Blockchain-Aided Edge Computing Market: Smart Contract and Consensus Mechanisms

Yu Du, *Student Member, IEEE,* Zhe Wang, *Member, IEEE,* Jun Li, *Senior Member, IEEE,* Long Shi, *Member, IEEE,* Dushantha Nalin K. Jayakody, *Senior Member, IEEE,* Quan Chen, Wen Chen, *Senior Member, IEEE,* and Zhu Han, *Fellow, IEEE*

Abstract—Building upon the prevailing concept of edge computing (EC), a distributed EC market requires decentralized and verified transaction management to trade computing resources. Towards this goal, we study a blockchain-aided EC market wherein each data service operator (DSO) rents a group of edge computing nodes (ECNs) and leases the ECNs to the user terminals (UTs) to provide computation offloading services. A trustworthiness model is introduced to evaluate the quality of each network entity throughout the transactions. We develop a two-level trading mechanism over smart contract to enable the automatic and efficient transactions among the network entities and provide high quality services. First, we propose a smart contract based matching mechanism to establish the renting association between the DSOs and ECNs with the aim of maximizing the social welfare. Second, we propose a social welfare improved double auction (SWIDA) mechanism to build up the leasing association between the DSOs and UTs, and determine the pricing of the winners. We show that the proposed double auction mechanism can achieve individual rationality, balanced budget, truthfulness in expectation, and an improved social welfare than the benchmark mechanisms. Moreover, we put forth a trustworthiness driven Proof-of-Stake (PoS) consensus mechanism to enable verified transaction and fair allocation of block generation reward. Following the principle of PoS, we formulate the block generation as a coalitional game, wherein each stakeholder votes according to its trustworthiness and coinage, and shares the reward among the coalition according to the Shapley values. The simulation results show that the proposed PoS consensus mechanism can reduce the wealth inequality among the network entities compared with the conventional consensus mechanisms.

Index Terms—Edge computing, blockchain, smart contract, matching, double auction, proof-of-stake, Shapley value.

1 Introduction

It is envisioned that tens of billions of smart devices will emerge in the next few years, creating a host of delay-sensitive services such as virtual/augmented reality and autonomous driving [1]. In particular, if massive computation-intensive tasks are processed in user terminals (UTs), it is bound to accelerate energy consumption and shorten their service lifetime [2]. The recent advances in edge computing (EC) tackles these challenges by allowing the UTs to offload the computational tasks to the edge computing nodes (ECNs) deployed in close proximity [3], [4].

Building upon the EC network, a typical EC market provides a trading platform on which the ECNs sell their resources to the UTs [5]–[7] where the UTs rent the computing resources of the ECNs for computational task offloading [5], [6] or content storage [7]. Most existing trading mechanisms for EC market require a central authority to enable the transactions and resource allocation across the network devices. However, the central authority may not be trusted and is vulnerable to the single point of failure. To avoid the intervention of central authority, blockchain is proposed to manage the transactions in a distributed and tamper-proof ledger.

Currently, the research about blockchain and EC market can be classified into two categories: EC market aided blockchain and blockchain aided EC market. First, for EC market aided blockchain [8]–[13], the resource-constrained UTs rent the computing resources from the ECNs to improve the mining efficiency for Proof of Work (PoW) in blockchain. Second, for blockchain aided EC market, the blockchain can verify and recall the EC market transactions that are automatically executed by smart contract [14]. Specically, smart contracts are lines of code that are stored in the blockchain and are automatically executed when predetermined conditions in the contract are met. Ref. [15] designed a smart contract based double auction mechanism to maximize the total amount of offloading tasks between the UTs and ECNs. In [16], the authors adopt blockchain based smart contracts to construct an autonomous content caching market that helps the UTs download content from the ECNs. In [17], a resource pricing and trading scheme is proposed based

- *Y. Du, J. Li, and L. Shi are with the School of Electronic and Optical Engineering, Nanjing University of Science and Technology, Nanjing 210094, China. E-mail: {yudu, and jun.li}@njust.edu.cn, slong1007@gmail.com.*
- *Z. Wang is with the School of Computer Science and Engineering, Nanjing University of Science and Technology, Nanjing 210094, China. E-mail: zwang@njust.edu.cn.*
- *D. N. K. Jayakody is with the Autónoma Techlab da, Centro de Investigação em Tecnologias, Universidade Autónoma de lisboa, 1169-023 lisbon Portugal, and also with the School of Computer Science and Robotics, Tomsk Polytechnic University, Tomsk 634050, Russia. E-mail: nalin@tpu.ru.*
- *Q. Chen is with the Department of Computer Science and Engineering, Shanghai Jiao Tong University, Shanghai 200240, China. E-mail: chenquan@cs.sjtu.edu.cn.*
- *W. Chen is with the Department of Electronic Engineering, Shanghai Jiao Tong University, Shanghai 200240, China. E-mail: wenchen@sjtu.edu.cn.*
- *Z. Han is with the University of Houston, Houston, TX 77004, and also with the Department of Computer Science and Engineering, Kyung Hee University, Seoul 446-701, South Korea. E-mail: zhan2@uh.edu.*

Manuscript received XXX, XXX; revised XXX XXX, XXX; accepted XXX XXX, XXX.(Corresponding authors: Zhe Wang, Jun Li.)

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2021.3140080, IEEE Transactions on Mobile Computing

on Stackelberg dynamic game to allocate edge computing resources between ECNs and drone UTs, where blockchain technology is applied to record and protect the security and privacy of the trading process. However, in practice, as the resources of the distributed ECNs are usually invisible to the UTs, it is difficult for the UTs to directly purchase the services from the ECNs. The data service provider (DSO), acting as an agent, can coordinate the transactions between UTs and ECNs. The most recent work in [18] proposes a blockchain-aided two-level Stackelberg game-based computing resource trading mechanism, where the DSO first rents the computing resources from the ECNs and then leases the resources to the UTs. However, the method proposed in [18] is only applicable to the blockchain aided EC trading with a single DSO. For EC market with multiple DSOs, the key challenge is to handle the competition among multiple DSOs when establishing the trading association among DSOs, ECNs, and UTs in such a blockchain aided EC market. This motivates us to propose a two-level trading association mechanism (i.e., matching based ECNs-DSOs association, and double auction based DSOs-UTs association) in the EC market that aims to achieve system efficiency (i.e., social welfare maximization) while ensuring the truthfulness of the trustless devices.

In addition, a consensus mechanism is a fault-tolerant mechanism to achieve a common agreement on the valid transactions ruled by smart contracts [19]. Proof of Work (PoW) is one of the most prevailing consensus mechanism in many blockchain networks [20]. With PoW, all entities compete to solve an mathematical puzzle to generate the blocks and earn the rewards. Ref. [21] applied the PoW mechanism to manage the data and energy in the blockchain-aided electric vehicle network. In [22], the authors applied PoW in the industrial IoT network to manage the credit value of each entity. However, the process of PoW is extremely computation-consuming, which is not applicable in the EC market. Proof of stake (PoS), has been proposed to address the limitation of PoW. Differing from PoW, the probability that an entity obtains the right to publish a block is determined by its stake, i.e., the coinage [23], [24]. More specifically, each entity earns a higher chance to publish a new block if it owns more coinage. Thus, PoS is beneficial for wealthy entities and may enlarge the wealth inequality among the entities. In addition, these conventional mechanisms incentivize the entities to aggregate either computing power or stake, but ignore service quality. This issue motivates us to propose a PoS design that achieves better fairness and service quality for blockchain-aided EC market.

In this work, we propose several mechanisms in a blockchain-aided EC market aiming to enable efficient and verified transactions among the network entities. The main contributions are summarized as follows.

- *Blockchain-aided EC market*: We study a blockchain-aided EC market consisting of multiple DSOs, ECNs and UTs. As an agent, each DSO first associates with a group of ECNs, and then leases these ECNs to the UTs that require computation offloading services. A trustworthiness model is introduced to evaluate the quality of each entity throughout the transactions. Without a central authority, we adopt blockchain to enable automatic, efficient and verified transactions in a decentralized EC market. More specifically, we propose the smart contract based trading mechanisms to enhance the system efficiency of the automatic transactions, and propose a PoS consensus mechanism to ensure fair and verified transactions.
- *Smart contract based trading mechanisms for automatic and efficient transaction*: We design the trading mechanisms of smart contract over the blockchain to automatically activate the transactions with the aim of maximizing the social welfare in the EC market. First, we design a smart-contract based matching mechanism to establish the one-to-many renting association between the DSOs and ECNs with the goal of maximizing the social welfare. Second, we propose a social welfare improved double auction (SWIDA) mechanism to establish the leasing association between the DSOs and UTs, and determine the pricing of the winners. We prove that the proposed SWIDA mechanism is individually rational and budget balanced. Moreover, we prove that SWIDA is truthful for the DSOs, and show that it is truthful in expectation for the UTs. Furthermore, SWIDA can improve the social welfare compared with the traditional double auction mechanism.
- *Trustworthiness-driven PoS mechanism for transaction verification and fair reward allocation*: We propose a trustworthiness-driven PoS mechanism to enable verified transactions and fair reward allocation in the blockchain-aided EC market. Following the principle of PoS mechanism, we first formulate the block generation as a coalitional game wherein each stakeholder votes according to its trustworthiness and coinage, and then allocate the reward among the coalition according to the Shapley values. Simulation results show that the proposed PoS mechanism can reduce wealth inequality among the entities than the conventional consensus mechanisms.

The rest of this paper is organized as follows. Section 2 describes the EC market model. In Sections 3 and 4, we propose two smart contract based mechanisms, and a trustworthiness-driven PoS mechanism, respectively. Section 5 shows the numerical results. Section 6 concludes this paper. We summarize the main notations in Table 1.

2 System Model

2.1 Distributed EC Market

Consider a typical EC market consisting of G DSOs, M ECNs and N UTs as shown in Fig. 1. Let $\mathcal{U} = \{U_1, U_2, \cdots, U_N\}$ denote the set of UTs with U_n being the n-th UT, $\mathcal{E} = \{E_1, E_2, \cdots, E_M\}$ denote the set of ECNs with E_m being the m-th ECN, and $\mathcal{D} = \{D_1, D_2, \cdots, D_G\}$ denote the set of DSOs with D_g being the g-th DSO.

With limited computing capability, each UT executes the delay-tolerant tasks locally, and offloads the delay-sensitive tasks to a proper ECN that owns more sufficient computing resources. Meanwhile, the ECNs can make profits by leasing their computing resources to the UTs. The DSOs, acting as agents, can coordinate the transactions between UTs

TABLE 1: Key Notations

Notation	Description
U_n	the n-th UT in UT set $\mathcal{U}$
D_g	the g-th DSO in DSO set $\mathcal{D}$
E_m	the m-th ECN in ECN set $\mathcal{E}$
ξ_i	the trustworthiness of entity i
$\Psi_{g,m}$	the estimated utility of DSO D_g
$R_{g,m}$	the rental that D_g promises to pay if the service of E_m is sold
$\gamma_{g,m}$	the estimated utility of ECN E_m
O_g	the maximum number of ECNs that DSO D_g can rent
$A_{g,m}$	D_g's ask on behalf of its rented ECN E_m
V_m^n	U_n's true valuation on ECN E_m
B_m^n	U_n's bid for ECN E_m
w_m^n	the utility of U_n by renting ECN E_m from the DSO
$\bar{w}_m^n$	the estimated utility of U_n if it rents ECN E_m from the DSO
$\pi_{g,m}$	the utility of DSO D_g by leasing ECN E_m to the UT
$\hat{\mathcal{U}}$	the winning UT set
$\hat{\mathcal{E}}$	the winning ECN set
$\sigma(\cdot)$	the association function that maps the UT to the ECN
P_m^n	the payment that the UT U_n is charged for renting ECN E_m
$I_{g,m}$	the reward that D_g receives for leasing ECN E_m.
$A_\jmath$	the median ask
$B_\imath$	the threshold bid
$\mathcal{B}_m$	the original bid list of E_m for payment determination
$\mathcal{B}_m$	the bid list of E_m for association determination
C_j	the coinage of entity j
X_j	the stake of entity j

and ECNs. As the number of ECNs and their computing resources are invisible to the UTs, the UTs can only purchase the computing services from the DSOs. In this context, the transactions in the EC market operate over two phases:

- **ECN association phase**: Each DSO builds the renting association with a group of ECNs, i.e., this DSO becomes an agent of these ECNs.
- **ECN leasing phase**: Each DSO sublets the ECNs to the UTs, and pays the rental to the ECN once it has sold this ECN's computation offloading service to a UT. We assume that each ECN can serve at most one UT, and each UT can be served by at most one ECN and does not move out of the coverage area of its associated ECN during the service time.

2.2 Trustworthiness Model

To build a trusted market, we employ a trustworthiness model to assess the entity. Due to the fact that the trustworthiness is more about whether the entity can fullfill the service task/payment as it promises, which is not directly related to the energy and computing power availability of this entity, we can regard any UT, DSO, or ECN as "entity i". We define the entity i's transaction reputation $\mathrm{Rep}(e_k^i) \in [0,1]$ as the normalized service quality of this entity in the k-th transaction evaluated by its trading partner. For example, the transaction reputation value of a service seller is the normalized service quality (e.g., timeliness of computing services) provided by this seller, and the transaction reputation value of the buyer is the normalized service quality (e.g., timeless of payment) of this buyer evaluated by its seller. After an ECN owned by a DSO computes a UT's task, this UT assesses the DSO's service by the transaction reputation value, and the DSO assesses this ECN with the same transaction reputation value. Meanwhile, this DSO also assesses the transaction reputation of the UT. Since different entity i may experience different number of transaction K_i, we adopt the average transaction reputation ϑ_i to reflect the long-term service quality of entity i, i.e., $\vartheta_i = \frac{1}{K_i}\sum_{k=1}^{K_i}\mathrm{Rep}(e_k^i)$. Note that once the entity i's average transaction reputation is low (e.g., below a threshold), it will also be added to the blacklist and prohibited from trading

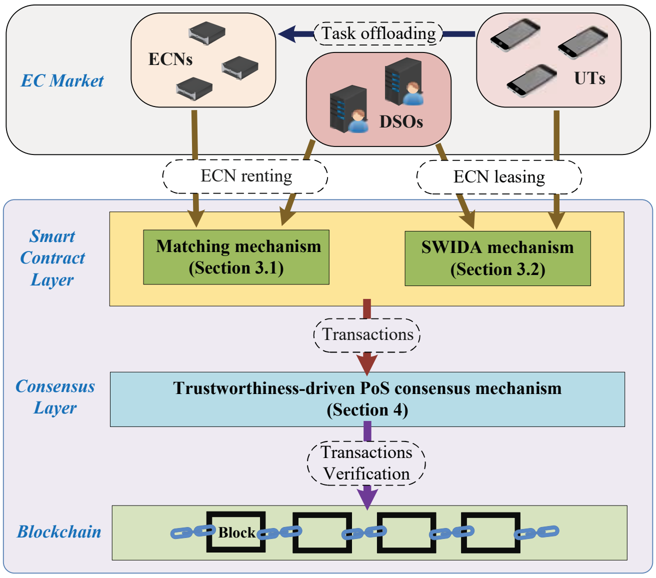


Fig. 1. The diagram of a blockchain-aided EC market.

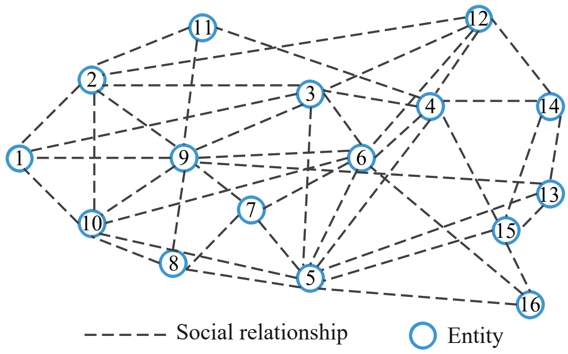


Fig. 2. Illustration of social relationships of entities in the distributed EC market.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2021.3140080, IEEE Transactions on Mobile Computing

with other entities.

From [25], the trustworthiness of entity i is expressed as

$$\xi_i = \mu\vartheta_i + (1-\mu)\beta_i, \tag{1}$$

where ϑ_i is the average transaction reputation, β_i is the betweenness, and $\mu \in [0,1]$ is a weight. As shown in Fig. 2, the pairs of entities are socially related if any transaction occurs between them. The betweenness of an entity reflects the proportion of the shortest path between all pairs of nodes passing through this entity. A large value of betweenness means that the entity is well-known as a bridge to interconnect other entities [26]. The betweenness of entity i is defined as

$$\beta_i = \sum\nolimits_{j=1}^{Q} \sum\nolimits_{j<b} \frac{y_{j,b}(i)}{Y_{j,b}}, \tag{2}$$

where $Q = N + M + G$ is the total number of all entities, $Y_{j,b}$ is the number of shortest links between entities j and b, and $y_{j,b}(i)$ is the number of shortest links between entities j and b that pass through entity i. As the weighted sum of the average transaction reputation and betweenness, the trustworthiness can be intuitively regarded as a "good fame", where the weight μ balances between "good" (i.e., the quality of transaction offered by this entity) and "fame" (i.e., the quantity of connections between this entity and other peers).

Transaction ID
Seller's Address
Buyer's Address
Payload data: <Computing frequency of the ECN, transaction reputation, betweenness, trustworthiness, payment value, transaction fee>
Buyer's signature

Fig. 3. Data structure of a single transaction.

2.3 Blockchain-Aided EC Market

In an EC market, a central authority for transaction recording may not be trusted and is vulnerable to the single point of failure. In this work, we employ the blockchain technology to manage the transactions (including both the ECN renting and leasing) in the distributed EC market. We consider each entity (i.e., a DSO, an ECN, or a UT) can be registered as both a client of the EC market transactions and a validator for these transactions on the blockchain. Note that the validation process is via voting-based PoS consensus mechanism, which consumes much less computation/energy on the energy-constrained devices compared with the traditional PoW consensus mechanism. As shown in Fig. 3, a transaction on the blockchain [16] can be structured as:

- Seller's address;
- Buyer's address;
- Payload data: payment value and the auxiliary information (e.g., computing frequency of the ECN, transaction reputations of seller and buyer, betweenness, and trustworthiness of the entity, payment value, transaction fee);
- Buyer's signature: the publicly verifiable digital signature of the buyer.

After each transaction, the information of the ECNs, DSOs, and UTs (e.g., computing frequency, transaction reputation, betweenness, and trustworthiness) will be stored on the blockchain in form of the payload data of this transaction. As shown in Fig. 1, we first design the trading mechanisms of the smart contract over the blockchain to automatically activate the transactions with the aim of maximizing the social welfare in the EC market (Section 3). Moreover, we develop the consensus mechanism to incentivize the entities towards fair and verified transactions (Section 4).

3 Smart Contract Based Trading Mechanisms

In this section, we design two smart contract enabled trading mechanisms for the ECN renting and leasing phases, respectively.

3.1 Matching Mechanism for ECN Association Phase

In the ECN association phase, each DSO targets at renting a group of ECNs that provide the highest estimated utilities, and the ECN aims to be rented by the DSO that provides the highest estimated utility.

Each DSO has different preferences over the ECNs depending on the ECN's trustworthiness and computing capabilities. Let $\tau_{g,m}$ denote the preference of DSO D_g for ECN E_m, i.e.,

$$\tau_{g,m} = \alpha_g \xi_m + (1-\alpha_g) f_m, \tag{3}$$

where ξ_m is the trustworthiness in (1), f_m is the computing frequency of ECN E_m, and $\alpha_g \in [0,1]$ is the weighting factor. We define the estimated utility of DSO D_g as

$$\Psi_{g,m} = \Lambda_g(\tau_{g,m}) - R_{g,m}, \tag{4}$$

where $\Lambda_g(\cdot)$ is estimated income function that positively correlates with $\tau_{g,m}$ and varies across DSOs. Moreover, $R_{g,m}$ is the rental that D_g promises to pay if the service of E_m is sold to a UT later. We define it as

$$R_{g,m} = \mho_g(\tau_{g,m}), \tag{5}$$

where $\mho_g(\cdot)$ is the rental function of D_g. We further define the estimated utility of ECN E_m, if it is rented by DSO D_g and leased to a UT by D_g, as

$$\gamma_{g,m} = R_{g,m} - \zeta\kappa(f_m)^2, \tag{6}$$

where ζ is the reference cost caused per CPU cycle, and $\kappa(f_m)^2$ is the energy consumed by a CPU cycle [27].

To build the renting association between the DSOs and UTs, we design a smart contract based matching mechanism aiming at maximizing the social welfare (i.e., sum of estimated utilities of all DSOs and ECNs), where the social welfare for the ECN association phase is given by

$$\begin{aligned}\Re_{\text{wel}} &= \sum_{g=1}^{G}\sum_{m=1}^{M} x_{g,m}(\Psi_{g,m}+\gamma_{g,m}) \\ &= \sum_{g=1}^{G}\sum_{m=1}^{M} x_{g,m}(\Lambda_g(\tau_{g,m})-\zeta\kappa(f_m)^2). \qquad (7)\end{aligned}$$

Let $x_{g,m} \in \{0,1\}$ denote the association between DSO D_g and ECN E_m. Concretely, $x_{g,m} = 1$ if E_m is rented. Otherwise, $x_{g,m} = 0$. The association problem for the ECN association phase is given by

$$\max_{x_{g,m}} \Re_{\text{wel}} \qquad (8a)$$

$$\text{s.t. } x_{g,m} \in \{0,1\} \qquad (8b)$$

$$\sum_{m=1}^{M} x_{g,m} \le O_g, \quad g = 1,\cdots,G, \qquad (8c)$$

$$\sum_{g=1}^{G} x_{g,m} \le 1, \qquad m = 1,\cdots,M, \qquad (8d)$$

where (8c) indicates that DSO D_g can rent at most O_g number of ECNs, and (8d) tells that an ECN can only be rented by at most one DSO.

The smart contract usually provides many functions, and the entity can invoke the functions by sending messages to the smart contract. To establish the association between DSOs and ECNs, we design three main functions of smart contract in the association phase as follows:

1) An "upload" function that enables ECN and DSO to upload messages to the smart contract.
2) A "matching" function that enables each DSO to rent a group of ECNs.

As shown in Algorithm 1, we propose a smart contract based matching mechanism to solve the association problem in (8).

Algorithm 1 Smart Contract Based Matching Mechanism.

1: Each ECN E_m calls the "upload" function and sends f_m and $\gamma_{g,m}$ to smart contract
2: Each DSO D_g in $\mathcal{D}$ calls the "upload" function and reports its estimated rental and estimated utility function pairs for ECN E_m, i.e., $(R_{g,m}, \Psi_{g,m})$, and its maximum allowable number of matched ECNs O_g to smart contract
3: Smart contract verifies f_m and calculates ξ_m, and then executes the matching function:
4: Initialize $O_g^{(t)} = O_g, \forall g \in \{1,2,...,G\}$ and $\mathcal{D}^{(t)} = \mathcal{D}$
5: **while** either $O_g^{(t)} \neq 0, \forall g \in \{1,2,...,G\}$ or existing ECNs are unmatched **do**
6:   Calculate each unmatched ECN's utility from each DSO in $\mathcal{D}^{(t)}$ according to $\gamma_{g,m}$ and identifies DSO $D_{g'}$ that provides the highest utility
7:   Identify the ECNs that obtain the highest utility from D_g, $\forall D_g \in \mathcal{D}^{(t)}$, and calculate the number of these ECNs $l_g^{(t)}$
8:   **for** $\forall D_g \in \mathcal{D}^{(t)}$ **do**
9:     **if** $l_g^{(t)} > O_g^{(t)}$ **then**
10:       Match DSO D_g with the top $O_g^{(t)}$ ECNs that provide the largest utilities to D_g
11:       Update $O_g^{(t+1)} = 0$, $\mathcal{D}^{(t+1)} = \mathcal{D}^{(t)} \setminus \{D_g\}$
12:     **else**
13:       Match DSO D_g with all $l_g^{(t)}$ ECNs
14:       Update $O_g^{(t+1)} = O_g^{(t)} - l_g^{(t)}$
15:     **end if**
16:   **end for**
17:   Go to step 5
18: **end while**

First, each ECN E_m calls the "upload" function by sending the computing frequency f_m and estimated utility function ($\gamma_{g,m}$ in (6)) to smart contract. Second, each DSO calls the "upload" function by reporting the estimated rental function ($R_{g,m}$ in (5)) and the estimated utility function ($\Psi_{g,m}$ in (4)) to smart contract (lines 1-2). Furthermore, smart contract verifies computing frequency[1] f_m and calculates trustworthiness ξ_m of ECN E_m according to social network (e.g., in Fig. 2) and Eqns. (1) and (2), and then calls the "matching" function to establish the association between the ECNs and DSOs (lines 3 to 17). In round t, smart contract estimates each unmatched ECN's utility obtained by associating with each DSO in the set of $\mathcal{D}^{(t)}$ according to $\gamma_{g,m}$, and then identifies the DSO that provides the highest utility (e.g., $D_{g'}$). Since the number of ECNs $l_g^{(t)}$ that obtain the highest utility from a DSO D_g may exceed its association constraint $O_g^{(t)}$, we discuss the matching process in the following two cases. If $l_g^{(t)} > O_g^{(t)}$ (lines $9-11$), smart contract matches D_g with the top $O_g^{(t)}$ ECNs that provide the highest estimated utilities for D_g from above $l_g^{(t)}$ ECNs. If $l_g^{(t)} \le O_g^{(t)}$ (lines 12 to 15), smart contract matches D_g with all $l_g^{(t)}$ ECNs and updates $O_g^{(t+1)} = O_g^{(t)} - l_g^{(t)}$. Smart contract terminates if either $O_g^{(t)} = 0, \forall g \in \{1,2,...,G\}$ or all ECNs are matched.

Based on Algorithm 1, we have established the one-to-many associations between the DSOs and ECNs. If a DSO successfully sells the computational resource of its associated ECN to a UT in the ECN leasing phase later, it will pay this ECN with the amount of money that equals to its rental $R_{g,m}$. Otherwise, the DSO pays zero rental to this ECN and there is no need for the ECN to serve for UTs.

3.2 Double Auction Mechanism for ECN Leasing Phase

In this subsection, we propose a double auction mechanism for the ECN leasing phase. First, we design two main functions of the smart contract in this phase as follows:

1) An "upload" function that enables DSOs and UTs to upload messages to smart contract.
2) A "double auction" function that enables the UTs to purchase the ECNs' computing services from the DSOs for these ECNs.

3.2.1 *Smart Contract Based Double Auction*

We now introduce the smart contract based double auction design in details. As shown in Fig. 4, the DSOs and UTs are sellers and buyers, respectively, and the smart contract can work as an auctioneer to establish the automatic transactions between the DSOs and UTs via the proposed double auction mechanism.

1. If the computing frequency f_m reported by ECN E_m is inconsistent with that stored in past transactions, E_m will be added to the blacklist.

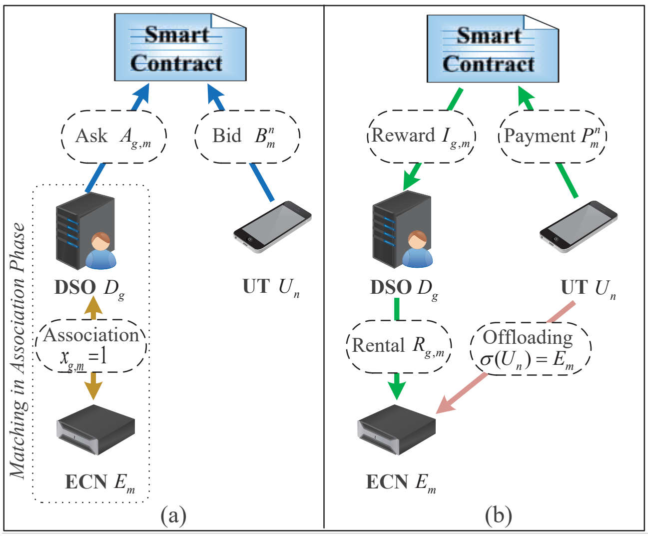


Fig. 4. Double auction mechanism of smart contract.

First, as shown in Fig. 4(a), each DSO D_g calls the "upload" function by sending ask $A_{g,m}$ and computing frequency f_m of each of its associated ECN E_m (obtained from the ECN association phase) to the smart contract. Here, $A_{g,m}$ is the ask submitted by DSO D_g on behalf of its associated ECN E_m, which can be different from its rental $R_{g,m}$ for E_m. The ask matrix consisting of the ask vectors of all DSOs is denoted by $\mathbf{A}$.

Second, smart contract verifies computing frequency f_m and calculates trustworthiness ξ_m of each ECN E_m from the blockchain, and broadcasts them to the UTs.

Third, with the knowledge of f_m and ξ_m, each UT U_n calculates its true valuation V_m^n for each ECN E_m, i.e.,

$$V_m^n = (1-\chi_n)\left\{\delta_n\left(1-\frac{f_n}{f_m}\right)\right\} + \chi_n\xi_m, \tag{9}$$

where f_n is the computing frequency of U_n, δ_n is the reference profit achieved by the computation speedup of a CPU cycle, $\delta_n(1-f_n/f_m)$ is the income of procedural acceleration, and $\chi_n \in [0,1]$ is a weighting factor. Then, each U_n selects the ECNs that have higher computing frequencies than itself, i.e., ECN E_m with $f_m > f_n$, and then calls the "upload" function by sending its bid B_m^n for each of its intended ECN E_m to the smart contract. Note that B_m^n can be different from its true valuation V_m^n. We will discuss whether or not the DSOs and UTs would truthfully report their asks (i.e., $A_{g,m} = R_{g,m}$) and bids (i.e., $B_m^n = V_m^n$) to the smart contract under the proposed double auction mechanism in Section 3.2.4. The bid matrix consisting of the bids of all UTs is denoted by $\mathbf{B}$.

Last, given $\mathbf{A}$ and $\mathbf{B}$, the smart contract calls the "double auction" function to decide winning UT set $\hat{\mathcal{U}} \subseteq \mathcal{U}$, winning ECN set $\hat{\mathcal{E}} \subseteq \mathcal{E}$, the association between $\hat{\mathcal{U}}$ and $\hat{\mathcal{E}}$, i.e., $\sigma(U_n \in \hat{\mathcal{U}}) = E_m \in \hat{\mathcal{E}}$, payment P_m^n that winning UT $U_n \in \hat{U}$ is charged for renting ECN E_m, and reward $I_{g,m}$ that DSO D_g receives for leasing ECN $E_m \in \hat{\mathcal{E}}$. Consider DSO D_g finally sells the computing resources of ECN E_m to a proper UT U_n as shown in Fig. 4(b). We have the following transactions: 1) UT U_n pays P_m^n to the smart contract; 2) the smart contract pays the reward $I_{g,m}$ to DSO D_g; 3) DSO pays rental that equals to its estimated rental $R_{g,m}$ to ECN E_m. Furthermore, the utility of UT U_n is given by

$$w_m^n = \begin{cases} V_m^n - P_m^n, & \text{if } U_n \in \hat{\mathcal{U}}, \\ 0, & \text{otherwise}. \end{cases} \tag{10}$$

The utility of DSO D_g by leasing ECN E_m is given by

$$\pi_{g,m} = \begin{cases} I_{g,m} - R_{g,m}, & \text{if } E_m \in \hat{\mathcal{E}}, \\ 0, & \text{otherwise}. \end{cases} \tag{11}$$

3.2.2 *Desired Properties for Auction Mechanism*

A feasible auction mechanism should first satisfy the following desirable properties.

1) *Balanced Budget*: A double auction is budget balanced if the smart contract (auctioneer) does not lose money in the transaction. In other words, the sum rewards paid to all DSOs should be no more than the sum payment from all UTs, i.e., $\sum_{U_n \in \hat{\mathcal{U}}} P_m^n - \sum_{E_m \in \hat{\mathcal{E}}} I_{g,m} \geq 0$.
2) *Truthfulness*: A double auction mechanism is strongly truthful when the UTs or DSOs cannot improve their utilities by untruthfully submitting their bids or asks to the smart contract [28]. Specifically, in our auction model, UT U_n's utility w_m^n is maximized when its bid for ECN E_m equals to its true valuation for this ECN, i.e., $B_m^n = V_m^n$. Moreover, DSO D_g's utility $\pi_{g,m}$ is maximized when its reported ask on behalf of ECN E_m in the ECN leasing phase equals to the rental paid to E_m in the ECN association phase, i.e., $A_{g,m} = R_{g,m}$. Furthermore, a weaker truthfulness is the truthfulness in expectation, where the UTs or DSOs cannot improve their expected utilities via untruthful bidding [29].
3) *System Efficiency*: The system efficiency is measured in terms of social welfare in our double auction, which is defined as the sum utility of all DSOs and UTs, i.e.,
$$\Pi_{\text{auc}} = \sum_{U_n \in \hat{\mathcal{U}}} w_m^n + \sum_{E_m \in \hat{\mathcal{E}}} \pi_{g,m}. \tag{12}$$
4) *Individual Rationality*: We assume all the buyers and sellers are rational, i.e., no one should lose from joining the auction. Particularly, the payment P_m^n of any winning UT U_n should be no more than its bid B_m^n, i.e., $P_m^n \leq B_m^n$. Moreover, for any DSO D_g that owns a winning ECN, its received reward $I_{g,m}$ should be no less than its ask $A_{g,m}$, i.e., $I_{g,m} \geq A_{g,m}$.

However, the well-known result in [30] reveals that it is next to impossible to design a strongly truthful, efficient, and budget-balanced double auction, even putting individual rationality aside. For example, McAfee double auction in [31] is individually rational and truthful, but neither budget-balanced nor efficient. Moreover, VCG double auction in [32] is individually rational, truthful, and efficient, but not budget-balanced. However, for the blockchain system, smart contract cannot insert money into the system to meet the budget-balanced requirement. To solve this problem, the incentive-compatible auction mechanism (ICAM) in [33] is individually rational, budget-balanced, and strongly truthful, but suffers from a relatively low social welfare. To further improve the system efficiency, we propose a social welfare improved double auction

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2021.3140080, IEEE Transactions on Mobile Computing

Algorithm 2 Determination of Candidates.

Input: $\mathbf{A}$, $\mathbf{B}$
Output: $\mathcal{A}, \mathcal{B}$
$\mathcal{A} \leftarrow \emptyset, \mathcal{B} \leftarrow \emptyset$
Sort all the asks from matrix $\mathbf{A}$ in ascending order, i.e., $\mathcal{A} = \{A_1, A_2, ..., A_\lambda\}$, $A_1 \leq A_2 \leq ... \leq A_\lambda$, where λ is total number asks
Sort all the bids from matrix $\mathbf{B}$ in descending order, i.e., $\mathcal{B} = \{B_1, B_2, ..., B_\eta\}$, $B_1 \geq B_2 \geq ... \geq B_\eta$, where η is total number of all bids
Find the median of $\mathcal{A}$ and define it as $A_\jmath$, where $\jmath = \left\lfloor \frac{\lambda+1}{2} \right\rfloor$
$\mathcal{A} \leftarrow \{A_1, A_2, ..., A_\jmath\}$
Find the minimum bid that is no less than $A_\jmath$ and define it as the threshold bid $B_\imath$
$\mathcal{B} \leftarrow \{B_1, B_2, ..., B_\imath\}$

(SWIDA) mechanism that achieves individual rationality, balanced budget, truthfulness and a higher social welfare than the traditional ICAM. The discussions and proofs will be detailed in Section 3.2.4.

3.2.3 Proposed SWIDA Mechanism

Next, we introduce the main procedures of the proposed SWIDA mechanism, which includes two stages, i.e., determination of candidates (Algorithm 2), determination and pricing of winners (Algorithm 3).

Determination of Candidates: In this stage, the smart contract shortlists the candidates of ECNs and UTs. To reduce the computational complexity, we first remove the ECNs with high asks and the UTs with low bids from the candidate set via Breakeven mechanism in Algorithm 2. First, we sort all the asks from $\mathbf{A}$ in ascending order to obtain a new set $\mathcal{A}$. Then, we sort all the bids from $\mathbf{B}$ in descending order and get a new set $\mathcal{B}$. We find the median of $\mathcal{A}$ and denote the median ask by $A_\jmath$, where $\jmath = \left\lfloor \frac{\lambda+1}{2} \right\rfloor$. Then, we remove the reported asks that are higher than $A_\jmath$ from set $\mathcal{A}$, and delete the bids that are less than $A_\jmath$ from set $\mathcal{B}$. We denote the smallest bid that is no less than median ask $A_\jmath$ as threshold bid $B_\imath$. The UTs whose bids are in the updated set of $\mathcal{B}$ and ECNs whose asks are in the updated set of $\mathcal{A}$ become the candidate buyers and sellers, respectively.

Determination and Pricing of Winners: In this seal-bid double auction, it is possible that one UT bids for multiple ECNs and one ECN receives bids from multiple UTs. We adopt Algorithm 3 to determine the one-to-one pairing between the UT and the pricing rules for the winners, which consists of two steps as follows.

Step 1: Winning UT determination (lines 4-10). In Algorithm 2, we have obtained the ECN candidates' ask set $\mathcal{A}$ and UT candidates' bid set $\mathcal{B}$. For each ECN E_m whose ask is in set $\mathcal{A}$, we first select its received bids from set $\mathcal{B}$, and then construct a set $\mathcal{B}_m$ by sorting these bids in descending order. In the case of a tie (i.e., more than one bid has the same value), we arrange these bids with the equal value in a random order. We denote the highest bid in set $\mathcal{B}_m$ by $B_m^{(1)}$ and regard the corresponding UT $U_m^{(1)}$ as the potential winning UT for ECN E_m. We then add this bid to the highest bid set $\mathcal{B}^{(1)}$.

Step 2: Winning ECN determination and pricing (lines 11-36). It is possible that one potential winning UT is the

Algorithm 3 Determination and Pricing of Winners.

1: **Input:** $\mathcal{B}, \mathcal{A}$
2: **Output:** $\hat{\mathcal{U}}$, $\hat{\mathcal{E}}$, σ
3: $\hat{\mathcal{U}} \leftarrow \emptyset$, $\hat{\mathcal{E}} \leftarrow \emptyset$, $\bar{\mathcal{B}} \leftarrow \mathcal{B}$
Winning UT Determination:
4: **for** E_m with its ask in $\mathcal{A}$ **do**
5: Construct a set consisting of the bids for E_m in $\bar{\mathcal{B}}$ $\bar{\mathcal{B}}_m = \{\bar{B}_m^{(q)} : \forall \bar{U}_m^{(q)}$ whose bid for E_m in $\bar{\mathcal{B}}\}$, where $\bar{B}_m^{(q)}$ is the q-th highest bid for E_m submitted by $\bar{U}_m^{(q)}$ in $\bar{\mathcal{B}}_m$.
6: **end for**
7: **for** E_m with its ask in $\mathcal{A}$ **do**
8: Construct a set consisting of the bids for E_m in $\mathcal{B}$ $\mathcal{B}_m = \{B_m^{(q)} : \forall U_m^{(q)}$ whose bid for E_m in $\mathcal{B}\}$, where $B_m^{(q)}$ is the q-th highest bid for E_m submitted by $U_m^{(q)}$ in $\mathcal{B}_m$.
9: **end for**
10: Construct highest bid set $\mathcal{B}^{(1)} = \{B_m^{(1)} : \forall E_m$ with its ask in $\mathcal{A}\}$
Winning ECN Determination and Pricing:
11: **if** $\mathcal{B}^{(1)} \neq \emptyset$ **then**
12: Randomly select a bid $B_m^{(1)} \in \mathcal{B}^{(1)}$
13: Construct a set consisting of all $U_m^{(1)}$'s bids that are in $\mathcal{B}^{(1)}$ $\{B_m^{(1)} : m \in \{1, 2, ..., H\}\}$
14: Find $B_m^{(1)}$'s rank r in $\bar{\mathcal{B}}_m$
15: **for** $m = 1$ to H **do**
16: **if** $r = \mid \bar{\mathcal{B}}_m \mid$ **then**
17: $P_m^{(1)} = B_\imath$
18: **else**
19: $P_m^{(1)} = \bar{B}_m^{(r+1)}$
20: **end if**
21: $\bar{w}_m^{(1)} = B_m^{(1)} - P_m^{(1)}$
22: **end for**
23: $m' = \underset{m \in \{1, \cdots, H\}}{\text{argmax}} \ \bar{w}_m^{(1)}$
24: $\sigma(U_m^{(1)}) = E_{m'}$
25: $\hat{\mathcal{U}} \leftarrow \hat{\mathcal{U}} \bigcup U_m^{(1)}$
26: $\hat{\mathcal{E}} \leftarrow \hat{\mathcal{E}} \bigcup E_{m'}$
27: $I_{g,m'} = A_\jmath$
28: DSO D_g pays rental $R_{g,m}$ to its associated ECN $E_{m'}$
29: Remove all bids submitted by $U_m^{(1)}$ from $\mathcal{B}$
30: Remove the ask regarding ECN $E_{m'}$ from $\mathcal{A}$
31: Go to step 7
32: **else**
33: Go to step 36
34: **end if**
35: **return** $(\hat{\mathcal{U}}, \hat{\mathcal{E}}, \sigma)$

highest bidder for multiple ECNs, but it can only choose one ECN to associate with. A simple and efficient solution is to associate this UT with the ECN that provides the highest utility. In order to prevent untruthful bidding, we start with a randomly ordered list of the potential winning UTs. More specifically, we randomly select a bid from the highest bid set $\mathcal{B}^{(1)}$, e.g., we choose the highest bid $B_m^{(1)}$ for ECN E_m submitted by UT $U_m^{(1)}$ (line 12). Suppose that UT $U_m^{(1)}$ is the highest bidder for H ECNs, where the integer $H \in [1, |\mathcal{B}^{(1)}|]$. We add UT $U_m^{(1)}$'s bids for its intended ECNs into a new set $\{B_m^{(1)} : m \in \{1, 2, ..., H\}\}$ (line 13). We then evaluate the estimated utility $\bar{w}_m^{(1)}$ that UT $U_m^{(1)}$ obtains by

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2021.3140080, IEEE Transactions on Mobile Computing

associating with each of these ECNs, which is expressed as $\bar{w}_m^{(1)} = B_m^{(1)} - P_m^{(1)}$ (line 21), where $P_m^{(1)}$ is the payment from UT $U_m^{(1)}$ to the smart contract if it associates with E_m. Note that, since the smart contract does not know the true valuation of the UTs, the estimated utility for each UT is based on its bid instead of the true valuation.

To further enhance the truthfulness, we construct another set $\bar{\mathcal{B}}_m$ with the same initial components as $\mathcal{B}_m$ for price determination. Note that we use set $\mathcal{B}_m$ to determine the winning UT for ECN E_m, and adopt another set $\bar{\mathcal{B}}_m$ to decide the payment for the UTs, respectively. We keep updating $\mathcal{B}_m$ by eliminating the UTs that have been successfully associated with other ECNs in each iteration, while adopting the same $\bar{\mathcal{B}}_m$ throughout the whole game. As a result, the highest bid (i.e., $B_m^{(1)}$) in $\mathcal{B}_m$ may not also be the highest in $\bar{\mathcal{B}}_m$ at an arbitrary iteration. We now assume $B_m^{(1)}$ is the r-th highest bid in set $\bar{\mathcal{B}}_m$, where $r = 1$ or $r = |\bar{\mathcal{B}}_m|$ means the $B_m^{(1)}$ is the highest or lowest bid, respectively. We will determine its payment $P_m^{(1)}$ in the following two cases. If $r = |\bar{\mathcal{B}}_m|$, UT $U_m^{(1)}$ pays the threshold bid of $B_\imath$. Otherwise, it pays the highest bid that is no greater than its own bid in $\bar{\mathcal{B}}_m$, i.e., $P_m^{(1)} = \bar{B}_m^{(r+1)}$ (lines 16-20).

Finally, UT $U_m^{(1)}$ associates with ECN $E_{m'}$ that provides the highest estimated utility among all ECNs, where the association between $U_m^{(1)}$ and $E_{m'}$ is represented by $\sigma(U_m^{(1)}) = E_{m'}$ (line 23). Once the association is established, UT $U_m^{(1)}$ pays $P_{m'}^{(1)}$ to the smart contract. To guarantee the strongly truthfulness of the DSO, the DSO that owns ECN $E_{m'}$ is rewarded by median ask $A_\jmath$, i.e., $I_{g,m'} = A_\jmath$ (line 27). In the meanwhile, the DSO pays the rental to its associated ECN $E_{m'}$ (line 28). Finally, this UT and its associated ECN are added to the winning UT set $\hat{\mathcal{U}}$ and winning ECN set $\hat{\mathcal{E}}$, respectively (lines 25-26). Moreover, their corresponding bids and asks will be removed from the bid set $\mathcal{B}$ and ask set $\mathcal{A}$ to avoid double association (lines 29-30).

By iteratively operating steps 1 and 2 until $\mathcal{B}^{(1)} = \emptyset$, the smart contract finally establishes the one-to-one association between the UTs and ECNs.

3.2.4 Proof of Desirable Properties

In the following, we prove several key properties for the proposed SWIDA mechanism.

Theorem 1. *SWIDA is individually rational.*

Proof: For any winning UT U_n, we can deduce that its payment $P_m^n \in [B_\imath, B_m^n]$ according to lines 16-20 of Algorithm 3. Since its payment P_m^n is less than its bid B_m^n, we have proved that SWIDA is individual rational for any winning UT U_n.

For each DSO D_g that owns a winning ECN, its received reward from the smart contract for leasing this ECN to the corresponding UT equals the median ask $A_\jmath$, which is no less than their own asks, i.e., $I_{g,m} = A_\jmath \geq A_{g,m}$ according to Algorithms 2 and 3. As a result, SWIDA is individually rational for any DSO that owns the winning ECNs.

Moreover, for the UT (or DSO) that does not win the auction, its payment (or reward) and utility are zero. Therefore, it also does not lose money by participating in the auction. □

Theorem 2. *SWIDA is budget-balanced.*

Proof: According to Algorithm 2, we know that the threshold bid $B_\imath$ is no less than the medium ask $A_\jmath$. Based on the proof of Theorem 1, we can deduce that $P_m^n \geq B_\imath \geq A_\jmath = I_{g,m}$ holds for each association pair of UT U_n and ECN E_m. We therefore can further deduce that the total reward paid to all DSOs is no more than the total payment received from all UTs, i.e.,

$$\sum_{U_n \in \hat{\mathcal{U}}} P_m^n - \sum_{E_m \in \hat{\mathcal{E}}} I_{g,m} \geq 0. \tag{13}$$

□

Theorem 3. *SWIDA is truthful for DSOs.*

Proof: We will prove that each DSO D_g achieves the maximum utility $\pi_{g,m}$ when it truthfully reports the ask of its rented ECN E_m to the smart contract, i.e., $A_{g,m} = R_{g,m}$ (D_g's ask on behalf of E_m in the ECN leasing phase equals the estimated rental it promised to pay to E_m in the ECN association phase). To prove this property, we first define $\mathcal{E}^{\mathrm{o}}$ as the set of all ECNs that have been associated with the DSOs in the ECN association phase. Assuming all the DSOs truthfully report the asks of their rented ECNs, we divide $\mathcal{E}^{\mathrm{o}}$ into two subsets: $\mathcal{E}^{\mathrm{d}}$ and $\mathcal{E}^{\mathrm{o}} \setminus \mathcal{E}^{\mathrm{d}}$, where $\mathcal{E}^{\mathrm{d}}$ is the ECN candidate set obtained via Algorithm 2, and $\mathcal{E}^{\mathrm{o}} \setminus \mathcal{E}^{\mathrm{d}}$ is the set of the remaining ECNs. In the rest of the proof, we will see whether or not the DSOs can improve their utilities by untruthfully reporting the asks.

Case 1: $E_m \in \mathcal{E}^{\mathrm{o}} \setminus \mathcal{E}^{\mathrm{d}}$. In this case, we consider ECN E_m is not a candidate via Algorithm 2 given DSO D_g truthfully reports its ask (i.e., $A_{g,m} = R_{g,m}$). This implies that the truthful ask of this ECN is higher than the medium ask, i.e., $R_{g,m} > A_\jmath$, and the utility of D_g by leasing this ECN is $\pi_{g,m}(R_{g,m}) = 0$. Based on Algorithms 2 and 3, the DSO will lose the auction by increasing the ask, and it may win the game by decreasing the ask. In the following two subcases, we will discuss whether or not the DSO can improve its utility by decreasing its ask.

1) If DSO D_g decreases the ask of ECN E_m and this ECN finally becomes the winning ECN via Algorithm 3, the DSO receives the medium ask as the reward (i.e., $I_{g,m} = A_\jmath$). However, since $R_{g,m} > A_\jmath$, the utility of D_g becomes negative which is smaller than that of truthful reporting, i.e., $\pi_{g,m}(A_{g,m}) = I_{g,m} - R_{g,m} = A_\jmath - R_{g,m} < 0 = \pi_{g,m}(R_{g,m})$.
2) If DSO D_g decreases the ask and E_m does not become the winning ECN, the utility of D_g is still zero, which is the same as that of truthful reporting.

Case 2: $E_m \in \mathcal{E}^{\mathrm{d}}$. In this case, we consider ECN E_m becomes the ECN candidate via Algorithm 2 given DSO D_g truthfully reports its ask. This implies that the DSO has a non-negative utility, where it has positive utility if it wins the game and it obtains zero utility if it does not win via Algorithm 3.

1) If DSO D_g untruthfully reports the ask and ECN E_m is not a candidate via Algorithm 2, the DSO receives zero utility which is no more than that of truthful reporting, i.e., $\pi_{g,m}(A_{g,m}) = 0 \leq \pi_{g,m}(R_{g,m})$.
2) If DSO D_g untruthfully reports the ask and ECN E_m is still one of candidates after Algorithm 2, the utility of D_g is the same as that of truthful reporting, i.e., $\pi_{g,m}(A_{g,m}) =$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2021.3140080, IEEE Transactions on Mobile Computing

$\pi_{g,m}(R_{g,m})$. This is because whether or not this ECN can become a winning ECN does not depend on its ask but decided by the bids of the UTs, as shown in Algorithm 3.

To sum up, the DSO cannot improve its utility by untruthfully reporting, which completes the proof. □

Different from the DSOs that satisfy strong truthfulness, the UTs in the proposed SWIDA mechanism only satisfy weak truthfulness. In the strongly truthful double auction like ICAM [33], truthful reporting is the weakly dominant strategy for all UTs, i.e., no UT can improve its utility by untruthfully submitting its bids. However, the social welfare is relatively low in such a mechanism. Compared with the ICAM, our proposed SWIDA mechanism improves the social welfare at the cost of certain degree of truthfulness. The definition of the weak truthfulness is similar to that of [29], i.e., no buyer can improve its expected utility via untruthful bidding. In the proposed SWIDA, each UT can always obtain a non-negative utility by truthful bidding, but may receive a non-positive utility by bidding untruthfully. Moreover, a UT may lose an auction with a lie while it should have won with a truthful bid. Compared with efficient design of auction (EDA) mechanism in [29], we adopt social welfare instead of successful trading pairs as the objective, making the design of the double auction mechanism even more challenging. Furthermore, we provide a comprehensive proof to illustrate the risks for the UTs to lie in the following proposition.

Proposition 1. *For the proposed SWIDA mechanism, any untruthful bidding strategy (i.e., underbidding or overbidding) that potentially leads to a positive improvement in a UT's utility also imposes a risk to reduce its utility.*

Proof: We denote the valuation and bid of UT U_n for ECN E_m by V_m^n and B_m^n, respectively. Given each UT truthfully submits its bid to the smart contract (i.e., $B_m^n = V_m^n$), we can divide UT set $\mathcal{U}$ into two subsets: winning UT set $\hat{\mathcal{U}}$, and non-winning UT set $\mathcal{U}\backslash\hat{\mathcal{U}}$. A UT U_n is does not win any ECN if it is not a UT candidate via Algorithm 2, or its bid is not the highest for any ECN via Algorithm 3. Given U_n is not a winning UT, its payment and utility are both zero.

If U_n becomes a UT candidate via Algorithm 2, there are two conditions for U_n to win an ECN E_m via Algorithm 3. First, UT U_n's bid B_m^n should be the highest among all UTs in set $\mathcal{B}_m$. In the meanwhile, U_n's estimated utility obtained from E_m (i.e., $\bar{\omega}_m^n = B_m^n - P_m^n$) should be the highest among all ECNs. Given U_n wins E_m, its utility is $\omega_m^n = V_m^n - P_m^n$, and payment P_m^n equals either the highest bid that is no more than B_m^n in $\bar{\mathcal{B}}_m$ (line 19, Algorithm 3), or the threshold price $B_\imath$ if there is no other smaller bid in $\bar{\mathcal{B}}_m$ (line 17, Algorithm 3). Note that payment P_m^n may change if U_n untruthfully bids.

Recall that U_n's truthful bid $B_m^n = V_m^n$ is the z-th highest bid in set $\bar{\mathcal{B}}_m$, i.e., $V_m^n = \bar{B}_m^{(z)}$. We denote $\bar{B}_m^{(z-1)}$ as the smallest bid that is no less than V_m^n in $\bar{\mathcal{B}}_m$, and $\bar{B}_m^{(z+1)}$ as the highest bid that is no more than V_m^n in $\bar{\mathcal{B}}_m$.[2] We can deduce that U_n's payment P_m^n increases if $B_m^n > \bar{B}_m^{(z-1)}$, and decreases if $B_m^n < \bar{B}_m^{(z+1)}$, respectively, compared with that of truthful bidding.

We will discuss whether or not the UTs can improve their utilities by untruthful bidding in the following two cases.

Case 1: $U_n \in \mathcal{U}\backslash\hat{\mathcal{U}}$. In this case, we consider UT U_n does not become the winning UT given it truthfully submits its bids. We can deduce that its utility is zero.

(1) If UT U_n underbids for ECN E_m (i.e., $B_m^n < V_m^n$), it still cannot win this ECN and obtains zero utility.

(2) If UT U_n overbids for E_m (i.e., $B_m^n > V_m^n$), it may become the highest bidder in $\mathcal{B}_m$ and wins E_m. Then U_n's utility may have two possible outcomes. First, if $B_m^n \leq \bar{B}_m^{(z-1)}$, its payment does not change and is less than the true valuation, i.e., $P_m^n \leq V_m^n$. In this case, its utility $\omega_m^n = V_m^n - P_m^n \geq 0$, which is higher than that of truthful bidding. Second, if $B_m^n > \bar{B}_m^{(z-1)}$, it pays at least $\bar{B}_m^{(z-1)}$ which exceeds its true valuation, i.e., $P_m^n \geq \bar{B}_m^{(z-1)} > V_m^n$. Its utility is thus non-positive, i.e., $w_m^n = V_m^n - P_m^n \leq 0$, which is even worse than that of truthful bidding.

Case 2: $U_n \in \hat{\mathcal{U}}$. In this case, we consider UT U_n wins an ECN (e.g., E_m) given it truthfully submits its bids. We can deduce that U_n obtains a non-negative utility from E_m by truthful bidding, i.e., $w_m^n \geq 0$. Now we discuss whether or not U_n can improve its utility by untruthfully bids for E_m or any other ECNs.

(1) If UT U_n underbids for E_m (i.e., $B_m^n < V_m^n$), it may not be able to become the winning UT. We have the following two subcases.

 1) If B_m^n is no longer the highest bid in $\mathcal{B}_m$, UT U_n cannot win ECN E_m. In this case, there are two possible outcomes. First, if U_n has not won any other ECNs, its utility is zero. Second, if it wins another EC-N, its utility may increase or decrease from truthful bidding.

 2) If B_m^n is still the highest bid in $\mathcal{B}_m$, we are not sure if it wins E_m since its estimated utility obtained from E_m has changed and we need to compare it with that obtained from other ECNs.

  i) If U_n's estimated utility obtained from E_m is greater than that obtained from any other ECNs, it still wins E_m. Its utility has two possible outcomes. First, if $B_m^n \geq \bar{B}_m^{(z+1)}$, the payment of U_n does not change and its utility remains the same. Second, if $B_\imath \leq B_m^n < \bar{B}_m^{(z+1)}$, U_n's payment P_m^n is smaller than $\bar{B}_m^{(z+1)}$, and its utility is improved than truthful bidding.

  ii) If U_n wins another ECN that provides a larger estimated utility than E_m, its utility may increase or decrease from truthful bidding.

(2) If UT U_n overbids for E_m (i.e., $B_m^n > V_m^n$), its bid B_m^n is still the highest in $\mathcal{B}_m$. However, since U_n's estimated utility obtained from E_m has changed, it may not still win E_m.

 1) If $B_m^n \leq \bar{B}_m^{(z-1)}$, U_n's estimated utility at E_m increases as B_m^n increases. We have the following two subcases.

  i) If U_n truthfully bids for all other ECNs, it still wins E_m and its payment and utility do not

2. If $V_m^n < B_\imath$, we let $\bar{B}_m^{(z-1)} = B_\imath$ and $\bar{B}_m^{(z+1)} = 0$. If $V_m^n \geq B_\imath$ and there does not exist any bid no less than V_m^n in $\bar{\mathcal{B}}_m$, we let $\bar{B}_m^{(z-1)} = \infty$. If $V_m^n \geq B_\imath$ and there does not exist any bid no more than V_m^n in $\bar{\mathcal{B}}_m$, we let $\bar{B}_m^{(z+1)} = B_\imath$.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2021.3140080, IEEE Transactions on Mobile Computing

changed compared with those of truthful bidding.

ii) If U_n also untruthfully bids for some other ECNs, U_n will win the ECN that provides the highest estimated utility. First, if U_n wins E_m, its utility does not change. Second, if U_n wins another ECN, and its utility may increase or decrease from truthful bidding (similar to Case 2-(1)-1)).

2) Once $B_m^n > \bar{B}_m^{(z-1)}$, UT U_n's estimated utility returns to zero before it further increases. In this case, U_n wins the ECN that provides the highest estimated utility. If U_n wins E_m, it pays at least $\bar{B}_m^{(z-1)}$ which is greater than its valuation. In this case, U_n obtains negative utility, i.e., $w_m^n \leq V_m^n - \bar{B}_m^{(z-1)} < 0$. Otherwise, U_n wins another ECN and its utility may increase or decrease from truthful bidding (similar to Case 2-(1)-1)).

□

Remark 1. *From the above two cases, we see that U_n needs to precisely know the other UTs' bids in order to increase its own utility. For example, in Case 1, B_m^n should become the highest bid in $\mathcal{B}_m$ while without exceeding $\bar{B}_m^h$. However, since no UT knows the bids of other UTs in such a seal-bid double auction, the UT's utility may be decreased in practice if the UT takes the untruthful bidding strategy that it expects to increase its utility. On the one hand, placing a blind overbid may help it win an auction, but may also cause it paying more than its valuation when its bid exceeds other UTs' bids, which results in a non-positive utility. On the other hand, placing a blind underbid may reduce its payment, but may also be at risk of losing the auction. Moreover, even if the UT knows the other UTs' bids in the original $\bar{\mathcal{B}}_m$, it still does not know which case it falls into due to the lack of information about the randomly ordered list of the potential winning UTs (line 12 of Algorithm 3).*

Furthermore, our simulation results in Section 5.2 illustrate that, for the proposed SWIDA mechanism, no UT can obtain a positive improvement in its expected utility by bidding untruthfully.

4 PROPOSED POS CONSENSUS MECHANISM

Using the smart contract designs, transactions are automatically generated in the blockchain-aided EC market. In this section, we propose a modified PoS consensus mechanism to determine an unique block generator and ensure a fair allocation of the block generation reward among stakeholders.

The traditional PoW/PoS mechanisms suffer from the limitation of wealth inequality, i.e., the entity that possesses either dominated computing power in PoW or massive coinage in PoS earns the right to verify the transaction and is rewarded induced by this verification. As a result, the conventional PoW/PoS mechanisms incentivize the entities in this market towards either computing power or coinage, but ignore the service quality. To cope with this issue, we redefine the stake of each entity as a weighted sum of coinage and trustworthiness accumulated in the transactions, and propose a trustworthiness-driven PoS mechanism for the blockchain-aided EC market. The entity that obtains the highest ballot proportion is given the right to generate a block.

4.1 Conventional Consensus Mechanisms

In the blockchain-aided EC market, all payments are made in blockchain digital coins. New digital coins are supplied to the blockchain as a reward for publishing every new block. To get the reward, the entities compete to publish the new block. Currently, the consensus mechanism used in the Bitcoin blockchain is called PoW. As [20] unveiled, the PoW mechanism is particularly resource-consuming during the block generation. However, PoW is not suitable for the blockchain-aided EC network, since the entities have limited computing capability [34]. Differing from PoW, the PoS mechanism selects the entity to publish a new block according to its stake (e.g., the coinage in [23]).

Definition 1. *The coinage of entity i is defined as*

$$C_i = \mathrm{num}_i \cdot t_i, \tag{14}$$

where num_i is the the amount of entity i's coins, and t_i is the period that the coins are possessed by entity i.

In each slot, an entity is elected to publish a single block. The PoS based election scheme can be implemented by a standard Follow-the-Satoshi algorithm [23], where the entity j is elected to publish the block in each slot with probability of

$$\varphi_j = \frac{C_j}{\sum_{i=1}^{J} C_i}, j \in \{1, 2, ..., J\}, \tag{15}$$

where J is the number of entities that participate in the block generation. For example, in the Delegated Proof of Stake (DPoS) mechanism, all entities each with coinage can vote for multiple trusted entities to publish the blocks in order [24].

Compared with PoW that requires to solve extremely complicated hash puzzles, the voting process for PoS greatly reduces the energy and computing power consumptions on the battery constrained devices [19]. Moreover, the communication overhead for the proposed PoS is similar to that of PoW. Different from PoW, the voting information is the only additional information exchanged among stakeholders, which does not introduce much communication overhead due to its small size.

However, the conventional PoS mechanism in [23] [24] is not well-suited to the proposed blockchain market due to two critical reasons [19]. First, the right of publishing blocks in PoS is determined by the coinage. In the context of the blockchain-enabled market, it might cause the evil of rich entities by forking and double spending. Second, from (15), it is more likely for the entity with larger coinage to win the right of publishing a new block. Thus, PoS is beneficial for the wealthy entities, and may enlarge the wealth inequality among the entities. Driven by these two issues, we propose a trustworthiness-driven PoS consensus mechanism in the following section.

4.2 PoS Based Consensus Mechanism

Consider that a single block is published in each time slot and multiple consecutive time slots form an epoch. At the beginning of each epoch, a genesis block records the entities that hold stakes and intend to publish the blocks as the stakeholders. Since the operation of our proposed PoS mechanism does not require much energy and computing power consumptions on each entity, any UT, DSO or ECN can participate in the blockchain as a stakeholder relying on its stake (i.e., the weighted sum of coinage or trustworthiness defined in Eqn. (16)). Note that only the entity whose normalized trustworthiness is greater than the threshold can register as a stakeholder and the stake of each entity in the current epoch is accumulated over all the previous epochs. Any change of stake within the current epoch does not affect the stakeholder election of publishing a block in this epoch. The proposed mechanism within every single epoch operates in three steps:

Step 1 (Ballot allocation): At the beginning of each epoch, different stakeholders take different ballot proportions based on their stakes (i.e., the weighted sum of coinage and trustworthiness as Eqn. (16)).

Step 2 (Voting of stakeholders): Every stakeholder votes for its trusted stakeholder, and the stakeholder that receives the highest ballots publishes the block. In this context, a tie may occur wherein multiple stakeholders receive the same highest ballots. Without loss of generality, let $\bar{\mathcal{S}} = \{\mathrm{PK}_1, \mathrm{PK}_2, \cdots, \mathrm{PK}_S\}$ be a set that collects the public keys of the stakeholders involved in the tie in the k-th block height. Inspired by [35] and [36], we use cryptographic hash function to promote distributed randomness for the selection of block publisher in the tie. Since the number of ballots and associated public keys are transparent, each stakeholder in $\bar{\mathcal{S}}$ first calculates the hash values, i.e., $H(\langle \mathrm{PK}_1, H(L^{k-1})\rangle)$, $H(\langle \mathrm{PK}_2, H(L^{k-1})\rangle)$,$\cdots$, $H(\langle \mathrm{PK}_S, H(L^{k-1})\rangle)$, where $H(L^{k-1})$ denotes the hash value of the previous block and $\langle\cdot\rangle$ is a concatenation operation. Here, $H(L^{k-1})$ serves as a random and unpredictable selection seed. Then, the stakeholder associated with the minimum hash value becomes the block publisher. Since the number of ballots and associated public keys are transparent and verifiable, all the stakeholders can verify the legitimacy of the block publisher.

Step 3 (Reward allocation): The stakeholder that votes for the block publisher is rewarded according to its Shapley value.

The details of steps 1, 2, and 3 are given in Section 4.2.1, Section 4.2.2, and Section 4.2.3, respectively.

4.2.1 Vote Allocation

In the conventional PoS mechanism, the stake only relies on the coinage. However, the blockchain-aided EC market emphasizes not only the coinage of each entity but also its trustworthiness that reflects the service quality. In this context, we define the stake of stakerholder j in the proposed PoS mechanism as a weighted sum of trustworthiness in (1) and coinage in (14):

$$X_j = (1 - u_\xi)C_j + u_\xi \xi_j, \tag{16}$$

where $u_\xi \in [0, 1]$ denotes the weight of trustworthiness. Note that the effect of different u_ξ on the wealth inequality will be illustrated in Section 5. The ballot proportion taken by stakeholder j is defined as

$$\Omega_j = \frac{X_j}{\sum_{q=1}^{J} X_q}. \tag{17}$$

From (17), the entity with a larger stake occupies a higher ballot proportion.

4.2.2 Voting of Stakeholders

In this step, each stakeholder uses up its votes for a single stakeholder it trusts. The stakeholder that receives the highest ballots publishes the new block, and all stakeholders that vote for the block publisher are rewarded due to the contributions of their own stakes. In this paper, we formulate the voting process as a coalitional game.

Definition 2. *Let $\mathcal{J} = \{1, ..., J\}$ be a set that collects all the stakeholders that vote for the block publisher, and the stakeholder joins in the $\mathcal{J}$ randomly. A coalition $\mathcal{K}$ is defined as a subset of $\mathcal{J}$. The coalitional game is given by pair $(\mathcal{J}, v(\mathcal{K}))$, where function $v(\mathcal{K})$ measures the total reward produced by coalition $\mathcal{K}$.*

We define a coalition $\mathcal{K}$ as the key coalition if the total ballot proportion of the stakeholders in $\mathcal{K}$ is the most. In other words, the stakeholder supported by the key coalition can get the right to publish the block. We define

$$v(\mathcal{K}) = \begin{cases} \varrho, & \text{if } \mathcal{K} \text{ is the key coalition,} \\ 0, & \text{otherwise,} \end{cases} \tag{18}$$

where $\mathcal{K} \subseteq \mathcal{J}$. From (18), only the key coalition $\mathcal{K}$ earns the reward ϱ for all stakeholders in $\mathcal{J}$, and each stakeholder in $\mathcal{J}$ is rewarded according to its voting proportion.

4.2.3 Reward Allocation

Unlike the conventional PoS mechanism in which only the stakeholder that publishes the block is rewarded, we employ the Shapley-based reward allocation strategy to reward all stakeholders in $\mathcal{J}$ that vote for the block publisher. According to the coalitional game, the reward of each stakeholder corresponds to its Shapley value [37]. The following theorem verifies the fairness of the Shapley based reward allocation.

Theorem 4. *The Shapley-based reward allocation strategy that rewards each stakeholder j according to its Shapley vlaue*

$$\phi(\mathcal{J}, j) = \frac{1}{J!} \sum_{\mathcal{K} \subseteq \mathcal{J} \backslash \{j\}} |\mathcal{K}|!(J - |\mathcal{K}| - 1)![v(\mathcal{K} \cup \{j\}) - v(\mathcal{K})] \tag{19}$$

is fair, where $\mathcal{K}$ can be any coalition in $\mathcal{J}$ and $|\mathcal{K}|$ denotes the number of stakeholders in coalition $\mathcal{K}$.

Proof: Consider that all stakeholders join in the coalitional game with the same probability p. Suppose that p is a random variable uniformly distributed over $[0, 1]$. The probability of forming a coalition $\mathcal{K} \subseteq \mathcal{J} \backslash \{j\}$ is given by

$$P(\mathcal{K} \subseteq \mathcal{J} \backslash \{j\}) = p^{|\mathcal{K}|}(1 - p)^{J - |\mathcal{K}| - 1}, \tag{20}$$

where $\mathcal{J} \backslash \{j\}$ denotes set $\mathcal{J}$ that excludes j and $j \notin \mathcal{K}$.

When stakeholder j joins coalition $\mathcal{K}$ by using up its votes for the same stakeholder voted by coalition $\mathcal{K}$, the rewards earned by $\mathcal{K}$ increases from $v(\mathcal{K})$ to $v(\mathcal{K} \cup \{j\})$. Note that the increased rewards belong to stakeholder j. Define the marginal contribution of stakeholder j to coalition $\mathcal{K}$ as

$$\phi^{'}(\mathcal{J}, j) = v(\mathcal{K} \cup \{j\}) - v(\mathcal{K}). \tag{21}$$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2021.3140080, IEEE Transactions on Mobile Computing

As such, the average marginal contribution of j to all possible coalitions $\mathcal{K}$ with respect to p is defined as the Shapley value of j, given by

$$\begin{aligned}\phi(\mathcal{J},j) &= \sum_{\mathcal{K}\subseteq\mathcal{J}\setminus\{j\}} \int_0^1 \phi_p(\mathcal{J},j)\mathrm{d}p \\ &= \sum_{\mathcal{K}\subseteq\mathcal{J}\setminus\{j\}} \int_0^1 p^{|\mathcal{K}|}(1-p)^{J-|\mathcal{K}|-1}[\phi^{'}(\mathcal{J},j)]\mathrm{d}p \\ &= \frac{1}{J!}\sum_{\mathcal{K}\subseteq\mathcal{J}\setminus\{j\}} |\mathcal{K}|!(J-|\mathcal{K}|-1)![v(\mathcal{K}\cup\{j\})-v(\mathcal{K})]. \end{aligned} \quad (22)$$

From (22), the allocation strategy based on the Shapley value is fair, since the reward allocated to the stakeholder is the stakeholder's average marginal contribution to all possible coalitions [38]. This completes the proof. □

In addition, as Section 5.2 will show, the proposed allocation strategy can reduce the wealth inequality compared with conventional PoS.

5 Simulation Results

In this section, we present the simulation results of smart contract design in Section 5.1 and the proposed PoS mechanism in Section 5.2, respectively.

5.1 Simulations of Smart Contract

In Fig. 5, we plot the network social welfare that is defined as the summation of the utilities of all the DSOs, ECNs and UTs obtained from both the matching-based ECN assocaition phase (in Section 3.1) and the auction-based ECN leasing phase (in Section 3.2). The system parameters are given in Table 2. With reference to [25], DSO D_g's weight of trustworthiness α_g and UT U_n's weight of trustworthiness χ_n are both randomly selected over [0,1]. With reference to [39], we consider UT U_n's computing frequency f_n and E_m's computing frequency f_m are both randomly selected over [0,4] GHz. We set the DSO number as $G = 10$ and the maximum number of rented ECNs by the DSOs as 40. In Fig. 5(a), we fix the number of ECNs as 60 and observe that the network social welfare first increases with the number of UTs, which is due to the increased number of successful ECN-UT association pairs. For large UT numbers (e.g., higher than 140), the network social welfare decreases with the further increase of the number of UTs. This is because the fiercer competition among UTs contributes to higher payment, which finally reduces the social welfare. In Fig. 5(b), by fixing the number of UT as 40, we see that the network social welfare first increases and then decreases with the number of ECNs. When the number of ECNs is small, the social welfare increases with the number of ECNs due to the increased number of successful trading pairs. As the number of ECNs further increases, the network social welfare decreases. This is mainly due to the fact that the fiercer competition among the ECNs improves the service standard (e.g., only the ECNs with higher computing frequency survive), which implicitly increases the UTs' payment and finally reduces the social welfare. Furthermore, for both Fig. 5(a) and Fig. 5(b), the proposed SWIDA achieves higher social welfare than the traditional ICAM since it is able to establish more successful trading pairs among the entities. In ICAM, when a UT's bid is the highest for multiple ECNs, only the ECN that provides the highest utility for this UT is rented and other ECNs are discarded from the candidate list until the double auction algorithm terminates. Therefore, the traditional ICAM suffers from relative low association efficiency. To cope with this problem, the proposed SWIDA does not immediately discard these ECNs but allows them to continue searching for the association opportunities along with other UTs in the next iterations, which improves the number of successful trading pairs and thus increases the social welfare. However, Fig. 5(b) shows that the gap between SWIDA and ICAM slightly decreases as the number of ECN increases. This is due to the fact that the benefit gained from more successful trading pair is less significant as the number of successful trading pair stabilizes.

TABLE 2: Simulation parameters for Fig. 5

Description	Notations	Value
Computing frequency of UT U_n	f_n	[0,4]GHz
Computing frequency of ECN E_m	f_m	[0,4]GHz
D_g's weight of trustworthiness	α_g	[0,1]
U_n's weight of trustworthiness	χ_n	[0,1]
Effective switched capacitance	κ	10^{-26}

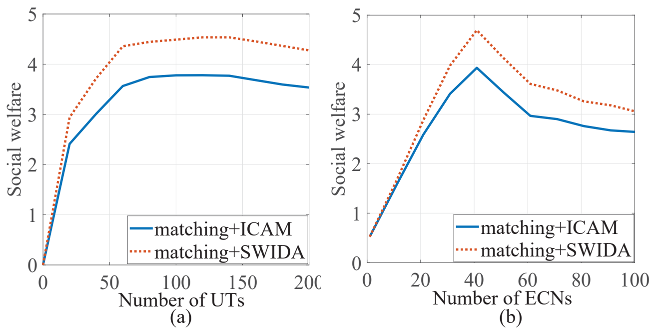


Fig. 5. Social welfare vs. (a) different numbers of UTs with $M = 60$ ECNs and (b) different numbers of ECNs with $N = 40$ UTs.

Fig. 6 validates the weak truthfulness of the UTs under the proposed SWIDA mechanism. We take the expectation of each UT's utility over 1,000 realizations, where each is initialized with a randomly ordered list of the potential winning UTs (see line 12 of Algorithm 3). The medium ask of the ECNs is set to be $A_j = 0.55$. Without loss of generality, we randomly select three typical UTs U_3, U_9 and U_7, where their true valuations on their intended ECNs of E_2, E_6 and E_{11} are $V_2^3 = 0.4$, $V_6^9 = 0.7$ and $V_{11}^7 = 0.9$, respectively. We now discuss the effects of untruthful bidding on the UTs' expected utilities. First, since $V_2^3 = 0.4 < A_j = 0.55$, UT U_3 is not even a UT candidate by using Algorithm 2. If U_3 underbids with $B_2^3 < 0.4$ or overbids with $B_2^3 \in [0, 4, 0.92]$, it still loses the auction and obtains zero utility. If U_3 overbids with $B_2^3 > 0.92$, it becomes the winning UT for ECN E_2. However, the utility for U_3 is negative since it pays more than its valuation for this ECN. Second, we observe

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2021.3140080, IEEE Transactions on Mobile Computing

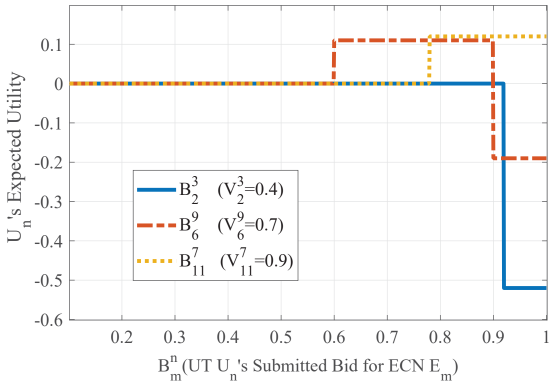


Fig. 6. Truthfulness in expectation of UTs with SWIDA.

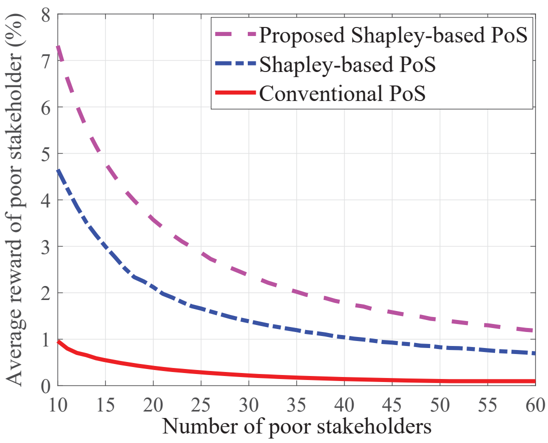


Fig. 7. Average reward allocation of each poor stakeholder under different consensus mechanisms.

that UTs U_9 and U_7 win ECNs E_6 and E_{11} by truthful bidding, respectively. If U_9 underbids with $B_6^9 < 0.6$, it is no longer the highest bidder for E_6 and loses auction with zero utility. If U_9 overbids with $B_6^9 > 0.9$, it pays more than its valuation and obtains negative utility. For $B_6^9 \in [0.6, 0.9]$, U_9 wins the auction without changing its payment and its utility remains the same as that of truthful bidding. Similarly, U_7 cannot improve its expected utility by either overbidding or underbidding. To sum up, no UT can improve its expected utility by bidding untruthfully, which confirms our conclusions on the UTs' weak truthfulness.

5.2 Simulations of Consensus Mechanism

Fig. 7 compares the average reward of each poor stakeholder of our proposed Sharpley-based PoS mechanism with the other two benchmark PoS mechanisms, where the reward of each stakeholder corresponds to the coinage proportion, the stake of coinage, and the stake of weighted sum of coinage and trustworthiness for the conventional PoS, Shapley-based PoS, and our proposed Sharpley-based PoS mechanisms, respectively. We consider that the reward for publishing a new block is fixed. The entity whose trustworthiness value is within [0.7,1] can register as the stakeholder and we set the weight of the trustworthiness

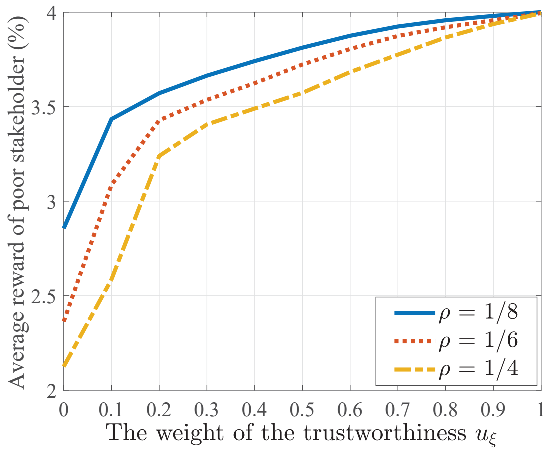


Fig. 8. Average reward allocation of each poor stakeholder of the proposed mechanism under different weights (two rich stakeholders, each with coinage proportion ρ).

$u_\xi = 0.5$. Moreover, we consider two rich stakeholders each with $1/4$ of the total coinage. First, compared with the other two mechanisms, we observe that our proposed Shapley based PoS mechanism can allocate more reward to the poor stakeholders. In this context, under a fixed total reward, the rich stakeholders reap less reward under the proposed mechanism than that under the other benchmark PoS mechanisms. Therefore, the proposed mechanism can reduce the wealth inequality among the entities. Second, we see that the average reward of poor stakeholder decreases as the number of poor stakeholders increases.

Fig. 8 shows the impact of different weights of trustworthiness on the reward allocation of the proposed Shapley based PoS mechanism. Consider that there are 20 poor stakeholders and two rich stakeholders. We discuss the following three cases, where each of the rich stakeholders has $1/4$, $1/6$, and $1/8$ of the total coinage, respectively. We observe that the average reward of poor stakeholders goes up as the weight of u_ξ increases. Therefore, the wealth equality among the entities is reduced under the fixed total reward. First, when the weight of trustworthiness u_ξ is small (e.g., $u_\xi \in [0, 0.2]$), the average reward allocation for poor stakeholder grows faster as u_ξ increases. Since the average reward of the poor stakeholder is positively correlated with its average ballot proportion, the ballot proportion of the rich stakeholders decreases sharply and that of the poor stakeholder increases sharply in this region. Second, when u_ξ is large (e.g., $u_\xi \in [0.2, 1]$), the average reward allocation for the poor stakeholders increases slowly since the trustworthiness value plays a more dominant role than coinage in the ballot proportion. For $u_\xi = 1$, the three curves coincide since the allocation of the stakeholder's ballot proportion only depends on the trustworthiness. The determination of the weight u_ξ depends on the aim of the blockchain system. If the target is to encourage more participants, it is better to use a large u_ξ as an incentive for the poor stakeholders. If the aim is to encourage the participation of the stakeholders with more coinage, it is better to use a small weight u_ξ that ensures the benefits of the rich.

6 Conclusion

In this paper, we proposed several mechanisms in a blockchain-aided EC market aiming to enable automatic, efficient and verified transactions among the decentralized network entities. First, we proposed a smart contract based matching mechanism to establish the one-to-many association between the DSOs and ECNs with the aim of maximizing the social welfare in the ECN association phase. Second, we proposed a double auction mechanism named SWIDA to build up the association between the DSOs and UTs, and determine the pricing of the winners. We proved that the proposed SWIDA mechanism is individually rational and budget balanced. Meanwhile, SWIDA is truthful for the DSOs and can ensure the truthfulness in expectation for the UTs. Most importantly, we showed that SWIDA can significantly improve the social welfare of the network compared with the traditional double auction mechanism. Third, we proposed a trustworthiness-driven PoS mechanism to fairly allocate the reward during the block generation. It is shown that the proposed PoS mechanism can reduce wealth inequality among the entities compared with the conventional consensus mechanisms. In future work, we will extend the smart contract based double auction algorithm design to one-to-many ECN-UT association scenario, where each ECN can potentially serve multiple UTs at the same time.

Acknowledgments

This work was supported in part by National Key Project 2020YFB1807700, the Natural Science Foundation of Jiangsu Province under Grant BK20210331, the National Natural Science Foundation of China under Grant 61872184, and the Competitiveness Enhancement Program of Tomsk Polytechnic University.

References

[1] H. Guo, J. Zhang, J. Liu, and H. Zhang, “Energy-aware computation offloading and transmit power allocation in ultradense IoT networks,” *IEEE Internet Things J.*, vol. 6, no. 3, pp. 4317–4329, Jun. 2019.

[2] S. Azodolmolky, P. Wieder, and R. Yahyapour, “Cloud computing networking: Challenges and opportunities for innovations,” *IEEE Commun. Mag.*, vol. 51, no. 7, pp. 54–62, Jul. 2013.

[3] W. Shi, J. Cao, Q. Zhang, Y. Li, and L. Xu, “Edge computing: Vision and challenges,” *IEEE Internet Things J.*, vol. 3, no. 5, pp. 637–646, Oct. 2016.

[4] Y. Du, J. Li, L. Shi, T. Liu, F. Shu, and Z. Han, “Two-tier matching game in small cell networks for mobile edge computing,” *IEEE Trans. Serv. Comput., early access*, 2019.

[5] H. Zhang, Y. Xiao, S. Bu, D. Niyato, F. R. Yu, and Z. Han, “Computing resource allocation in three-tier IoT fog networks: A joint optimization approach combining Stackelberg game and matching,” *IEEE Internet Things J.*, vol. 4, no. 5, pp. 1204–1215, Oct. 2017.

[6] H. Zhang, Y. Zhang, Y. Gu, D. Niyato, and Z. Han, “A hierarchical game framework for resource management in fog computing,” *IEEE Commun. Mag.*, vol. 55, no. 8, pp. 52–57, Aug. 2017.

[7] J. Li, S. Chu, F. Shu, J. Wu, and D. N. K. Jayakody, “Contract-based small-cell caching for data disseminations in ultra-dense cellular networks,” *IEEE Trans. Wireless Commun.*, vol. 18, no. 5, pp. 1042–1053, May 2019.

[8] Z. Zhou, B. Wang, M. Dong, and K. Ota, “Secure and efficient vehicle-to-grid energy trading in cyber physical systems: Integration of blockchain and edge computing,” *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 50, no. 1, pp. 43–57, Jan. 2020.

[9] Z. Chang, W. Guo, X. Guo, Z. Zhou, and T. Ristaniemi, “Incentive mechanism for edge-computing-based blockchain,” *IEEE Trans. Ind. Informat.*, vol. 16, no. 11, pp. 7105–7114, Nov. 2020.

[10] Y. Fan, L. Wang, W. Wu, and D. Du, “Cloud/Edge computing resource allocation and pricing for mobile blockchain: An iterative greedy and search approach,” *IEEE Trans. Computat. Social Syst.*, vol. 8, no. 2, pp. 451–463, Apr. 2021.

[11] Z. Xiong, Y. Zhang, D. Niyato, P. Wang, and Z. Han, “When mobile blockchain meets edge computing,” *IEEE Commun. Mag.*, vol. 56, no. 8, pp. 33–39, Aug. 2018.

[12] S. Guo, Y. Dai, S. Guo, X. Qiu, and F. Qi, “Blockchain meets edge computing: Stackelberg game and double auction based task offloading for mobile blockchain,” *IEEE Trans. Veh. Technol.*, vol. 69, no. 5, pp. 5549–5561, May 2020.

[13] Y. Jiao, P. Wang, D. Niyato, and Z. Xiong, “Social welfare maximization auction in edge computing resource allocation for mobile blockchain,” in *Proc. IEEE ICC*, Kansas City, MO, USA, May 2018.

[14] Q. Wang, R. Y. K. Lau, and X. Mao, “Blockchain-enabled smart contracts for enhancing distributor-to-consumer transactions,” *IEEE Consum. Electron. Mag.*, vol. 8, no. 6, pp. 22–28, Nov. 2019.

[15] W. Sun, J. Liu, Y. Yue, and P. Wang, “Joint resource allocation and incentive design for blockchain-based mobile edge computing,” *IEEE Trans. Wireless Commun.*, vol. 19, no. 9, pp. 6050–6064, Sept. 2020.

[16] W. Wang, D. Niyato, P. Wang, and A. Leshem, “Decentralized caching for content delivery based on blockchain: A game theoretic perspective,” in *Proc. IEEE ICC*, Kansas City, MO, USA, May 2018.

[17] H. Xu, W. Huang, Y. Zhou, D. Yang, M. Li, and Z. Han, “Edge computing resource allocation for unmanned aerial vehicle assisted mobile network with blockchain applications,” *IEEE Trans. Wireless Commun.*, vol. 20, no. 5, pp. 3107–3121, May 2021.

[18] Z. Yang, K. Liu, Y. Chen, W. Chen, and M. Tang, “Two-level Stackelberg game for IoT computational resource trading mechanism: A smart contract approach,” *IEEE Trans. Serv. Comput., early access*, 2020.

[19] B. Cao, Y. Li, L. Zhang, L. Zhang, S. Mumtaz, Z. Zhou, and M. Peng, “When Internet of things meets blockchain: Challenges in distributed consensus,” *IEEE Netw.*, vol. 33, no. 6, pp. 133–139, Nov. 2019.

[20] D. Tosh, S. Shetty, X. Liang, C. Kamhoua, and L. L. Njilla, “Data provenance in the cloud: A blockchain-based approach,” *IEEE Consum. Electron. Mag.*, vol. 8, no. 4, pp. 38–44, Jul. 2019.

[21] H. Liu, Y. Zhang, and T. Yang, “Blockchain-enabled security in electric vehicles cloud and edge computing,” *IEEE Netw.*, vol. 32, no. 3, pp. 78–83, May. 2018.

[22] J. Huang, L. Kong, G. Chen, M. Wu, X. Liu, and P. Zeng, “Towards secure industrial IoT: Blockchain system with credit-based consensus mechanism,” *IEEE Trans. Ind. Informat.*, vol. 15, no. 6, pp. 3680–3689, Jun. 2019.

[23] A. Kiayias, A. Russell, B. David, and R. Oliynykov, “Ouroboros: A provably secure proof-of-stake blockchain protocol,” Santa Barbara, CA, Aug. 2017, pp. 357–388.

[24] D. Wang and X. Zhang, “Secure ride-sharing services based on a consortium blockchain,” *IEEE Internet Things J.*, vol. 8, no. 4, pp. 2976–2991, Feb. 2021.

[25] K. Chen, H. Shen, K. Sapra, and G. Liu, “A social network based reputation system for cooperative P2P file sharing,” *IEEE Trans. Parallel Distrib. Syst.*, vol. 26, no. 8, pp. 2140–2153, Aug. 2015.

[26] C. Ma, M. Ding, H. Chen, Z. Lin, G. Mao, Y. Liang, and B. Vucetic, “Socially aware caching strategy in device-to-device communication networks,” *IEEE Trans. Veh. Technol.*, vol. 67, no. 5, pp. 4615–4629, May 2018.

[27] W. Zhang, Y. Wen, K. Guan, D. Kilper, H. Luo, and D. O. Wu, “Energy-optimal mobile cloud computing under stochastic wireless channel,” *IEEE Trans. Wireless Commun.*, vol. 12, no. 9, pp. 4569–4581, Sep. 2013.

[28] Z. Wang, T. Alpcan, J. Evans, and S. Dey, “Truthful mechanism design for wireless powered networks,” *IEEE Trans. Commun.*, vol. 67, no. 11, pp. 7966–7979, Nov. 2019.

[29] A. Jin, W. Song, P. Wang, D. Niyato, and P. Ju, “Auction mechanisms toward efficient resource sharing for cloudlets in mobile cloud computing,” *IEEE Trans. Serv. Comput.*, vol. 9, no. 6, pp. 895–909, Nov.-Dec. 2016.

[30] V. Krishna, *Auction Theory, 2nd ed.* New York, NY, USA: Academic, Aug. 2009.

[31] R. P. McAfee, “A dominant strategy double auction,” *J. Econ. Theory*, vol. 56, no. 2, pp. 434–450, Apr. 1992.

[32] V. V. Vazirani, N. Nisan, T. Roughgarden, and E. Tardos, *Algorithmic game theory*. Cambridge, UK: Cambridge Univ. Press, 2007.

[33] A. Jin, W. Song, and W. Zhuang, "Auction-based resource allocation for sharing cloudlets in mobile cloud computing," *IEEE Trans. Emerg. Topics Comput.*, vol. 6, no. 1, pp. 45–57, Jan.-Mar. 2018.

[34] H. Xing, L. Liu, J. Xu, and A. Nallanathan, "Joint task assignment and resource allocation for d2d-enabled mobile-edge computing," *IEEE Trans. Commun.*, vol. 67, no. 6, pp. 4193–4207, Jun. 2019.

[35] Y. Gilad, R. Hemo, S. Micali, G. Vlachos, and N. Zeldovich, "Algorand: Scaling byzantine agreements for cryptocurrencies," in *Proc. ACM SOSP*, ShangHai, China, Oct. 2017.

[36] S. Das, V. Krishnan, I. M. Isaac, and L. Ren, "Spurt: Scalable distributed randomness beacon with transparent setup," Cryptology ePrint Archive, Report 2021/100, 2021, https://ia.cr/2021/100.

[37] Z. Han and H. V. Poor, "Coalition games with cooperative transmission: a cure for the curse of boundary nodes in selfish packet-forwarding wireless networks," *IEEE Trans. Commun.*, vol. 57, no. 1, pp. 203–213, Jan. 2009.

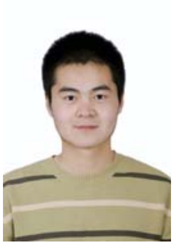
[38] S. Sharma and A. R. Abhyankar, "Loss allocation for weakly meshed distribution system using analytical formulation of shapley value," *IEEE Trans. Power Syst.*, vol. 32, no. 2, pp. 1369–1377, Mar. 2017.

[39] M. Sun, X. Xu, Y. Huang, Q. Wu, X. Tao, and P. Zhang, "Resource management for computation offloading in D2D-aided wireless powered mobile-edge computing networks," *IEEE Internet Things J.*, vol. 8, no. 10, pp. 8005–8020, May 2021.

Yu Du received his B.S. degree from Nanjing University of Science and Technology, P. R. China in 2015. From September 2015 to now, he is a Ph.D. student at the School of Electronic and Optical Engineering, Nanjing University of Science and Technology, Nanjing, China. His research interests include blockchain, mobile edge computing, and game theory.

Zhe Wang (M'14) received the Ph.D. degree in electrical engineering from The University of New South Wales, Sydney, Australia, in 2014. From 2014 to 2020, she was a research fellow with The University of Melbourne, Australia, and Singapore University of Technology and Design, Singapore, respectively. She is currently a professor with the School of Computer Science and Engineering at Nanjing University of Science and Technology, Nanjing, China. Her research interests include applications of optimization, game theory, and machine learning to resource allocation in communications and networking.

Jun Li (M'09-SM'16) received Ph. D degree in Electronic Engineering from Shanghai Jiao Tong University, Shanghai, P. R. China in 2009. From January 2009 to June 2009, he worked in the Department of Research and Innovation, Alcatel Lucent Shanghai Bell as a Research Scientist. From June 2009 to April 2012, he was a Postdoctoral Fellow at the School of Electrical Engineering and Telecommunications, the University of New South Wales, Australia. From April 2012 to June 2015, he was a Research Fellow at the School of Electrical Engineering, the University of Sydney, Australia. From June 2015 to now, he is a Professor at the School of Electronic and Optical Engineering, Nanjing University of Science and Technology, Nanjing, China. He was a visiting professor at Princeton University from 2018 to 2019. His research interests include network information theory, game theory, distributed intelligence, multiple agent reinforcement learning, and their applications in ultra-dense wireless networks, mobile edge computing, network privacy and security, and industrial Internet of things. He has co-authored more than 200 papers in IEEE journals and conferences, and holds 1 US patents and more than 10 Chinese patents in these areas. He was serving as an editor of IEEE Communication Letters and TPC member for several flagship IEEE conferences. He received Exemplary Reviewer of IEEE Transactions on Communications in 2018, and best paper award from IEEE International Conference on 5G for Future Wireless Networks in 2017.



Long Shi (S'10-M'15) received the Ph.D. degree in Electrical Engineering from University of New South Wales, Sydney, Australia, in 2012. From 2013 to 2016, he was a Postdoctoral Fellow at the Institute of Network Coding, Chinese University of Hong Kong, China. From 2014 to 2017, he was a Lecturer at Nanjing University of Aeronautics and Astronautics, Nanjing, China. From 2017 to 2020, he was a Research Fellow at Singapore University of Technology and Design. Currently, he is a Professor at School of Electronic and Optical Engineering, Nanjing University of Science and Technology, Nanjing, China. His research interests include blockchain networks, mobile edge computing, and wireless network coding.

Dushantha Nalin K. Jayakody (S'09-M'14-SM'18) received the Ph. D. degree in Electronics, Electrical, and Communications Engineering, from the University College Dublin, Ireland in 2013. From 2014 - 2016, he was a Postdoc Research Fellow at the Institute of Computer science, University of Tartu, Estonia and Department of Informatics, University of Bergen, Norway. From 2016, he is a Professor at the School of Computer Science & Robotics, National Research Tomsk Polytechnic University (TPU), Russia. In addition, since 2019, he also serves as the Dean/ School of Postgraduate & Research, Sri Lanka Technological Campus (SLTC), Padukka Sri Lanka and Founding director of Centre of Telecommunication Research, SLTC, Sri Lanka. He has received the best paper award from the IEEE International Conference on Communication, Management and Information Technology (ICCMIT) in 2017 and International Conference on Emerging Technologies of Information and Communications, Bhutan, March 2019. In July 2019, He received the Education Leadership Award from the World Academic Congress in 2019. In 2017 and 2018, he received the outstanding faculty award by National Research Tomsk Polytechnic University, Russia. He also received Distinguished Researcher in Wireless Communications in Chennai, India 2019. He has published over 140 international peer reviewed journal and conference papers and books. His research interests include PHY and NET layer prospective of 5G communications technologies such as NOMA for 5G etc, Cooperative wireless communications, device to device communications, LDPC codes, Unmanned Ariel Vehicle etc. He is a Senior Member of IEEE and he has organized or co-organized more than 20 workshops and special sessions of various IEEE conferences. He also served as chair, session chair or technical program committee member for various international conferences, such as IEEE PIMRC 2013-2019, IEEE WCNC 2014-2018, IEEE VTC 2015-2018 etc. He currently serves as an Area Editor the Elsevier Physical Communications Journal, MDPI Information journal and Wiley Internet of Technology Letters. Also, he serves as a reviewer for various IEEE Transactions and other journals.

Quan Chen is a tenure-track associate professor in the Department of Computer Science and Engineering, Shanghai Jiao Tong University, China. His research interests include High performance computing, Task Scheduling in various architectures, Resource management in Datacenter, Runtime System and Operating System. He got his Ph.D. degree at June 2014 from the Department of Computer Science and Engineering, Shanghai Jiao Tong University, China.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2021.3140080, IEEE Transactions on Mobile Computing

Wen Chen (M'03-SM'11) is a tenured Professor with the Department of Electronic Engineering, Shanghai Jiao Tong University, China, where he is the director of Broadband Access Network Laboratory. He is a fellow of Chinese Institute of Electronics and the distinguished lecturers of IEEE Communications Society and IEEE Vehicular Technology Society. He is the Shanghai Chapter Chair of IEEE Vehicular Technology Society, an Editors of IEEE Transactions on Wireless Communications, IEEE Transactions on Communications, IEEE Access and IEEE Open Journal of Vehicular Technology. His research interests include multiple access, wireless AI and meta-surface communications. He has published more than 110 papers in IEEE journals and more than 120 papers in IEEE Conferences, with citations more than 6000 in google scholar.

Zhu Han (S'01-M'04-SM'09-F'14) received the B.S. degree in electronic engineering from Tsinghua University, in 1997, and the M.S. and Ph.D. degrees in electrical and computer engineering from the University of Maryland, College Park, in 1999 and 2003, respectively.

From 2000 to 2002, he was an R&D Engineer of JDSU, Germantown, Maryland. From 2003 to 2006, he was a Research Associate at the University of Maryland. From 2006 to 2008, he was an assistant professor at Boise State University, Idaho. Currently, he is a John and Rebecca Moores Professor in the Electrical and Computer Engineering Department as well as in the Computer Science Department at the University of Houston, Texas. His research interests include wireless resource allocation and management, wireless communications and networking, game theory, big data analysis, security, and smart grid. Dr. Han received an NSF Career Award in 2010, the Fred W. Ellersick Prize of the IEEE Communication Society in 2011, the EURASIP Best Paper Award for the Journal on Advances in Signal Processing in 2015, IEEE Leonard G. Abraham Prize in the field of Communications Systems (best paper award in IEEE JSAC) in 2016, and several best paper awards in IEEE conferences. Dr. Han was an IEEE Communications Society Distinguished Lecturer from 2015-2018, AAAS fellow since 2019 and ACM distinguished Member since 2019. Dr. Han is 1% highly cited researcher since 2017 according to Web of Science. Dr. Han is also the winner of 2021 IEEE Kiyo Tomiyasu Award, for outstanding early to mid-career contributions to technologies holding the promise of innovative applications, with the following citation: "for contributions to game theory and distributed management of autonomous communication networks."