Computer Networks 185 (2021) 107631

Contents lists available at ScienceDirect

Computer Networks

journal homepage: www.elsevier.com/locate/comnet

Incentivizing cooperative relay in UTXO-based blockchain network

Xu Wang [a], Yanjiao Chen [a,*], Qian Zhang [b]

[a] School of Computer Science, Wuhan University, China
[b] Department of Computer Science and Engineering, The Hong Kong University of Science and Technology, Hong Kong, China

ARTICLE INFO

Keywords:
Blockchain
Incentive
Transaction propagation
UTXO model

ABSTRACT

Today's blockchain system provides two incentives for participants: block reward and transaction fee. However, these incentives benefit block miners more than ordinary participants. Besides, the message propagation relies on the volunteer work of these participants. We consider that there could be a third incentive to reward the relay process in the blockchain network. Nodes should be paid for corresponding relay work and incentives to expedite message propagation. We take advantage of the payment nature of blockchain and the characteristics of the UTXO Model, and propose a relay payment scheme to tackle the relay incentive issue. UTXO (unspent transaction output) is an abstract ownership of cryptocurrency and it represents a chain of payment relations between transactions. Any legal transaction should have valid UTXO(s) and generate new UTXO(s) to accomplish payment process. Native token in blockchain provides a straight and favorable payment method, and the UTXO model binds the interest of relay nodes and message originator. Then we model a cooperative, competitive relay game based on our scheme. We analyze the equilibrium results and run massive simulations on different datasets. This incentive scheme is free-rider resilient and could further expand to meet diverse relay requirement via transaction scripts in UTXO model. Our discussion and results help explore the relay incentives in the blockchain network and draw an example for the design of new blockchain systems.

1. Introduction

In the past decade, blockchain technology has attracted tremendous attention from both academia and commercial companies. A blockchain is a decentralized, distributed and public digital ledger that is used to record transactions across multiple nodes so that any involved record cannot be altered retroactively, without the alteration of all subsequent blocks. The blockchain was first introduced in [1] to serve as a crypto-currency transaction ledger and is currently widely adopted for a large number of crypto-currencies. With broad research over the years, the blockchain technology is recently applied in a wide range of scenarios far beyond crypto-currencies, like the Internet of Things (IoT), healthcare, and tax.

A blockchain is managed autonomously using a peer-to-peer network and achieving the consensus among those nodes, and it allows participants to verify and audit transactions independently and relatively inexpensively. The verified transaction data is stored in a chain of blocks, and the chain grows in an append-only manner with all new transactions to it. This process involves several operations such as verifying transactions, disseminating blocks, and attaching blocks to the blockchain.

In today's mainstream blockchain implementations, there are two incentives for network participants: block rewards and transaction fees. However, both incentives are granted to those mining nodes who have powerful calculation resources or pivotal stake. Thus most of the nodes inside blockchain networks have little probability of obtaining rewards of mining a block which forces many nodes to turn to pool mining to share these rewards [2,3]. As the block rewards are fixed and dwindle to zero, some work tries to redesign the transaction fee market. Moreover, further study shows that only incentive of transaction fees in the future contributes to undesirable security issues [4].

With the rapid development of blockchain technology, some proposed schemes make a breakthrough in both scaling the blockchain network and boosting the throughput up to thousands TPS level via sharding [5] or consensus zone [6]. We could expect that the bandwidth resource will be comparable to present computation resources according to the current research progress since the blockchain network is getting more significant, the information dissemination is more frequent, and the throughput is greater. Bandwidth resource could be as scarce as computation resources in the future. It is especially true in the IoT scenario where blockchain technology has been applied to.

[*] Corresponding author.
E-mail addresses: wangxu298@whu.edu.cn (X. Wang), chenyanjiao@whu.edu.cn (Y. Chen), qianzh@cse.ust.hk (Q. Zhang).

https://doi.org/10.1016/j.comnet.2020.107631
Received 24 June 2020; Received in revised form 3 October 2020; Accepted 19 October 2020
Available online 6 November 2020
1389-1286/© 2020 Published by Elsevier B.V.

Moreover, many mainstream blockchains choose PBFT-based consensus [7], and those systems maintain their consistency via voting. The voting process and consensus confirmation bring more network traffic into the blockchain network, which makes bandwidth more valuable.

In this paper, we consider that there could be the third incentive for those blockchain network participants with valuable bandwidth resources in the future. Nodes in the blockchain network should be paid for their relaying massive transactions just like they devote their computation resources into block mining. Since the message dissemination process is not as onefold as block mining, we first propose a basic scheme to handle the multi-agent relay process. In this scheme, the reward, a.k.a. relay fee, is paid in native blockchain token as the transaction fee does. We consider the relay process as a multi-hop relay cooperation, which is a message originator wants to send a transaction to a receiver via other nodes residing in the underlying blockchain network. Here the receiver may refer to a mining pool in PoW-based blockchain or a committee leader in a PBFT-based blockchain. What is more, in blockchain systems over a sharding architecture, transactions should be delivered to the shard which has the same hash suffix to be confirmed. And among these transactions, cross-shard transactions account for a large proportion. Those transactions should be relayed to relevant shards for validation and audit. We form the process of a cooperative but competitive contest that benefits the originator of faster transaction delivery and reward the participant relay in the native token as an incentive.

Rewarding schemes often suffer from the problem of free-riding and fairness. The free-riding problem is a type of market failure that occurs when those who benefit from resources, public goods or service of a communal nature do not pay for them, which cause the Tragedy of the Commons. We show our scheme guarantees that there would not be any free-rider and any reward paid comes from labor exchange, that is relay in context. On the other hand, the reward allocation among relays influences their enthusiasm in participation greatly. A node is willing to participate in relaying only if it can charge more than the cost for forwarding messages. Thus, we then model the relay scheme and form the problem in a game-theoretic fashion. We derive some insights through the model, which can guide and regulate the proposed scheme. Furthermore, we seek equilibrium under such model and solve the incentive problem on paid relays in the blockchain network.

Since the relay problem is strongly relevant to underlying network topology, we evaluate our proposed scheme to different datasets. We choose snapshots of the Bitcoin Simulator [8] and Gnutella peer-to-peer network [9]. We have run massive experiments over this two topologies and have some basic observations.

Here we summarize our main contributions:

- We design a new payment scheme to provide incentive for message propagation in UTXO-based blockchain networks. And nodes within the network are both cooperative and competitive in different manner.
- We model the payment scheme in a game-theoretic method, and provide best response for different types of nodes in theory.
- We derive the no-regret learning framework to compare different strategies in such incentive mechanism, and have some basic observations.

The rest of the paper is organized as follows. In Section 2, we introduce some preliminaries of our work and offer an overview of present blockchain technology. The main scheme is presented in Section 3 as well as concise analyses. In Section 4, we utilize game-theoretic tools to model the scheme, and corresponding evaluation is given in Section 5. Related work is reviewed in Section 6. Finally in Section 7 are the conclusions.

2. Preliminary

2.1. Relay in blockchain networks

Blockchain is typically managed by a peer-to-peer network collectively adhering to a protocol for inter-node communication and validating new blocks. Bitcoin uses a simple broadcast network to propagate transactions and blocks [10]. The current Bitcoin protocol implements various bandwidth optimizations, and measures in order to sustain its scalability and correct operation in spite of ever-increasing use. It adopts an advertisement-based request management system to minimize the information spread in the network. Namely, if a node receives information about a new block or transaction from another node, it will advertise this object to its other connections by sending them an *inv* message which is much smaller in size since it only contains the hash and the type of object which is advertised. Moreover, an internal reputation management system is maintained in the overlay network. The request and reputation management system is designed to reduce traffic in the blockchain network and to resist DDoS attack, that is, a receiving node locally assigns a penalty to peers who broadcast ill-formed transactions or blocks. Further, running atop an overlay network, full nodes bear the burden of information propagation, and the propagation delay is not negligible. There is no such penalty or reputation punishment upon selfish behavior such as free-riders.

In some other sharding blockchain networks, like [11], the gossip [12] as well its derivative protocol is most frequently adopted. The gossip algorithms are schemes which distribute the computational burden and in which a node communicates with a randomly chosen neighbor. Distributed systems user peer-to-peer gossip to ensure that data is routed to all members of an ad hoc network since there is no central registry and the only way to spread common data is to rely on each member to pass it along to their neighbors.

2.2. Record-keeping models

There are two mainstream types of record-keeping models in today's blockchain platform. One is the UTXO (Unspent Transaction Output) Model applied to Bitcoin [1], and the other is Account/Balance Model applied to Ethereum [13]. The Account/Balance Model records the balance of each account as a global state. The balance of an account is checked to make sure it covers the corresponding transaction amount. In the UTXO model, each transaction spends output from prior valid transactions and generates brand new outputs that can be spent by transactions in the future. All the unspent outputs are kept in each node's local database, and therefore, this model is called UXTO. A user's wallet keeps track of a list of unspent transactions associated with all addresses owned by the user, and the balance of the wallet is calculated as the sum of those unspent transactions output. For example, in Fig. 1, Transaction A and B have unique output, and both outputs are spent in Transaction C since the inputs of Transaction C reference this two via cryptographical signature [14]. Being spent, the output in Transaction A and Transaction B (gray one) became invalid and the new output in Transaction C (green one) is valid now and waiting to be spent. Note that a valid UTXO can only be spent once during its lifespan.

3. Scheme design

In this section, we introduce our whole relay payment scheme design. We first describe the transaction propagation scenario and briefly demonstrate the rationale upon UTXO Model. Then we introduce the concrete scheme from two aspects: cooperation and competition. We also have some further discussions with auxiliary and possible collusion.

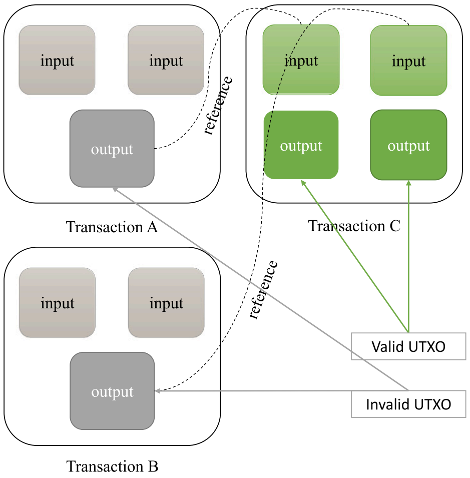


Fig. 1. Illustration of UTXO model.

3.1. Scenario

In large-scale networks consisting of selfish agents, incentive mechanism issues usually come along with competition and service/resource pricing. But in those system design, the authors mainly talks about the pricing scheme or reward protocol, but not how to pay the corresponding reward. They assume that there is a feasible way to handle the payment of service or resource. We take the payment nature of blockchain to deal with the incentive issues within the blockchain network.

Suppose there is a message originator who wants to transfer money to its vendor or some other payees as soon as possible. Then he should send a transaction out to announce this balance alternation. In today's blockchain implementation, this transaction message is flooded into the network via the gossip protocol [12], and most of the agents are volunteering in this protocol for the time being. We assume that this transaction message has its own receiver. It is quite common because transactions in blockchain should be confirmed via being packed into a block and appended to blockchain to reach consensus throughout the network. Those agents bearing powerful ability often play the role of transaction packers. In a PoW-based blockchain network, there may be many of those which we call them mining pools. While in some PBFT-based blockchain network, there is often only one leader who is responsible for the packing assignment. For simplicity, we assume that there is one receiver of this transaction message.

We choose the UTXO model in our design. Note that when the prior transaction has not been packed into a confirmed block, the newly constructed transaction specifying this prior transaction's output as input would be an invalid transaction and would be deprecated by nodes in the blockchain network. Besides, in the Bitcoin network, there is a scripting system for transactions. Script is simple, stack-based, and intentionally not Turing-complete, with no loops. The scripting system provides us more freedom and more ability to construct diverse and versatile transactions by the combination of its opcodes [15].

3.2. Cooperation in relay

Definition 1. Payment setup: there should be a connection construction step between every two nodes in the network. During the construction, nodes are ought to swap their account addresses for later incentive payment.

We assume that all the nodes have finished the connection construction step with payment setup. We show the relay process in Fig. 2. Denote the originator as s, and the receiver as w and relay R_1 is one neighbor of s which means R_1 has a direct connection with s. In order to make its transaction confirmed as soon as possible, s would like to pay for the relay nodes which help relay the transaction message. We stress on the relay process to boost the confirmation but do not discuss the possible waiting interval caused by packing strategy of w, which is beyond our scope. In the beginning, s would construct Transaction 0 specifying a TxInput and two TxOutputs where TxInput is a valid unspent transaction output of s. As for TxOutputs, node s's payee gets part of the token and relay R_1 gets the other part as its relay fee. Once Transaction 1 gets confirmed, then these TxOutputs become valid, and the dealings all settle down.

Note that relay R_1 may not be a receiver of this transaction message since he is not a valid leader or a powerful miner. But for relay R_1, in order to get the reward from s, it attempts to relay Transaction 1 to the receiver so that it can receive the reward. So relay R_1 and s achieve an agreement on relaying Transaction 1 to the receiver. Then relay R_1 encounters the problem of how it can relay Transaction 1 to receiver successfully. De facto, relay R_1 may have no idea where is the receiver in this blockchain network. Thus it turns to its neighbors to help relay Transaction 1. However, those neighbors would not help relay Transaction 1 for free since there is nothing incentive at all.

Hence, relay R_1 would like to pay its neighbor for helping relay Transaction 0. Note that relay R_1 cannot simply construct a transaction to pay. Denote this transaction as Transaction 1, if Transaction 1's TxInput is a valid unspent transaction output in relay R_1's balance, then the neighbors of relay R_1 would relay Transaction 1 but not Transaction 0. This will cause relay R_1 cannot get the reward from s and even lose some amount of token paid to its neighbors. In our scheme, Transaction 1's TxInput should be the TxOutput of Transaction 0. In Fig. 2, say R_2 is one of R_1's neighbors, R_2 could get the reward from R_1 only under the condition that both Transaction 0 and Transaction 1 are confirmed. So in Transaction 1, the TxInput is one part of TxOutput in Transaction 0 which R_1 will obtain. There are two TxOutputs in Transaction 1, too. One part of the token transfer to R_1 self which is the net reward for R_1, the other part transfer to R_2 as its transaction relay fee. Thus, when R_2 receives Transaction 0 and Transaction 1, R_2 would encounter the same problem as R_1 does one step before and if R_2 is not the receiver it would ask its neighbor for help to relay Transaction 0 and Transaction 1. For simplicity, we call Transaction 0, Transaction 1, etc. as message batch. This message batch would be relayed among those relay nodes, and the number of transactions inside it would increase one by one until it reached the receiver.

Theorem 1. *The message batch size is a linear function of hop count.*

Proof. In the first hop, there is one transaction in message batch including payment for sender's vendor and first hop relay. In the next hops, once a relay joins the cooperative relay, it constructs a transaction to pay for next hop relay, and collects this transaction into message batch. □

Suppose the message batch arrives at relay R_N, and there are already $N-1$ transactions in the batch. Different from former relays, R_N knows that it has a direct connection with the receiver since all transactions are plain text and R_N can read out the receiver from Transaction 0. Thus R_N would not turn to its neighbor for help but construct Transaction N instead. Similar to other transactions, the TxInput of Transaction N is part of the TxOutput of Transaction $N-1$. Also, the TxOutput of Transaction N goes to R_N and w. R_N gets its reward for helping relay the message and w gets its reward for packing those transactions into the blockchain. So when w receives the message batch from relay R_N, it first checks the validity of Transaction 0 to make sure that the TxInput of Transaction 0 is valid, and then follow the order until Transaction N. Node w would add this message batch to its transaction memory pool waiting for packing. Generally, validators, as well as miners, have their own packing strategies like the highest fee

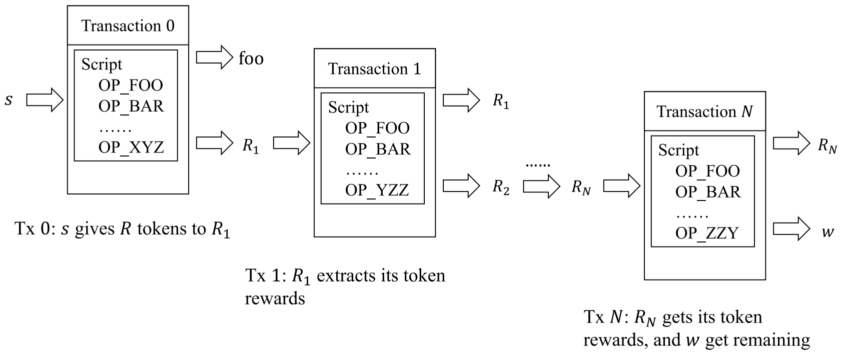


Fig. 2. Illustration of transaction relations. s initiates Transaction 0 to pay its vendor 'foo', and pay R_1 for relaying this transaction. Likewise, R_1 constructs Transaction 1 to pay B for relaying Transaction 0 and Transaction 1. This process continues along with the multi-hop relay until all transactions arrive at w.

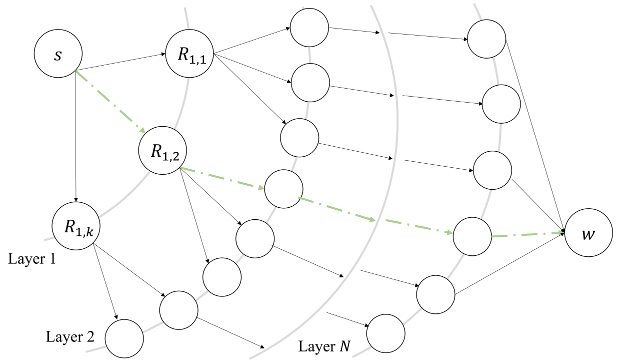


Fig. 3. A sample relay tree and winning path. Nodes can choose cooperate by relaying transactions to its offsprings or not. Only one path of relays/transactions will be chosen due to the constraint of UTXO model (as indicated by the green dotted line).

rates in the Bitcoin mining scenario. Since packing strategy is not the main concern of this work, we assume that w would always be willing to include those transactions in a block.

Note that all nodes in the relay network are not forced to join this relay. Nodes have sufficient freedom in both participation and bid determination. For example, a node might choose to defect and simply drop a message batch if the remaining reward is too low. This freedom puts pressure on upstream nodes, and brings tradeoff for each relay.

Theorem 2. *The message batch is like a simple record and thus there would not be any loop in the propagation tree.*

Proof. The message batch is a plain record which tracks all unconfirmed transactions, thus a relay is able to read the whole relay chain. If one of its neighbors' address is listed in these transactions, then this relay learns that which one has already heard about this message batch, and never relays back to this neighbor. So there would not be any loop in transaction relay tree. □

3.3. Competition in relay

The discussion above has formed a single relay path as well as a transaction chain. However, in the actual situation, there should be a relay tree and a transaction tree rooting at s and Transaction 0. Note that we assume only relay R_N knows it has a direct connection with w and relay R_1, etc does not have such connection with w. Inspired by the gossip protocol, these relays should relay message batch to its neighbors. Assume there are k neighbors of R_1(except node s), R_1 would send k different Transaction 1 out to its neighbors. Because R_1 should pay to different account address of those neighbors.

Meanwhile, those transactions have common TxInput as the UTXO of Transaction 0. In the UTXO model, all the unspent transaction outputs can only be spent once. We have seen many security-related papers seek feasible methods to form a double-spent attack. Thus we call those k transactions are conflicting to each other which means if one transaction has been confirmed by the network consensus, the remaining $k-1$ transactions become invalid at once. Like the iterative process we discussed above, it expands the single relay path to a relay tree. One thing that remains unchanged is that R_N would still relay only to w. So in this relay tree, nodes in the $N-1$ layer have only one out-degree to leaf node w.

Thus we show the relay tree in Fig. 3. Those transactions in the same layer jointly sharing a predecessor are conflicting with each other. Facing with those transactions, even though w wants to collect all the transactions fee but it could only choose one single path from s to w due to the mandatory restriction of the UTXO model. We show the green path of winners in Fig. 3. That is exactly why we choose the UTXO model in our scheme. Naturally, there is sharp competition among those relay nodes. We call the whole relay process a contest because all candidate relays try to be on the winning path, so that retrieve corresponding relay reward. In this contest, nodes in the same layer are in a competitive relationship while nodes in one specific path are in a cooperative relationship.

The competition among relay nodes benefits the originator s in two aspects. On the one hand, s leaves the reward allocation problem to subsequent relay nodes and extract its own reward in only one transaction. On the other hand, which is more critical, these subsequent relay nodes try to perform better in relay efficiency in order to be chosen by w causing the original Transaction 0 relayed to its destination as soon as possible.

As for how many rewards should each extract, namely the reward allocation among those relays, it associates the incentive issues. In Fig. 4, we model the interactions among last hop of relay and the receiver as an auction (the aqua part). And the interactions among intermediate hop of relays (the green part) are relevant to their position of the relay path. We show some intuitive consideration here. Suppose relay R_1 gives itself most of the reward in Transaction 1, then those subsequent relays are not willing to help relay Transaction 0 and Transaction 1 since relay R_1 is too greedy. Nevertheless, if R_1 chooses to extract little reward, its own reward would decrease and even lose some of its own balance due to relay cost. Those subsequent nodes face the same problem as R_1, too (if they decide to participate). So the decision on reward obtaining is quite crucial in our scenario. We assume that all the relay nodes are selfish, which means they want to maximize their profits. The modeling of communication networks consisting of multiple selfish agents requires tools from game theory. We leave the detailed discussion in Section 4.

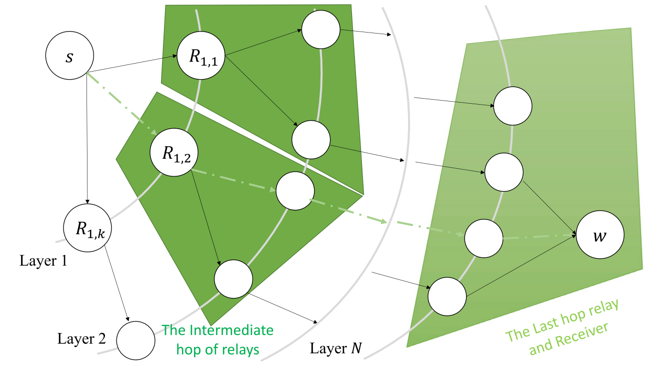

s
$R_{1,1}$
$R_{1,2}$
$R_{1,k}$
Layer 1
Layer 2
The Intermediate hop of relays
Layer N
w
The Last hop relay and Receiver

Fig. 4. The two parts of the game. The green field indicates the interactions among intermediate hop of relays, and the aqua field indicates the interaction among last hop of relays and the receiver. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

3.4. Auxiliary

In an actual situation, take the PBFT-based consensus blockchain network, for instance, the leader has its term limit in general. Thus the role of a leader might be performed by another node in a consensus committee after several epochs. It brings somewhat risk for s since it specified the receiver of its transaction at the beginning. The standard transaction to the Bitcoin address is the pay-to-pubkey-hash script. Meanwhile, the Script language provides us many useful opcodes like flow control, crypto, and locktime features. Take this scenario for example, we could construct the timeliness script using locktime opcodes.

Note that all the transactions in the message batch finally would be included in our blockchain which means that this record incorporates the relay path from to via a series of account addresses. When the validator or miner prepare and commit blocks comprised of vast relay message batch, these blocks could be further duplicated by all nodes inside the blockchain network to keep consistency. That is the core characteristic of blockchain technology summarized as non-repudiation. We consider those records can play another role as a routing reference table.

3.5. Possible collusion

We now consider the possible case when one relay node, relay R_2 in Fig. 5. For instance, it is in collusion with receiver w. R_2 could be able to intercept its subsequent relay nodes' reward through advance consultation. For example, w receives a message batch and reads out that R_2 participated in, then w would only choose those transactions ending at R_2 and does not handle other transactions. So relay node C and so on would not get their corresponding reward due to the intentional transaction loss conducted by w. In order to compensate w, since w would not get any reward from Transaction N, relay R_2 might bribe w via the advance consultation.

However, we show that this kind of collusion hardly ever appears in our native blockchain network, which means this collusion highly depends on a third-party payment method. We assume R_2 colludes with w, so there is no direct connection (if any, this collusion is unnecessary). Obviously w will not work for R_2 if R_2 never compensates w's reward loss of Transaction N. Once R_2 tends to bribe w with native blockchain token, this bribery transaction cannot be confirmed without other cooperative relay and R_2 should pay for its relay fee in the same manner unless R_2 chooses a third-party payment method for compensation. From a practical perspective, to establish a connection between R_2 and w is a low cost, and this collusion is a remote possibility.

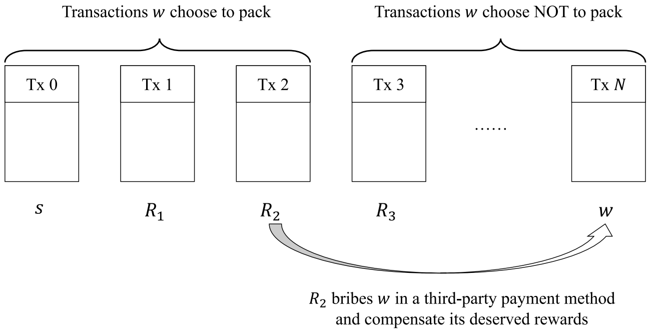

Transactions w choose to pack
Transactions w choose NOT to pack
Tx 0
Tx 1
Tx 2
Tx 3
......
Tx N
s
R_1
R_2
R_3
w
R_2 bribes w in a third-party payment method
and compensate its deserved rewards

Fig. 5. Illustration of collusion. R_2 bribes w, and asks w not to pack following transactions so that R_2 can acquire all remaining reward.

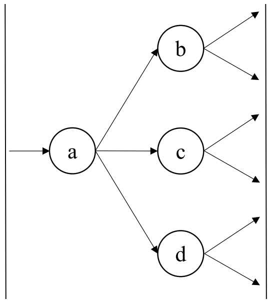


Fig. 6. Illustration of node relations. Node a as a predecessor, has 3 offsprings: b, c and d. And the three are siblings to each other. $\mathcal{O}(a) = \{b, c, d\}$, $\mathcal{P}(b) = \{a\}$.

4. Network model and problem formulation

In this section, we first model the player's strategies of different types. Then we manage to solve and prove the best response of each type. For a more special case, we formulate a Stackelberg Game into analysis.

4.1. Network and competitive relay

We consider an underlying blockchain network represented by a directed graph $\mathcal{G} = (\mathcal{N}, \mathcal{E})$ with one message originator s and one message receiver w, and a set of relays $\mathcal{I}$ which would help relay the message in a multi-hop fashion. We assume that there is no direct link between s and w, which means the message from s to w has to be routed via these relays. Since this paper mainly deals with the incentive perspective of the transaction relay, we also assume that nodes in our blockchain system are selfish and rational. Such assumption is pretty common under the topic of mechanism design and incentive analysis [4,16]. Besides, the network has an average diameter denoted by N (see Fig. 6).

We assume that each node in the network is aware of its neighbors' address, both the network address and native token account address. That is, each node can send a message to their neighbors and send the native token to them as well. The originator s, as well as relays, sends a message batch to their neighbors inside which is a series of unconfirmed transactions. These transactions include the bidding decision of present relay's predecessors and reveal the remaining reward for subsequent relays and receiver w. The relay process is noninteractive, so if a relay choose not to participate, they just drop the message batch.

4.2. Multi-hop relay and relay cost

By the previous assumption, since there is no direct link between s and w, the relay task cannot accomplish without routing among relays in $\mathcal{I}$. However, in this multi-hop relay game, there is a competitive but also cooperative relationship among these relays.

We assume each transaction has a fixed and identical length of bytes, and further assume that each single transaction cost at c_i denoting by the number of native token. Moreover, we assume that a relay would relay k_i times of the message batch to its offsprings. As the relay proceeds, the number of transaction is a linear function of the node's distance from originator s. Suppose a relay node i who received a message batch telling him of distance n, the cost function for this relay is

$$C_i = n \cdot c_i \cdot k_i \tag{1}$$

It can represent the bandwidth cost for relay node i and measure the transmission power cost if the links between nodes are wireless.

4.3. Competitive cooperative relay game

The relay game starts with the originator sends a specially constructed transaction to its neighbor to seek cooperation in order to deliver the transaction to a destination. Then those neighbors receive the transaction message and decide on whether it participates in this relay contest in a way to maximize its expected reward. Since the nodes are unaware of the topology of the underlying network, they do not know the best routing path to the receiver. For simplicity, we assume if a relay choose to join the contest, it would relay the message to all its offsprings or just ignore the message if defect to join. We further assume that before a relay node making its decision, it always get all offering bids that it would receive.

4.3.1. Strategy of the receiver

We assume that this message receiver is selfish like any other peer node in the network. It can represent, for example, the destination committee leader in a sharding blockchain network with PBFT-like voting consensus. As another example, if in a PoW consensus blockchain network, the receiver may be a specific mining pool bearing powerful calculation resource. Being selfish, the receiver would choose the most profitable message batch, which means the message batch with most remaining token would be packed into the blockchain. Thus, those transactions within the message batch would be confirmed altogether, and all relays associated in these transactions are rewarded corresponding tokens whom we call them winners of the relay contest. And all transactions in any other message batch of this relay contest would be orphan transactions and no longer make sense.

To summarize, the receiver's strategy is to pick one from its predecessors' bids, which we denote by $r(\cdot)$. Thus, the strategy space is $\Pi_{i\in\mathcal{P}(w)} r_i \times \{0,1\}$. And the optimal strategy is $\max_{i\in\mathcal{P}(w)} r_i$.

4.3.2. Last hop of relay

We first consider those special relays who have straight links with receiver w. Suppose a relay i that satisfying $w \in \mathcal{O}(i)$. Since all the transactions are plain text, i can read that one of its offsprings is originator's assigned receiver. Moreover, i also knows that this message batch has traveled how many hops and how many tokens are remaining for him and w. To win this relay contest, i must take other last hop relays into consideration because w would choose the most profitable offer reaching him.

Since our relay contest is non-cooperative games with incomplete information [17], those strategic relays make their decision not only on that public information but also the decisions of other peer relays. For simplicity, we assume that the last hop relays' bids follow a distribution denoted by CDF $\Phi(\cdot)$ and PDF $\phi(\cdot)$. Moreover, it is common knowledge for all last hop relays. We further assume that $\Phi(\cdot)$ is a normal distribution. Our simulation in Section 5.2 confirms that in real peer-to-peer dataset evaluation, there indeed exists such consistent distribution.

We consider relay i at last hop making its cooperation decision. When i receives the message batch, and he reads out all the information given to him: the number of transactions inside the message batch reveals how many hops have gone through and how many tokens are remaining denote by r_{i-1}. Moreover, i would not relay this message batch to any other offsprings but w. So $k_i = 1$ in the cost equation of i. Then relay i 's strategy is to extract $r_{i-1} - r_i$ tokens and leave r_i to receiver w. Given the distribution of last hop relay bids, we denote that the winning probability as $P(\cdot)$. Intuitively, $P(\cdot)$ is increasing in r_i since the more remaining left, w would more likely choose i's message batch to pack into blockchain and resulting i's winning the contest. The utility function of i can be written as follows:

$$u_i = \begin{cases} P(r_i|r_{i-1})(r_{i-1} - r_i) - C_i, & \text{if relay} \\ 0, & \text{if not relay} \end{cases} \tag{2}$$

The last hop relay contest is like a first-price sealed-bid auction [18] where relay who make the greatest bid wins the contest. So the win condition is that in $r_i = max(r_i, r_{-i})$, given the number of siblings of i as s_i, combined with the i.i.d assumption of $\Phi(\cdot)$, the winning probability can be written as:

$$P = P\{r_j < r_i | \forall r_j \in r_{-i}\} = \Phi^{s_i-1}(r_i) \tag{3}$$

Thus the problem relay i encounters with is:

$$\max_{r_i\in[0,r_{i-1}]} u_i = (r_{i-1} - r_i)\Phi^{s_i-1}(r_i) - nc_i \tag{4}$$

The information set for i is $\mathcal{I}_i = \{\mathcal{R}_i, n\}$ where $\mathcal{R}_i = \{r_j | j \in \mathcal{P}_i\}$, and the strategy space for i is $\mathcal{S} = \Pi_{j\in\mathcal{P}_i} r_j \times [0, r_j]$.

4.3.3. Intermediate hop of relay

Next, we consider those relays which do not have straight links with the receiver w. Those intermediate relays have less information than last hop relays since they have no idea about the network address of w, nor how far w is. To assist intermediate relays' decision-making process, the characteristic parameters of the underlying network are to help. We assume those intermediate relays have their own estimation of remaining hops denote by $\hat{m} = N - n$, and that the average amount of tokens each hop extracts is c.

Thus, the intermediate relay, say node j, makes the estimation that when this message batch reaches receiver w, the remaining reward for w is $f(r_j, \hat{m}, c)$. We map the intermediate hop relays into last hop relays through such estimation. On the one hand, the relay detail is highly related to the underlying network structure, and we hope our formulation is independent with specific node relation topology. On the other hand, such distance estimation has been widely studied like [19] and [20]. The estimation is feasible and succinct. For simplicity, we adopt the linear estimation as$f(r_j, \hat{m}, c) = r_j - \hat{m}c$. This is because relay j knows that all his offsprings and offsprings' offsprings and so on are rational game players. We make these assumptions so as to reduct the intermediate hop of relay problem into last hop of relay problem so that we can derive the winning probability from the intermediate relay's point of view.

Unlike last hop relays, since j do not which its offspring has shortest path to w, even there is no path to w in extreme case, so j would send the message batch to all his offsprings as the gossip protocol does. Thus we have the intermediate relay's problem as:

$$\max_{r_j\in[0,r_{j-1}]} u_j = (r_{j-1} - r_j)\Phi^{s_j-1}(r_j - \hat{m}c) - nk_jc_j \tag{5}$$

The strategy space for intermediate relays is identical to last hop relays since we assume each intermediate would offer bid to all its offsprings indistinguishably.

To sum up, we now have formed a relay game $RG(\mathcal{I}, \mathcal{S}, U)$.

4.4. Best response

Given the problem intermediate relay j and last hop relay i with Eq. (4) and Eq. (5), we now try to derive their best response.

Theorem 3. *There is a best response r_i^* in Eq. (4).*

Proof. For simplicity and without loss of generality, we take $s_i = 2$. Note that r_{i-1} is a given information for relay i, in order to avoid notation confusion, we denote r_{i-1} as R, so in equilibrium, we have the optimal first-order condition:

$$g(r_i) = \frac{\partial u_i}{\partial r_i} = -\Phi(r_i) + (R - r_i)\phi(r_i) \tag{6}$$

Assume the distribution has mean μ and variance σ^2. From three-sigma rule of thumb, we assume that $0 < \mu - 3\sigma < \mu + 3\sigma < R$. Here we consider that in interval $[\mu, R]$, $g(r_i)$ has unique root.

Note that $g(R) = -\Phi(R) < 0$, and

$$\begin{aligned} g(\mu) &= -\Phi(\mu) + (R-\mu)\phi(\mu) \\ &= -\frac{1}{2} + (R-\mu)\cdot\frac{1}{\sigma\sqrt{2\pi}} \\ &> \frac{3\sigma}{\sigma\sqrt{2\pi}} - \frac{1}{2} \\ &> 0 \end{aligned}$$

Moreover, denote

$$g'(r_i) = \frac{\partial g}{\partial r_i} = (R - r_i)\phi'(r_i) - 2\phi(r_i) \tag{7}$$

In interval $[\mu, \mu+\sigma]$, $\phi(\cdot)$ is a decreasing and strict concave function. Thus $\phi(r_{i-1}) - \phi(r_i) < \phi'(r_i)(r_{i-1} - r_i)$. Substitute to Eq. (7), we have $g'(r_i) < 3\phi'(r_i)(R - r_i)$, and $\phi'(r_i)\langle 0, R - r_i\rangle 0$, so we prove in $[\mu, \mu+\sigma]$, $g'(r_i) < 0$.

While in interval $[\mu+\sigma, R]$, note that $\phi(\cdot)$ is a decreasing and strict convex function. Thus $\phi(r_{i-1}) - \phi(r_i) > \phi'(r_i)(r_{i-1} - r_i)$. And we have $g'(r_i) < \phi(r_{i-1}) - 3\phi(r_i) < 0$.

And in $[0, \mu]$, $g(\cdot) > 0$ always holds. We left the proof in Appendix.

So we prove that $g(r_i)$ is a monotonically decreasing function in $[\mu, R]$, and $g(\mu) > 0, g(R) < 0$, according to intermediate value theorem, there exists a point r_i^* where u_i has its maxima and it is the best response. □

We denote the best response of relay i and relay j as $u_i^*(r_i^*)$ and $u_j^*(r_j^*)$, thus we have that $r_i^* = r_i^*(r_{j-1})$ and $r_j^* = r_j^*(r_{j-1})$, which means the offspring's decision is based on the offering bid of its directly connected predecessor. It is due to the iterative nature of our designed scheme that constructs such decision chain. This sequential decision making process gives those individual participant relay much freedom of their decision. The trust between relays is based on the sunk cost of relaying the message batch, that is, when a relay chooses to cooperate, he relays the message batch and undertakes the relay cost in advance before the settlement of this relay contest. So he makes the commitment of cooperation, tries to convince and incentive its offsprings to join the relay contest and share the reward together.

$$\max u_i = u_i(r_i^*) \tag{8}$$

The best response of relay largely depends on the detail distribution of $\phi(\cdot)$, we will show our numerical solution and corresponding analysis in Section 5.1.1.

4.5. Iterative Stackelberg game

The previous discussion illustrates that winning probability plays a crucial role in our relay contest since our scheme stipulates that winners take all and losers obtain nothing. We assume that all participants are risk-neutral. From Eq. (5), we know that the decision of last hop relay highly depends on the bids of the predecessor. Our discussion on the best response in Section 4.4 is a demonstration of relay's best response. That is all relays have no idea about how many hops left to reach the receiver but last hop relays. Even the direct predecessor of last hop relay does not know, either. The estimation of those relays might be inaccurate due to its information limits. However, things get different when the scheme runs for a period of time according to the discussion in Section 3.4. Relays may have some heuristics about the position of their neighborhood.

We now consider the situation when relay i is directly linked with w, and relay j participates in by relaying the message to i. What is more, j knows that there is practicable and reliable connection between i and w. Starting from this situation, we can formulate a Stackelberg game between i and j, where j acts as a leader and i as a follower. Review Eq. (4), so r_{i-1} is identical with r_j. The best response of i is a response function $r_i^* = r_i^*(r_j)$. Note that our assumption of last hop bid distribution still holds. Thus relay j could revise his winning probability according to the response function $r_i^*(r_j)$, that is, the probability of $r_i^*(r_j)$ is the max bids over all bids to receiver. We substitute this result to Eq. (5), so the estimation $r_j - c\hat{m}$ comes to $r_i^*(r_j)$.

Formally, the problem relay j faces now is below:

$$\max_{r_j \in [0, r_{j-1}]} u_j = (r_{j-1} - r_j)\Phi(r_i^*(r_j)) - nk_j c_j \tag{9}$$

Accordingly, the best response of relay j has corresponding alternation. We evaluate this result in Section 5.1.1 as well.

We have discussed the Stackelberg game in the last two hops. However, this interaction could happen between two arbitrary connected relays. Generally, we notice that the response of one single relay depends on the remaining reward its predecessor left to him, like the relay i in the above discussion. So its predecessor could predict the response in advance. Moreover, its prediction can expand to farther relays. In a more complicated situation, it is a multi-stage game with observed actions, and the equilibrium comprises all subgame perfection.

We consider the problem now for relay is below:

$$\max_{r_p \in [0, r_{p-1}]} u_p = (r_{p-1} - r_p)\Phi(r_q^*(r_{q-1}^*(\ldots(r_{p+1}^*(r_p))))) - nk_p c_p \tag{10}$$

Theorem 4. *In our Stackelberg model of RG, there still has a unique equilibrium.*

Proof. According to Eq. (6), We know that the best response function requires that $g(r_i) = 0$. Denote the function between r_{i-1} and r_i^* as $r_i^* = h(r_{i-1})$. So we have

$$\begin{aligned} \frac{\partial h}{\partial r_{i-1}} &= -\frac{g_{r_{i-1}}(r_i^*, r_{i-1})}{g_{r_i^*}(r_i^*, r_{i-1})} \\ &= -\frac{\phi(r_i^*)}{g'(r_i^*)} \\ &> 0 \end{aligned} \tag{11}$$

Without loss of generality, we show that $r_i^* = h(r_{r-1})$ is a monotonically increasing function. That is, in Eq. (10), no matter how many layer functions are nested, $r_q^*(\cdot)$ is monotonically increasing all along. In addition, the best response is always within the domain interval. Thus the unique equilibrium still holds. □

5. Evaluation

In this section, our discussions on the evaluation of our relay contest are two-fold. On the one hand, we first study the influence of environmental parameters via numerical analysis. We have derived the best response for two kinds of relays but not the closed-form. On the other hand, we conducted some experiments on the real Gnutella dataset, and cast a retrospective glance.

5.1. Numerical analysis

5.1.1. Simple estimation

From the analysis of previous sections, we have derived the best response of two different kinds of relays as equation. We assume the bid of last hop relays to the receiver obey the normal distribution. In more detail, we assume that each transaction cost a unit token, and the originator grants one hundred tokens as a reward to the relay contest. We evaluate how those environmental parameters influence the equilibrium, such as the hypothetical assumption of bids distribution and relay distance, etc.

Same as before, we first pay attention to the last hop relays. In Fig. 7(a), we have the response curve of last hop relays which tell us how many rewards would last hop relays deal out to the receiver given the reward from its predecessor. Comparing the solid line and the dash–dot line in Fig. 7(a), it shows that when the number of last hop relays increase (that is the total siblings), their decision to deal out also increases. This demonstrates that when the receiver has a higher in-degree in the network, competition among its direct predecessors becomes more intense. This competition benefits the receiver because those competitive relays have to deal out more to win the contest, that is, the receiver would earn more. Besides, the network condition also accounts for the response. In a high density network that corresponds to a higher variance in distribution, like the black lines in Fig. 7(a), those last hop relays have to deal out more as well.

In Fig. 8(a), we arrive at the same conclusion as the dash–dot line being atop the solid line. Moreover, this figure illustrates that the expect utility for long-distance last hop relays is less than those closer relays. It is because as the contest gets longer, the winning probability gets less. So the ideal situation for last hop relays is they are closer to the message originator, which means that the distance between originator and receiver is very close as well.

Then for the intermediate relays, the response curve is much closer to the straight line with a slope of 1 compared with last hop relays. Thus, those intermediate relays extract less reward. Intuitively, the last hop relays master more information and better location than intermediate relays. That superior condition allows them to get more. The variance influence on response and utility on intermediate relays are almost the same as last hop relays. However, the utility curve has a relatively obvious difference. In Fig. 8(b), the interval where utility greater than 0 in the curve significantly reduced, the solid blue line is underneath zero throughout, even worse. This means the early relays are afraid of losing the contest and those intermediate relays are concerned and sensitive to their predecessors' decision.

Next, we derive the sequential decision each layer relays would make in the contest. Fig. 9(a) shows us the remaining reward curve and Fig. 9(b) shows us the corresponding reward each layer relays would obtain. We can tell the reward for the latter relays is higher than the former. It is obviously true since the closer the relay is, the higher is their winning probability. For example, if the layer one has four siblings, then each has a winning probability of 25%. Thus, it is a lower risk and lower reward for front relays, and higher risk, higher reward for rear relays.

5.1.2. Stackelberg game result

For simplicity, we calculate the Stackelberg game in the last two hops for illustration. The previous discussion reveals that our Stackelberg game model helps revise the winning probability via early prediction. We show the result alteration in Fig. 10. According to Fig. 10(b) where we sampled 100 rounded numerical results, the best response to different remaining rewards from predecessor has not changed much when the value is less than 35 unit token. The response change is no more than 10 unit token. Now let us pay attention to a single given fixed remaining reward in Fig. 10(a). The maximum utility of the leader decreased by about 10 unit token due to the change of response. The penultimate hop's previous linear estimation is not that accurate, and last hop relay extracts more token in fact. So the utility increase of the last hop requires the utility decrease of penultimate hop relay given the fixed remaining reward to penultimate hop relay. That is, in the interaction between last two hops, the penultimate hop relays have to concede to last hop relays. The leader makes such a compromise to maximize its utility, and it is rational and reasonable in Stackelberg games.

5.2. Real dataset experiment

Furthermore, we also migrate our scheme to real dataset networks. Note that our model is sensitive to the underlying network topology, however, the real topology of today's mainstream blockchain system is hard to perceive due to various protections. We choose Bitcoin Simulator [8] and Gnutella network snapshot [9] from the Stanford Network Analysis Project [21]. Both two networks have fair numbers of nodes, while they are quite different in connectivity condition. We first show the connectivity of the datasets.

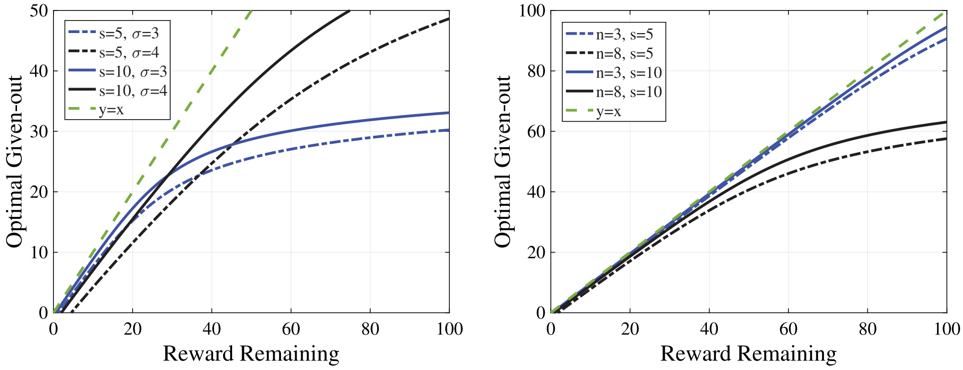
In Fig. 11, we can see that most nodes in Gnutella network have out-degree of 10 (we omit the isolated nodes), and that why we choose $k = 10$ for ordinary relay nodes. However, in the simulated bitcoin network, there are considerable super nodes with hundreds of neighbors. Moreover, most nodes have less than ten in-degrees and a small part of nodes with high in-degree. In reality, those well-connected nodes may refer to the receiver role in our scheme.

In order to further carry more evaluations in Gnutella dataset, we implemented our relay strategy along with some baseline strategies to rational social learning process. The first baseline, we call it dumb strategy, means that a single relay extracts a fixed amount of reward regardless of its position. And the second, selfish strategy, means that a relay extracts half of the remaining reward every time. To accomplish this, we run massive rounds with random originator and receiver pair in both datasets. Relayers in our evaluation perform no-regret learning, a standard learning that is popular in game theoretic works. The learning algorithms maintain a weight for every strategy, and keep adjusting the weights of the strategies from relay round to round on which strategy whey choose and what result this strategy brings. We choose EXP3 algorithm [22,23] for learning with adversarial bandits in our evaluation (see Fig. 13).

The convergence in two different networks is different as well. We found that, in a more distributed network like Gnutella, our proposed strategy performs better than the simulated bitcoin network where there are many super nodes there. It is because we assumed a normal distribution to instantiate our model. As shown in Fig. 12, our model fits distributed network more in Gnutella, and the distribution of last hop bids is just alike what we assumed before modeling. Things get quite different in simulated bitcoin network. Super nodes brings centralize the whole network and makes the network diameter shorter. However, such super nodes also bring threat to the whole blockchain network in term of security. From a theoretic point of view, it is a threat as well since those super nodes could be dominant nodes and compromise other relays' normal decision-making process (see Fig. 14).

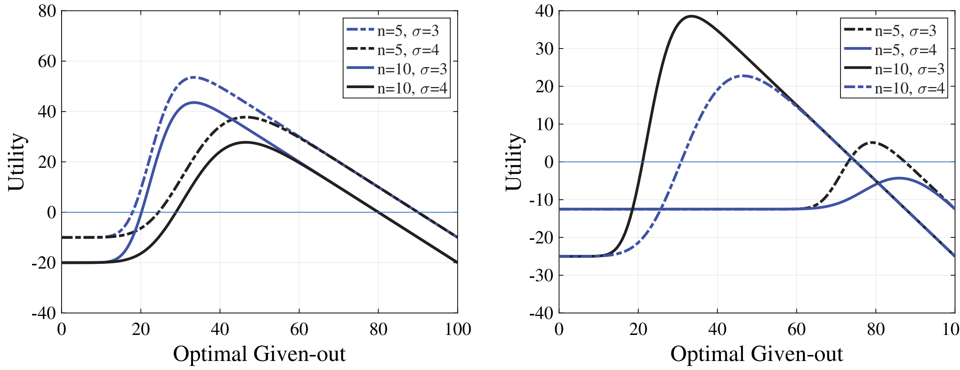
We also found that, our baseline schemes are not stable across these datasets. As shown in Fig. 15, the Selfish strategy beats Dumb strategy thoroughly in Gnutella dataset. However, this result flipped in Bitcoin Simulator dataset. Selfish strategy wins in Gnutella dataset because the dataset is more distributed. Nodes, to get a maximum expect reward, should claim a greedy reward since the subsequent competition would be more fierce. All in all, our proposed strategy beats these two strategies and showed a more stable performance across datasets.

Note that we proposed a general game-theoretic framework for relay incentive in UTXO-based blockchain network, and the strategy is sensitive to the underlying topology. The network topology makes its impact on the strategy via bids distribution. In a blockchain system, decentralization is pretty important. Although our proposed strategy is not that optimal in a less distributed network, we believe our strategy is expectant and promising for ideal distributed blockchain networks.



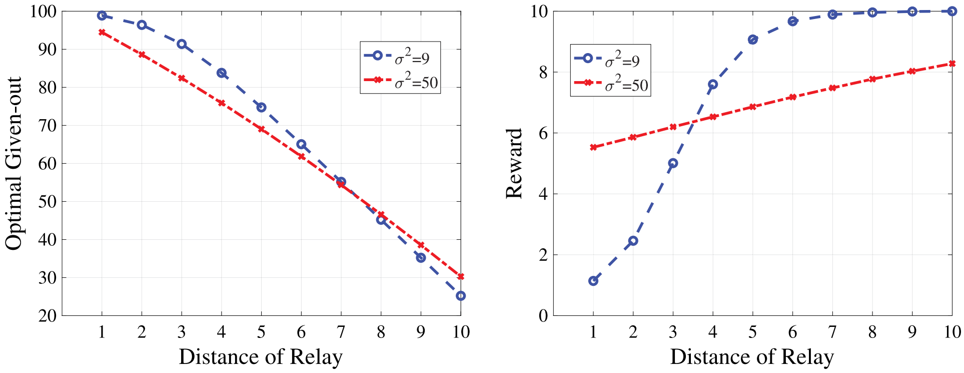
(a) (b)

Fig. 7. Numerical result of Eq. (8): differences of best response. $R = 100, N = 10, c_i = 1$.



(a) (b)

Fig. 8. Numerical result of Eq. (8): differences of utility. $R = 100, N = 10, c_i = 1$. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)



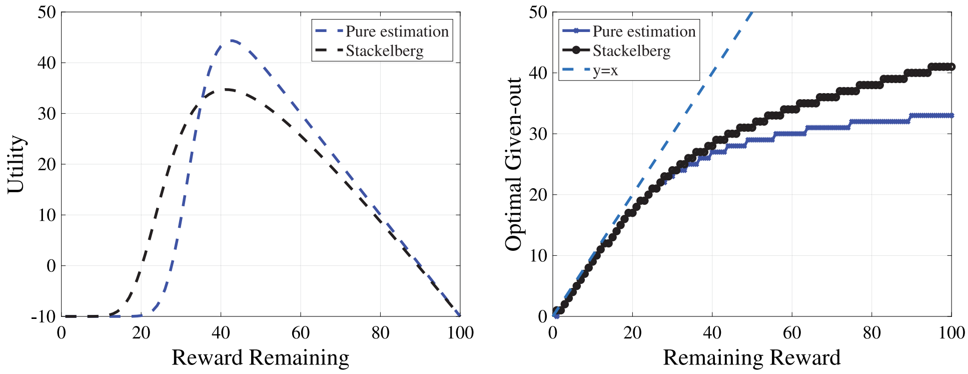
(a) (b)

Fig. 9. Numerical result of Eq. (8): differences of node's bids. $R = 100, N = 10, c_i = 1$.

6. Related work

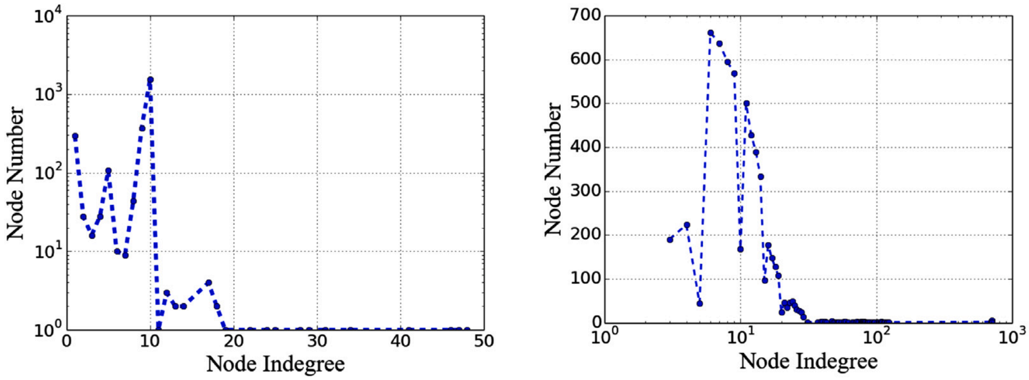
6.1. Blockchain network relay

Message dissemination in a blockchain network is a hot topic. On the one hand, several works seek to upgrade the existing protocol to improve relay efficiency and performance. The current version of the Bitcoin transaction relay protocol propagates messages among nodes using diffusion [24], which is a variation on random flooding. [25] proposes a new transaction dissemination protocol, Erlay, that not only reduces the bandwidth consumption but also keeps the bandwidth use almost constant. [26] offers a new relay network for Bitcoin blocks called Falcon to speed up block transmission. On the other hand, some works try to study and model the existing blockchain network message dissemination. [27] analyzes how Bitcoin uses a multi-hop broadcast to propagate transactions and blocks and gathered information to verify the conjecture that blockchain forks blame on propagation relay. There are some open discussions [28] in Ethereum research forum focusing on the incentive relay problem specifically in the Ethereum context. They propose Strawman Scheme which incorporate multiple encryption keys to each transaction in a onion-like fashion. [29] proposes a incentive propagation and smart routing when there is a leader.



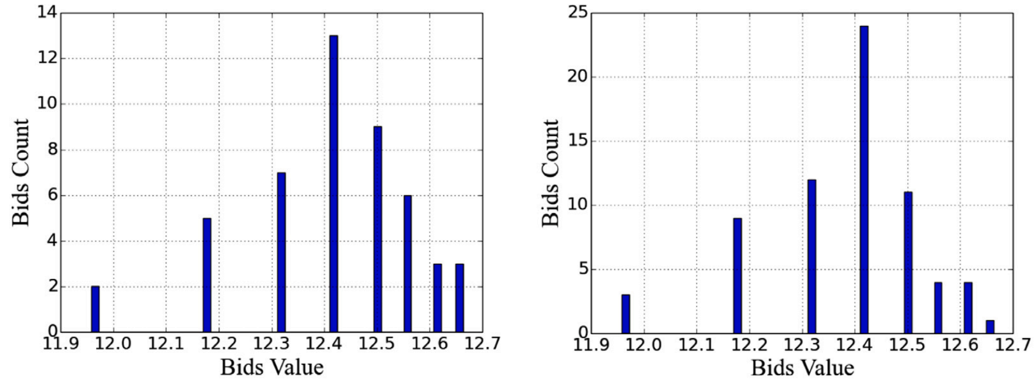
(a) (b)

Fig. 10. Alteration in Stackelberg Game. $R = 100, N = 10, c_i = 1$.



(a) (b)

Fig. 11. Basic connectivity of Gnutella (left) and Bitcoin Simulator (right) datasets.



(a) (b)

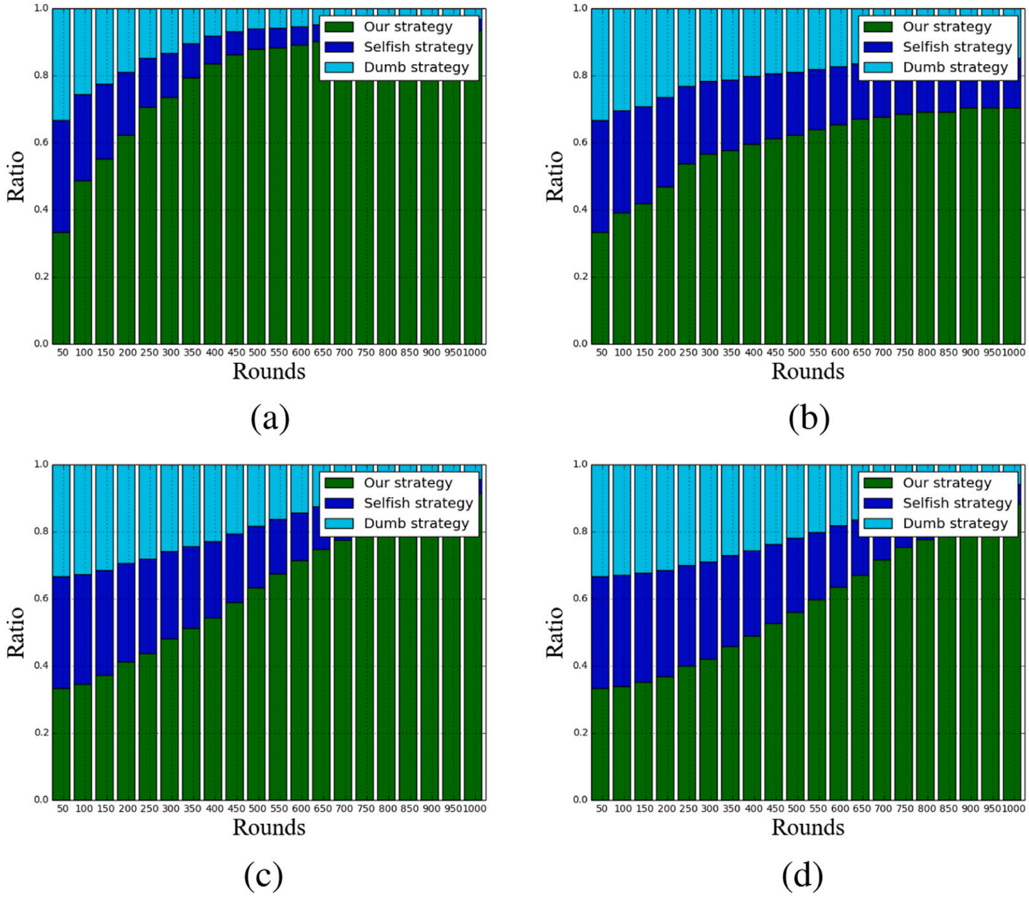
Fig. 12. Last bids distribution. We choose random sample destination nodes as w. The results follow a bell distribution due to network connectivity.

6.2. Incentive in relay and routing

Incentive issues have been widely studied in communication networks via game theory tools. [30] considers the problem of routing traffic to optimize the performance of a congested network and demonstrates that network performance degradation due to selfish routing. [31] studies pricing games in multi-hop relay networks where nodes price their services and route their traffic selfishly and strategically. [32] study the efficiency implications of competition among profit-maximizing service providers in communication networks. There are some proposed mechanism for bolckchain system. [33] studied the incentive problem where rational node has no incentive to propagate a transaction. [34] proposed a scheme, Solidus, offering an incentive to propagate transactions and validated blocks.

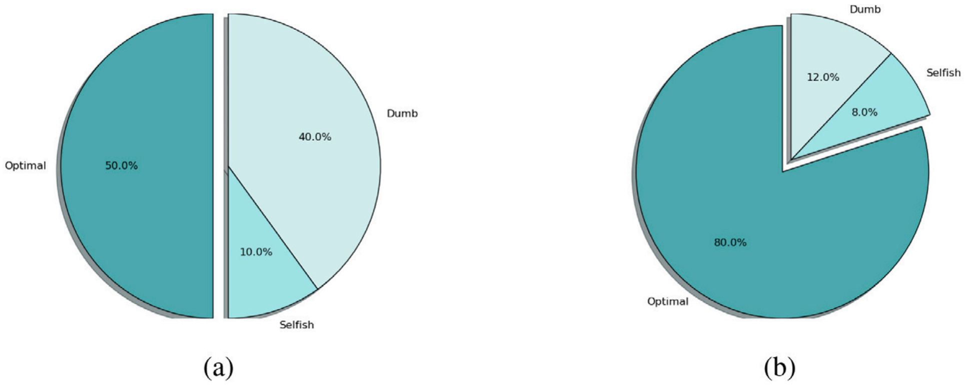
6.3. Incentive mechanism for blockchain

Recently, several incentive-based application in blockchain have been proposed. [35] presented a Stackelberg game and double auction based task offloading mechanism in edge computing to deal with blockchain mining process. [36] proposed a blockchain-based truthful incentive mechanism for P2P applications. [37] studied incentivizing



(a) (b) (c) (d)

Fig. 13. Strategy convergence in Gnutella dataset with random originator–receiver pair. At the beginning, all strategies have identical weights. After 1000 rounds, all cooperative nodes converge at choosing our proposed strategy other than the baseline strategies.



(a) (b)

Fig. 14. Convergence statistics in two networks. In a distributed network (right) like Gnutella network, our strategy performs far better than other baseline strategies. However, our strategy is not that dominant in a relatively centralized network (left) like Bitcoin Simulator.

the consensus propagation considering the tradeoff between delay and transaction fee. [38] proposed a secure, decentralized IoV data-trading system using blockchain technology via debt–credit mechanism. [39] studied the interaction between miners and cloud service providers and modeled it to a Stackelberg game. Most of those incentive-based researches use blockchain as a tool, yet not dive into the blockchain technology itself.

7. Conclusion

In this article, we consider that there could be a third incentive to reward the relay process in the blockchain network. Nodes should be paid for corresponding relay work and incentives to expedite message propagation. We take advantage of the payment nature of blockchain and the characteristics of the UTXO Model, and propose a relay payment scheme to tackle the relay incentive issue. Native token in blockchain provides a straight and favorable payment method, and the UTXO model binds the interest of relay nodes and message originator. We focus on the problem of transactions organization and bidding strategy, that is, embedding the payment in the whole relay process, rather than proposing a routing method in blockchain network. Then we model a cooperative, competitive relay game based on our scheme. Our analysis starts from the last hop of relays where these nodes directly connect to the destination node, and then extend the scenario where intermediate

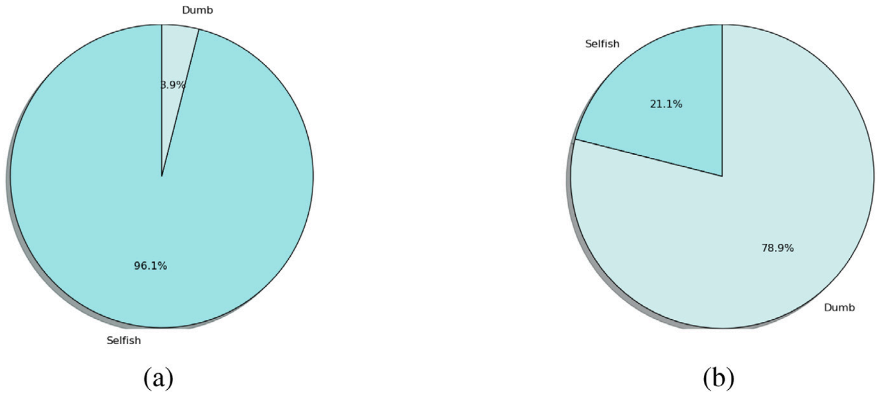


Fig. 15. The comparison of baseline strategy in two datasets.

relays do not have connection with the destination node. We analyze the equilibrium results and run massive simulations on real datasets. This incentive scheme is free-rider resilient and could further expand to meet diverse relay requirement via transaction scripts in UTXO model.

CRediT authorship contribution statement

Xu Wang: Conceptualization, Methodology, Software, Writing - original draft. **Yanjiao Chen:** Conceptualization, Writing - review & editing, Supervision. **Qian Zhang:** Supervision.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgment

This work is partially supported by the National Natural Science Foundation of China under Grants 61972296, and 61702380, Wuhan Advanced Application Project under Grant 2019010701011419, RGC under Contract CERG 16204418, 16203719, R8015, and the Guangdong Natural Science Foundation under Grant 2017A030312008.

Appendix. The remaining proof of Theorem 3

We split the proof into two pieces in the same way. First, in $[0, \mu-\sigma]$, $\phi(\cdot)$ is a increasing and strict convex function, we have proved that is $\phi(\cdot)$ is convex, then $g'(r_i)$ in Eq. (7) is negative. While in $[\mu-\sigma, \mu]$ where $\phi(\cdot)$ is concave, it is obvious that both $R-r_i$ and $\phi'(\cdot)$ is decreasing, and $\phi(\cdot)$ is increasing. Thus $g'(\cdot)$ is decreasing.

References

[1] N. Satoshi, Bitcoin: A peer-to-peer electronic cash system, 2008, http://www.bitcoin.org/bitcoin.pdf.

[2] O. Schrijvers, J. Bonneau, D. Boneh, T. Roughgarden, Incentive compatibility of bitcoin mining pool reward functions, in: International Conference on Financial Cryptography and Data Security, Springer, 2016, pp. 477–498.

[3] M. Rosenfeld, Analysis of bitcoin pooled mining reward systems, 2011, arXiv preprint arXiv:1112.4980.

[4] M. Carlsten, H. Kalodner, S.M. Weinberg, A. Narayanan, On the instability of bitcoin without the block reward, in: Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, ACM, 2016, pp. 154–167.

[5] L. Luu, V. Narayanan, C. Zheng, K. Baweja, S. Gilbert, P. Saxena, A secure sharding protocol for open blockchains, in: Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, ACM, 2016, pp. 17–30.

[6] J. Wang, H. Wang, Monoxide: scale out blockchain with asynchronous consensus zones, in: Proceedings of the 16th USENIX Conference on Networked Systems Design and Implementation, USENIX Association, 2019, pp. 95–112.

[7] M. Castro, B. Liskov, et al., Practical byzantine fault tolerance, in: OSDI, 99, (1999) 1999, pp. 173–186.

[8] A. Gervais, G. Karame, K. Wüst, V. Glykantzis, H. Ritzdorf, S. Capkun, On the security and performance of proof of work blockchains, in: Proceedings of the 23nd ACM SIGSAC Conference on Computer and Communication Security (CCS), ACM, 2016.

[9] J. Leskovec, A. Krevl, SNAP datasets: Stanford large network dataset collection, 2014, http://snap.stanford.edu/data.

[10] A. Gervais, H. Ritzdorf, G.O. Karame, S. Capkun, Tampering with the delivery of blocks and transactions in bitcoin, in: Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security, ACM, 2015, pp. 692–705.

[11] M. Zamani, M. Movahedi, M. Raykova, Rapidchain: Scaling blockchain via full sharding, in: Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security, ACM, 2018, pp. 931–948.

[12] S. Boyd, A. Ghosh, B. Prabhakar, D. Shah, Randomized gossip algorithms, IEEE/ACM Trans. Netw. 14 (SI) (2006) 2508–2530.

[13] G. Wood, et al., Ethereum: A Secure Decentralised Generalised Transaction Ledger, Vol. 151, Ethereum Project Yellow Paper, 2014, pp. 1–32.

[14] Bitcoin script, 2020, https://en.bitcoin.it/wiki/Script.

[15] R. Kumaresan, I. Bentov, How to use bitcoin to incentivize correct computations, in: Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security, ACM, 2014, pp. 30–41.

[16] I. Eyal, E.G. Sirer, Majority is not enough: Bitcoin mining is vulnerable, in: Financial Cryptography, 2014.

[17] M.J. Osborne, A. Rubinstein, A Course in Game Theory, MIT press, 1994.

[18] V. Krishna, Auction Theory, Academic press, 2009.

[19] M. Costa, M. Castro, R. Rowstron, P. Key, PIC: Practical internet coordinates for distance estimation, in: 24th International Conference on Distributed Computing Systems, 2004. Proceedings., IEEE, 2004, pp. 178–187.

[20] T.E. Ng, H. Zhang, Predicting internet network distance with coordinates-based approaches, in: Proceedings. Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies, 1, IEEE, 2002, pp. 170–179.

[21] J. Leskovec, R. Sosič, SNAP: A general-purpose network analysis and graph-mining library, ACM Trans. Intell. Syst. Technol. (TIST) 8 (2016) 1.

[22] A. Blum, Y. Mansour, From external to internal regret, J. Mach. Learn. Res. 8 (Jun) (2007) 1307–1324.

[23] P. Auer, N. Cesa-Bianchi, Y. Freund, R.E. Schapire, The nonstochastic multiarmed bandit problem, SIAM J. Comput. 32 (1) (2002) 48–77.

[24] Bitcoin core commit 5400ef, 2015, https://github.com/bitcoin/bitcoin/commit/5400ef6bcb9d243b2b21697775aa6491115420f3.

[25] G. Naumenko, G. Maxwell, P. Wuille, S. Fedorova, I. Beschastnikh, Bandwidth-efficient transaction relay for bitcoin, 2019, arXiv preprint arXiv:1905.10518.

[26] Falcon: A fast new bitcoin backbone relay network, 2016, https://www.falcon-net.org.

[27] C. Decker, R. Wattenhofer, Information propagation in the bitcoin network, in: IEEE P2P 2013 Proceedings, IEEE, 2013, pp. 1–10.

[28] Incentivizing a robust P2p network/relay layer, 2018, https://ethresear.ch/t/incentivizing-a-robust-p2p-network-relay-layer/1438.

[29] O. Ersoy, Z. Ren, Z. Erkin, R.L. Lagendijk, Transaction propagation on permissionless blockchains: incentive and routing mechanisms, in: 2018 Crypto Valley Conference on Blockchain Technology (CVCBT), IEEE, 2018, pp. 20–30.

[30] T. Roughgarden, É. Tardos, How bad is selfish routing? J. ACM 49 (2) (2002) 236–259.

[31] Y. Xi, E.M. Yeh, Pricing, competition, and routing for selfish and strategic nodes in multi-hop relay networks, in: IEEE INFOCOM 2008-the 27th Conference on Computer Communications, IEEE, 2008, pp. 1463–1471.

[32] D. Acemoglu, A. Ozdaglar, Competition in parallel-serial networks, IEEE J. Sel. Areas Commun. 25 (6) (2007) 1180–1192.

[33] M. Babaioff, S. Dobzinski, S. Oren, A. Zohar, On bitcoin and red balloons, in: Proceedings of the 13th ACM Conference on Electronic Commerce, ACM, 2012, pp. 56–73.

[34] I. Abraham, D. Malkhi, K. Nayak, L. Ren, A. Spiegelman, Solidus: An incentive-compatible cryptocurrency based on permissionless byzantine consensus, 2016, CoRR, abs/1612.02916.

[35] S. Guo, Y. Dai, S. Guo, X. Qiu, F. Qi, Blockchain meets edge computing: Stackelberg game and double auction based task offloading for mobile blockchain, IEEE Trans. Veh. Technol. 69 (5) (2020) 5549–5561.

[36] Y. He, H. Li, X. Cheng, Y. Liu, C. Yang, L. Sun, A blockchain based truthful incentive mechanism for distributed P2p applications, IEEE Access 6 (2018) 27324–27335.

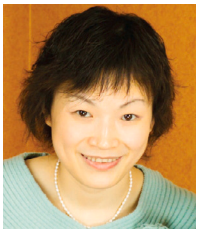
[37] J. Kang, Z. Xiong, D. Niyato, P. Wang, D. Ye, D.I. Kim, Incentivizing consensus propagation in proof-of-stake based consortium blockchain networks, IEEE Wirel. Commun. Lett. 8 (1) (2019) 157–160.

[38] K. Liu, W. Chen, Z. Zheng, Z. Li, W. Liang, A novel debt-credit mechanism for blockchain-based data-trading in internet of vehicles, IEEE Internet Things J. 6 (5) (2019) 9098–9111.

[39] H. Yao, T. Mai, J. Wang, Z. Ji, C. Jiang, Y. Qian, Resource trading in blockchain-based industrial internet of things, IEEE Trans. Ind. Inf. 15 (6) (2019) 3602–3609.

Xu Wang received his B.E. degree in Information Security from Wuhan University, China, in 2018. He is currently working towards the M.S. degree in the School of Computer Science, Wuhan University. His research interests include blockchain technology, network economics and network security.

Yanjiao Chen received her B.E. degree of electronic engineering from Tsinghua University in 2010 and Ph.D. degree of computer science and engineering in Hong Kong University of Science and Technology in 2015. She is currently a Professor in Wuhan University, China. Her research interests include network security, wireless network resource allocation, and network economics.



Qian Zhang joined Hong Kong University of Science and Technology in Sept. 2005 where she is a full Professor in the Department of Computer Science and Engineering. Before that, she was in Microsoft Research Asia, Beijing, from July 1999, where she was the research manager of the Wireless and Networking Group. Dr. Zhang has published about 300 refereed papers in international leading journals and key conferences in the areas of wireless/Internet multimedia networking, wireless communications and networking, wireless sensor networks, and overlay networking. She is a Fellow of IEEE for ‘‘contribution to the mobility and spectrum management of wireless networks and mobile communications’’. Dr. Zhang has received MIT TR100 (MIT Technology Review) world’s top young innovator award. She also received the Best Asia Pacific (AP) Young Researcher Award elected by IEEE Communication Society in year 2004. Her current research is on cognitive and cooperative networks, dynamic spectrum access and management, as well as wireless sensor networks. Dr. Zhang received the B.S., M.S., and Ph.D. degrees from Wuhan University, China, in 1994, 1996, and 1999, respectively, all in computer science.