org/. Accessed September 10, 2018.

[23] Etherscan the ethereum block explorer. https://etherscan.io. Accessed: 2018-08-19.

Appendix A Scalibility Evaluation

Table 4: Gas Cost of Smart Contract Calls

Operation	Description	Gas Used
Announce	Uploader announces possession of a file	62434
Request	Downloader requests uploader	112222
Payment	Uploader requests payment of one file piece	62964
Withdraw	Downloader cancels request after timeout	30250

We implement sample smart contracts for proposed system protocol in *Solidity* [21] on *Ethereum* platform. The cost of execution of EVM-based smart contracts is measured in *gas* and different operations have various gas cost. Therefore gas cost is an important metric to indicate the complexity of the operation and scalability. We deploy smart contracts on local private Ethereum network with *geth* client [22]. We deploy brand new contracts and record the gas cost of the first and successful smart contract call in Table 4. *Request* operation constructs a new conditional payment (step 3 in Section 3.1). *Payment* (step 5 in Section 3.1) verifies one payment token and the execution contains several assertions and one signature verification. Also, we can package *Payment* in larger function to transfer multiple payments at one time. Then the gas cost is about $42000 + 25000k$ in which k denotes the number of payment tokens uploader provides. *Withdraw* accords with step 6 in Section 3.1. In general, such a

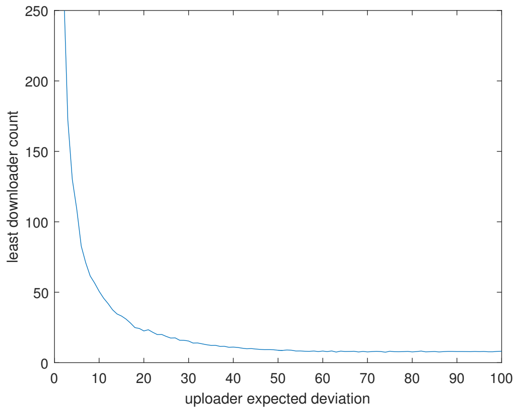


Fig. 6: Simulation of system startup: relationship between uploader's deviation and the least number of downoaders to copy a file (downloaders' deviation in Gaussian distribution)

smart contract is definitely feasible to implement with the acceptable gas cost. Note that gas cost in Table 4 is estimated value. Different execution path and the status of the contract may cause various gas usage. Unfortunately, we won't cover all those possibilities here.

Actually current popular smart contract platform, *Ethereum* for example, has relative slow transaction rate. The theoretical maximum Transaction Per Second (TPS) for *Ethereum* is only about 15 [23], which means when a large volume of transactions flood in, the latency of transaction calls will significantly increase. However, conditional payments allow off-chain data transfer to be independent with on-chain payments. Uploaders can request payments whenever the conditional payment is active therefore off-chain data transfer won't be blocked by slow on-chain operations. Connection setup requires on-chain operation *Request* only once and operation *Payment* can be done whenever uploaders have payment tokens and conditional payment is in an active state.

Appendix B Simulation of System Startup

Suppose one peer uploads a new file and at the beginning and there is only one uploader and multiple downloaders. We want to see how many downloads at least are able to copy one file in the system. Assume the file still has 100 pieces. Note that downloaders randomly choose pieces to download.

In the extreme situation, uploader and downloaders are definitely greedy and they can finish 100 piece exchange. So it only needs one downloader to make one file copy. If we suppose uploader has an initial expected deviation while downloaders'deviation distribution (Gaussian distribution) $X \sim (\mu, \sigma)$. Since all downloaders don't have any piece of the file at the beginning, their demand can be set 100. For convenience, we set $\mu = 50$ $\sigma = 10$ and choose different uploader deviation for simulation (Figure 6). The simulation shows that the later uploader's deviation, the less download count needed to make a copy. With the above parameters, at least 7 downloads are needed to copy a file in the system.