User-Centric Blockchain for Industry 5.0 Applications

Hulin Yang[1], Alia Asheralieva[1], Jin Zhang[1], Md Monjurul Karim[1], Dusit (Tao) Niyato[2], and Khuhawar Arif Raza[1]
[1]*Department of Computer Science and Engineering*, *Southern University of Science and Technology*, Shenzhen, China,
e-mail:{12132369@mail.sustech.edu.cn, aasheralieva@gmail.com, zhangj4@sustech.edu.cn,
karim@mail.sustech.edu.cn, khuhawar@mail.sustech.edu.cn}
[2]*School of Computer Science and Engineering*, *Nanyang Technological University*, Singapore, e-mail:{dniyato@ntu.edu.sg}

2022 IEEE International Conference on Communications Workshops (ICC Workshops) | 978-1-6654-2671-8/22/$31.00 ©2022 IEEE | DOI: 10.1109/ICCWORKSHOPS53468.2022.9814562

***Abstract*—The forthcoming Industry 5.0 era reinforces the role of the “human(user)-centric” approach for future technologies, services and applications, where the key objective is to satisfy the quality of service (QoS), security and privacy requirements of each individual user. To meet this objective, blockchain is considered as one of the prime enablers, as it allows secure, reliable, verifiable, and transparent management of private user data. However, contemporary blockchains are not suitable for Industry 5.0 applications due to their inability to ensure high throughput while maintaining reasonable security levels. Hence, in this paper, we design a user-centric sharded blockchain that enables generating, verifying, and storing blocks of data related to individual users with the aim to satisfy their QoS, security and privacy requirements. By analyzing the impact of user allocations to shards on the block period, we devise the user-shard allocation algorithm to minimize the block period or, equivalently, maximize the system throughput, and demonstrate the superior performance of our framework via simulations.**

***Index Terms*—blockchain, game theory, Industry 5.0, optimization, resource allocation, sharding.**

I. Introduction

A recent European Commission report [1] emphasizes the role of the “human(user)-centric” technologies, services, and applications in the upcoming Industry 5.0 era, where the key objective is to satisfy the QoS, security and privacy requirements of each individual user. To reach this objective, a distributed ledger – blockchain, is considered as one of the prime enablers [2]. The data in the blockchain are organized as a linked list of blocks (e.g., records of digital transactions). Each generated block is verified and stored by the group of blockchain nodes (or peers) using a decentralized consensus mechanism, and protected by the strong cryptographic techniques to ensure that the data contained in blocks are managed securely, privately, and reliably [3], [4].

Unfortunately, many current blockchains cannot meet the stringent QoS requirements of future Industry 5.0 applications due to their inability to ensure high throughput while maintaining reasonable security levels. For example, highly-secure blockchains (e.g., based on proof-of-work [PoW], proof-of-stake [PoS], etc.) have very long block period of 5 minutes and, consequently, low throughput. On the other hand, lightweight blockchains (e.g., based on practical Byzantine fault tolerance [pBFT], delegated PoS [DPoS], proof-of-authority [PoA], etc.) compromise on security – less than 33% of malicious peers are tolerated [2], [4]–[6]. To increase the blockchain throughput, the concept of sharding [5], [7]–[16] has been proposed. In the sharded blockchain, peers are divided into groups called shards, each of which is responsible for processing specific transactions (or parts of transactions), that allows increasing the system throughput in proportion to the total number of shards. To improve the system security, most existing sharded blockchains, such as OmniLedger [4] and RapidChain [8], randomly update the shard structure to avoid collusions among members of the same shard. Nonetheless, the resulting security is still rather low – less than 33% malicious peers are tolerated in each shard. Furthermore, in such blockchains, the data associated with a single user can be processed, verified and stored in different shards and, hence, accessing this data can be problematic due to increased delay/signaling costs.

Accordingly, in this paper, we design a user-centric sharded blockchain with the reduced block period (and, hence, increased throughput) to satisfy the users’ QoS requirements. In particular, unlike the sharding systems in [9] and [17], in our blockchain, all data associated with a single user are processed and verified in the same shard which reduces the delay/signaling costs for accessing these data. Moreover, to reduce the block period while preserving the system security, we propose a reputation-based sharding model that adopts a self-organized shard/coalition formation algorithm to reach the reputation-based stable shard structure minimizing the block verification delays of the peers.

The main contributions of the paper are as follows:

- We propose a user-centric blockchain for Industry 5.0 applications based on the self-organized reputation-based sharding model where the data associated with a single user is processed, verified, and stored in the same shard to reduce the delay/signaling costs for accessing this data.
- We present an analytical model of the reputation-based shard formation process and analyze the relationships between the user-shard allocations and the block period.
- We represent the interactions of users and peers in shards as a Stackelberg game. In the game, the users are allocated to shards by the blockchain system (as a leader) to minimize the block period. The followers (blockchain peers) respond to each user-shard allocation by forming a reputation-based stable shard structure.
- We propose the algorithm to find the Nash equilibrium (NE) that minimizes the block period or, equivalently,

maximizes the system throughput given a reputation-based stable shard structure of the Stackelberg game.

II. User-Centric Sharded Blockchain

A. System Model

The proposed user-centric blockchain (shown in Fig. 1) is formed by the set $\mathbf{N} = \{1, \ldots, n, \ldots, N\}$ of peers, i.e., blockchain nodes, labeled as $\mathrm{P}_1, \ldots, \mathrm{P}_n, \ldots, \mathrm{P}_N$ that provide blockchain services to the set $\mathbf{M} = \{1, \ldots, m, \ldots, M\}$ users labeled as $u_1, \ldots, u_m, \ldots, u_M$. Similar to [3], the peers can be represented by edge devices, e.g., personal computers, gateways, servers and smart phones, located close to the users. The proposed blockchain system basically represents a lightweight software interface (similar to a browser) downloaded by users when joining the system. After joining, each peer is assigned with a unique ID to access the system. If the peer refuses to follow the system rules, e.g., fails to verify its transaction in time, it is removed from the system. In the blockchain, data generated by each individual user are packaged into transactions which are then processed and verified by the peers in the form of blocks. As such, the blockchain services provided by the peers comprise processing, verification, and storage of user transactions contained in the blocks.

The process of mining the blockchain is divided into a number of stages denoted as $t = 0, \ldots, T$. The duration of each stage is equal to the block period Δt, i.e., time to process, verify, and append the user transactions contained in a block. During one stage, all network parameters (e.g., the number of peers and users, number of user transactions, and computing powers of peers) are considered to be fixed, but can change at the next stage. At every stage t, the peers in the sharded blockchain are distributed into K shards denoted as $\mathbf{S}_1, \ldots, \mathbf{S}_K$ and each shard $\mathbf{S}_k \subseteq \mathbf{N}, k = 1, \ldots, K$, is associated with some groups of users whose transactions are included in the blocks generated and verified in the shard. In particular, we let $\mathbf{y} = \{y_{mk} \in \{0, 1\}\}$, for $m \in \mathbf{M}$ and $k = 1, \ldots, K$, denote the user-shard allocation plan proposed by the blockchain system, such that $y_{mk} = 1$ if the transactions of user u_m are included in the blocks generated and verified in the shard $\mathbf{S}_k$ and $y_{mk} = 0$ otherwise, such that

$$\sum_{k=1}^{K} y_{mk} = 1, \forall m \in \mathbf{M}, \tag{1}$$

to ensure that each user is assigned to one shard. Accordingly, at any stage t, a certain shard structure $\mathbf{O} = \{\mathbf{S}_1, \ldots, \mathbf{S}_K\}$, such that $\mathbf{S}_k \cap \mathbf{S}_j = \emptyset$, for all $j \neq k$, is formed by peers, so that each shard $\mathbf{S}_k$ includes exactly $S_k = |\mathbf{S}_k|$ peers. Any block generated inside the shard $\mathbf{S}_k$ is appended to the blockchain only if all user transactions contained in the block are processed and verified by all peers in the shard $\mathbf{S}_k$. The block verification process is based on the voting when each peer P_n in the shard $\mathbf{S}_k$ must submit its vote $v_n \in \{0, 1\}$, i.e., $v_n = 1$ if the peer votes to accept the block, or $v_n = 0$ if the peer votes to reject the block. For every transaction correctly appended within the block to the blockchain, the users pay a certain transaction fee $\mathbf{r}$. The transaction fee $\mathbf{r}$ is distributed across all peers that have participated in the respective block generation and verification process based on their reputations that are estimated as explained in the next subsection.

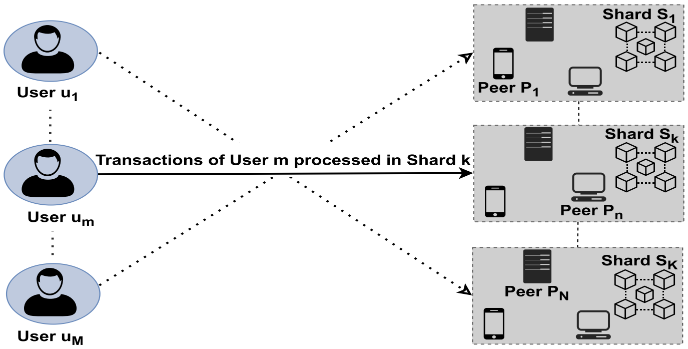


Figure 1: Proposed system model.

B. Reputation of Peers

Note that any peer in the set $\mathbf{N}$ can be malicious, in which case it will incorrectly process user transactions and generate false block verification results' vote to accept the erroneous blocks and reject the correct blocks. Unfortunately, in the practical blockchain systems, no peers can know exactly whether the other peers are malicious or trustworthy [4], [8], [9], [17]. Nevertheless, if peers are allowed to form and record their opinions about other peers in the blockchain system, we can deploy a certain reputation mechanism [5], [6], [17] letting each peer P_n to assess the trustworthiness of any other peer P_i by combining its own opinion about this peer with the opinions of other peers about the peer P_i. For example, similar to [5], [17], to express the opinion of peer P_i about another peer P_j in subjective logic, we can use the opinion vector $\chi_{i\to j} = \{b_{i\to j}, d_{i\to j}, u_{i\to j}\}$, where $b_{i\to j}, d_{i\to j}$ and $u_{i\to j}$ stand for the belief, distrust, and uncertainty, respectively. $b_{i\to j} + d_{i\to j} + u_{i\to j} = 1, b_{i\to j}, d_{i\to j}, u_{i\to j} \in [0, 1]$, and

$$\begin{cases} b_{i\to j} = (1 - u_{i\to j}) \frac{T^{+}_{i\to j}}{T^{+}_{i\to j} + T^{-}_{i\to j}}, \\ d_{i\to j} = (1 - u_{i\to j}) \frac{T^{-}_{i\to j}}{T^{+}_{i\to j} + T^{-}_{i\to j}}, \\ \quad u_{i\to j} = 1 - q_{i\to j}. \end{cases} \tag{2}$$

In (2), $T^{+}_{i\to j}$and $T^{-}_{i\to j}$represent, respectively, the numbers of positive and negative opinions about peer P_j submitted by peer P_i that can be determined by peer P_i, as in [5], from the history of interactions between the peers P_i and P_j. $q_{i\to j}$ is the quality of communication link between peers P_i and P_j, i.e., the probability that the opinion is received correctly.

As such, the reputation ρ_n (or trustworthiness), of any peer P_n in the blockchain network can be found based on the opinions of other peers about peer P_n, as

$$\rho_n = \sum_{i \in \mathbf{N} \setminus \{n\}} \rho_{i\to n} = \sum_{i \in \mathbf{N} | \{n\}} (b_{i\to n} + \lambda u_{i\to n}), \forall n \in \mathbf{N}, \tag{3}$$

where $\rho_{i\to n} = b_{i\to n} + \lambda u_{i\to n}$ expresses the amount of trust to peer P_n by another peer P_i; λ is a given constant indicating the effect of the uncertainty on the peer's reputation. Given the peers' reputations in (3), the reputation $\rho_{\mathbf{S}_k}$ of every shard $\mathbf{S}_k$ can be expressed by the average reputation of peers belonging o the shard, as

$$\rho_{\mathbf{S}_k} = \frac{1}{|\mathbf{S}_k|} \sum_{n \in \mathbf{S}_k} \rho_n, \forall k = 1, \ldots, K. \tag{4}$$

Thus, the higher are the reputations, i.e., trustworthiness, of the peers verifying the blocks generated in the shard, the higher is the shard's reputation. Block generated by the shard can only be appended to the sub-chain maintained by the shard if it is verified by all peers in the shard before becoming an orphan. In other words, the peers must vote on the validity of each block within the shard to which they belong. Then, to enhance security of the block verification process in the shard $\mathbf{S}_k$, the voting power or weight ω_n of each peer P_n in the shard $\mathbf{S}_k$, can be set proportionally to its reputation ρ_n, i.e., as

$$\omega_n = \frac{\rho_n}{\sum_{i \in \mathbf{S}_k} \rho_i}, \forall n \in \mathbf{S}_k, \tag{5}$$

so that the more trustworthy peers will have a greater voting power in their shards. As such, each block generated by the members of the shard $\mathbf{S}_k$ is verified only if $\sum_{n \in \mathbf{S}_k} \omega_n v_n >$ 0.5, i.e., the weighted majority of the peers in the subset $\mathbf{S}_k$ belonging to and assigned to the shard $\mathbf{S}_k$ votes to accept the block generated in the shard $\mathbf{S}_k$.

Lastly, note that by deploying the considered reputation mechanism, every peer P_n can assess the trustworthiness of any other peer P_i by combining its own opinion about this peer with the opinions of other peers about peer P_i, as

$$B_{n \to i} = \gamma \rho_{n \to i} + (1 - \gamma) \sum_{j \in \mathbf{N} \setminus \{i, n\}} \rho_{j \to i}, \tag{6}$$

where $B_{n \to i} \in [0, 1]$ is the assessment of the trustworthiness of peer P_i by peer P_n that expresses the probability that the peer P_i is trustworthy from the point of view of the peer P_n so that $1 - B_{n \to i}$ expresses the probability that the peer P_i is malicious from the point of view of the peer $\mathrm{P}_n; \gamma \in [0, 1]$ is the metric that indicates how much the peer P_n values its own opinion compared to the opinions of other peers.

C. Payoffs of Peers

We now estimate the expected payoffs of the peers from the provided blockchain services. First, recall that if the block including transactions of users associated with the shard $\mathbf{S}_k$ is correctly appended to the blockchain, all peers in the shard $\mathbf{S}_k$ are rewarded proportionally to their reputations. As such, the expected reward R_n of peer P_n at stage t is given by

$$\begin{aligned} R_n(\mathbf{y}, \mathbf{O} \mid \mathbf{B}_n, \boldsymbol{\rho}) &= \sum_{k=1}^{K} \mathbf{1}_{n \in \mathbf{S}_k} \Pr\left\{ \sum_{i \in \mathbf{S}_k} \omega_i v_i > 0.5 \right\} \frac{\rho_n \mathbf{r} l_{\mathbf{S}_k}}{\sum_{i \in \mathbf{S}_k} \rho_i} \\ &= \sum_{k=1}^{K} \mathbf{1}_{n \in \mathbf{S}_k} P^{\text{success}}_{n \to \mathbf{S}_k}(\mathbf{B}_n) \frac{\rho_n \mathbf{r} \sum_{m \in \mathbf{M}} y_{mk} l_m}{\sum_{i \in \mathbf{S}_k} \rho_i}, \forall n \in \mathbf{N}. \end{aligned} \tag{7}$$

In (7), $\mathbf{B}_n = \{B_{n \to i}\}_{i \in \mathbf{N} \setminus \{n\}}$ are the assessments by the peer P_n of the trustworthiness of all other peers; $\boldsymbol{\rho} = \{\rho_n\}_{n \in \mathbf{N}}$ are the peers' reputations; $\mathbf{1}_x = 1$ if x is true and $\mathbf{1}_x = 0$ otherwise; $l_m \in [l_{\min}, l_{\max}]$ is the number of transactions of user u_m that must be processed and verified in a block, $l_{\min}$ and $l_{\max}$ are minimal and maximal possible numbers of transactions, respectively, so that the total number of user transactions appended in a block of shard $\mathbf{S}_k$ is a sum $l_{\mathbf{S}_k} = \sum_{m \in \mathbf{M}} y_{mk} l_m$; $\Pr\left\{\sum_{i \in \mathbf{S}_k} \omega_i v_i > 0.5\right\} = \mathrm{P}^{\text{success}}_{n \to \mathbf{S}_k}$ is the probability that every correct block in the shard $\mathbf{S}_k$ is successfully appended to the blockchain.

Note that although no peer knows the probability $\mathrm{P}^{\text{success}}_{n \to S_k}$ exactly, any trustworthy peer P_n can estimate this probability based on its assessments of the trustworthiness of all other peers in the shard $\mathbf{S}_k$ as

$$P^{\text{success}}_{n \to \mathbf{S}_k}(\mathbf{B}_n) = \sum_{i \in \mathbf{S}_k \setminus \{n\}} \omega_i B_{n \to i} + \omega_n. \tag{8}$$

That is, if the peer P_n is trustworthy, it votes $v_n = 1$ i.e., accept the correct block, similar to all other trustworthy peers in the shard $\mathbf{S}_k$. Otherwise, if the peer P_n is malicious, it votes $v_n = 0$ to reject the correct block, similar to all other malicious peers in the shard $\mathbf{S}_k$.

Given its expected reward R_n, the peer P_n can estimate its expected payoff V_n (also called the value) from the blockchain services represented by the difference

$$V_n(\mathbf{y}, \mathbf{O} \mid \mathbf{B}_n, \boldsymbol{\rho}) = R_n(\mathbf{y}, \mathbf{O} \mid \mathbf{B}_n, \boldsymbol{\rho}) - C_n(\mathbf{y}, \mathbf{O}), \forall n \in \mathbf{N}, \tag{9}$$

where C_n is the total expected cost incurred by the peer P_n on generating and verifying the block, given by

$$C_n(\mathbf{y}, \mathbf{O}) = z_n \sum_{k=1}^{K} \mathbf{1}_{n \in \mathbf{S}_k} l_{\mathbf{S}_k} = z_n \sum_{k=1}^{K} \sum_{m \in \mathbf{M}} \mathbf{1}_{n \in \mathbf{S}_k} y_{mk} l_m, \tag{10}$$

with z_n denoting the average cost incurred by the peer P_n on processing and verifying a single user transaction.

III. Shard Formation Process and Block Period

A. Self-Organized Reputation-Based Shard Formation

We consider that peers to form shards in a self-organizing way, i.e., they choose shards independently. We model the self-organized shard formation process as a reputation-based coalition formation game, where a subset of peers that manage the same shard is a coalition. Unlike the classical coalition formation game without reputations, in the game, peers need to consider not only their payoffs, but also the reputation of the shards they belong to, i.e., the coalitional reputation, given by

$$\tilde{\rho}_n(\mathbf{O}) = \sum_{k=1}^{K} \mathbf{1}_{n \in \mathbf{S}_k} \rho_{\mathbf{S}_k}. \tag{11}$$

As such, in any shard structure $\mathbf{O}$, each peer P_n selects the shard that maximizes its payoff V_n and coalitional reputation $\tilde{\rho}_n$ without affecting the payoffs and coalitional reputations of the other peers in this shard. The reputation-based coalition formation game eventually reaches a reputation-based stable shard structure, defined as follows:

Definition 1 (Reputation-based stable shard structure)**.** *A shard structure $\mathbf{O}^*$ represents a reputation-based stable shard structure of the reputation-based coalition formation game played by the peers if and only if there is no other structure $\mathbf{O}$, such that $\exists \mathbf{S}_k \in \mathbf{O}, \exists n \in \mathbf{S}_k, \forall i \in \mathbf{S}_k \backslash \{n\}$, we have:*

$$\begin{aligned} V_n(\mathbf{y}, \mathbf{O} \mid B_n, \boldsymbol{\omega}) &> V_n\left(\mathbf{y}, \mathbf{O}^* \mid B_n, \boldsymbol{\omega}\right) \\ \tilde{\rho}_n(\mathbf{O}) &= \rho_{\mathbf{S}_k} \geq \tilde{\rho}_n\left(\mathbf{O}^*\right), \\ V_i(\mathbf{y}, \mathbf{O} \mid B_i, \boldsymbol{\omega}) &\geq V_i\left(\mathbf{y}, \mathbf{O}^* \mid B_i, \boldsymbol{\omega}\right) \text{ and} \\ \tilde{\rho}_i(\mathbf{O}) &= \rho_{\mathbf{S}_k} \geq \tilde{\rho}_i\left(\mathbf{O}^*\right). \end{aligned} \tag{12a}$$

or

$$\begin{aligned} V_n(\mathbf{y}, \mathbf{O} \mid B_n, \boldsymbol{\omega}) &\geq V_n\left(\mathbf{y}, \mathbf{O}^* \mid B_n, \boldsymbol{\omega}\right) \\ \tilde{\rho}_n(\mathbf{O}) &= \rho_{\mathbf{S}_k} > \tilde{\rho}_n\left(\mathbf{O}^*\right) \\ V_i(\mathbf{y}, \mathbf{O} \mid B_i, \boldsymbol{\omega}) &\geq V_i\left(\mathbf{y}, \mathbf{O}^* \mid B_i, \boldsymbol{\omega}\right) \text{ and} \\ \tilde{\rho}_i(\mathbf{O}) &= \rho_{\mathbf{S}_k} \geq \tilde{\rho}_i\left(\mathbf{O}^*\right). \end{aligned} \tag{12b}$$

That is, a shard structure is a reputation-based stable shard structure if and only if there is no other shard structure which can increase the payoff or coalitional reputation of at least one peer without reducing the payoffs and coalitional reputations of the other peers in the same shard. More details about such a shard formation can be found in [5].

B. Block Period

Recall that at each stage t, each shard $\mathbf{S}_k$ needs to process and verify exactly $l_{\mathbf{S}_k}$ transactions of users. As in [17], we consider that the size of each user transaction is defined by the tuple $\sigma = \left(\sigma_t^{\mathrm{P}}, \sigma_t^{V}\right)$, where σ_t^{P} and σ_t^{V} are the numbers of CPU cycles required to process and verify the transaction, respectively. Similar to other sharding systems, e.g., [5], [6], [8], [9], each transaction must be processed entirely by a single peer, i.e., the processing of a transaction is indivisible. As such, the time required to process a transaction is equal to $\sigma_t^{\mathrm{P}} / \min_{n \in \mathbf{S}_k} x_n$, where $x_n \in [x_{\min}, x_{\max}]$ is the computing power of peer P_n in CPU cycles per time unit, $x_{\min}$ and $x_{\max}$ are the minimal and maximal possible computing powers, respectively. Then, the time $t^{\mathrm{P}}_{\mathbf{S}_k}$ to generate a block in the shard $\mathbf{S}_k$ that is equal to the time to process all transactions appended in a block can be estimated according to

$$t^{\mathrm{P}}_{\mathbf{S}_k}(\mathbf{y}) = \frac{\sigma_t^{\mathrm{P}} l_{\mathbf{S}_k}}{\min_{n \in \mathbf{S}_k} x_n} = \frac{\sigma_t^{\mathrm{P}}}{\min_{n \in \mathbf{S}_k} x_n} \sum_{m \in \mathbf{M}} y_{mk} l_m. \tag{13}$$

After the block is generated, it must be verified by all peers in the shard $\mathbf{S}_k$. Thus, the time $t^{\mathrm{P}}_{\mathbf{S}_k}$ required to verify a block generated in the shard $\mathbf{S}_k$ is given by

$$t^{V}_{\mathbf{S}_k}(\mathbf{y}) = \frac{\sigma_t^{V} l_{\mathbf{S}_k}}{\min_{n \in \mathbf{S}_k} x_n} = \frac{\sigma_t^{V}}{\min_{n \in \mathbf{S}_k} x_n} \sum_{m \in \mathbf{M}} y_{mk} l_m, \tag{14}$$

and, hence, the total time required to generate and verify a block including all transactions of users associated with the shard $\mathbf{S}_k$ is given by

$$t^{In}_{\mathbf{S}_k}(\mathbf{y}) = t^{\mathrm{P}}_{\mathbf{s}_k}(\mathbf{y}) + t^{V}_{\mathbf{s}_k}(\mathbf{y}) = \frac{\sigma_t^{\mathrm{P}} + \sigma_t^{V}}{\min_{n \in \mathbf{S}_k} x_n} \sum_{m \in \mathbf{M}} y_{mk} l_m. \tag{15}$$

Algorithm 1: User-Shard Allocation Algorithm

```
Input: $\mathbf{N}, \mathbf{U}$
Output: Reputation-based stable shard structure $\mathbf{O}^*$ and
        optimal user-shard allocation plan $\mathbf{y}^*$
Randomly initialize $\mathbf{y}(0)$ and set number of iterations $t = 0$
repeat
Blockchain system broadcasts user-shard allocation plan $\mathbf{y}(\mathrm{t})$
Peers form a reputation-based stable shard structure $\mathbf{O}(\mathrm{t})$
  through Reputation-Based Coalition Formation Algorithm
$\mathbf{y}(t+1) \leftarrow \underset{\mathbf{y} \in \mathbf{Y}}{\operatorname{argmin}} \Delta t(\mathbf{y}, \mathbf{O}(t))$
$t = t + 1$
until No more players update strategies
```

From (15), the block period Δt - time required to generate and verify all blocks in the sharding blockchain, is given by

$$\Delta t(\mathbf{y}, \mathbf{O}) = \max_{k=1,\ldots,K} t^{In}_{\mathbf{S}_k}(\mathbf{y}). \tag{16}$$

From (16), the block period Δt (or, equivalently, the system throughput that is inversely proportional to the block period) depends on both the shard structure $\mathbf{O} = \{\mathbf{S}_1, \ldots, \mathbf{S}_K\}$ formed by the peers according to the self-organized shard formation process outlined in Section III-A and the user-shard allocation plan $\mathbf{y}$ proposed by the blockchain system.

IV. Stackelberg Game between the Blockchain System and Peers

A. Game Model and Existence of the Nash Equilibrium

In this section, we model interactions between the blockchain system and peers as a two-stage Stackelberg game. In the game, blockchain system as a leader decides on the user-shard allocation plan $\mathbf{y}$ that minimizes the block period Δt. In response, peers as followers form the shard structure $\mathbf{O}$ to improve their payoffs and coalitional reputations. We show that if the game is played repeatedly by all game participants, i.e., the leader and followers, at each block period t, it will eventually reach the NE state $(\mathbf{y}^*, \mathbf{O}^*)$, where $\mathbf{y}^*$ denotes the optimal user-shard allocation plan which minimizes the block period Δt; $\mathbf{O}^*$ is the reputation-based stable shard structure in which no participant can be better off without negatively affecting other participants.

The proposed Stackelberg game is played repeatedly at each block period t in two stage. In stage I, the leader, i.e., blockchain system, determines the user-shard allocation plan $\mathbf{y}^*$ that minimizes the current block period Δt according to

$$\mathbf{y}^* = \underset{\mathbf{y} \in \mathbf{Y}}{\operatorname{argmin}} \Delta t(\mathbf{y}, \mathbf{O}) = \underset{\mathbf{y} \in \mathbf{Y}}{\operatorname{argmin}} \left(\max_{k=1,\ldots,K} t^{In}_{\mathbf{S}_k}(\mathbf{y}) \right), \tag{17a}$$

where from (1), the set $\mathbf{Y}$ of possible values of $\mathbf{y}$ is defined by

$$\mathbf{Y} = \left\{ \mathbf{y} = \{y_{mk}\} \mid \begin{array}{c} y_{mk} \in \{0,1\}, k \in \mathbf{K}, m \in \mathbf{M} \\ \sum_{k=1}^{K} y_{mk} = 1 \end{array} \right\}. \tag{17b}$$

In stage II, the followers, i.e., peers, play the reputation-based coalition formation game described in Section III-A to form

Algorithm 2: Reputation-Based Coalition Formation

```
Input: $\mathbf{N}, \mathbf{y}$
Output: Reputation-based stable shard structure $\mathbf{O}^*$
    Initialize: $\mathbf{O}, \mathbf{N}_v \leftarrow \emptyset, \Delta \leftarrow 0$
while true do
    for $i = 1 : |\mathbf{N}|$ do
        Randomly choose one of non-visited players $\mathrm{P}_n$
          from the subset $\mathbf{N}\backslash\mathbf{N}_v$
        for $\mathbf{S}_j \in \mathbf{O}\backslash\{\mathbf{S}_k\}$ do
            if $\mathbf{O}_n^{k\to j}$ satisfies condition (12a) or (12b) then
                $\mathbf{O} \leftarrow O_n^{k\to j}$
                $\Delta \leftarrow \Delta + 1$
                break
        $\mathbf{N}_v \leftarrow \mathbf{N}_v \cup \{n\}$
    if $\Delta = 0$ then
        break
    $\mathbf{N}_v \leftarrow \emptyset$, $\Delta \leftarrow 0$
```

a reputation-based stable shard structure $\mathbf{O}^*$. Accordingly, the main aim of the game is to reach the NE state defined as follows:

Definition 2 (Nash equilibrium of the Stackelberg game)**.** *The tuple* $(\mathbf{y}^*, \mathbf{O}^*)$ *is the Nash equilibrium of the Stackelberg game played by the blockchain system and peers if and only if we have:*

$$\Delta t\left(\mathbf{y}^*, \mathbf{O}^*\right) \leq \Delta t\left(\mathbf{y}, \mathbf{O}^*\right), \forall \mathbf{y} \in \mathbf{Y}, \tag{18}$$

where $\mathbf{O}^*$ is the reputation-based stable shard structure.

Proposition 1 (Nash equilibrium existence in the Stackelberg game)**.** *The Stackelberg game played by the blockchain system and peers admits at least one* $\mathrm{NE}\left(\mathbf{y}^*, \mathbf{O}^*\right)$.

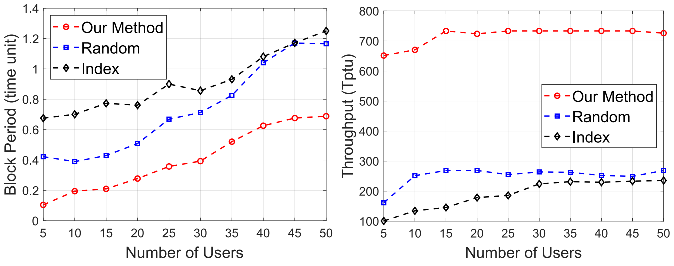
Proposition 1 establishes the existence of the NE in the game. The proof of Proposition 1 is provided in Appendix.

B. Finding Nash Equilibrium of the Game

We now explain the proposed method to find the NE of our Stackelberg game. In this method, to find the NE at stage I of the game played by the leader, i.e., blockchain system, we utilize Algorithm 1. In the algorithm, given the current shard structure $\mathbf{O}$, the blockchain system decides on the user-shard allocation plan $\mathbf{y}$ by solving (17a). At stage II, the followers, i.e., peers, form a reputation-based stable shard structure $\mathbf{O}^*$ by deploying a distributed iterative Algorithm 2 where at each iteration, we randomly select a peer, e.g., peer P_n, belonging to some shard, e.g., shard $\mathbf{S}_k$. If the peer P_n can find another shard $\mathbf{S}_j \in \mathbf{O}\backslash\{\mathbf{S}_k\}$ where the peer's coalitional reputation and/or payoff can be increased without reducing coalitional reputations and payoffs of other peers in the shard $\mathbf{S}_j$, peer P_n joins another shard $\mathbf{S}_j$, so that a new shard structure

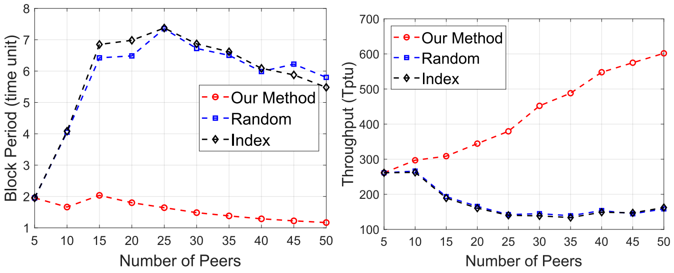
$$\mathbf{O}_n^{k\to j} = \left\{ \hat{\mathbf{S}}_1,\dots,\hat{\mathbf{S}}_K \mid \begin{array}{c} \hat{\mathbf{S}}_k = \mathbf{S}_k\backslash\{n\}, \hat{\mathbf{S}}_j = \mathbf{S}_j \cup \{n\}, \\ \hat{\mathbf{S}}_i = \mathbf{S}_i, \forall i \in \mathbf{K}\backslash\{k,j\} \end{array} \right\}, \tag{19}$$

is formed. Then, it is straightforward to verify that if the game is played repeatedly according to Algorithms 1 and 2, it will eventually converge to the NE $(\mathbf{y}^*, \mathbf{O}^*)$.



(a) (b)

Figure 2: Block period (a) and throughput (b) depending on the number of users M for N = 50.



(a) (b)

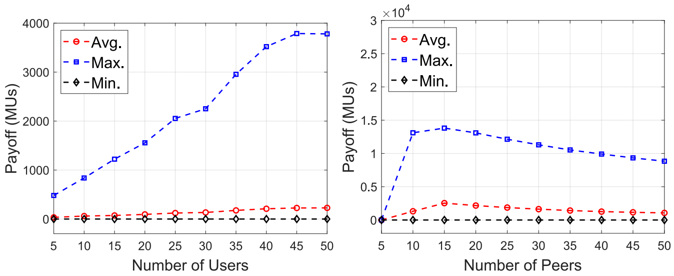
Figure 3: Block period (a) and throughput (b) depending on the number of peers N for M = 50.

V. Performance Evaluation

We begin by presenting technical specifications of the simulated environment and then demonstrate the detailed evaluation in terms of block period, throughput and payoffs of the peers. The performance of our proposed system is compared with existing shard-based blockchain systems, such as OmniLedger [4], which is based on random updates of the shard structure to avoid collusions among peers. We conduct simulations on MATLAB by considering two scenarios: i) Random, and ii) Index. The first scenario updates shard structure with random user-shard allocations while the second updates shard structure with fixed user-allocations. In simulations, the computing powers of peers are distributed uniformly in the interval [1,2000] CPU cycles per time unit. The number of transactions follow the Poisson distribution with the mean 20 transactions per user. To simplify the simulation, the quality of the communication link $q_{i\to j}$ between any two peers P_i and P_j is 0.99 and λ is 0.3. For a given reputation-based shard structure $\mathbf{O}^*$, we use the **fmincon** function in MATLAB to find the corresponding optimal user-shard allocation plan $\mathbf{y}^*$.

A. Result Discussions

Figures 2a and 2b show, respectively, the block period (in time unit) and blockchain throughput (in transactions per time unit or Tptu) depending on the number of users M for a fixed number of peers N = 50 peers. Figures 3a and 3b show the block period and throughput, respectively, depending on the number of peers N for a fixed number of users M = 50. From these figures, our method outperforms other schemes, i.e., Random and Index, in most scenarios. The reason is that



(a) (b)

Figure 4: Payoff of a peer depending on (a) the number of users M for N = 50 and (b) the number of peers N for M = 50.

in our method, the user-shard allocation plan is obtained by minimizing the block period according to (17a). As a result, the method yields the lower block period and, hence, higher blockchain throughput than those in Random and Index that do not optimize the user-shard allocation. We also evaluate the payoffs of the peers (in monetary units – MUs) in our method. Figure 4a shows the average, maximal, and minimal payoff of a peer depending on the number user M for the fixed number of peers N = 50. Figure 4b shows the average, maximal, and minimal payoff of a peer depending on the number of peers N for the fixed number of users M = 50. From Figure 4a, the peers' payoffs are growing with M. The reason is that when the number of users M increases while the number of peers N is fixed, each peer has more transactions to be processed and, thus, its payoff is growing. On the other hand, from Figure 4b, the payoff of a peer is decreasing with N. The reason is that when the number of peers N increase while the number of users M (and, hence, the number of transactions) is fixed, the payoff of a peer is reducing.

VI. Conclusion

We have proposed a user-centric sharded blockchain for Industry 5.0 applications. We have introduced a self-organized shard formation mechanism where peers form shards based on their reputations to minimize the block verification delays while preserving the system security. We have formulated a Stackelberg game to represent interactions between the users and peers in shards, and developed the algorithms to find its solution. Simulation results show that out scheme outperforms the existing sharding systems in terms of throughput and block period. In our future work, we plan to implement the proposed blockchain in practical Industry 5.0 system.

Acknowledgement: This work was supported by the Characteristic Innovation Project No. 2021KTSCX110 of Guangdong Provincial Department of Education. The Corresponding Author is Alia Asheralieva.

References

[1] J. Müller, "Enabling technologies for industry 5.0, results of a workshop with europe's technology leaders," *Directorate-General for Research and Innovation*, 2020.

[2] P. K. R. Maddikunta *et al.*, "Industry 5.0: A survey on enabling technologies and potential applications," *Journal of Industrial Information Integration*, p. 100257, Aug. 2021.

[3] L. Luu *et al.*, "A secure sharding protocol for open blockchains," in *Proc. in ACM SIGSAC CCS*, 2016, pp. 17–30.

[4] E. Kokoris-Kogias *et al.*, "Omniledger: A secure, scale-out, decentralized ledger via sharding," in *2018 IEEE Symposium on Security and Privacy*. IEEE, 2018, pp. 583–598.

[5] A. Asheralieva and D. Niyato, "Reputation-Based Coalition Formation for Secure Self-Organized and Scalable Sharding in IoT Blockchains With Mobile-Edge Computing," *IEEE IoT Journal*, vol. 7, no. 12, pp. 11 830–11 850, Dec. 2020.

[6] W. Wang *et al.*, "A Survey on Consensus Mechanisms and Mining Strategy Management in Blockchain Networks," *IEEE Access*, vol. 7, pp. 22 328–22 370, 2019.

[7] M. H. Manshaei *et al.*, "A Game-Theoretic Analysis of Shard-Based Permissionless Blockchains," *IEEE Access*, vol. 6, pp. 78 100–78 112, 2018.

[8] M. Zamani, M. Movahedi, and M. Raykova, "RapidChain: Scaling Blockchain via Full Sharding," in *Proc. of the 2018 ACM SIGSAC Conference on CCS*. ACM, Oct. 2018, pp. 931–948.

[9] J. Kang *et al.*, "Blockchain for secure and efficient data sharing in vehicular edge computing and networks," *IEEE IoT Journal*, vol. 6, no. 3, pp. 4660–4670, 2018.

[10] J. Yu *et al.*, "RepuCoin: Your Reputation Is Your Power," *IEEE Transactions on Computers*, vol. 68, no. 8, pp. 1225–1237, Aug. 2019.

[11] J. Wang and H. Wang, "Monoxide: Scale out Blockchains with Asynchronous Consensus Zones," in *16th USENIX Symposium on NSDI*, 2019, pp. 95–112.

[12] H. Chen and Y. Wang, "SSChain: A full sharding protocol for public blockchain without data migration overhead," *Pervasive and Mobile Computing*, vol. 59, p. 101055, Oct. 2019.

[13] G. Wang, "RepShard: Reputation-based Sharding Scheme Achieves Linearly Scaling Efficiency and Security Simultaneously," in *2020 IEEE International Conference on Blockchain*, Nov. 2020, pp. 237–246.

[14] Z. Hong *et al.*, "Pyramid: A Layered Sharding Blockchain System," in *IEEE INFOCOM 2021*, May 2021, pp. 1–10.

[15] C. Huang *et al.*, "RepChain: A Reputation-Based Secure, Fast, and High Incentive Blockchain System via Sharding," *IEEE IoT Journal*, vol. 8, no. 6, pp. 4291–4304, Mar. 2021.

[16] S. Li *et al.*, "PolyShard: Coded Sharding Achieves Linearly Scaling Efficiency and Security Simultaneously," *IEEE Transactions on Information Forensics and Security*, vol. 16, pp. 249–261, 2021.

[17] A. Asheralieva and D. Niyato, "Distributed dynamic resource management and pricing in the iot systems with blockchain-as-a-service and uav-enabled mobile edge computing," *IEEE IoT Journal*, vol. 7, no. 3, pp. 1974–1993, 2019.

Appendix

First, note that the existence of a reputation-based stable shard structure $\mathbf{O}^*$ has already been proven in [5]. Hence, we only need to proof that our Stackelberg game admits at least one NE $(\mathbf{y}^*, \mathbf{O}^*)$, i.e., there exists at least one user-allocation plan $\mathbf{y}^*$ satisfying (18). Indeed, from (16), we have

$$\Delta t\left(\mathbf{y}, \mathbf{O}^*\right) = \max_{k=1,\ldots,K} t^{In}_{\mathbf{S}_k}(\mathbf{y}) = \max_{k=1,\ldots,K} \left(\frac{\sigma_t^{\mathrm{P}} + \sigma_t^{V}}{\min_{n \in \mathbf{S}_k} x_n} \sum_{m \in \mathbf{M}} y_{mk} l_m \right)$$
$$\in \left[\frac{\left(\sigma_t^{\mathrm{P}} + \sigma_t^{V}\right) l_{\min}}{x_{\max}}, \frac{\sigma_t^{\mathrm{P}} + \sigma_t^{V}}{x_{\min}} \sum_{m \in \mathbf{M}} l_m \right].$$

The above means that the function $\Delta t\left(\mathbf{y}, \mathbf{O}^*\right)$ defined on the set $\mathbf{y} \in \mathbf{Y}$ is bounded from the above and below. Furthermore, note that from (17b), the set $\mathbf{Y}$ of possible values of $\mathbf{y}$ is finite. Hence, there exists at least one user-shard allocation plan $\mathbf{y}^*$ that satisfies

$$\frac{\left(\sigma_t^{\mathrm{P}} + \sigma_t^{V}\right) l_{\min}}{x_{\max}} \leq \Delta t\left(\mathbf{y}^*, \mathbf{O}^*\right) = \min_{\mathbf{y} \in Y} \Delta t\left(\mathbf{y}, \mathbf{O}^*\right) \leq \Delta t\left(\mathbf{y}, \mathbf{O}^*\right)$$
$$\leq \frac{\sigma_t^{\mathrm{P}} + \sigma_t^{V}}{x_{\min}} \sum_{m \in \mathbf{M}} l_m, \forall \mathbf{y} \in \mathbf{Y},$$

where $\Delta t\left(\mathbf{y}^*, \mathbf{O}^*\right) = \min_{\mathbf{y} \in \mathbf{Y}} \Delta t\left(\mathbf{y}, \mathbf{O}^*\right)$ is exactly the NE solution in (18). ■