Reputation-Based Coalition Formation for Secure Self-Organized and Scalable Sharding in IoT Blockchains With Mobile-Edge Computing

Alia Asheralieva and Dusit Niyato, *Fellow, IEEE*

***Abstract*—We propose a fully distributed system architecture and a scalable self-organized sharding scheme for the Internet-of-Things (IoT) blockchains that can guarantee system security without reducing its throughput. In the system, the IoT devices are supported by the set of blockchain peers that gather, process, verify, and store the blocks of IoT transaction records. To support communications among peers, the system is realized in the mobile-edge computing (MEC) network. We design a new consensus mechanism in which each peer votes on the outputs of each block task in its shard. The peer's voting power is computed from its reputation, i.e., trustworthiness in the system. By adopting a reputation-based coalitional game model, we formulate a novel self-organized shard formation algorithm in which each peer acts as a rational player aiming to maximize both its payoff and the coalitional reputation. We prove that the algorithm converges to the reputation-based stable shard structure, i.e., a structure that maximizes the payoff and coalitional reputation of each peer without negatively affecting other peers. The algorithm shows a superior performance in terms of system security and throughput when compared to state-of-the-art sharding schemes and reputation-based blockchains.**

***Index Terms*—Blockchains, edge computing, game theory, Internet of Things (IoT), reputation management, security, self-organization, trust.**

I. Introduction

WITH the widespread deployment of Internet-of-Things (IoT) applications (e.g., smart city, connected vehicles, smart grid, e-health, smart home, etc.) enormous volumes of data which requires massive amounts of computing, communication, and storage resources, will be generated repeatedly by the rapidly growing number of IoT devices. This puts an extraordinary strain on all kinds of networking and data management services causing increased concerns about security, privacy, and performance degradation [1]. Indeed, many current IoT applications, such as e-health or connected vehicles, are time sensitive, compute intensive, and involve private data. Therefore, the IoT data must be collected, processed, and utilized very fast in an anonymous and secure manner. However, when the vast data volumes are produced continuously, it is very hard to efficiently exchange, process, verify, and store these data, or to detect numerous malicious attacks and cyber threats on a large scale [2]. Due to the absence of transparency, single points of failure, and limited scalability and reliability, traditional centralized management approaches cannot meet these challenges [3]. Instead, a distributed ledger technology called blockchain has been recently proposed to address the main limitations of centralized approaches [1], [2]. In blockchains, data are organized in the form of blocks, e.g., records of IoT transactions, to preserve logical relations in the appended blocks. The copies of blocks are distributed across the entire blockchain network—a connected system formed by geodistributed blockchain peers. The process by which the blocks are verified and appended to a blockchain is referred to as a mining [3]. During mining, a blockchain peer performs a compute-intensive task, e.g., block processing or verification, and broadcasts its output to the other peers. The transaction is validated, and a new block is appended to the blockchain only if the output reaches a consensus, which ensures improved data integrity and security comparing to centralized approaches [4]. As such, blockchain can become a powerful platform for data management and networking in IoT systems, since it is capable of fulfilling the main requirements of current IoT applications, including decentralization, anonymity, reliability, transparency, and low operational costs [1]–[4].

Although blockchains have already been deployed in some distributed system scenarios, e.g., connected vehicles, content delivery networks, and smart grids [5]–[7], they are still not widely adopted in the IoT and other mobile applications. In order to support these applications, the blockchain system must scale well with the number of peers [8]. In particular, in the blockchain systems, scalability is measured with respect to three key performance metrics: 1) throughput—the number of

Manuscript received February 3, 2020; revised May 16, 2020; accepted June 10, 2020. Date of publication June 17, 2020; date of current version December 11, 2020. The work of Dusit Niyato was supported in part by the National Natural Science Foundation of China under Project 61950410603; in part by the National Research Foundation (NRF), Singapore, through Singapore Energy Market Authority, Energy Resilience under Grant NRF2017EWT-EP003-041; in part by Singapore NRF under Grant NRF2015-NRF-ISF001-2277; in part by Singapore NRF National Satellite of Excellence, Design Science and Technology for Secure Critical Infrastructure NSoE under Grant DeST-SCI2019-0007; in part by the A*STAR-NTU-SUTD Joint Research Grant on Artificial Intelligence for the Future of Manufacturing under Grant RGANS1906; and in part by the Wallenberg AI, Autonomous Systems and Software Program and Nanyang Technological University under Grant M4082187 (4080). *(Corresponding author: Alia Asheralieva.)*

Alia Asheralieva is with the Department of Computer Science and Engineering, Southern University of Science and Technology, Shenzhen 518055, China (e-mail: aasheralieva@gmail.com).

Dusit Niyato is with the School of Computer Science and Engineering, Nanyang Technological University, Singapore 639798 (e-mail: dniyato@ntu.edu.sg).

This article has supplementary downloadable material available at https://ieeexplore.ieee.org, provided by the authors.

Digital Object Identifier 10.1109/JIOT.2020.3002969

transactions verified per time unit; 2) storage efficiency—the size of blockchain which can be handled by peers with limited storage capacities; and 3) security—the number of malicious peers that the system can tolerate. Scalability means that all metrics can improve or, at least, do not deteriorate as the number of peers increases [8], [9]. Poor scalability is the main drawback of the current blockchain systems, i.e., Ethereum and Bitcoin, that are based on classical Proof-of-Work (PoW) and Proof-of-Stake (PoS) consensus algorithms [9], [10]. The reason is that both PoW and PoS require full replication of the computation and storage—each appended block must be verified and stored by all peers in the system. Full replication results in high system security (up to 49% of adversarial peers is tolerated). Yet, the throughput and storage efficiency are compromised—the number of operations and the number of blocks stored by each peer per time unit increase linearly with the number of peers, which causes throughput and storage efficiency to drop [11]. In addition to the aforementioned problems, PoW is highly compute-intensive and demands large energy consumption of peers; PoS is unfair to users with low stakes leading to "rich get richer" phenomenon [8]. Other popular alternatives to PoS and PoW are the practical Byzantine Fault Tolerance (pBFT) and Delegated PoS (DPoS) that do not require each block to be verified and stored by all peers [11]. As such, the consensus must be reached among the preselected validators represented by known or "trustworthy" peers in pBFT and the peers with large stakes in DPoS. This increases throughput and storage, and reduces energy costs comparing to PoS and PoW. However, this approach limits security, because less than 33% of adversaries can be tolerated. In addition, in DPoS, malicious peers can collude with compromised high-stake peers, whereas pBFT needs a great amount of communications among peers putting restrictions on the size of the system (otherwise, the latency will grow which further reduces throughput). However, smaller sizes can lead to "sybil attacks" performed by forging the number of malicious nodes to exceed 33% [11].

Up to now, one of the leading solutions to address the scalability of the blockchains is via sharding [12]–[16]. The idea is to split an entire "global" blockchain into a number of local subchains or "shards" which are then replicated, processed, and stored in parallel. In this way, the computation and storage requirements can be reduced by a factor is equal to the number of shards. In order to allow the throughput and storage efficiency to scale up with the system size, the number of shards must grow linearly with the number of peers which results in the constant number of peers managing each shard. Consequently, when the system expands, it becomes more susceptible to adversaries. Existing sharding models maintain security with system solutions, e.g., by randomly sampling and regularly updating the subsets of peers associated with the same shards [12]. Such solutions can achieve a near-optimal security in the cloud-based blockchains, but are not suitable for IoT and mobile applications because: 1) peers are assigned to shards by the leader, i.e., central authority node, which leads to a single failure point; 2) any randomly assigned peer can be malicious; and 3) peers assigned to the same shards according to a random sampling can be located very far from each other, which increases the block propagation time and, hence, reduces blockchain throughput [3]. As such, the alternative sharding models are required to support IoT/mobile blockchains. For example, to maximize the system throughput in a decentralized way, peers can be allowed to select shards that maximize their individual throughputs, whereas, to ensure shards' security, peers can be admitted to the shards based on their reputations or "trustworthiness" [5] in the system.

In this article, we propose a decentralized and self-organized implementation of sharding for IoT applications which enables enhancing both the blockchain throughput and the security. In particular, we study the IoT-blockchain with sharding, where IoT devices are served by the set of blockchain peers that collect, process, verify, and store the blocks of IoT transaction records. Similar to [17], [18], to support the communications among peers, the blockchain is realized in the mobile-edge computing (MEC) network formed by the set of base stations (BSs) equipped with MEC servers. Shards are formed in a self-organized way, i.e., each peer independently selects its shard with the aim to maximize its payoff which depends on the peer's throughput. To ensure the security of the local shards' subchains, peers are admitted to shards based on their reputations [5], [19]. The reputation is determined based on the past behavior of the peer from its trustworthiness by the peer's customers, i.e., the users of IoT devices associated with the peer, as well as by the other peers in the same shard. As such, a block is appended to the shard's subchain only if all peers in this shard reach a consensus on the block's validity. On the other hand, to improve the global blockchain security, each block of the local subchain is also validated by at least one randomly assigned peer outside of the shard which minimizes collusions among the malicious peers operating in the same shards.

The main contributions of the article are as follows.

1) We design a novel fully decentralized system architecture and a sharding scheme for the blockchain with MEC (BC-MEC) to facilitate IoT applications. The proposed sharding scheme enhances both the system throughput and security with a two-stage solution: a) self-organized reputation-based shard formation to improve throughput and security of the local shards and b) random assignment of peers to guarantee the global blockchain security.
2) We formulate a new consensus methods for the shards in which the peers take a vote on the output of each block task in their shards. The peer's voting power is calculated based on its reputation. The peer's reputation is defined according to a subjective logic model [5], [19] by considering both the opinions submitted by the peer's customers and the opinions of other peers.
3) We formulate a comprehensive analytical model of the BC-MEC system which takes into account: a) shard structure and the subsets of randomly assigned peers; b) parameters of the peers' tasks; c) locations of the peers and BSs; d) wireless channel parameters and co-channel interference; and e) waiting time in the processor buffers. Based on the model, we derive the closed-form expressions of: a) exact and expected block

delays, i.e., the sum of processing, transmission, and verification delays for a task; b) exact and expected energy consumption of peers, i.e., the sum of the energies consumed on processing, transmitting and verifying a task; c) orphaning probability, i.e., probability that the task is discarded due to long block delay; and d) exact and expected payoffs of the peers in any shard structure.

4) We model a self-organized shard formation by the peers as a reputation-based coalitional game where coalitions represent the subsets of players/peers managing every shard. Unlike conventional nonreputational games in which a coalition is determined only by its value, i.e., expected payoff to the players, in our game, a coalition is determined by both the value and the coalitional reputation defined by the average reputation of all players/peers validating task outputs in the respective shard. Thus, the coalitional reputation measures the trustworthiness of block confirmations in the shard.

5) We propose a novel distributed reputation-based coalition formation algorithm where each peer acts independently as a rational player aiming to select the best shard, i.e., the coalition of peers which maximizes both: a) peer's payoff determined by its throughput and b) coalitional reputation. Accordingly, each consecutive shard structure formed as a result of such a coalition formation is better (or, at least, not worse) than the previous one in terms of the total throughput and coalitional reputations, i.e., trustworthiness of block confirmations. We prove that the proposed algorithm converges to a reputation-based stable shard structure, i.e., structure which maximizes both the payoff and coalitional reputation of every player/peer without negatively affecting all other players/peers. We also show that the algorithm has a linear rate of convergence and polynomial time complexity.

The remainder of the article is organized as follows. In Section II, we review existing research on blockchains for IoT and mobile systems, sharding systems, reputation-based coalitional games, and other prospective solutions. In Section III, we present the system model of BC-MEC with sharding for IoT applications. In Section IV, we develop the analytical model of the BC-MEC system. In Section V, we define a reputation-based coalitional game and formulate the algorithm for self-organized shard formation. In Section VI, we evaluate the performance of the proposed algorithm and draw conclusions in Section VII.

II. Related Work

A. Blockchains for IoT and Mobile Systems

Recently, there has been a surge of interest in integrating blockchain with IoT and mobile systems, as well as the design of suitable consensus protocols [3]–[8]. In particular, the classical PoW and PoS algorithm which ensures optimal security (up to 49% of malicious peers is tolerated) are not suitable for IoT blockchains due to reduced throughput and storage efficiency caused by full replication of computation and storage—each appended block must be verified and cached by all peers. Thus, a variety of alternative consensus methods has been proposed to support IoT/mobile applications. In this regard, a variety of lightweight blockchain models have been proposed (e.g., [5], [7], and [19]–[30]) with the objective to minimize the total amount of required computations and storage. To reach this objective, peers have been separated into two groups: 1) lightweight peers that can issue transactions and store only the block headers in their local caches and 2) validators that can store the full blocks and are capable of confirming transactions. The main question arising in these models is how to select the validators so as to satisfy the computation and storage constraints. For example, in [20], [22], [23], [25], [27], and [28], validators are chosen based on their resources, i.e., any peer with adequate processing and caching capabilities can be a validator. In [7] and [21], validators are peers with the highest stakes, whereas, in [5], [19], [24], [26], [29], and [30], validators are selected from the "trustworthy" peers. The "trustworthiness" is determined based on either: 1) credits and certificates issued by certain external blockchain authorities [24], [26] or 2) peers' reputations and voting [5], [19], [29], [30].

In this way, the lightweight blockchains can reduce the total amount of computations and storage in the system. However, they still have several limitations which are listed as follows.

1) In [20], [22], [23], [25], [27], and [28], i.e., the first group of works, no additional measures are taken to secure protection against adversarial nodes during the block verification. As a result, system security can degrade significantly (comparing to PoW and PoS protocols) when the number of validators is small [3]. The reason is that any validator can be malicious, as the validators are selected based only on their computing and storage resources.
2) In [7] and [21], i.e., the second group of works, the adversaries can collude with high-stake peers to generate false results during block verification, or even to launch double-spending attacks, which are very hard to prevent [3], [5].
3) In [5], [19], [24], [26], [29], and [30], i.e., the third group of works, challenges lie in determining the trustworthiness of the peers. In particular, in [24] and [26], the models fully rely on the credits and certificates of external blockchain authorities that are hard to evaluate and trace (given the absence of other evaluation metrics). In [5], [19], [29], and [30], peers earn the right to become validators, so there are incentives to retain positions they have gained by reducing the number of potential competitors, e.g., by voting out or by providing negative opinions about other validators or candidate peers even if they are "well-behaved." This leads to reduced security (only up to 33% malicious peers can be tolerated) comparing to baseline PoW and PoS methods [3]. Besides, in many reputational models, e.g., based on pBFT [5], [19] or Proof-of-Authority (PoA) [29], [30], the validators run a software allowing them to add transactions in blocks. The process is automated and does not require the validators to constantly monitor their computers. However, it requires maintaining a computer,

an authority node, uncompromised [3], [31]. In addition, PoA only allows the nonconsecutive block approval from the same validators, i.e., the risk of serious damage is centralized to the authority node [3], [31]. As such, there is a single failure point, i.e., authority node, which also threatens security. A detailed description of the security issues in current reputation-based blockchains can be found, e.g., in [3] and [31].

To summarize, in general, the lightweight blockchains can increase the overall throughput and storage efficiency, but with the compromise on security. Unlike these models, in this article, system security is enhanced with a two-stage solution: 1) self-organized reputation-based shard formation aiming to improve throughput and security of local shards and 2) random assignment of peers to secure a global blockchain. More specifically, we have the following.

1) In the first stage, both the system throughput and security are enhanced by allowing each peer to independently select the best shard, i.e., the coalition of peers which maximizes both: a) peer's expected payoff determined by its throughput and b) peer's coalitional reputation that depends on reputations of the other shard members and measures trustworthiness of block confirmations in the shard. Hence, each consecutive shard structure formed as a result of such a self-organized shard formation is better (or, at least, not worse) than the previous one in terms of the throughput and the coalitional reputations, i.e., trustworthiness of block confirmations.
2) In the second stage, each block of a local shard is validated by at least one randomly assigned peer outside of the shard with the aim to minimize collusions among malicious peers operating in the same shards and improve global blockchain security.
3) Unlike existing reputation-based blockchains, e.g., [5], [19], [24], [26], [29], and [30], in our model, peers have no incentives to provide false opinions about other peers, as they do not compete with each other to become validators (every peer is the validator of the blocks in its shard). Instead, each peer is incentivized in providing true opinion scores so as to select the best shard and increase its coalitional reputation (that depends on the reputations of all shard's members). Such a cooperative reputation-based approach allows achieving the coalitional structure that maximizes both: a) expected payoff or, equivalently, throughput of each peer and b) coalitional reputations. Moreover, since the shard formation process is fully decentralized, i.e., does not dependent on the authority node, there is no single point of failure.

B. Sharding Systems

Recent studies of sharding systems to enhance scalability, i.e., ability to simultaneously maintain the desired levels of throughput, storage efficiency, and security when the size of the blockchain network increases, have been conducted in [12]–[16]. In [12], a sharding scheme called OmniLedger is devised based on the bias-resistant distributed randomness generation model to sample and update subsets of peers managing shards aiming to improve security. In [13], a blockchain with static domain-based sharding is designed where peers competitively generate PoW blocks. The top peers are selected as validators verifying the blocks according to the pBFT algorithm. As the composition of validators changes dynamically depending on the results of PoW competition, the system security is maintained. Forestier et al. [14] proposed a blockchain architecture referred to as Blockclique, which shards transactions in a block graph with multiple threads in order to parallelize block creation, and does not rely on the network sharding. As such, Blockclique is more similar to a traditional blockchain: every participant is randomly selected according to a PoS consensus to record and verify blocks of all threads. Manshaei *et al.* [15] analyzed the strategic behaviors of peers to design the appropriate incentives to foster cooperation among peers and prevent free-riding. The problem is modeled as a static noncooperative game where each peer aims to maximize its reward at a minimum cost. It is shown that depending on the reward sharing mechanism, the peer can increase its payoff by defecting unilaterally, which leads to a social dilemma. To address this issue, the authors formulate an incentive-compatible reward sharing scheme that promotes cooperation and improves system performance. Li *et al.* [16] developed a coded sharding model called PolyShard that allows injecting the computation redundancy in the unorthodox codes. Unlike the models in [12]–[15] that are susceptible to dynamic adversaries which corrupt peers after they have been assigned to shards, PolyShard ensures security against erroneous results generated by both fixed and dynamic malicious peers.

To summarize, the works in [12]–[16] presented a variety of sharding solutions to improve scalability. Unfortunately, these solutions are not applicable to IoT applications, since: 1) peers are assigned to shards by the leader, i.e., central authority node, which leads to a single failure point; 2) any randomly assigned peer can be malicious; and 3) peers assigned to the same shards according to a random sampling can be located far from each other, which increases the block propagation time and reduces system throughput [3]. The impact of block propagation time on system throughput has been studied in several prior works (e.g., [32]–[36]). In particular, the block propagation delay can be very high in large-scale mobile/IoT blockchains where geodistributed peers can be connected through multihop wireless links, in which case the throughput can drop significantly [3], [32], [37]–[39]. Thus, unlike existing sharding schemes, in our model, both the system throughput and security are enhanced with a two-stage solution: 1) self-organized shard formation to improve throughput and security inside the formed shards and 2) random assignment of peers for the global blockchain security. Self-organized shard formation allows a peer to independently select the best shard, i.e., a coalition of peers that maximizes both: a) the peer's payoff determined by its throughput and b) peer's coalitional reputation which measures the trustworthiness of block confirmation in the shard. Therefore, each consecutive shard structure formed as a result of shard formation is better (or, at least, not worse) than the previous one in terms of the throughput and coalitional reputations, i.e., trustworthiness of block confirmations. In the second stage, each block of a local shard is validated by at least one randomly assigned peer outside of the shard. This allows minimizing collusions among malicious peers operating in the same shards and improves global blockchain security.

C. Reputation-Based Coalitional Games and Other Solutions

A few reputation-based coalitional games have already been analyzed in the past (e.g., in [40]–[43]). However, prior works have several limitations. For example, with the exception of [42], they analyze the existence of a reputation-based reward sharing mechanism which leads to a formation of a traditional stable structure. As such, these works do not suggest any new solution concepts for the game and do not define a coalition formation strategy for its players. On the other hand, in [42], a search algorithm aiming is developed to find a coalition with the maximal average reputation and payoff. This is achieved by iteratively removing players with minimal reputations from the formed coalitions. Nevertheless, the proposed algorithm is not suitable for a distributed shard formation because of the following reasons: 1) only one coalition is formed as a result of the algorithm; 2) players with minimal reputations are forced out of each formed coalition, even if this reduces their payoffs; and 3) the algorithm is fully centralized. We also note that there are other methods, such as static and dynamic clustering [44]–[47] or learning-based coalition formation [48]–[50], that have a potential to facilitate sharding in IoT and mobile blockchains. However, the main disadvantage of such methods is that they do not provide any means to track the reputations of peers in the system. Hence, it is hard to measure the trustworthiness of block confirmations in the formed shards. In addition, there are other disadvantages, e.g., low convergence rate and high computational and communication complexities [44], [48], [49], which make these methods inappropriate for use in large-scale blockchains.

III. BC-MEC with Sharding for IoT Applications

A. Architecture and System Model

Consider the BC-MEC system for IoT applications realized in the network formed by BSs equipped with MEC servers. In the system, the IoT devices are served by the set $\mathbf{N} = \{1, \ldots, N\}$ of BC-MEC nodes or peers labeled as $P_1, \ldots, P_N$. Peers can be represented by small-cell BSs, mobile terminals (e.g., laptops, smartphones, and tablets) or desktop computers equipped with processors, as shown in Fig. 1. The main functions of peers are to: 1) gather data, e.g., sensing records, from the associated IoT devices which are connected to one or more BC-MEC customers, i.e., users of IoT devices; 2) process IoT data and record processing outputs as unconfirmed transactions; 3) validate unconfirmed transactions; and 4) store confirmed records of transactions to control associated IoT devices. The macrocell BSs are reserved for basic control/forwarding functions and for providing typical communication/offloading services for their cellular customers. Macrocell BSs are labeled as $\text{BS}_1, \ldots, \text{BS}_M$, with $\mathbf{M} = \{1, \ldots, M\}$ denoting the set of BSs' labels. Every macrocell BS is connected to other macrocell BSs via fiber links.

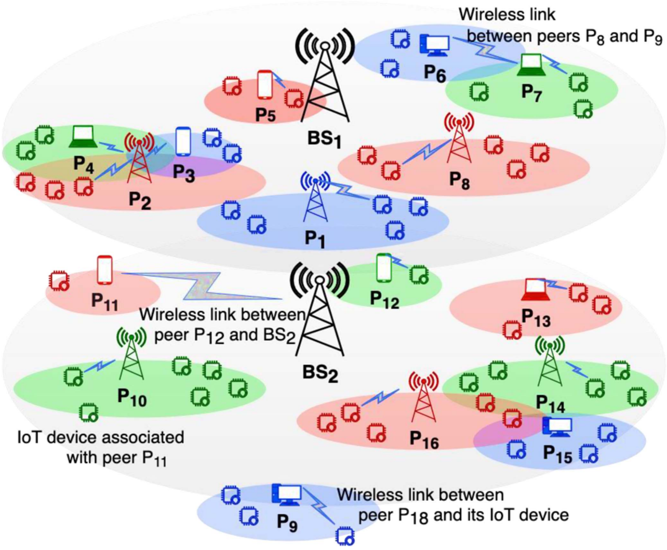

Fig. 1. BC-MEC system with two macrocell BSs, BS_1 and BS_2, and 16 peers, $P_3, \ldots, P_{18}$, represented by the small-cell BSs, mobile terminals and desktop computers. Each peer serves a set of IoT devices in its service range associated with one or more BC-MEC customers.

In order to be registered in the system, each peer $P_n, n \in \mathbf{N}$, submits a digital signature for its identification and a deposit, i.e., stake, that will be withdrawn if the peer fails to follow the rules specified in the system, e.g., refusing to perform its BC-MEC task. Upon registering, peer P_n is assigned with a set of IoT devices in its service range and allotted an orthogonal full-duplex (FD) channel of some bandwidth B_n. The bandwidth B_n can be allocated, e.g., based on the number and/or expected demands of the IoT devices associated with peer P_n. The peer's channel can be used for: 1) transmissions between peer P_n and IoT devices and 2) forwarding the peer's tasks to other peers. The channels allotted to the peers may overlap with each other and with the channels reserved for cellular customers of the macrocell BSs. Also, note that because the number of IoT devices is usually large, it is rather inefficient (e.g., in terms of bandwidth costs and spectrum utilization) to assign them with a separate spectrum [48], [49]. Instead, transmissions between any peer P_n and its associated IoT devices can be implemented (e.g., as in [48] and [49]) over the device-to-device (D2D) [50], [51] links overlaying the uplink (UL) bandwidth of the peer's channel. The UL is favored because it is underloaded compared to the downlink (DL) direction, since most transmissions between peer P_n and other peers or the associated macrocell BS are carried over a DL channel.

The process of mining the IoT blockchain is divided into a number of stages, denoted as $t = 0, \ldots, T$, during which each subsequent block of the entire IoT blockchain is processed and verified. The peers and IoT devices remain in the system for an indefinitely long time, i.e., even if the duration of their stay in the system is finite, it is unknown. Hence, the total number of stages is $T \rightarrow \propto$. Within one stage, all network parameters (e.g., channel quality, numbers/locations of peers, and IoT devices) are constant, but can change as the system transits to the next stage. At every stage t, peer P_n collects, records, and processes IoT data at its local processor of some computing power $\mathcal{X}^n$ in CPU cycles per second. The processed task output represents an unconfirmed transaction that must be distributed to and verified by other peers. If verified, the transaction is appended to a blockchain, in which case peer P_n receives some positive reward $r^n > 0$ that may depend, e.g., on the number of BC-MEC customers connected to IoT devices associated with this peer.

B. Block Mining With Sharding

In order to improve scalability, i.e., reduce the total amount of computations, the concept of sharding [16] can be utilized to verify transactions. With sharding, an entire global blockchain can be partitioned into K subchains, each of which forms a so-called "shard" managed by the disjoint subset of peers, which reduces the amount of computations by a factor of K. To scale up with the number of peers N, the number of shards K must increase with N, i.e., $K = \mathcal{O}(N)$, which results in the constant number of peers in each shard [16]. Therefore, as the sharding system expands, it becomes more susceptible to the adversaries which compute/communicate erroneous results to disrupt the block mining process [11]. Existing sharding models improves security with system solutions, e.g., by randomly sampling and updating subsets of peers managing shards [12]. Although such solutions can achieve near-optimal security for cloud-based blockchains, they are not suitable for IoT applications because: 1) peers are assigned to shards by the leader, i.e., central authority node, which leads to a single point of failure; 2) any randomly assigned peer can be malicious; and 3) peers assigned to the same shards based on random sampling can be located far from each other, which increases the block propagation time and reduces throughput [3]. Hence, since the output of every task in the shard is distributed to and verified by all peers managing this shard, the task may be orphaned, i.e., discarded, due to long transmission delay which further reduces the throughput [11], [32], [37]–[39]. For example, in a scenario in Fig. 1, peers P_6 and P_9 are located far from each other. In particular, in order to send the task output to peer P_9, peer P_6 must, first, transmit this output to BS_1 via a wireless link. After receiving this output, BS_1 must forward it to BS_2 through a fiber link. In its turns, BS_2 must forward the output to peer P_9 via a wireless link. As such, communication between peers P_6 and P_9 is realized through a three-hop link, $P_6 - \mathrm{BS}_1, \mathrm{BS}_1 - \mathrm{BS}_2$, and $\mathrm{BS}_2 - P_9$.

In order to guarantee security of the sharding system without its unnecessary centralization and without reducing the system throughput, the following approaches can be considered.

1) To facilitate a decentralized blockchain implementation, the shards can be formed in a self-organized way so that every peer can independently select a shard which maximizes its expected payoff (determined by the expected throughput of the peer).
2) To ensure security of local shards' subchains, peers can be admitted to shards based on their reputations [5], [19]. The peer's reputation can be determined from its trustworthiness, e.g., by the BC-MEC customers connected to IoT devices associated with the peer, or by the other peers managing this shard, based on the peer's past behavior in the system. As such, the output of the shard's task is added to a block of a local shard's subchain only if all peers in this shard reach a consensus on its validity before the task is orphaned. As a result, the peers must take a vote on the output of each task in their shard.
3) To minimize collusions among malicious peers in the same shard, each block of the local shard's subchain can also be verified by one or more randomly assigned peers outside of the shard. The block is appended to a global blockchain only if randomly assigned peers reach consensus on its validity. This further enhances the global blockchain security.

The above considerations enable a fully distributed BC-MEC network model with minimal control over its nodes, such as determining the maximal number of peers in a shard and the random assignment of peers outside shards. Consequently, at any stage t, there exists some shard structure $\mathbf{\Pi} = \{\mathbf{N}_1, \dots, \mathbf{N}_K\} \in \mathbf{\Omega}$ that represents a partition of set $\mathbf{N}$ into K subsets of peers managing each shard, where the set $\mathbf{\Omega}$ holds all feasible shard structures. A subset $\mathbf{N}_k \in \mathbf{\Pi}$ contains $N_k = |\mathbf{N}_k| \in [1, N_{sh\_\max}]$ peers associated with the shard k, where $N_{sh\_\max} \in (1, N]$ is a predefined maximal number of peers that can be assigned per shard. Note that since the shards are formed in a self-organized manner, the number of peers in different shards may vary. As such, the set $\mathbf{\Omega}$ is defined by

$$\mathbf{\Omega} = \left\{ \mathbf{\Pi} = \{\mathbf{N}_1, \dots, \mathbf{N}_K\} \;\middle|\; \begin{array}{c} N_k = |\mathbf{N}_k| \in \left[1, N_{sh\_\max}\right] \\ \mathbf{N}_k \cap \mathbf{N}_j = \emptyset \\ \forall k, j \in \{1, \dots, K\} \\ \bigcup_{k \in \{1,\dots,K\}} \mathbf{N}_k = \mathbf{N} \end{array} \right\}. \quad (1)$$

Note that by fixing the maximal number of peers assigned per shard, we reduce the total number of computations in the BC-MEC system by a factor of $N/N_{sh\_\max}$, which scales up with the number of peers N, since $N/N_{sh\_\max} = \mathcal{O}(N)$. As such, we can preserve the system scalability without any need for setting and updating the number of shards K. That is, the number of shards K depends on the peers' decisions about the shards. The number $N_{sh\_\max}$ can be defined based on considerations of the trade-off between security and complexity of block validation process in the shard. In particular, by increasing $N_{sh\_\max}$, we can improve security of block validations in shards, but this comes together with the growing computational complexity.

C. Block Mining Process and Reputation of Peers

In order to be appended to a local shard's subchain, the output of any task in the shard must be verified by all shard's peers before the task is orphaned. In other words, peers in the shard must reach a consensus on the output validity. Hence, the peers must vote on the task output. In many IoT-blockchain applications, the voting is based on the peers' stakes, as in PoS consensus [3]. As such, the peers with high stakes have higher voting powers. However, this approach does not ensure system security. The reason is that the peers with high stakes can be compromised by malicious peers, i.e., attackers [5], [19]. The attackers can launch voting collusion with the high-stake peers that have greater voting power, e.g., ask them to confirm false results. To eliminate such cases, voting must be based on some other metrics, e.g., reputations or trustworthiness of the peers. As such, at any stage t, the voting power of peer P_n, $n \in \mathbf{N}_k$, in the shard k managed by the set $\mathbf{N}_k$ of peers can be defined by its normalized reputation in this shard, i.e.,

$$\omega^n(\mathbf{N}_k|\boldsymbol{\rho}) = \frac{\rho^n}{\sum_{i\in\mathbf{N}_k}\rho^i} \in [0,\ 1] \tag{2}$$

where $\omega^n(\mathbf{N}_k|\boldsymbol{\rho}) \in [0,\ 1]$ is the voting power of peer P_n in the shard k managed by the subset of peers $\mathbf{N}_k$; $\boldsymbol{\rho} = \{\rho^n\}_{n\in\mathbf{N}}$ are the peers' reputations.

Note that any peer in the system can be compromised. That is, any peer may generate a false opinion about other peers to increase/decrease their reputation scores. Therefore, the peers' opinions alone cannot be used as an objective measure of the reputations of the other peers. Consequently, in addition to the opinions of peers, the system can collect opinions of parties which are not interested in falsifying their scores, e.g., the BC-MEC customers connected to IoT devices associated with each peer. As such, both the BC-MEC customers and other peers can be asked to rank their experience of interacting with the peer as a "positive" or a "negative." Then, the reputation ρ^n of peer P_n can be updated according to a subjective logic model [5], [19]. In the model, the opinion about peer P_n submitted by its BC-MEC customers is represented by the tuple $o^{0\to n} = (t^{0\to n}, d^{0\to n}, u^{0\to n})$, where $t^{0\to n}$, $d^{0\to n}$ and $u^{0\to n}$ define the customers' trust, distrust and uncertainty about peer P_n, respectively. Similarly, the opinion about peer P_n submitted by another peer P_i is the tuple $o^{i\to n} = (t^{i\to n}, d^{i\to n}, u^{i\to n})$, where $t^{i\to n}$, $d^{i\to n}$, and $u^{i\to n}$ are the trust, distrust, and uncertainty about peer P_n by peer P_i. Note that for all $i \in \{0\} \cup \mathbf{N}\backslash\{n\}$, the trust, distrust, and uncertainty are such that $t^{i\to n} + d^{i\to n} + u^{i\to n} = 1$ and $t^{i\to n}, d^{i\to n}, u^{i\to n} \in [0,\ 1]$.

Then, given that all customers and all peers have the same evaluation criteria for generating opinions, at the end of every stage t, the trust and distrust can be updated according to

$$\begin{cases} t^{i\to n} = \left(1 - u^{i\to n}\right)\frac{\chi_+^{i\to n}}{\chi_+^{i\to n}+\chi_-^{i\to n}} \\ d^{i\to n} = \left(1 - u^{i\to n}\right)\frac{\chi_-^{i\to n}}{\chi_+^{i\to n}+\chi_-^{i\to n}} \end{cases} \tag{3}$$

for all $i \in \{0\}\cup\mathbf{N}\backslash\{n\}$, where $\chi_+^{i\to n}$ and $\chi_-^{i\to n}$ are, respectively, the total numbers of positive and negative opinions about peer P_n submitted by its customers (for $i = 0$) or another peer P_i (for $i \in \mathbf{N}\backslash\{n\}$) by the end of stage t. From (3), the reputation ρ^n of peer P_n can be defined (e.g., as in [19]) by the weighted sum of expected trusts of other peers and customers to the peer, i.e.,

$$\begin{aligned} \rho^n &= w_{\mathcal{C}}\rho^{0\to n} + (1 - w_{\mathcal{C}}) \sum_{i\in\mathbf{N}\backslash\{n\}} \rho^{i\to n} \\ &= w_{\mathcal{C}} t^{0\to n} + (1 - w_{\mathcal{C}}) \sum_{i\in\mathbf{N}\backslash\{n\}} t^{i\to n} \\ &\quad + \phi\left(w_{\mathcal{C}} u^{0\to n} + (1 - w_{\mathcal{C}}) \sum_{i\in\mathbf{N}\backslash\{n\}} u^{i\to n}\right) \end{aligned} \tag{4}$$

where $w_{\mathcal{C}} \in [0, 1]$ is the weight of customers' opinions in the peer's assessment within the system that indicates how much these opinions are valued in the peer's reputation compared to the opinions of peers; $\rho^{i\to n} = t^{i\to n} + \phi u^{i\to n}$ is the expected trust of the peer's customers (for $i = 0$) or another peer P_i (for $i \in \mathbf{N}\backslash\{n\}$) to peer P_n; $\phi \in [0, 1]$ is a given constant indicating the effect of the uncertainty on the peer's reputation.

If the output of the task in the shard k is confirmed by the shard's peers before the task is orphaned, it can be appended to a global IoT blockchain. For this, the output must be validated by $\hat{N}_k = N_{sh\_\max} + 1 - N_k$ peers not associated with the shard k that are assigned uniformly at random from the subset $\mathbf{N}\backslash\mathbf{N}_k$ by the BC-MEC system. As such, in order to be appended to a global blockchain, the task must be confirmed by $\hat{N}_k + N_k = N_{sh\_\max}+1$ peers. Let $\widehat{\mathbf{N}}_t^k \in \widehat{\boldsymbol{\Omega}}^k$ be the subset of peers which are randomly assigned to the shard k at stage t, where $\widehat{\boldsymbol{\Omega}}^k = \{\widehat{\mathbf{N}}^k|\widehat{\mathbf{N}}^k \subseteq \mathbf{N}\backslash\mathbf{N}_k, |\widehat{\mathbf{N}}^k| = \hat{N}_k = N_{sh\_\max}+1 - N_k\}$ is a set that comprises all possible subsets of peers which can be randomly assigned to the shard k. To verify the task output, every peer P_i, $i \in \widehat{\mathbf{N}}_t^k$, in the subset $\widehat{\mathbf{N}}_t^k$ must take a vote. Similar to the local task validation, voting is based on the peers' reputations. Thus, at any stage t, the voting power of peer P_i in the subset $\widehat{\mathbf{N}}_t^k$, randomly assigned to the shard k can be defined as

$$\omega^i\left(\widehat{\mathbf{N}}_t^k \cup \mathbf{N}_k|\boldsymbol{\rho}\right) = \frac{\rho^i}{\sum_{n\in\widehat{\mathbf{N}}_t^k\cup\mathbf{N}_k}\rho^n} \in [0,\ 1]. \tag{5}$$

If the task output is appended to the global IoT blockchain, the peer which has recorded the task receives a certain positive reward $r^n > 0$ that may depend, e.g., on the number of BC-MEC customers connected to IoT devices associated with this peer. Otherwise, the peer receives no reward.

Based on the above, at any stage t, the trustworthiness of the block confirmation in any shard k depends on trustworthiness of all peers validating outputs of the shard's tasks, i.e., every peer P_n, $n \in \mathbf{N}_k$, managing shard k and each peer P_i, $i \in \widehat{\mathbf{N}}_t^k$, randomly assigned to shard k. As such, the reputation of shard k at stage t can be defined by the average reputation of peers verifying outputs of the shard's tasks, i.e.,

$$\rho\left(\mathbf{N}_k\middle|\widehat{\mathbf{N}}_t^k\right) = \frac{\sum_{n\in\mathbf{N}_k}\rho^n + \sum_{i\in\widehat{\mathbf{N}}_t^k}\rho^i}{N_{sh\_\max}+1} \quad \forall\mathbf{N}_k \in \boldsymbol{\Pi}. \tag{6}$$

From (6), the expected reputation $\rho^{\mathbf{N}_k}$ of shard k is given by

$$\begin{aligned} \rho^{\mathbf{N}_k} &= \mathrm{E}\left\{\rho\left(\mathbf{N}_k\middle|\widehat{\mathbf{N}}_t^k\right)\middle|\widehat{\mathbf{N}}_t^k\right\} \\ &= \frac{\sum_{n\in\mathbf{N}_k}\rho^n + \left(N_{sh\_\max}+1 - N_k\right)\widehat{\rho}^{-\mathbf{N}_k}}{N_{sh\_\max} + 1} \end{aligned} \tag{7}$$

where $\widehat{\rho}^{-\mathbf{N}_k}$ is the expected reputation of the peer randomly assigned to shard k, which is defined by the average reputation of the peers outside of subset $\mathbf{N}_k$, i.e.,

$$\widehat{\rho}^{-\mathbf{N}_k} = \frac{\sum_{i\in\mathbf{N}\backslash\mathbf{N}_k}\rho^i}{N - N_k}. \tag{8}$$

The expected shard's reputation $\rho^{\mathbf{N}_k}$ measures trustworthiness of the block confirmation process in shard k. In particular, the higher is the value of $\rho^{\mathbf{N}_k}$, the most trustworthy is the block confirmation in the corresponding shard.

IV. Analytical Model of the BC-MEC With Sharding

A. Block Delays and Energy Consumption of Peers

In order to be appended as a block of the local shard's subchain, data from IoT devices associated with the shard must be collected, processed, and verified by the shard's peers before the block's tasks are orphaned. The blocks in a local shard's subchain can be utilized to control the associated IoT devices, and for data management and analytics inside the shard. On the other hand, the block of a local shard's subchain is appended to the global IoT blockchain only if it is verified by at least one additional peer, randomly assigned to the shard. The blocks stored in a global IoT blockchain can be used for control, data management, and analytics in the entire BC-MEC network. As such, data gathered by peer P_n, $n \in \mathbf{N}$, from its associated IoT devices at the beginning of stage t is recorded as a mining task defined by the tuple $\theta_t^n = (\theta_t^{n(P)}, \theta_t^{n(O)}, \theta_t^{n(V)})$ with following parameters: 1) processing size $\theta_t^{n(P)} \geq 0$, i.e., the number of CPU cycles required to process the task; 2) output size $\theta_t^{n(O)} \geq 0$, i.e., the size of the task output in bits; and 3) verification size $\theta_t^{n(V)} \geq 0$, i.e., the number of CPU cycles required to verify the task output. Without loss of generality, $\theta_t^n = (0, 0, 0)$, if no task has been recorded at stage t.

Consider peer P_n that has recorded a mining task θ_t^n at stage t. Upon recording the task, the peer processes it locally. Note that the task must be processed by the peer within one mining stage. Otherwise, the task is discarded due to orphaning. Thus, at the beginning of every stage, the local processor buffer of peer P_n is empty. Then, given that the peer prioritizes its own tasks, i.e., processes its own tasks prior to verifying the outputs of the tasks of other peers, the processing delay for the task θ_t^n of peer P_n is given by

$$D^{n(P)}\left(\theta_t^n\right) = \theta_t^{n(P)}/x^n \tag{9}$$

and the energy spent on processing the task is given by

$$E^{n(P)}\left(\theta_t^n\right) = \vartheta^n \theta_t^{n(P)} \tag{10}$$

where ϑ^n is the energy consumption of peer P_n per CPU cycle. If peer P_n operates in a nonsingleton shard k, i.e., $N_k > 1$, it must distribute the task output to other peers in the shard, i.e., the peers in the subset $\mathbf{N}_k \backslash \{n\} \neq \emptyset$. Otherwise, i.e., if peer P_n belongs to a singleton shard, it must distribute the task output to randomly assigned peers, i.e., peers in the subset $\widehat{\mathbf{N}}_t^k \in \widehat{\mathbf{\Omega}}^k$. Thus, the transmission delay for the task θ_t^n of peer P_n is given by

$$\begin{aligned} D^{n(O)}&\left(\mathbf{\Pi}, \widehat{\mathbf{\Pi}}_t, \theta_t^n\right) \\ &= \sum_{\mathbf{N}_k \in \mathbf{\Pi}} \mathbf{1}_{n \in \mathbf{N}_k} \theta_t^{n(O)} \Bigg(\mathbf{1}_{N_k > 1} \min_{i \in \mathbf{N}_k \backslash \{n\}} R^{n,i}\left(l^n, l^i\right) \\ &\qquad\qquad + \mathbf{1}_{N_k = 1} \min_{i \in \widehat{\mathbf{N}}_t^k} R^{n,i}\left(l^n, l^i\right)\Bigg)^{-1} \end{aligned} \tag{11}$$

where $\mathbf{1}_x = 1$, if x is true and 0, otherwise; the structure $\widehat{\mathbf{\Pi}}_t = \{\widehat{\mathbf{N}}_t^1, \ldots, \widehat{\mathbf{N}}_t^K\}$ contains the subsets of peers randomly assigned to all shards at stage t; $l^n \in \mathbf{R}$ is the location of peer P_n, where $\mathbf{R} \in \mathbb{R}^2$ is the service range of the BC-MEC system, such that $\mathbf{R} = \bigcup_{m \in \mathbf{M}} \mathbf{R}_m$, where $\mathbf{R}_m$ is the coverage area of BS_m; $R^{n,i}$ is the rate at which the output of the task of peer P_n is transmitted to peer P_i that depends on locations l^n and l^i of peers P_n and P_i.

In particular, if peer P_i is in the transmission range of peer P_n, the task output can be transmitted directly to peer P_i over the DL channel of peer P_n. Otherwise, if peer P_i is not in the transmission range of peer P_n, the output is transmitted to the associated macrocell BS. The BS either: 1) transmits the task output directly to peer P_i, if this peer is in its coverage area or 2) forwards the output to the BS associated with peer P_i via fiber link, if this peer is not in its coverage area. Thus, given that the output is forwarded over the DL channel of peer P_n, we have

$$\begin{aligned} R^{n,i}\left(l^n, l^i\right) &= \mathbf{1}_{l^i \in \mathbf{R}(l^n)}\, R_{P2P}^{n,i(n)} + \mathbf{1}_{l^i \notin \mathbf{R}(l^n)} \sum_{m \in \mathbf{M}} \mathbf{1}_{b^n = m} \\ &\quad \times \left(R_{P2B}^{n,m(n)} + \sum_{j \in \mathbf{M}} \mathbf{1}_{b^i = j}\left(R_{P2B}^{i,j(n)} + \mathbf{1}_{b^i \neq m} R_{B2B}^{m,j}\right)\right) \\ &= R^{n,i(n)}\left(l^n, l^i\right) + \mathbf{1}_{l^i \notin \mathbf{R}_n(l^n)} \sum_{m \in \mathbf{M}} \sum_{j \in \mathbf{M}} \mathbf{1}_{b^n = m, b^i = j} \\ &\quad \times \left(R_{P2B}^{i,j(n)} + \mathbf{1}_{b^i \neq m}\, R_{B2B}^{m,j}\right). \end{aligned} \tag{12a}$$

In (12a), $\mathbf{R}(l^n)$ denotes the transmission range of peer P_n that depends on its location l^n in the system; $b^n \in \mathbf{M}$ indicates the BS associated with peer P_n; $R_{P2P}^{n,i(n)}$ is the rate of a wireless link between peers P_n and P_i realized over the DL channel of peer P_n; $R_{P2B}^{i,j(n)}$ is the rate of a wireless link between peer P_i and BS_j realized over a DL channel of peer P_n; $R_{B2B}^{m,j}$ is the rate of a fiber link between BS_m and BS_j that can be estimated (e.g., as in [52]) from fiber link parameters; $R^{n,i(n)}$ is the rate at which peer P_n transmits the output of its task to peer P_i (if peer P_i is in the transmission range of peer P_n) or the associated BS_m (if peer P_i is beyond the transmission range of peer P_n), given by

$$\begin{aligned} R^{n,i(n)}\left(l^n, l^i\right) = \mathbf{1}_{l^i \in \mathbf{R}(l^n)} R_{P2P}^{n,i(n)} \\ + \mathbf{1}_{l^i \notin \mathbf{R}(l^n)} \sum_{m \in \mathbf{M}} \mathbf{1}_{b^n = m} R_{P2B}^{n,m(n)}. \end{aligned} \tag{12b}$$

Assuming that the multicast transmission [53] can be used to forward the task output over a DL channel of peer P_n, the rates $R_{P2P}^{n,i(n)}$ and $R_{P2B}^{i,m(n)}$ are given by

$$R_{P2P}^{n,i(n)} = B_n \log_2\left(1 + \frac{p^n G^{n,i(n)}}{\sum_{j \in \mathbf{N} \cup \mathbf{M} \backslash \{n\}} b^{n,j} p^j G^{j,i(n)} + \sigma^2}\right) \tag{13a}$$

and

$$R_{P2B}^{i,m(n)} = B_n \log_2\left(1 + \frac{p^i G^{i,m(n)}}{\sum_{j \in \mathbf{N} \cup \mathbf{M} \backslash \{n\}} b^{i,j} p^j G^{j,m(n)} + \sigma^2}\right) \tag{13b}$$

respectively, where $b^{i,j} \in \{0,\ 1\}$, for all $i \in \mathbf{N}$, $j \in \mathbf{N} \cup \mathbf{M} \backslash \{i\}$, is the band overlap factor, such that $b^{i,j} = 1$, if the spectrum allotted to peer P_n overlaps with the spectrum allotted to peer P_j (for $j \in \mathbf{N}$) or BS_j (for $j \in \mathbf{M} \backslash \{i\}$); p^j is the transmit power of peer P_j or BS_j; $G^{j,i(n)} \in \{0,\ 1\}$ is the gain of a wireless link between peer P_j or BS_j and peer P_i realized over a DL channel of peer P_n; σ^2 is the variance of a zero-mean additive white Gaussian noise (AWGN) power. Then, we can compute the energy consumed by peer P_n on sending the output of its task θ_t^n for verification. From (12a), this energy is given by

$$\begin{aligned} &E^{n(O)}\big(\mathbf{\Pi}, \widehat{\mathbf{\Pi}}_t, \theta_t^n\big) \\ &= \sum_{\mathbf{N}_k \in \mathbf{\Pi}} \mathbf{1}_{n \in \mathbf{N}_k} p^n \theta_t^{n(O)} \Bigg(\mathbf{1}_{N_k > 1} \min_{i \in \mathbf{N}_k \setminus \{n\}} R^{n,i(n)}\big(l^n, l^i\big) \\ &\qquad + \mathbf{1}_{N_k = 1} \min_{i \in \widehat{\mathbf{N}}_t^k} R^{n,i(n)}\big(l^n, l^i\big) \Bigg)^{-1}. \qquad (14) \end{aligned}$$

Any peer P_i which has received the task output $\theta_t^{n(O)}$ must validate it by utilizing its local processor. Note that the peer processes its own task θ_t^i prior to verifying the task outputs of other peers. However, the peer does not differentiate among the tasks of other peers. As such, the tasks of other peers have the same priorities. Hence, the probability that the output $\theta_t^{n(O)}$ is placed after the task of peer P_i is equal to the probability that the output is placed at the end of a local processor buffer of peer P_i. Hence, the verification delay for the task θ_t^n of peer P_n in the shard k is given by

$$\begin{aligned} &D^{n(V)}\big(\mathbf{\Pi}, \widehat{\mathbf{\Pi}}_t, \boldsymbol{\theta}_t\big) \\ &= \sum_{\mathbf{N}_k \in \mathbf{\Pi}} \mathbf{1}_{n \in \mathbf{N}_k} \Bigg(\mathbf{1}_{N_k > 1} \max_{i \in \mathbf{N}_k \setminus \{n\}} D^{n,i(V)}\big(\mathbf{N}_k, \widehat{\mathbf{\Pi}}_t, \boldsymbol{\theta}_t\big) \\ &\qquad + \mathbf{1}_{N_k = 1} \max_{i \in \widehat{\mathbf{N}}_t^k} D^{n,i(V)}\big(\mathbf{N}_k, \widehat{\mathbf{\Pi}}_t, \boldsymbol{\theta}_t\big) \Bigg) \qquad (15a) \end{aligned}$$

where $\boldsymbol{\theta}_t = \{\theta_t^n\}_{n \in \mathbf{N}}$ are the parameters of the tasks of peers; $D^{n,i(V)}(\mathbf{N}_k, \widehat{\mathbf{\Pi}}_t, \boldsymbol{\theta}_t)$ is the verification delay for the task of peer P_n at the local processor buffer of peer P_i, given by

$$\begin{aligned} &D^{n,i(V)}\big(\mathbf{N}_k, \widehat{\mathbf{\Pi}}_t, \boldsymbol{\theta}_t\big) \\ &= \frac{1}{x^i} \Bigg(\theta_t^{i(P)} + \theta_t^{n(V)} \\ &\qquad + \frac{1}{2} \Bigg(\sum_{j \in \mathbf{N}_k \setminus \{i,n\}} \theta_t^{j(V)} + \sum_{\widehat{\mathbf{N}}_t^r \in \widehat{\mathbf{\Pi}}_t} \mathbf{1}_{i \in \widehat{\mathbf{N}}_t^r} \sum_{j \in \mathbf{N}_r} \theta_t^{j(V)} \Bigg) \Bigg). \qquad (15b) \end{aligned}$$

In the above equation, the last two terms are represented by the sum $\mathbf{1}_{N_k > 2} \sum_{j \in \mathbf{N}_k \setminus \{i,n\}} \theta_t^{j(V)} + \sum_{\widehat{\mathbf{N}}_t^r \in \widehat{\mathbf{\Pi}}_t} \mathbf{1}_{i \in \widehat{\mathbf{N}}_t^r} \sum_{j \in \mathbf{N}_r} \theta_t^{j(V)}$ that takes into account both: 1) verification of tasks generated in the shard k of peer P_i, i.e., $\sum_{j \in \mathbf{N}_k \setminus \{i,n\}} \theta_t^{j(V)}$ and 2) verification of tasks generated in shards to which peer P_i is assigned randomly, i.e., $\sum_{\widehat{\mathbf{N}}_t^r \in \widehat{\mathbf{\Pi}}_t} \mathbf{1}_{i \in \widehat{\mathbf{N}}_t^r} \sum_{j \in \mathbf{N}_r} \theta_t^{j(V)}$. On the other hand, the energy spent by peer P_n on verifying the tasks of other peers, i.e., peers in the shard k of peer P_n and peers in shards to which peer P_n is assigned randomly, is given by

$$\begin{aligned} &E^{n(V)}\big(\mathbf{\Pi}, \widehat{\mathbf{\Pi}}_t, \boldsymbol{\theta}_t\big) \\ &= \sum_{\mathbf{N}_k \in \mathbf{\Pi}} \mathbf{1}_{n \in \mathbf{N}_k} \Bigg(\sum_{i \in \mathbf{N}_k \setminus \{n\}} \vartheta^n \theta_t^{i(V)} \\ &\qquad + \sum_{\widehat{\mathbf{N}}_t^r \in \widehat{\mathbf{\Pi}}_t} \mathbf{1}_{n \in \widehat{\mathbf{N}}_t^r} \sum_{i \in \mathbf{N}_r} \vartheta^n \theta_t^{i(V)} \Bigg). \qquad (16) \end{aligned}$$

B. Expected Task Delays and Energy Consumption of Peers

Apparently, the total delay for the task θ_t^n of peer P_n given the shard structures $\mathbf{\Pi}$ and $\widehat{\mathbf{\Pi}}_t$, and task parameters $\boldsymbol{\theta}_t$ is the sum of processing, transmission and verification delays, i.e.,

$$\begin{aligned} D^n\big(\mathbf{\Pi}, \widehat{\mathbf{\Pi}}_t, \boldsymbol{\theta}_t\big) &= D^{n(P)}\big(\theta_t^n\big) + D^{n(O)}\big(\mathbf{\Pi}, \widehat{\mathbf{\Pi}}_t, \boldsymbol{\theta}_t\big) \\ &\quad + D^{n(V)}\big(\mathbf{\Pi}, \widehat{\mathbf{\Pi}}_t, \boldsymbol{\theta}_t\big). \qquad (17) \end{aligned}$$

Note that the voting delay (or, equivalently, time on returning the votes, i.e., verification results, to peer P_n) is negligible, as the vote of any peer P_i verifying the task output of peer P_n can be encoded with only one bit, e.g., "1," if the output is verified and "0," otherwise. Therefore, the delay on returning the vote, denoted as $D^{n(R)}$, is such that

$$\begin{aligned} D^{n(R)}\big(\mathbf{\Pi}, \widehat{\mathbf{\Pi}}_t\big) &= \sum_{\mathbf{N}_k \in \mathbf{\Pi}} \mathbf{1}_{n \in \mathbf{N}_k} \Bigg(\mathbf{1}_{N_k > 1} \min_{i \in \mathbf{N}_k \setminus \{n\}} R^{n,i}\big(l^n, l^i\big) \\ &\qquad + \mathbf{1}_{N_k = 1} \min_{i \in \widehat{\mathbf{N}}_t^k} R^{n,i}\big(l^n, l^i\big) \Bigg) \\ &\ll D^{n(O)}\big(\mathbf{\Pi}, \widehat{\mathbf{\Pi}}_t, \theta_t^n\big) \end{aligned}$$

i.e., very small compared to the transmission delay $D^{n(O)}$ in (11). On the other hand, the total energy spent by peer P_n per stage on processing and transmitting its own task θ_t^n and verifying the tasks of other peers given the shard structures $\mathbf{\Pi}$ and $\widehat{\mathbf{\Pi}}_t$, and task parameters $\boldsymbol{\theta}_t$ is given by

$$\begin{aligned} E^n\big(\mathbf{\Pi}, \widehat{\mathbf{\Pi}}_t, \boldsymbol{\theta}_t | \ell\big) &= E^{n(P)}\big(\theta_t^n\big) + E^{n(O)}\big(\mathbf{\Pi}, \widehat{\mathbf{\Pi}}_t, \theta_t^n | \ell\big) \\ &\quad + E^{n(V)}\big(\mathbf{\Pi}, \widehat{\mathbf{\Pi}}_t, \boldsymbol{\theta}_t\big). \qquad (18) \end{aligned}$$

Then, assuming that the task parameters $\boldsymbol{\theta}_t$ follow the certain probability distributions with the means $\theta = (\theta^P, \theta^O, \theta^V)$, we can obtain the closed-form expressions of the expected task delay and expected energy consumption per stage for peer P_n in any shard structure $\mathbf{\Pi} \in \mathbf{\Omega}$.

Proposition 1: The expected delay $\mathcal{D}^n(\mathbf{\Pi})$ for a task of peer P_n in the shard structure $\mathbf{\Pi} = \{\mathbf{N}_1, \ldots, \mathbf{N}_K\} \in \mathbf{\Omega}$ is a sum

$$\mathcal{D}^n(\mathbf{\Pi}) = \mathcal{D}^{n(P)} + \mathcal{D}^{n(O)}(\mathbf{\Pi}) + \mathcal{D}^{n(V)}(\mathbf{\Pi}) \qquad (19a)$$

of the expected processing delay $\mathcal{D}^{n(P)}$, expected transmission delay $\mathcal{D}^{n(O)}$, and expected verification delay $\mathcal{D}^{n(V)}$, given by

$$\mathcal{D}^{n(P)} = \theta^P / x^n \qquad (19b)$$

$$\begin{aligned} &\mathcal{D}^{n(O)}(\mathbf{\Pi}) \\ &= \sum_{\mathbf{N}_k \in \mathbf{\Pi}} \mathbf{1}_{n \in \mathbf{N}_k} \theta^O \Bigg(\mathbf{1}_{N_k > 1} \min_{i \in \mathbf{N}_k \setminus \{n\}} R^{n,i}\big(l^n, l^i\big) \\ &\qquad + \mathbf{1}_{N_k = 1} \frac{\sum_{\widehat{\mathbf{N}}^k \subseteq \widehat{\mathbf{\Omega}}^k} \min_{i \in \widehat{\mathbf{N}}^k} R^{n,i}\big(l^n, l^i\big)}{(N-1)^{N_{sh\_\max}}} \Bigg)^{-1} \qquad (19c) \end{aligned}$$

and

$$\begin{aligned} &\mathcal{D}^{n(V)}(\mathbf{\Pi}) \\ &= \sum_{\mathbf{N}_k \in \mathbf{\Pi}} \mathbf{1}_{n \in \mathbf{N}_k} \Bigg(\mathbf{1}_{N_k > 1} \max_{i \in \mathbf{N}_k \setminus \{n\}} \mathcal{D}^{n,i(V)}(\mathbf{\Pi} | \mathbf{N}_k) \\ &\qquad + \mathbf{1}_{N_k = 1} \frac{\sum_{\widehat{\mathbf{N}}^k \subseteq \widehat{\mathbf{\Omega}}^k} \max_{i \in \widehat{\mathbf{N}}_t^k} \mathcal{D}^{n,i(V)}(\mathbf{\Pi} | \mathbf{N}_k)}{(N-1)^{N_{sh\_\max}}} \Bigg) \qquad (19d) \end{aligned}$$

respectively, where $\mathcal{D}^{n,i(V)}(\mathbf{\Pi}|\mathbf{N}_k)$ is the expected verification delay for the tasks of peer P_n at the local processor buffer of peer P_i in the shard k, given by

$$\mathcal{D}^{n,i(V)}(\mathbf{\Pi}|\mathbf{N}_k) = \frac{1}{x^i}\left(\theta^P + \frac{\theta^V}{2}\left(2 + \lceil N_k - 2\rceil^+ + \mathcal{Z}(\mathbf{\Pi})\right)\right) \tag{19e}$$

for $\lceil x\rceil^+ = \max\{0, x\}$, and

$$\mathcal{Z}(\mathbf{\Pi}) = \sum_{\mathbf{N}_r\in\mathbf{\Pi}} N_r(N - N_r)^{N_{sh\_\max}-N_r-1}. \tag{19f}$$

Proposition 2: The expected energy $\mathcal{E}^n(\mathbf{\Pi})$ spent by peer P_n per stage in the shard structure $\mathbf{\Pi} = \{\mathbf{N}_1, \ldots, \mathbf{N}_K\} \in \mathbf{\Omega}$ is a sum

$$\mathcal{E}^n(\mathbf{\Pi}) = \mathcal{E}^{n(P)} + \mathcal{E}^{n(O)}(\mathbf{\Pi}) + \mathcal{E}^{n(V)}(\mathbf{\Pi}) \tag{20a}$$

of the expected energy consumed on task processing $\mathcal{E}^{n(P)}$, expected energy spent on task transmission $\mathcal{E}^{n(O)}$ and expected energy spend on task verification $\mathcal{E}^{n(V)}$, given by

$$\mathcal{E}^{n(P)} = \vartheta^n\theta^P \tag{20b}$$

$$\begin{aligned}\mathcal{E}^{n(O)}&(\mathbf{\Pi})\\ &= \sum_{\mathbf{N}_k\in\mathbf{\Pi}} \mathbf{1}_{n\in\mathbf{N}_k} p^n\theta^O\left(\mathbf{1}_{N_k>1}\min_{i\in\mathbf{N}_k\backslash\{n\}} R^{n,i(n)}\right.\\ &\qquad\left. + \mathbf{1}_{N_k=1}\frac{\sum_{\widehat{\mathbf{N}}^k\subseteq\widehat{\mathbf{\Omega}}^k}\min_{i\in\widehat{\mathbf{N}}^k} R^{n,i(n)}}{(N-1)^{N_{sh\_\max}}}\right)^{-1}\end{aligned} \tag{20c}$$

and

$$\mathcal{E}^{n(V)}(\mathbf{\Pi}) = \sum_{\mathbf{N}_k\in\mathbf{\Pi}} \mathbf{1}_{n\in\mathbf{N}_k}\vartheta^n\theta^V((N_k - 1) + \mathcal{Z}(\mathbf{\Pi})) \tag{20d}$$

respectively.

The proofs of Propositions 1 and 2 are given in Appendices A and B, respectively. These propositions present the closed-form expressions of the expected task delay and expected energy consumption per stage for every peer in any possible shard structure. However, the computation of these expressions is intractable for a large number of peers N. In particular, since

$$\left|\widehat{\mathbf{\Omega}}^k\right| \subseteq \left\{\widehat{\mathbf{N}}^k|\widehat{\mathbf{N}}^k \subseteq \mathbf{N}\backslash\{n\}, \hat{N}_k = \left|\widehat{\mathbf{N}}^k\right| = N_{sh\_\max}\right\} \leq (N-1)!$$

the time complexity of estimating $\mathcal{D}^n$ and $\mathcal{E}^n$ is $\mathcal{O}(|\widehat{\mathbf{\Omega}}^k|) = \mathcal{O}(N!)$. Accordingly, in Corollaries 1 and 2, we establish the tight (i.e., least) upper bounds or suprema of expected delay $\mathcal{D}^n$ and energy $\mathcal{E}^n$ which are computable in polynomial time.

Corollary 1: The expected delay $\mathcal{D}^n(\mathbf{\Pi})$ for a task of peer P_n in the shard structure $\mathbf{\Pi} = \{\mathbf{N}_1, \ldots, \mathbf{N}_K\} \in \mathbf{\Omega}$ is tightly bounded from above by the value

$$\widetilde{\mathcal{D}}^n(\mathbf{\Pi}) = \sup\mathcal{D}^n(\mathbf{\Pi}) \ = \mathcal{D}^{n(P)} + \widetilde{\mathcal{D}}^{n(O)}(\mathbf{\Pi}) + \widetilde{\mathcal{D}}^{n(V)}(\mathbf{\Pi}) \tag{21a}$$

where $\widetilde{\mathcal{D}}^{n(O)}$ is the least upper bound of expected transmission delay $\mathcal{D}^{n(O)}$ for the task, given by

$$\begin{aligned}\widetilde{\mathcal{D}}^{n(O)}(\mathbf{\Pi}) &= \sum_{\mathbf{N}_k\in\mathbf{\Pi}} \mathbf{1}_{n\in\mathbf{N}_k}\theta^O\left(\mathbf{1}_{N_k>1}\min_{i\in\mathbf{N}_k\backslash\{n\}} R^{n,i}\left(l^n, l^i\right)\right.\\ &\qquad\left. + \mathbf{1}_{N_k=1}\mathcal{Z}(1)\min_{i\in\mathbf{N}\backslash\{n\}} R^{n,i}\left(l^n, l^i\right)\right)^{-1}\\ &= \sup\mathcal{D}^{n(O)}(\mathbf{\Pi})\end{aligned} \tag{21b}$$

and $\widetilde{\mathcal{D}}^{n(V)}$ is the least upper bound of expected verification delay $\mathcal{D}^{n(V)}$ for the task, given by

$$\begin{aligned}\widetilde{\mathcal{D}}^{n(V)}(\mathbf{\Pi}) &= \sum_{\mathbf{N}_k\in\mathbf{\Pi}} \mathbf{1}_{n\in\mathbf{N}_k}\left(\mathbf{1}_{N_k>1}\max_{i\in\mathbf{N}_k\backslash\{n\}} \mathcal{D}^{n,i(V)}(\mathbf{\Pi}|\mathbf{N}_k)\right.\\ &\qquad\left. + \mathbf{1}_{N_k=1}\mathcal{Z}(1)\max_{i\in\mathbf{N}\backslash\{n\}} \mathcal{D}^{n,i(V)}(\mathbf{\Pi}|\mathbf{N}_k)\right)\\ &= \sup\mathcal{D}^{n(V)}(\mathbf{\Pi})\end{aligned} \tag{21c}$$

with $\mathcal{Z}(x)$ given by

$$\mathcal{Z}(x) = \frac{1}{(N-x)^{N_{sh\_\max}+1-x}}\binom{N-x}{N_{sh\_\max}+1-x}. \tag{21d}$$

Corollary 2: The expected energy $\mathcal{E}^n(\mathbf{\Pi})$ spent by peer P_n per stage in the shard structure $\mathbf{\Pi} = \{\mathbf{N}_1, \ldots, \mathbf{N}_K\} \in \mathbf{\Omega}$ is tightly bounded from above by the value

$$\widetilde{\mathcal{E}}^n(\mathbf{\Pi}) = \sup\mathcal{E}^n(\mathbf{\Pi}) = \mathcal{E}^{n(P)} + \widetilde{\mathcal{E}}^{n(O)}(\mathbf{\Pi}) + \mathcal{E}^{n(V)}(\mathbf{\Pi}) \tag{22a}$$

where $\widetilde{\mathcal{E}}^{n(O)}(\mathbf{\Pi})$ is the least upper bound of expected energy $\mathcal{E}^{n(O)}$ spent on task transmission, given by

$$\begin{aligned}\widetilde{\mathcal{E}}^{n(O)}(\mathbf{\Pi}) &= \sum_{\mathbf{N}_k\in\mathbf{\Pi}} \mathbf{1}_{n\in\mathbf{N}_k} p^n\theta^O\left(\mathbf{1}_{N_k>1}\min_{i\in\mathbf{N}_k\backslash\{n\}} R^{n,i(n)}\right.\\ &\qquad\left. + \mathbf{1}_{N_k=1}\mathcal{Z}(1)\min_{i\in\mathbf{N}\backslash\{n\}} R^{n,i(n)}\right)^{-1}\\ &= \sup\mathcal{E}^{n(O)}(\mathbf{\Pi}).\end{aligned} \tag{22b}$$

The proofs of Corollaries 1 and 2 are given in Appendices C and D, respectively. These corollaries present the least upper bounds $\widetilde{\mathcal{D}}^n$ and $\widetilde{\mathcal{E}}^n$ of the expected task delay and expected energy consumption per stage for peer P_n in any shard structure $\mathbf{\Pi} \in \mathbf{\Omega}$. Unlike the exact expected values $\mathcal{D}^n$ and $\mathcal{E}^n$, their least upper bounds can be estimated in polynomial time, since the worst case complexity of computing $\widetilde{\mathcal{D}}^n$ and $\widetilde{\mathcal{E}}^n$ is $\mathcal{O}(NK)$.

V. Self-Organized Shard Formation by Peers

A. Expected Rewards and Payoffs of the Peers

From the expressions of the expected delay $\mathcal{D}^n$ and its least upper bound $\widetilde{\mathcal{D}}^n$ in (19a) and (21a), peer P_n can estimate the probability of orphaning $\mathcal{P}^n_{\mathcal{O}}(\mathbf{\Pi})$ for its tasks, i.e., probability that the peer's task is discarded due to long delay, in any shard structure $\mathbf{\Pi} \in \mathbf{\Omega}$. In particular, according to [36], [54], we have

$$\begin{aligned}\mathcal{P}^n_{\mathcal{O}}(\mathbf{\Pi}) = 1 - e^{-\mathcal{D}^n(\mathbf{\Pi})/\Delta t} &\leq \widetilde{\mathcal{P}}^n_{\mathcal{O}}(\mathbf{\Pi}) = \sup\mathcal{P}^n_{\mathcal{O}}(\mathbf{\Pi})\\ &= 1 - e^{-\widetilde{\mathcal{D}}^n(\mathbf{\Pi})/\Delta t}\end{aligned} \tag{23}$$

where Δt is the expected block interval time or stage duration; $\widetilde{\mathcal{P}}^n_{\mathcal{O}}$ is the least upper bound or supremum of $\mathcal{P}^n_{\mathcal{O}}$. Note that if all peers in the system were faithful, i.e., if $\mathbf{N}_{\mathcal{F}} = \mathbf{N}$, where $\mathbf{N}_{\mathcal{F}}$ is a subset of faithful peers, both the expected throughput $\mathcal{T}^n_{\mathcal{F}}(\mathbf{\Pi})$ and expected reward $\mathcal{R}^n_{\mathcal{F}}(\mathbf{\Pi})$ of peer P_n would depend only on the orphaning probability $\mathcal{P}^n_{\mathcal{O}}(\mathbf{\Pi})$ in a shard structure $\mathbf{\Pi} \in \mathbf{\Omega}$. In particular, the expected throughput $\mathcal{T}^n_{\mathcal{F}}(\mathbf{\Pi})$, i.e., the number of transactions of peer P_n verified per time unit, and its greatest lower bound or infimum $\widetilde{\mathcal{T}}^n_{\mathcal{F}}(\mathbf{\Pi})$ would take the form

$$\begin{aligned}\mathcal{T}^n_{\mathcal{F}}(\mathbf{\Pi}) &= \big(1-\mathcal{P}^n_{\mathcal{O}}(\mathbf{\Pi})\big)/\Delta t \geq \widetilde{\mathcal{T}}^n_{\mathcal{F}}(\mathbf{\Pi})\\ &= \inf \mathcal{T}^n_{\mathcal{F}}(\mathbf{\Pi}) \ = \big(1-\widetilde{\mathcal{P}}^n_{\mathcal{O}}(\mathbf{\Pi})\big)/\Delta t.\end{aligned} \tag{24a}$$

The expected reward $\mathcal{R}^n_{\mathcal{F}}(\mathbf{\Pi})$ is simply the product of expected throughput $\mathcal{T}^n_{\mathcal{F}}(\mathbf{\Pi})$, block reward r^n and block interval time Δt [36], [54]. Hence, we have

$$\begin{aligned}\mathcal{R}^n_{\mathcal{F}}(\mathbf{\Pi}) &= \mathcal{T}^n_{\mathcal{F}}(\mathbf{\Pi})r^n\Delta t = \big(1-\mathcal{P}^n_{\mathcal{O}}(\mathbf{\Pi})\big)r^n \geq \widetilde{\mathcal{R}}^n_{\mathcal{F}}(\mathbf{\Pi})\\ &= \inf \mathcal{R}^n_{\mathcal{F}}(\mathbf{\Pi}) \ = \widetilde{\mathcal{T}}^n_{\mathcal{F}}(\mathbf{\Pi})r^n\Delta t = \big(1-\widetilde{\mathcal{P}}^n_{\mathcal{O}}(\mathbf{\Pi})\big)r^n\end{aligned} \tag{24b}$$

where $\widetilde{\mathcal{R}}^n_{\mathcal{F}}(\mathbf{\Pi})$ is the greatest lower bound or infimum of $\mathcal{R}^n_{\mathcal{F}}(\mathbf{\Pi})$.

In general, however, we have $\mathbf{N}_{\mathcal{F}} \subseteq \mathbf{N}$, i.e., any subset $\mathbf{N}_{\mathcal{M}} = \mathbf{N}\backslash\mathbf{N}_{\mathcal{F}}$ of peers in the system can be malicious. The goal of malicious peers is to disrupt the process of block mining and minimize rewards of faithful peers by deliberately computing and communicating incorrect results. More specifically, if the faithful peer P_n, $n \in \mathbf{N}_{\mathcal{F}}$, transmits the correct task output to a malicious peer, the output may be not verified, in which case peer P_n does not receive a reward. Thus, both the throughput and the reward of peer P_n in shard k depend on faithfulness of peers verifying its task output, i.e., peers in shard k and peers that are randomly assigned to shard k. As such, at any stage t, given a subset $\mathbf{N}_{\mathcal{F}}$, the expected throughput $\mathcal{T}^n(\mathbf{\Pi}|\mathbf{N}_{\mathcal{F}}, \widehat{\mathbf{\Pi}}_t, \boldsymbol{\rho})$ and expected reward $\mathcal{R}^n(\mathbf{\Pi}|\mathbf{N}_{\mathcal{F}}, \widehat{\mathbf{\Pi}}_t, \boldsymbol{\rho})$ of peer P_n in the shard structure $\mathbf{\Pi} \in \mathbf{\Omega}$ for a given structure $\widehat{\mathbf{\Pi}}_t$ of randomly assigned peers and peers' reputations $\boldsymbol{\rho}$, take the forms

$$\begin{aligned}&\mathcal{T}^n\big(\mathbf{\Pi}|\mathbf{N}_{\mathcal{F}}, \widehat{\mathbf{\Pi}}_t, \boldsymbol{\rho}\big)\\ &\quad= \mathcal{T}^n_{\mathcal{F}}(\mathbf{\Pi}) \sum_{\mathbf{N}_k\in\mathbf{\Pi}} \mathbf{1}_{n\in\mathbf{N}_k}\\ &\quad\quad\times \Bigg(\sum_{i\in\mathbf{N}_k\backslash\{n\}} \mathbf{1}_{i\in\mathbf{N}_{\mathcal{F}}}\Pr\{i \in \mathbf{N}_{\mathcal{F}}\}\omega^i(\mathbf{N}_k|\boldsymbol{\rho})\\ &\quad\quad\quad+ \sum_{i\in\widehat{\mathbf{N}}^k_t} \mathbf{1}_{i\in\mathbf{N}_{\mathcal{F}}}\Pr\{i \in \mathbf{N}_{\mathcal{F}}\}\omega^i\Big(\widehat{\mathbf{N}}^k_t \cup \mathbf{N}_k|\boldsymbol{\rho}\Big)\Bigg)\end{aligned} \tag{25a}$$

and

$$\mathcal{R}^n\big(\mathbf{\Pi}|\mathbf{N}_{\mathcal{F}}, \widehat{\mathbf{\Pi}}_t, \boldsymbol{\rho}\big) = \mathcal{T}^n\big(\mathbf{\Pi}|\mathbf{N}_{\mathcal{F}}, \widehat{\mathbf{\Pi}}_t, \boldsymbol{\rho}\big)r^n\Delta t \tag{25b}$$

respectively, where $\Pr\{i \in \mathbf{N}_{\mathcal{F}}\}$ is the probability that peer P_i is faithful.

Unfortunately, no faithful peer P_n knows if another peer P_i is faithful, i.e., if $i \in \mathbf{N}_{\mathcal{F}}$. That is, no peer P_n can directly estimate its expected throughput $\mathcal{T}^n(\mathbf{\Pi}|\mathbf{N}_{\mathcal{F}}, \widehat{\mathbf{\Pi}}_t, \boldsymbol{\rho})$ and expected reward $\mathcal{R}^n(\mathbf{\Pi}|\mathbf{N}_{\mathcal{F}}, \widehat{\mathbf{\Pi}}_t, \boldsymbol{\rho})$ from (25a) and (25b). As such, the peer can only rely on its own experience of interacting with peer P_i, as well as on reputation scores submitted by BC-MEC customers of peer P_i and the other peers, to construct its estimate $\mathcal{B}^{n\to i} = \Pr_n\{i \in \mathbf{N}_{\mathcal{F}}\} \in [0,\ 1]$ of the probability that peer P_i is faithful. The estimate $\mathcal{B}^{n\to i}$ which, essentially, represents the "belief" of peer P_n in faithfulness of another peer P_i can be updated with a subjective logic model [5], [19], as

$$\begin{aligned}\mathcal{B}^{n\to i} &= \Pr_n\{i \in \mathbf{N}_{\mathcal{F}}\}\\ &= \big(1-w^n_{\mathcal{O}}\big)\Bigg(w^n_{\mathcal{C}}\rho^{0\to i} + \big(1-w^n_{\mathcal{C}}\big)\sum_{j\in\mathbf{N}\backslash\{n,i\}}\rho^{j\to i}\Bigg)\\ &\quad+ w^n_{\mathcal{O}}\rho^{n\to i}.\end{aligned} \tag{26}$$

In (26), $w^n_{\mathcal{O}} \in [0,\ 1]$ is the weight of the opinion of peer P_n in its assessment of another peer which indicates how much the peer's own opinion is valued compared to other opinions; $w^n_{\mathcal{C}} \in [0,\ 1]$ is the weight of the opinion of the peer's customer(s) in the assessment of peer P_n. As such, the greater is the weight $w^n_{\mathcal{O}}$, the more is the peer's own opinion is valued; the greater is the weight $w^n_{\mathcal{C}}$, the more is the customers' opinion is valued in the peer's assessment.

Based on beliefs $\mathcal{B}^n = \{\mathcal{B}^{n\to i}\}_{i\in\mathbf{N}\backslash\{n\}} \in [0,\ 1]^{N-1}$ of peer P_n in faithfulness of other peers updated according to (26), peer P_n can estimate its own expected throughput $\mathcal{T}^n(\mathbf{\Pi}|\mathcal{B}^n, \widehat{\mathbf{\Pi}}_t, \boldsymbol{\rho})$ and expected reward $\mathcal{R}^n(\mathbf{\Pi}|\mathcal{B}^n, \widehat{\mathbf{\Pi}}_t, \boldsymbol{\rho})$ in the shard structure $\mathbf{\Pi} \in \mathbf{\Omega}$ given a structure $\widehat{\mathbf{\Pi}}_t$ of randomly assigned peers and the peers' reputations $\boldsymbol{\rho}$. In particular, from (25a) and (25b), we have

$$\begin{aligned}\mathcal{T}^n\big(\mathbf{\Pi}|\mathcal{B}^n, \widehat{\mathbf{\Pi}}_t, \boldsymbol{\rho}\big) &= \mathcal{T}^n_{\mathcal{F}}(\mathbf{\Pi}) \sum_{\mathbf{N}_k\in\mathbf{\Pi}} \mathbf{1}_{n\in\mathbf{N}_k}\\ &\quad\times \Bigg(\sum_{i\in\mathbf{N}_k\backslash\{n\}} \mathcal{B}^{n\to i}\omega^i(\mathbf{N}_k|\boldsymbol{\rho})\\ &\quad\quad+ \sum_{i\in\widehat{\mathbf{N}}^k_t} \mathcal{B}^{n\to i}\omega^i\Big(\widehat{\mathbf{N}}^k_t \cup \mathbf{N}_k|\boldsymbol{\rho}\Big)\Bigg)\end{aligned} \tag{27a}$$

and

$$\mathcal{R}^n\big(\mathbf{\Pi}|\mathcal{B}^n, \widehat{\mathbf{\Pi}}_t, \boldsymbol{\rho}\big) = \mathcal{T}^n\big(\mathbf{\Pi}|\mathcal{B}^n, \widehat{\mathbf{\Pi}}_t, \boldsymbol{\rho}\big)r^n\Delta t. \tag{27b}$$

Then, the value $\mathcal{V}^n(\mathbf{\Pi}|\mathcal{B}^n, \boldsymbol{\rho})$, i.e., expected payoff of peer P_n in the shard structure $\mathbf{\Pi} \in \mathbf{\Omega}$ given its beliefs $\mathcal{B}^{n\to i}$ and peers' reputations $\boldsymbol{\rho}$ is the difference between its expected reward and expected cost. That is

$$\begin{aligned}\mathcal{V}^n\big(\mathbf{\Pi}|\mathcal{B}^n, \boldsymbol{\rho}\big) &= \mathcal{R}^n\big(\mathbf{\Pi}|\mathcal{B}^n, \boldsymbol{\rho}\big) - \varphi^n\mathcal{E}^n(\mathbf{\Pi})\\ &= \mathcal{T}^n\big(\mathbf{\Pi}|\mathcal{B}^n, \boldsymbol{\rho}\big)r^n\Delta t - \varphi^n\mathcal{E}^n(\mathbf{\Pi})\\ &= \mathrm{E}\big\{\mathcal{T}^n\big(\mathbf{\Pi}|\mathcal{B}^n, \widehat{\mathbf{\Pi}}_t, \boldsymbol{\rho}\big)|\mathbf{\Pi}, \mathcal{B}^n, \boldsymbol{\rho}\big\}r^n\Delta t - \varphi^n\mathcal{E}^n(\mathbf{\Pi})\\ &= \mathrm{E}\big\{\mathcal{R}^n\big(\mathbf{\Pi}|\mathcal{B}^n, \widehat{\mathbf{\Pi}}_t, \boldsymbol{\rho}\big)|\mathbf{\Pi}, \mathcal{B}^n, \boldsymbol{\rho}\big\} - \varphi^n\mathcal{E}^n(\mathbf{\Pi})\end{aligned} \tag{28}$$

where $\mathcal{T}^n(\mathbf{\Pi}|\mathcal{B}^n, \boldsymbol{\rho}) = \mathrm{E}\{\mathcal{T}^n(\mathbf{\Pi}|\mathcal{B}^n, \widehat{\mathbf{\Pi}}_t, \boldsymbol{\rho})|\mathbf{\Pi}, \mathcal{B}^n, \boldsymbol{\rho}\}$ is expected throughput and $\mathcal{R}^n(\mathbf{\Pi}|\mathcal{B}^n, \boldsymbol{\rho}) = \mathrm{E}\{\mathcal{R}^n(\mathbf{\Pi}|\mathcal{B}^n, \widehat{\mathbf{\Pi}}_t, \boldsymbol{\rho})|\mathbf{\Pi}, \mathcal{B}^n, \boldsymbol{\rho}\}$ is expected reward of peer P_n in the shard structure $\mathbf{\Pi} \in \mathbf{\Omega}$ given its beliefs $\mathcal{B}^n_t$ and the peer's reputations $\boldsymbol{\rho}$; φ^n is the cost per energy unit for peer P_n. Note that in order to estimate its value $\mathcal{V}^n(\mathbf{\Pi}|\mathcal{B}^n, \boldsymbol{\rho})$ in (28), the peer should be able to compute the expectation $\mathrm{E}\{\mathcal{R}^n(\mathbf{\Pi}|\mathcal{B}^n, \widehat{\mathbf{\Pi}}_t, \boldsymbol{\rho})|\mathbf{\Pi}, \mathcal{B}^n, \boldsymbol{\rho}\}$. Thus, we must obtain the closed-form expression of this expectation leading to the value $\mathcal{V}^n(\mathbf{\Pi}|\mathcal{B}^n, \boldsymbol{\rho})$, which we do in Proposition 3.

Proposition 3: The value or expected reward $\mathcal{V}^n(\mathbf{\Pi}|\mathcal{B}^n, \ell)$ of peer P_n in the shard structure $\mathbf{\Pi} = \{\mathbf{N}_1, \ldots, \mathbf{N}_K\} \in \mathbf{\Omega}$ given the beliefs $\mathcal{B}^n$ and reputations $\boldsymbol{\rho}$, is expressed by

$$\begin{aligned}\mathcal{V}^n\big(\mathbf{\Pi}|\mathcal{B}^n, \boldsymbol{\rho}\big) &= \mathcal{R}^n_{\mathcal{F}}(\mathbf{\Pi}) \sum_{\mathbf{N}_k\in\mathbf{\Pi}} \mathbf{1}_{n\in\mathbf{N}_k}\Bigg(\sum_{i\in\mathbf{N}_k\setminus\{n\}} \mathcal{B}^{n\to i}\omega^i(\mathbf{N}_k|\boldsymbol{\rho}) \\ &\quad + \frac{\sum_{\widehat{\mathbf{N}}^k\subseteq\widehat{\mathbf{\Omega}}^k}\sum_{i\in\widehat{\mathbf{N}}^k} \mathcal{B}^{n\to i}\omega^i\big(\widehat{\mathbf{N}}^k_t \cup \mathbf{N}_k|\boldsymbol{\rho}\big)}{(N-N_k)^{N_{sh\_\max}+1-N_k}}\Bigg) \\ &\quad - \varphi^n\mathcal{E}^n(\mathbf{\Pi}).\end{aligned} \tag{29}$$

The proof of Proposition 3 is provided in Appendix E, in the supplementary material. This proposition presents the closed-form expressions of the value of peer P_n in any shard structure $\mathbf{\Pi} \in \mathbf{\Omega}$ given the beliefs $\mathcal{B}^n$ and reputations $\boldsymbol{\rho}$. Unfortunately, the computation of this value is intractable for large N, because the complexity of estimating $\mathcal{R}^n_{\mathcal{F}}$ and $\sum_{\widehat{\mathbf{N}}^k\subseteq\widehat{\mathbf{\Omega}}^k}\sum_{i\in\widehat{\mathbf{N}}^k} \mathcal{B}^{n\to i}\omega^i(\widehat{\mathbf{N}}^k_t \cup \mathbf{N}_k|\boldsymbol{\rho})$ is $\mathcal{O}(|\widehat{\mathbf{\Omega}}^k|) = \mathcal{O}(N!)$. Thus, in Corollary 3, we provide the tight (i.e., greatest) lower bound or infimum of the value $\mathcal{V}^n$ which can be computed in polynomial time.

Corollary 3: The value or expected payoff $\mathcal{V}^n(\mathbf{\Pi}|\mathcal{B}^n, \ell)$ of peer P_n in the shard structure $\mathbf{\Pi} = \{\mathbf{N}_1, \ldots, \mathbf{N}_K\} \in \mathbf{\Omega}$ given the belief $\mathcal{B}^n$ and reputations $\boldsymbol{\rho}$, is tightly bounded from below by the value

$$\begin{aligned}\widetilde{\mathcal{V}}^n\big(\mathbf{\Pi}|\mathcal{B}^n, \boldsymbol{\rho}\big) &= \inf \mathcal{V}^n\big(\mathbf{\Pi}|\mathcal{B}^n, \boldsymbol{\rho}\big) \\ &= \widetilde{\mathcal{R}}^n_{\mathcal{F}}(\mathbf{\Pi}) \sum_{\mathbf{N}_k\in\mathbf{\Pi}} \mathbf{1}_{n\in\mathbf{N}_k}\Bigg(\sum_{i\in\mathbf{N}_k\setminus\{n\}} \mathcal{B}^{n\to i}\omega^i(\mathbf{N}_k|\boldsymbol{\rho}) \\ &\quad + \big(N_{sh\_\max}+1-N_k\big)\mathcal{Z}(N_k)\min_{i\in\mathbf{N}\setminus\mathbf{N}_k} \\ &\quad \times \mathcal{B}^{n\to i}\omega^i\Big(\widehat{\mathbf{N}}^k_t \cup \mathbf{N}_k|\boldsymbol{\rho}\Big)\Bigg) - \varphi^n\widetilde{\mathcal{E}}^n(\mathbf{\Pi}).\end{aligned} \tag{30}$$

The proof of Corollary 3 is presented in Appendix F, in the supplementary material. This corollary establishes the greatest lower bound $\widetilde{\mathcal{V}}^n(\mathbf{\Pi}|\mathcal{B}^n, \boldsymbol{\rho})$ of the value of peer P_n in any shard structure $\mathbf{\Pi} \in \mathbf{\Omega}$ given the belief $\mathcal{B}^n$ and reputations $\boldsymbol{\rho}$. Unlike the exact value $\mathcal{V}^n$, this bound can be estimated in polynomial time, since the worst case time complexity of computing $\widetilde{\mathcal{R}}^n_{\mathcal{F}}$ and the sum

$$\begin{aligned}&\sum_{\mathbf{N}_k\in\mathbf{\Pi}} \mathbf{1}_{n\in\mathbf{N}_k}\Bigg(\sum_{i\in\mathbf{N}_k\setminus\{n\}} \mathcal{B}^{n\to i}\omega^i(\mathbf{N}_k|\boldsymbol{\rho}) + \big(N_{sh\_\max}+1-N_k\big) \\ &\quad \times \mathcal{Z}(N_k)\min_{i\in\mathbf{N}\setminus\mathbf{N}_k} \mathcal{B}^{n\to i}\omega^i\Big(\widehat{\mathbf{N}}^k_t \cup \mathbf{N}_k|\boldsymbol{\rho}\Big)\Bigg)\end{aligned}$$

is $\mathcal{O}(NK)$. Thus, to presume the tractability of computations in the BC-MEC, we assume that each peer P_n estimates its value $\mathcal{V}^n$ based on the greatest lower bounds in (30).

B. Reputation-Based Coalitional Game for Shard Formation

In order to enable a distributed implementation of the BC-MEC system, shards should be formed in a self-organized way, so that the peers can select their shards independently. In this section, we introduce a reputation-based coalition formation game to model the process of self-organized shard formation. In the game, the subsets of peers operating in the same shards are regarded as coalitions. The proposed game can be defined as follows.

Definition 1 (Reputation-Based Coalition Formation Game): A reputation-based coalition formation game is the game defined by the tuple $\Gamma = (\mathbf{N}, \mathbf{\Omega}, \rho^n, \mathcal{B}^n, \mathcal{V}^n)$ that comprises the following elements: 1) a set of peers or players $\mathbf{N}$; 2) a finite space $\mathbf{\Omega}$ of feasible coalitional structures defined in (1), where each structure $\mathbf{\Pi} = \{\mathbf{N}_1, \ldots, \mathbf{N}_K\} \in \mathbf{\Omega}$ is a partition of set $\mathbf{N}$ into $K \geq 1$ coalitions, with coalition $\mathbf{N}_k \in \mathbf{\Pi}$ representing a subset of players/peers managing shard k and, for each player/peer P_n; 3) the peer's individual reputation ρ^n defined in (4); 4) the peer's belief $\mathcal{B}^n$in faithfulness of other peers defined in (26); and 5) the peer's value $\mathcal{V}^n(\mathbf{\Pi}|\mathcal{B}^n, \boldsymbol{\rho})$ in the structure $\mathbf{\Pi} \in \mathbf{\Omega}$ given its belief $\mathcal{B}^n$ and reputations $\boldsymbol{\rho}$ of all peers defined by the lowest bound of its expected payoff in (30).

Recall that in the conventional (nonreputational) coalitional games, players care only about their own values, i.e., expected payoffs. Therefore, in any structure $\mathbf{\Pi}$, each player/peer P_n will select such coalition $\mathbf{N}_k \in \mathbf{\Pi}$ which can maximize the player's value $\mathcal{V}^n$ while not hurting the other members of this coalition [re]. As such, the goal of coalition formation in conventional games is to reach a stable structure, as defined as follows.

Definition 2 (Stable Structure): A structure $\mathbf{\Pi} \in \mathbf{\Omega}$ is a stable structure of game $\Gamma = (\mathbf{N}, \mathbf{\Omega}, \rho^n, \mathcal{B}^n, \mathcal{V}^n)$ if and only if there is no other structure $\acute{\mathbf{\Pi}} \in \mathbf{\Omega}\setminus\{\mathbf{\Pi}\}$, such that $\exists\acute{\mathbf{N}}_k \in \acute{\mathbf{\Pi}}, \exists n \in \acute{\mathbf{N}}_k$**:**

$$\begin{aligned}&\mathcal{V}^n\Big(\acute{\mathbf{\Pi}}|\mathcal{B}^n, \boldsymbol{\rho}\Big) > \mathcal{V}^n\big(\mathbf{\Pi}|\mathcal{B}^n, \boldsymbol{\rho}\big) \\ &\mathcal{V}^i\Big(\acute{\mathbf{\Pi}}|\mathcal{B}^i, \boldsymbol{\rho}\Big) \geq \mathcal{V}^i\big(\mathbf{\Pi}|\mathcal{B}^i, \boldsymbol{\rho}\big) \quad \forall i \in \acute{\mathbf{N}}_k\setminus\{n\}.\end{aligned} \tag{31}$$

In other words, a structure is stable if and only if there is no other structure that can increase the value of at least one player without reducing the values of other members of its coalition. Unlike conventional games, in the reputation-based coalitional game, coalitions are not only characterized by their values to different players but also by the coalitional reputations [42]. In particular, in our game, the reputation of coalition $\mathbf{N}_k \in \mathbf{\Pi}$ is represented by the expected reputation $\rho^{\mathbf{N}_k}$ of a corresponding shard k defined in (7). That is, the coalitional reputation $\rho^{\mathbf{N}_k}$ measures trustworthiness of the block confirmation process in shard k. Accordingly, in any structure $\mathbf{\Pi}$, each player/peer P_n is characterized not only by its individual reputation ρ^n but also by its coalitional reputation, given by

$$\widetilde{\rho}^n(\mathbf{\Pi}) = \sum_{\mathbf{N}_k\in\mathbf{\Pi}} \mathbf{1}_{n\in\mathbf{N}_k}\,\rho^{\mathbf{N}_k}. \tag{32}$$

Note that unlike the individual reputation ρ^n that is defined by opinion scores about the overall performance of peer P_n in the game (possibly, in different shards), coalitional reputation $\widetilde{\rho}^n$ depends on the reputations of all peers verifying task outputs in the shard managed by peer P_n. As a result, in any structure $\mathbf{\Pi}$, player/peer P_n will select such coalition $\mathbf{N}_k \in \mathbf{\Pi}$ which can maximize both the player's value $\mathcal{V}^n$ and coalitional reputation $\widetilde{\rho}^n$ without hurting other members of this coalition.

Hence, the goal of coalition formation in a reputation-based game is to reach a so-called "reputation-based" stable structure defined below.

Definition 3 (Reputation-Based Stable Structure): A structure $\boldsymbol{\Pi} \in \boldsymbol{\Omega}$ is the reputation-based stable structure of game $\Gamma = (\mathbf{N}, \boldsymbol{\Omega}, \rho^n, \mathcal{B}^n, \mathcal{V}^n)$ if and only if there is no other structure $\acute{\boldsymbol{\Pi}} \in \boldsymbol{\Omega}\backslash\{\boldsymbol{\Pi}\}$, such that $\exists \acute{\mathbf{N}}_k \in \acute{\boldsymbol{\Pi}}, \exists n \in \acute{\mathbf{N}}_k\ \forall i \in \acute{\mathbf{N}}_k\backslash\{n\}$:

$$\begin{gathered}\mathcal{V}^n\left(\acute{\boldsymbol{\Pi}}|\mathcal{B}^n, \boldsymbol{\rho}\right) > \mathcal{V}^n(\boldsymbol{\Pi}|\mathcal{B}^n, \boldsymbol{\rho}),\ \widetilde{\rho}^n\left(\acute{\boldsymbol{\Pi}}\right) = \rho^{\acute{\mathbf{N}}_k} \geq \widetilde{\rho}^n(\boldsymbol{\Pi}) \\ \mathcal{V}^i\left(\acute{\boldsymbol{\Pi}}|\mathcal{B}^i, \boldsymbol{\rho}\right) \geq \mathcal{V}^i(\boldsymbol{\Pi}|\mathcal{B}^i, \boldsymbol{\rho}) \text{ and } \widetilde{\rho}^i\left(\acute{\boldsymbol{\Pi}}\right) = \rho^{\acute{\mathbf{N}}_k} \geq \widetilde{\rho}^i(\boldsymbol{\Pi})\end{gathered} \tag{33a}$$

or

$$\begin{gathered}\mathcal{V}^n\left(\acute{\boldsymbol{\Pi}}|\mathcal{B}^n, \boldsymbol{\rho}\right) \geq \mathcal{V}^n(\boldsymbol{\Pi}|\mathcal{B}^n, \boldsymbol{\rho}), \widetilde{\rho}^n\left(\acute{\boldsymbol{\Pi}}\right) = \rho^{\acute{\mathbf{N}}_k} > \widetilde{\rho}^n(\boldsymbol{\Pi}) \\ \mathcal{V}^i\left(\acute{\boldsymbol{\Pi}}|\mathcal{B}^i, \boldsymbol{\rho}\right) \geq \mathcal{V}^i(\boldsymbol{\Pi}|\mathcal{B}^i, \boldsymbol{\rho}) \text{ and } \widetilde{\rho}^i\left(\acute{\boldsymbol{\Pi}}\right) = \rho^{\acute{\mathbf{N}}_k} \geq \widetilde{\rho}^i(\boldsymbol{\Pi}).\end{gathered} \tag{33b}$$

In other words, a structure is a reputation-based stable structure if and only if there is no other structure that can increase the value or coalitional reputation of at least one player without reducing values and coalitional reputations of other members of the player's coalition.

Proposition 4: Game $\Gamma = (\mathbf{N}, \boldsymbol{\Omega}, \rho^n, \mathcal{B}^n, \mathcal{V}^n)$ admits at least one reputation-based stable structure $\boldsymbol{\Pi} \in \boldsymbol{\Omega}$.

The proof of Proposition 4 is provided in Appendix G, in the supplementary material. From Proposition 4, there exists at least one reputation-based stable structure $\boldsymbol{\Pi} \in \boldsymbol{\Omega}$. Unfortunately, an exhaustive search of this structure is nondeterministic polynomial time (NP) complete because the computational complexity of this search is at least the complexity of the search of a stable structure, which is known to be an NP-complete problem [42]. Hence, in the next section, we propose a distributed coalition formation algorithm that allows all players/peers to reach a reputation-based stable structure $\overline{\boldsymbol{\Pi}}$ by following a simple protocol. If some player/peer P_n refuses to follow the protocol, its stake is withdrawn from the player, and the player is removed from the system.

C. Distributed Reputation-Based Coalition Formation

The proposed coalition formation algorithm is realized at every stage t of the mining process assuming that at any stage, there exists some shard structure $\boldsymbol{\Pi} \in \boldsymbol{\Omega}$. To track the changes in the structure $\boldsymbol{\Pi}$ and determine if this structure is stable, we utilize a variable $\Delta_{\boldsymbol{\Pi}}$ and a subset of "visited" players $\mathbf{N}_{\mathcal{V}} \subseteq \mathbf{N}$, where $\Delta_{\boldsymbol{\Pi}}$ shows the number of modifications in the structure from the moment when $\mathbf{N}_{\mathcal{V}} = \emptyset$. A variable and a subset are initialized as $\Delta_{\boldsymbol{\Pi}} \leftarrow 0$ and $\mathbf{N}_{\mathcal{V}} \leftarrow \emptyset$, respectively, at the start of the algorithm (i.e., at $t = 0$) and each time when all players in the set $\mathbf{N}$ are visited, i.e., if $\mathbf{N}_{\mathcal{V}} = \mathbf{N}$. Any player can propose the changes to the current structure $\boldsymbol{\Pi} = \{\mathbf{N}_1, \ldots, \mathbf{N}_K\} \in \boldsymbol{\Omega}$. In particular, at the beginning of every stage t, if $\mathbf{N}_{\mathcal{V}} = \mathbf{N}$ and $\Delta_{\boldsymbol{\Pi}} = 0$, i.e., all players in the set $\mathbf{N}$ are visited, but there are no changes in the structure from the moment when $\mathbf{N}_{\mathcal{V}} = \emptyset$, the algorithm terminates. Otherwise, if $\mathbf{N}_{\mathcal{V}} = \mathbf{N}$ and $\Delta_{\boldsymbol{\Pi}}>0$, we set $\Delta_{\boldsymbol{\Pi}} \leftarrow 0$ and $\mathbf{N}_{\mathcal{V}} \leftarrow \emptyset$, so that $\mathbf{N}\backslash\mathbf{N}_{\mathcal{V}} \neq \emptyset$. After this, we randomly choose one of nonvisited players from the subset $\mathbf{N}\backslash\mathbf{N}_{\mathcal{V}}$ to make a proposal to join a new coalition $\mathbf{N}_j \in \boldsymbol{\Pi}\backslash\{\mathbf{N}_k\}$, where $\mathbf{N}_k \in \boldsymbol{\Pi}$ is the current player's coalition. If player/peer P_n is chosen, it is labeled as "visited" and the subset $\mathbf{N}_{\mathcal{V}}$ is updated as $\mathbf{N}_{\mathcal{V}} \leftarrow \mathbf{N}_{\mathcal{V}} \cup \{n\}$.

If chosen, player/peer P_n can only select such coalition $\mathbf{N}_j$ that can maximize both the player's value $\mathcal{V}^n$ and coalitional reputation $\widetilde{\rho}^n$ without hurting other members of coalition $\mathbf{N}_j$. The player determines this coalition by solving the following bicriteria optimization problem:

$$\underset{\mathbf{N}_j \in \boldsymbol{\Pi}^n_{k(f)}}{\arg\max} \left(\mathcal{V}^n\left(\boldsymbol{\Pi}^n_{k\to j}|\mathcal{B}^n, \boldsymbol{\rho}\right), \widetilde{\rho}^n\left(\boldsymbol{\Pi}^n_{k\to j}\right)\right) \tag{34}$$

where $\boldsymbol{\Pi}^n_{k\to j} \in \boldsymbol{\Omega}$ is the structure derived from the current one by moving peer P_n from its coalition $\mathbf{N}_k \in \boldsymbol{\Pi}$ into coalition $\mathbf{N}_j \in \boldsymbol{\Pi}$, given by

$$\boldsymbol{\Pi}^n_{k\to j} = \left\{\acute{\mathbf{N}}_r \;\middle|\; \acute{\mathbf{N}}_k = \mathbf{N}_k\backslash\{n\}, \acute{\mathbf{N}}_j = \mathbf{N}_j \cup \{n\}, \acute{\mathbf{N}}_i = \mathbf{N}_i \quad \forall i \notin \{k, j\}\right\} \tag{35}$$

$\boldsymbol{\Pi}^n_{k(f)} \subseteq \boldsymbol{\Pi}$ is the subset of feasible coalitions in the structure $\boldsymbol{\Pi}$, i.e., coalitions that can accept peer P_n without reducing the values and coalitional reputations of their members, given by

$$\boldsymbol{\Pi}^n_{k(f)} = \left\{\begin{matrix}\mathbf{N}_j \in \\ \boldsymbol{\Pi}\backslash\{\mathbf{N}_k\}\end{matrix} \;\middle|\; \begin{matrix}\widetilde{\rho}^i\left(\boldsymbol{\Pi}^n_{k\to j}\right) = \rho^{\mathbf{N}_j\cup\{n\}} \\ \geq \widetilde{\rho}^i(\boldsymbol{\Pi}) = \rho^{\mathbf{N}_j}, \\ \mathcal{V}^i\left(\boldsymbol{\Pi}^n_{k\to j} \;\middle|\; \mathcal{B}^i, \boldsymbol{\rho}\right) \geq \mathcal{V}^i\left(\boldsymbol{\Pi} \mid \mathcal{B}^i, \boldsymbol{\rho}\right) \\ \forall i \in \mathbf{N}_j\end{matrix}\right\}. \tag{36}$$

If $\boldsymbol{\Pi}^n_{k(f)} \neq \emptyset$, there is at least one coalition $\mathbf{N}_j \in \boldsymbol{\Pi}\backslash\{\mathbf{N}_k\}$ that satisfies (34). In this case, peer P_n joins coalition $\mathbf{N}_j$ if either

$$\begin{gathered}\mathcal{V}^n\left(\boldsymbol{\Pi}^n_{k\to j} \;\middle|\; \mathcal{B}^n, \boldsymbol{\rho}\right) > \mathcal{V}^n\left(\boldsymbol{\Pi} \mid \mathcal{B}^n, \boldsymbol{\rho}\right) \\ \widetilde{\rho}^i\left(\boldsymbol{\Pi}^n_{k\to j}\right) \geq \widetilde{\rho}^i(\boldsymbol{\Pi})\end{gathered} \tag{37}$$

or

$$\begin{gathered}\mathcal{V}^n\left(\boldsymbol{\Pi}^n_{k\to j} \;\middle|\; \mathcal{B}^n, \boldsymbol{\rho}\right) \geq \mathcal{V}^n\left(\boldsymbol{\Pi} \mid \mathcal{B}^n, \boldsymbol{\rho}\right) \\ \widetilde{\rho}^i\left(\boldsymbol{\Pi}^n_{k\to j}\right) > \widetilde{\rho}^i(\boldsymbol{\Pi}).\end{gathered} \tag{38}$$

If there are several coalitions which satisfy (34), peer P_n selects among them randomly. Otherwise, if $\boldsymbol{\Pi}^n_{k(f)} = \emptyset$, or

$$\begin{gathered}\mathcal{V}^n\left(\boldsymbol{\Pi}^n_{k\to j} \;\middle|\; \mathcal{B}^n, \boldsymbol{\rho}\right) \leq \mathcal{V}^n\left(\boldsymbol{\Pi} \mid \mathcal{B}^n, \boldsymbol{\rho}\right) \\ \widetilde{\rho}^i\left(\boldsymbol{\Pi}^n_{k\to j}\right) \leq \widetilde{\rho}^i(\boldsymbol{\Pi})\end{gathered} \tag{39}$$

peer P_n remains in its old coalition $\mathbf{N}_k$. If peer P_n moves into coalition $\mathbf{N}_j$, a modified structure $\boldsymbol{\Pi} \leftarrow \boldsymbol{\Pi}^n_{k\to j}$ is formed at the end of stage t, and a variable $\Delta_{\boldsymbol{\Pi}}$ is updated as $\Delta_{\boldsymbol{\Pi}} \leftarrow \Delta_{\boldsymbol{\Pi}}+1$. Otherwise, if peer P_n remains in coalition $\mathbf{N}_k$, the structure $\boldsymbol{\Pi}$ does not change.

The above algorithm is repeated at each mining stage t until it converges to some final stable structure $\boldsymbol{\Pi} \in \boldsymbol{\Omega}$. Note that the structure $\boldsymbol{\Pi} \in \boldsymbol{\Omega}$ is considered stable if it remains unchanged after all players in the set $\mathbf{N}$ are visited, i.e., if $\mathbf{N}_{\mathcal{V}} = \mathbf{N}$

and $\Delta_{\mathbf{\Pi}} = 0$. In other words, the algorithm terminates if the structure does not change after N stages from the moment when $\mathbf{N}_{\mathcal{V}} = \emptyset$. Proposition 5 below verifies that this algorithm converges to a reputation-based stable structure.

Proposition 5: With probability one, the proposed coalition formation algorithm converges to the reputation-based stable structure $\mathbf{\Pi} \in \mathbf{\Omega}$ of game $\Gamma = (\mathbf{N}, \mathbf{\Omega}, \rho^n, \mathcal{B}^n, \mathcal{V}^n)$.

The proof of Proposition 5 is provided in Appendix H, in the supplementary material. Next, Proposition 6 below establishes the computational complexity and convergence rate of the proposed algorithm.

Proposition 6: The proposed coalition formation algorithm has the worst case time complexity of $\mathcal{O}(N^3)$ and convergence rate is equal to $\|\mathbf{x}_{i+1} - \overline{\mathbf{x}}\| = \mathcal{O}(\|\mathbf{x}_i - \overline{\mathbf{x}}\|)$, where $\mathbf{x}_i = (\mathcal{V}_i, \widetilde{\rho}_i) = (\{\mathcal{V}_i^n\}_{n\in\mathbf{N}}, \{\widetilde{\rho}_i^n\}_{n\in\mathbf{N}})$ are the players' values and coalitional reputations after every ith algorithm iteration; $\overline{\mathbf{x}} = (\overline{\mathcal{V}}, \overline{\widetilde{\rho}}) = (\{\overline{\mathcal{V}}^n\}_{n\in\mathbf{N}}, \{\overline{\widetilde{\rho}}^n\}_{n\in\mathbf{N}})$ are the values and coalitional reputations of players in a reputation-based stable structure $\mathbf{\Pi} \in \mathbf{\Omega}$ of game $\Gamma = (\mathbf{N}, \mathbf{\Omega}, \rho^n, \mathcal{B}^n, \mathcal{V}^n)$.

The proof of Proposition 6 is provided in Appendix I, in the supplementary material. From Proposition 6, the proposed algorithm has a polynomial worst case complexity and a near-linear rate of convergence.

D. Theoretical Performance of Self-Organized Sharding

We now conduct the theoretical analysis of the performance of the proposed self-organized sharding model in terms of the system security and throughput, i.e., the number of transactions validated per time unit. We start by analyzing the local shard security and global blockchain security of our sharding model. In particular, based on common blockchain security definitions (e.g., [3] and [10]), we say that the for some given number $N_{\mathcal{M}}$ of malicious peers in the shard, the local shard security is ensured if both of the following conditions are satisfied:

1) all correct task outputs are accepted by the shard's members and appended to a local shard's subchain;
2) all incorrect outputs are rejected by the shard's members.

When the number of malicious peers in the shard does not exceed $N_{\mathcal{M}}$. Similarly, given up to $N_{\mathcal{M}}$ malicious peers in the sharding system, the global blockchain security is guaranteed if both of the following conditions are satisfied:

1) all correct task outputs are accepted by the shard's members and randomly assigned peers, and appended to the global blockchain;
2) all incorrect outputs are rejected by the shard's members and randomly assigned peers.

Proposition 7: For any $w_{\mathcal{C}} > 0$ and $\phi > 0.5$, given that the total number of malicious peers in the BC-MEC system does not exceed $\lfloor N/2 \rfloor$ or 50% of peers, a self-organized sharding model provides local shard security for up to $\lfloor N_k/2 \rfloor$ malicious peers in each shard k and global blockchain security for up to $N_k - \lceil (N_{sh\_\max} + 1)/2 \rceil$ malicious peers in every shard k.

The proof of Proposition 7 is provided in Appendix J, in the supplementary material. From Proposition 7, we obtain Corollary 4 that provides the optimal value of the maximal number of peers per shard, $N_{sh\_\max}$, i.e., the value that guarantees security against the maximal number of malicious peers.

Corollary 4: For any $w_{\mathcal{C}} > 0$ and $\phi > 0.5$, given that the total number of malicious peers in the BC-MEC system does not exceed $\lfloor N/2 \rfloor$ or 50% of peers, in a self-organized sharding model, the global blockchain security is optimal for $N_{sh\_\max} = \max_{\mathbf{N}_k \in \mathbf{\Pi}} N_k$. In this case, the model ensures protection against the maximal number of malicious peers—it tolerate up to

$$N - K\left\lceil \left(\max_{\mathbf{N}_k\in\mathbf{\Pi}} N_k + 1\right)/2 \right\rceil = \begin{cases} \left\lfloor \frac{N-K}{2} \right\rfloor, & \max\limits_{\mathbf{N}_k\in\mathbf{\Pi}} N_k = \min\limits_{\mathbf{N}_k\in\mathbf{\Pi}} N_k = N/K \\ \left\lfloor \frac{N}{2}\left(1 - \frac{\min\limits_{\mathbf{N}_k\in\mathbf{\Pi}} N_k}{\max\limits_{\mathbf{N}_k\in\mathbf{\Pi}} N_k}\right) \right\rfloor, & \max\limits_{\mathbf{N}_k\in\mathbf{\Pi}} N_k > \min\limits_{\mathbf{N}_k\in\mathbf{\Pi}} N_k \end{cases}$$

malicious peers in the entire BC-MEC system.

The proof of Corollary 4 is provided in Appendix K, in the supplementary material. From Corollary 4, the global blockchain security of our proposed self-organized sharding model with $w_{\mathcal{C}} > 0$ and $\phi > 0.5$ can be optimized if the maximal number of peers in every shard is adjusted dynamically, i.e., based on the current shard structure, as $N_{sh\_\max} = \max_{\mathbf{N}_k\in\mathbf{\Pi}} N_k$. In this case, the entire BC-MEC system tolerates up to $\lfloor (N-K)/2 \rfloor$ malicious peers if all the formed shards have the same size $\max_{\mathbf{N}_k\in\mathbf{\Pi}} N_k = \min_{\mathbf{N}_k\in\mathbf{\Pi}} N_k = N/K$, and up to $\lfloor N(1 - \min_{\mathbf{N}_k\in\mathbf{\Pi}} N_k / \max_{\mathbf{N}_k\in\mathbf{\Pi}} N_k)/2 \rfloor$ malicious peers if the shards have different sizes, i.e., $\max_{\mathbf{N}_k\in\mathbf{\Pi}} N_k > \min_{\mathbf{N}_k\in\mathbf{\Pi}} N_k$. Moreover, when all shards have the same size, the global blockchain security improves when the number of shards reduces. In particular, the system tolerates:

1) up to $\lfloor N/3 \rfloor$ malicious peers (or 33% of all peers) for $K = N/3$ (i.e., 3 peers in each formed shard);
2) up to $\lfloor 9N/20 \rfloor$ malicious peers (or 45% of all peers) for $K = N/10$ (i.e., 10 peers in each shard);
3) up to $\lfloor (N-1)/2 \rfloor$ malicious peers (or slightly less than 50% of all peers) for $K = 1$ (i.e., all peers form one shard).

When the size of the shards in the system varies, the global blockchain security improves when the ratio $\min_{\mathbf{N}_k\in\mathbf{\Pi}} N_k / \max_{\mathbf{N}_k\in\mathbf{\Pi}} N_k$ of the minimal to the maximal shard sizes reduces. That is, the system tolerates:

1) up to $\lfloor N/3 \rfloor$ or 33% malicious peers for $\min_{\mathbf{N}_k\in\mathbf{\Pi}} N_k / \max_{\mathbf{N}_k\in\mathbf{\Pi}} N_k = 1/3$;
2) up to $\lfloor 9N/20 \rfloor$ or 45% malicious peers for $\min_{\mathbf{N}_k\in\mathbf{\Pi}} N_k / \max_{\mathbf{N}_k\in\mathbf{\Pi}} N_k = 1/10$;
3) up to $\lfloor (N-1)/2 \rfloor$ or slightly less than 50% malicious peers for $\min_{\mathbf{N}_k\in\mathbf{\Pi}} N_k / \max_{\mathbf{N}_k\in\mathbf{\Pi}} N_k = 1/N$ (i.e., all peers form one shard).

Accordingly, our self-organized sharding model can provide enhanced security comparing with lightweight blockchains and other sharding techniques. In particular, with the exception of PolyShard (Lagrange polynomial-based coded sharding), most sharding schemes, e.g., OmniLedger [12], static domain-based sharding [13], and Blockclique [14], tolerate only one adversarial peer in each shard, i.e., up to K malicious nodes in the system. PolyShard tolerates up to $\lfloor (N-K)/2K \rfloor$ adversaries per shard and $\lfloor (N-K)/2 \rfloor$ malicious peers in the system [16] which is comparable to security guarantees of our model. On the other hand, the lightweight blockchains, including reputation-based systems utilizing PoA [29], [30] or pBFT [5], [19], can tolerate only up to $\lfloor N/3 \rfloor$ or 33% adversaries in best-case scenarios.

Next, in Proposition 8, we show that after converging to a reputation-based stable structure $\mathbf{\Pi} \in \mathbf{\Omega}$, the proposed self-organized sharding model can achieve the maximal blockchain throughput subject to the security guarantees in Proposition 7 and Corollary 4.

Proposition 8: After converging to a reputation-based stable structure $\mathbf{\Pi} \in \mathbf{\Omega}$, a self-organized sharding model achieves the maximal system throughput. In particular

$$\sum_{n\in\mathbf{N}} \mathcal{T}^n\left(\mathbf{\Pi} \mid \mathcal{B}^n, \boldsymbol{\rho}\right) \geq \sum_{n\in\mathbf{N}} \mathcal{T}^n\left(\acute{\mathbf{\Pi}} \,\middle|\, \mathcal{B}^n, \boldsymbol{\rho}\right) \forall \acute{\mathbf{\Pi}} \in \mathbf{\Omega}\backslash\{\mathbf{\Pi}\}$$

i.e., the aggregated throughput of peers in any shard structure $\acute{\mathbf{\Pi}} \in \mathbf{\Omega}\backslash\{\mathbf{\Pi}\}$ different from $\mathbf{\Pi}$ is less than or equal to that in the reputation-based stable structure $\mathbf{\Pi}$.

The proof of Proposition 8 is provided in Appendix L, in the supplementary material. From Proposition 8, there is no shard structure that can achieve the higher blockchain throughput than the reputation-based stable structure. Consequently, after converging to a reputation-based stable structure, our proposed sharding model can maximize the system throughput without compromising on security, i.e., subject to security guarantees established in Proposition 7 and Corollary 4. As a result, based on our analytical evaluation, this model is able to improve performance of existing sharding schemes in terms of both the throughput and the security (note that the storage efficiency of our model is the same as in other sharding schemes). To verify the theoretical claims of this work, in the next section, we provide a numerical evaluation of the performance of the proposed sharding model.

VI. Numerical Performance Evaluation

A. Simulation Model and Settings

A simulation model of the BC-MEC with sharding has been developed by using the OPNET package [55]. Comparing with other popular network simulators, such as Omnet [56] or NS2 [57], OPNET provides a more realistic and reliable simulation environment in all network types, as well as pre-built models of all standard wireless and wireline protocols, and network devices (e.g., servers, BSs, routers, switches, IoT devices, etc.) with easily reprogrammable interfaces [58]. The MEC network model has been implemented upon the long-term evolution—advanced (LTE-A) time division duplex (TDD) platform [59]. The MEC network comprises $M = 3$ BSs represented by macrocell LTE-A evolved Node Bs (eNBs) placed as shown in Fig. 2, each of which operates on the separate spectrum with the bandwidth 20 MHz. The macrocell eNBs are interconnected by the ITU-T G.657 brand B fiber (designated for the use in access networks, such as our LTE-A based MEC, in proximity to end-users [60]) with the capacity 20 wavelengths. The number of wavelengths is selected to support dedicated simultaneously fast forwarding of task outputs from up to 20 peers (one wavelength to forward the output of each peer). The peers are represented by small-cell eNBs placed randomly in the service area of the MEC network. Each peer P_n is allotted with the bandwidth $B_n = 2.5$ MHz that can be used by the peer to communicate with its IoT devices, the macrocell eNBs, and the other peers. Each peer serves 100 IoT devices represented by the typical temperature, smoke, image, and motion detector sensors simulated with the common parameters (listed, e.g., in [61]). The number of IoT devices connected to a peer is chosen to comply with bandwidth requirements of a typical IoT device (described, e.g., in [61]). To simplify our further performance evaluation, the IoT devices connected to the same peer are assumed to be associated with a single BC-MEC customer. All parameters related to the LTE-A model are set in accordance with the LTE-A 3rd Generation Partnership Project (3GPP) specifications [59]. For example, the values of transmit powers and cell radiuses of eNBs are the same as the standard values defined in [59], i.e., 42 dBm and 3000 m (for a macrocell eNB) and 23 dBm and 500 m (for a small-cell eNB). The wireless channel parameters, e.g., antenna gain, path loss, and noise and shadowing, are also based on the 3GPP specifications [59].

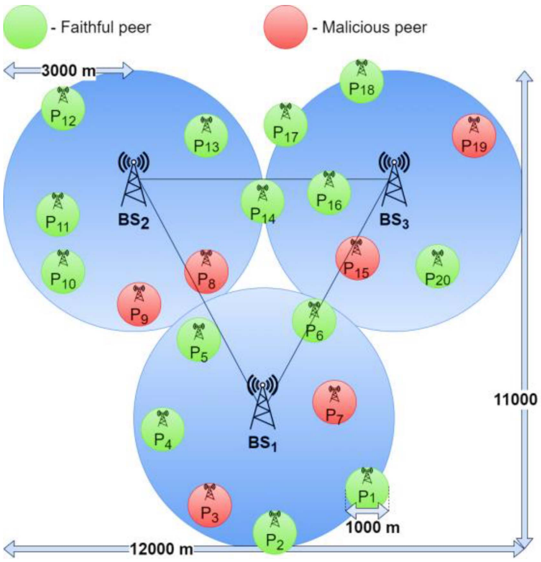

Fig. 2. Simulation model of the BC-MEC with three BSs, i.e., macrocell eNBs, and 20 peers, i.e., small-cell eNBs. The locations of BSs are fixed during all simulation runs, as shown in the figure. All numerical results are collected and averaged for three simulation runs with the randomized locations of peers. In particular, at the beginning of each simulation run, a peer is placed randomly inside the service area of the MEC network. The peer's location remains fixed during one simulation run, but can be changed at the beginning of the next run.

Similar to prior works on mobile/IoT blockchains (e.g., [49], [62] and [63]), we assume that: 1) the parameters of the mining tasks recorded by peers follow a Poisson distribution with the means $\theta = (\theta^P, \theta^O, \theta^V) =$ (1-G CPU cycle [Gc], 1 kb, 0.5 Gc) and 2) task interarrival times are distributed exponentially with the mean of 5 min. As such, the expected block interval time or stage duration (which is equal to the mean task interarrival time) is $\Delta t = 5$ min. Each peer P_n, i.e., small-cell eNB, is equipped with 7150N Dual Core Xeon server processor [64] of the computing power $x^n = 3.5$ Gc/s, 16-MB L2 Cache and the power consumption 150 W, i.e., the energy spent per CPU cycle by the peer is $\vartheta^n = 150/(3.5 \times 10^9) = 42.8$ nJ. All the payments made in the system are counted in units of a digital currency, i.e., bitcoin. Assuming that the peer pays a minimal possible cost, i.e., 1 bitcoin, for each kJ of energy, the cost per energy unit of peer P_n is set as $\varphi^n = 10^{-3}$. Next, noting that $r^n > \varphi^n \Delta t(\vartheta^n x^n + p^n \theta^O)$, i.e., the block reward r^n of peer P_n must be significantly higher than the expected energy costs $\varphi^n \Delta t(\vartheta^n x^n + p^n \theta^O)$ of the peer (otherwise, the peer has no incentives to participate in the block mining), the peer's reward is set equal to $r^n = 500 > \varphi^n \Delta t(\vartheta^n x^n + p^n \theta^O) = 105$. All reputation-related parameters are similar to those in the prior reputation-based systems, e.g., [5] and [19]. In particular, we set $\phi = 0.6$ and $w_{\mathcal{C}} = 0.5$; uncertainties $u^{i \to n}$, $i \in \{0\} \cup \mathbf{N} \backslash \{n\}$, are modeled according to a standard normal distribution; weights $w^n_{\mathcal{C}}$ and $w^n_{\mathcal{O}}$ follow Poisson distributions with means 0.5.

By default, the total number of peers is $N = 20$, the number of malicious peers is $N_{\mathcal{M}} = 6$ or 30% of all peers; the maximal number of peers per shard is $N_{sh\_\max} = \max_{\mathbf{N}_k \in \mathbf{\Pi}} N_k$, i.e., equal to the optimal number (see Corollary 4). All simulation results are collected and averaged for three simulation runs with the randomized peers' locations. At the beginning of a simulation run, each peer is placed randomly inside the service area of the MEC network. The peer's location remains fixed during one simulation run but can be changed at the beginning of the next run. The remainder of this section is organized as follows. In Section VI-B, we analyze the performance of our self-organized shard formation model to validate theoretical claims in Propositions 6–8 and Corollary 4. In Section VI-C, we compare our model with state-of-the-art sharding schemes and a reputation-based lightweight blockchain.

B. *Performance of the Self-Organized Sharding Model*

Recall that the goal of our work is to enhance the overall throughput and security of the BC-MEC system through self-organized reputation-based shard formation. Therefore, in the following, we evaluate our proposed sharding model in terms of its throughput, i.e., the number of transactions verified per time unit, and security, i.e., the number of malicious peers the system can tolerate. For this, let us, first, analyze the performance of our self-organized shard/ coalition formation algorithm during the first 50 stages starting from $t = 0$ and until convergence to a reputation-based stable structure $\mathbf{\Pi} \in \mathbf{\Omega}$ at $t = 50$. The main objectives here is to show that each consecutive shard structure formed as a result of the algorithm is better (or, at least, not worse) that the previous one in terms of the peers values, i.e., expected payoffs, and coalitional reputations which measure the trustworthiness of block confirmations inside formed shards. As such, we aim to: 1) show that the peers' values (and, hence, total blockchain throughput) and coalitional reputations (and, therefore, system security) improve consistently during self-organized shard formation until reaching the stable levels and 2) validate theoretical claims in Propositions 6–8 and Corollary 4.

Figs. 3–9 show the results of simulations with $N_{\mathcal{M}} = 2$, $N_{\mathcal{M}} = 4$, $N_{\mathcal{M}} = 6$, $N_{\mathcal{M}} = 8$, and $N_{\mathcal{M}} = 9$ or, respectively, 10%, 20%, 30%, 40%, and 45% malicious peers, each of which generates arbitrarily erroneous results, reject correct outputs and confirms incorrect outputs. Initially, at $t = 0$, all shards are singletons, i.e., each managed by one peer. First, to verify that the values and coalitional reputations of peers improve consistently until reaching the stable levels, in Figs. 3 and 4, we present the average peer's value and coalitional reputation during the first 50 stages of the algorithm. We observe that the values and coalitional reputations increase from $t = 0$ to $t = 40$ stages and stabilize after $t = 40$ stages. The reason is that a peer moves to another shard/coalition only if it can maximize both the peer's value and coalitional reputation

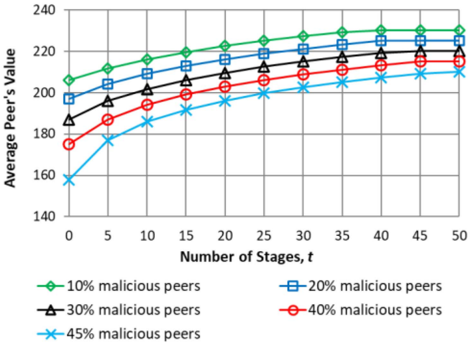

Fig. 3. Average peer's value during shard/coalition formation starting from $t = 0$ until convergence to a reputation-based stable structure at $t = 50$ for the varying number of malicious peers.

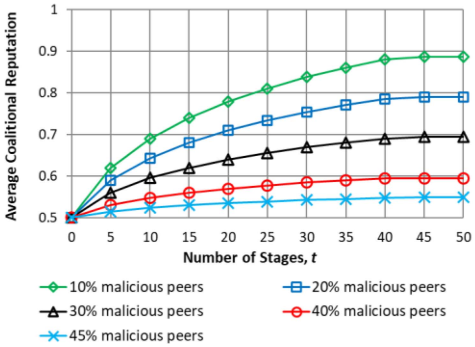

Fig. 4. Average coalitional reputation during shard/coalition formation starting from $t = 0$ until convergence to a reputation-based stable structure at $t = 50$ for the varying percent of malicious peers.

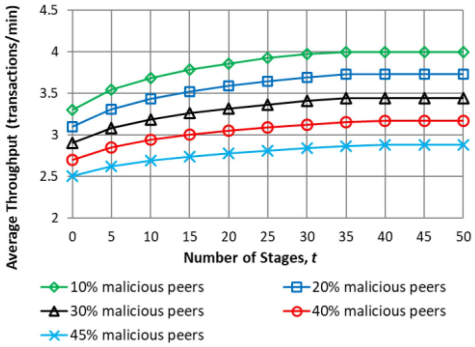

Fig. 5. Average system throughput (transactions/min) during shard/coalition formation starting from $t = 0$ until convergence to a reputation-based stable structure at $t = 50$ for the varying percent of malicious peers.

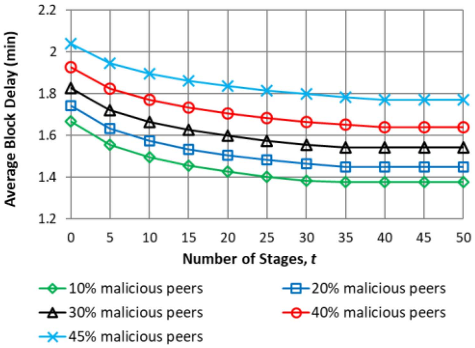

Fig. 6. Average block delay (min) during shard/coalition formation starting from $t = 0$ until convergence to a reputation-based stable structure at $t = 50$ for the varying percent of malicious peers.

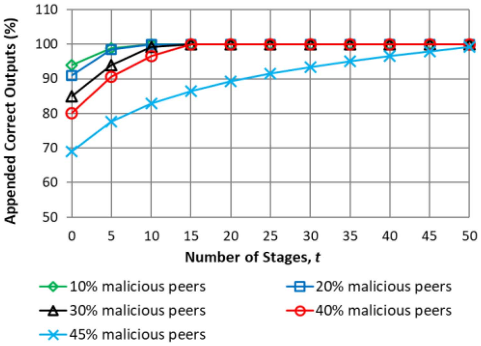

Fig. 7. Percent of appended correct outputs during shard/coalition formation starting from $t = 0$ until convergence to a reputation-based stable structure at $t = 50$ for the varying percent of malicious peers.

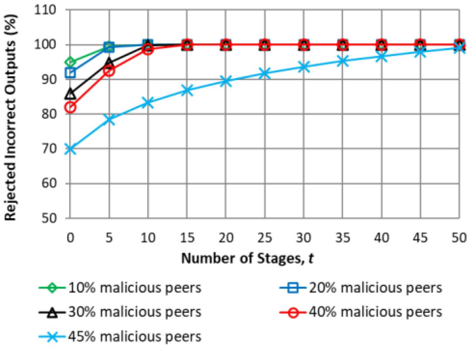

Fig. 8. Percent of rejected incorrect outputs during shard/coalition formation starting from $t = 0$ until convergence to a reputation-based stable structure at $t = 50$ for the varying percent of malicious peers.

without hurting other shard members, i.e., without reducing their values and coalitional reputations. Accordingly, prior to convergence to a reputation-based stable structure $\mathbf{\Pi}$, each consecutive shard structure formed as a result of the algorithm will be better than the previous one in terms of value and/or coalitional reputation of at least one peer. On the other hand, after convergence, the peers stop forming new shards and, therefore, their values and coalitional reputations reach stable levels, which concurs with the findings of Proposition 6. We also observe that the peers' values and coalitional reputations reduce with the number of malicious peers $N_{\mathcal{F}}$. The reason is that when the number of malicious peers is at most 50%, the

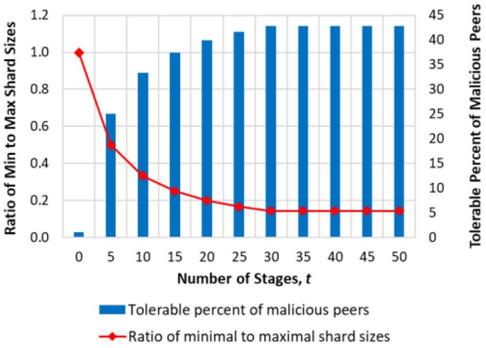

Fig. 9. Ratio of minimal to maximal shard sizes and theoretical security, i.e., percent of tolerable malicious peers, during shard/coalition formation starting from $t = 0$ until convergence to a reputation-based stable structure at $t = 50$.

reputation of a faithful peer in our system is always higher than that of a malicious peer (see the proof of Proposition 7 given in Appendix J, in the supplementary material). Therefore, the average reputations of peers in shards reduce with the growing percent of malicious peers. In its turn, the peer's value depends on the peer's beliefs about other members of its shard defined by the reputations of the shard's members [see (26) and (29)]. Accordingly, the peer's value decreases when the average reputation of the members of its shard reduces.

Next, to analyze the throughput improvements during self-organized shard formation, in Figs. 5 and 6, we present the average throughput, i.e., the number of outputs or transactions verified per time unit, and average block delay, i.e., expected delay per task of each peer, during the first 50 stages of the algorithm. We observe that the values of throughput increase, whereas, the value of the block delay decrease from $t = 0$ to $t = 40$ stages and stabilize after $t = 40$ stages. The reason is that in any shard/coalition, the peer's value is mainly defined by the peer's throughput [see (28)]. Hence, when moving to a shard that maximizes its value, the peer is able to increase its throughput without hurting other shard's members. Furthermore, from (24a), the throughput is determined by the orphaning probability which depends on the expected delay for the peer's task. That is, when moving to a coalition that increases its throughput, the peer can reduce the delay for its tasks. As such, initially, i.e., at $t = 0$, each shard in the structure is a singleton. That is, the output of a task in the shard is verified by the randomly assigned peers outside of the shard. Any of these randomly assigned peers can be malicious and/or located far from the peer that has produced the output, which leads to reduced throughput and increased block delay in the initial shard structure. Nonetheless, each consecutive shard structure that will be formed at $t = 1, 2, \ldots,$ as a result of self-organized shard formation will be better (or, at least, not worse) than the previous one in terms of the throughput and block delay, which concurs with the claims of Proposition 8.

Furthermore, to study the security bounds of our algorithm, in Figs. 7 and 8, we present the percent of appended correct outputs and percent of rejected incorrect outputs during self-organized shard formation. We note that with time, the percent of appended correct outputs and rejected incorrect

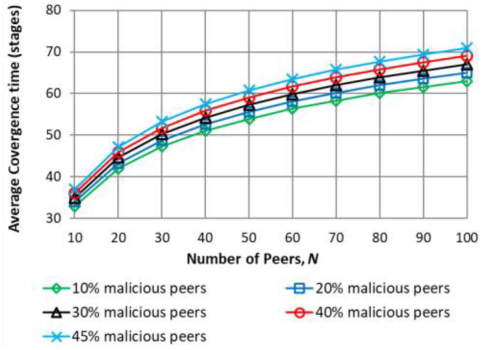

Fig. 10. Average convergence time (stages) as a function of the total number of peers for the varying percent of malicious peers in the system.

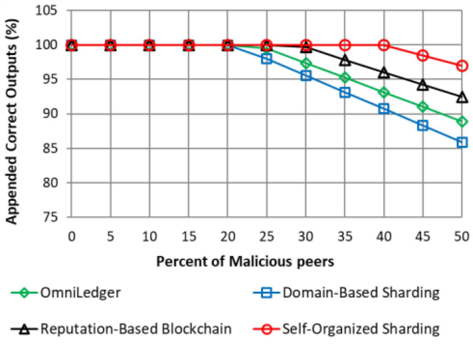

Fig. 11. Percent of appended correct outputs in different algorithms depending on the percent of malicious peers for the average number of shards $K = 5$.

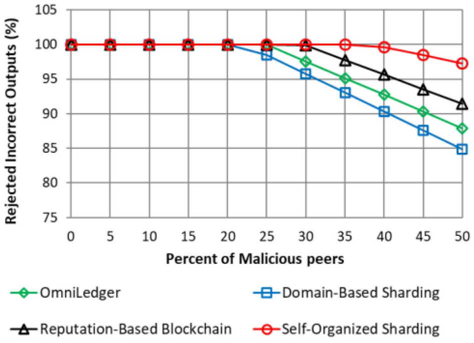

Fig. 12. Percent of rejected incorrect outputs in different algorithms depending on the percent of malicious peers for the average number of shards $K = 5$.

outputs increases reaching 100% after $t = 15$ stages (with up to 40% malicious peers) or $t = 50$ stages (with 45% malicious peers), which means that security improves until it reaches its optimal level. To understand such a performance, note that initially, i.e., at after $t = 0$, each shard is a singleton, i.e., the output of a task in the shard is verified only by the randomly assigned peers outside of the shard. Since any of the randomly assigned peers can be malicious, some of the correct outputs maybe not confirmed or, alternatively, some incorrect outputs maybe accepted. However, with a self-organized sharding, each peer will eventually move to a shard/coalition which maximizes its coalitional reputation (which depends on reputations of all the shard's members). In this case, the percent of malicious peers verifying the outputs in each shard can be minimized. Hence, every consecutive shard structure is better (or, at least, not worse) than the previous one in terms of shards' security.

To show that the results in Figs. 7 and 8 concur with the findings of Corollary 4, in Fig. 9, we present the ratio of the minimal to the maximal shard sizes $\min_{\mathbf{N}_k \in \mathbf{\Pi}} N_k \ / \max_{\mathbf{N}_k \in \mathbf{\Pi}} N_k$ during self-organized shard formation and the corresponding tolerable percent of malicious peers $\lfloor (1 - \min_{\mathbf{N}_k \in \mathbf{\Pi}} N_k / \max_{\mathbf{N}_k \in \mathbf{\Pi}} N_k)/2 \rfloor \times 100\%$ computed according to Corollary 4. From Fig. 9, at $t = 0$, only one malicious peer can be tolerated; at $t = 5$, up to 25% malicious peers are tolerated; at $t = 10$, up to 33% malicious peers are tolerated; at $t = 15$, up to 38% malicious peers are tolerated; at $t = 20$, up to 40% malicious peers are tolerated; at $t = 25$, up to 42% malicious peers are tolerated; after $t = 30$, up to 43% malicious peers can be tolerated, which supports numerical results in Figs. 7 and 8. Thus, after convergence, the proposed self-organized shard formation algorithm is able to enhance the throughput and security of our system. To study the algorithm convergence, in Fig. 10, we show the average convergence time for the increasing total number of peers N, with 10%, 20%, 30%, 40%, and 45% malicious peers. From Fig. 10, the algorithm converges relatively fast regardless of the percent of malicious peers. The algorithm convergence time is logarithmic $\mathcal{O}(\log N)$ with respect to the number of peers N.

C. Comparison With Other Related Algorithms

Let us now evaluate the performance of our self-organized sharding model after stabilization, i.e., after convergence to a reputation-based stable structure $\mathbf{\Pi} \in \mathbf{\Omega}$, by comparing it with the performance of the state-of-the-art sharding methods and a reputation-based lightweight blockchain system (described in Section II) applicable to our BC-MEC system: 1) OmniLedger [12] based on bias-resistant distributed randomness generation for sampling and updating subsets of peers managing shards to preserve blockchain security; 2) domain-based sharding [13] where the validators are selected based on the results of PoW competition, which enables changing the set of validators to improve security; and 3) reputation-based lightweight blockchain for vehicular IoT [5] where the validators are selected from the peers with the highest reputations. In order to be comparable with our sharding model where each task output in the shard is confirmed by $N_{sh\_\max}$ shard's peers, the number of shards in OmniLedger and domain-based sharding, and the number of validators in a reputation-based blockchain is adjusted as $K = N/N_{sh\_\max}$, so that every validator verifies $N_{sh\_\max}$ outputs, as in our sharding model. The main objective here is to show that our sharding model can outperform other simulated algorithms in terms of system security and blockchain throughput.

First, we compare the performance of our sharding scheme with the performance of other algorithms in terms of security, i.e., the tolerable number of malicious peers. For this, in Figs. 11 and 12, we show the percent of appended correct outputs and percent of rejected incorrect outputs in simulations with the increasing percent of malicious peers and the average

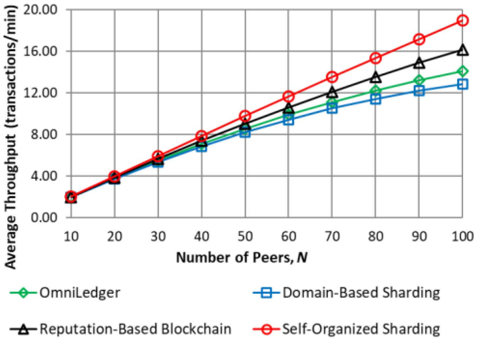

Fig. 13. Average system throughput (transactions/min) in different algorithms depending on the total number of peers for the percent of malicious peers fixed to 30%.

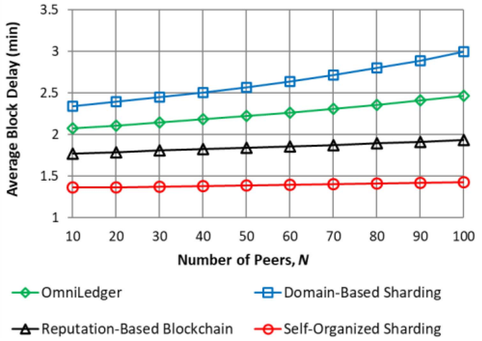

Fig. 14. Average block delay (min) in different algorithms depending on the total number of peers for the percent of malicious peers fixed to 30%.

number of shards $K = 5$. We observe that due to the growing number of malicious peers, the percent of appended correct outputs and percent of rejected incorrect outputs decrease in all algorithms, but with different rates. In particular, the decrease rates are the highest in the domain-based sharding and OmniLedger where the percent of accepted correct outputs and rejected incorrect outputs drops rapidly when there are more than 25% malicious peers, which means that these methods can only tolerate up to 25% or $N_{\mathcal{M}} = 5$ malicious peers. Such results concur with the theoretical security bounds of sharding models (reported, e.g., in [12], [13], and [16])—only one malicious peer is tolerated in each shard and, hence, up to $N_{\mathcal{M}} \leq K = 5$ malicious peers can be tolerated in the system. This is due to the fact that in the domain-based sharding and OmniLedger, the consensus inside each shard is reached only when all peers in the shard agree on the result. Therefore, if there is at least one malicious peers in the shard, the correct outputs can be rejected and the incorrect ones can be appended. On the other hand, as shown in [5], a reputation-based blockchain can tolerate up to 30% malicious peers, whereas, a self-organized sharding model can tolerate up to 40% malicious peers achieving the best performance among all algorithms. One of the reasons is that in a reputation-based blockchain, the consensus among validators is reached through conventional equal-weighted voting. Although the validators are selected from trustworthy peers, i.e., peers with the highest reputations, some of them can still be malicious. Accordingly, if the number of malicious validators is at least the number of faithful validators, the correct outputs can be rejected and the incorrect ones can be appended. On the contrary, in our self-organized sharding model, the consensus inside each shard is reached through weighted voting, in which the peer's weight is proportional to its normalized reputation. As such, the correct outputs will be rejected and incorrect ones will be appended only if there are shards where the total reputation of malicious peers is higher than that of faithful peers (which occurs much rarer than in the case with equal-weighted voting).

Next, we compare the simulated algorithms in terms of their throughputs. Figs. 13 and 14 show the overall throughput and average block delay in simulations with the increasing total number of peers N and percent of malicious peers fixed to 30%. We observe that because of the growing number of peers, the throughput increases in all algorithms, but with different rates—the rates of increase are lower in the domain-based sharding, OmniLedger, and reputation-based blockchain, and higher in our self-organized sharding scheme. The main reason is that in the domain-based sharding and OmniLedger, shards are formed based on results of PoW competition and random sampling, respectively, whereas, in a reputation-based system, validators are selected based on their reputations. As such, the peers' preferences (such as peers' locations and opinions about other peers in their shards) are ignored during shard formation. This leads to the growing block delay (Fig. 14), especially in scenarios with large numbers of peers. On the contrary, in our self-organized sharding model, each peer can select the shard that maximizes both: 1) peer's payoff which depends on its throughput and 2) peer's coalitional reputation which depends on reputations of other peers in the shard. This allows reducing the block delay (Fig. 14) and improving the throughput and security of the formed shards.

VII. Conclusion

We have designed a fully decentralized system architecture and a secure self-organized and scalable sharding scheme for the IoT-BC-MEC. We have proposed a new consensus method for the system where each peer votes on the outputs of block tasks in its shard. The peer's voting power is determined based on its reputation. By adopting a reputation-based coalitional game, we have developed a novel self-organized shard formation algorithm in which each peer acts as a rational player aiming to maximize both: 1) the peer's payoff which depends on its throughput and 2) the peer's coalitional reputation which depends on the reputations of other members of the peer's shard/coalition. We have shown that the algorithm converges to a reputation-based stable shard structure and achieves a superior performance in terms of the system throughput and security when compared to state-of-the-art sharding schemes and reputation-based blockchains.

References

[1] O. Novo, "Blockchain meets IoT: An architecture for scalable access management in IoT," *IEEE Internet Things J.*, vol. 5, no. 2, pp. 1184–1195, Mar. 2018.

[2] O. Novo, "Scalable access management in IoT using blockchain: A performance evaluation," *IEEE Internet Things J.*, vol. 6, no. 3, pp. 4694–4701, Jun. 2019.

[3] W. Wang *et al.*, "A survey on consensus mechanisms and mining strategy management in blockchain networks," *IEEE Access*, vol. 7, pp. 22328–22370, 2019.

[4] K. Yeow, A. Gani, R. W. Ahmad, J. J. P. C. Rodrigues, and K. Ko, "Decentralized consensus for edge-centric Internet of Things: A review, taxonomy, and research issues," *IEEE Access*, vol. 6, pp. 1513–1524, 2018.

[5] J. Kang, Z. Xiong, D. Niyato, D. Ye, D. I. Kim, and J. Zhao, "Toward secure blockchain-enabled Internet of vehicles: Optimizing consensus management using reputation and contract theory," *IEEE Trans. Veh. Technol.*, vol. 68, no. 3, pp. 2906–2920, Mar. 2019.

[6] W. Wang, D. Niyato, P. Wang, and A. Leshem, "Decentralized caching for content delivery based on blockchain: A game theoretic perspective," in *Proc. IEEE Int. Conf. Commun.*, Kansas City, MO, USA, May 2018, pp. 1–6.

[7] J. Kang, R. Yu, X. Huang, S. Maharjan, Y. Zhang, and E. Hossain, "Enabling localized peer-to-peer electricity trading among plug-in hybrid electric vehicles using consortium blockchains," *IEEE Trans. Ind. Informat.*, vol. 13, no. 6, pp. 3154–3164, May 2017.

[8] A. Reyna, C. Martín, J. Chen, E. Soler, and M. Díaz, "On blockchain and its integration with IoT. Challenges and opportunities," *Future Gener. Comput. Syst.*, vol. 88, pp. 173–190, Nov. 2018.

[9] Z. Zheng *et al.*, "Blockchain challenges and opportunities: A survey," *Int. J. Web Grid Services*, vol. 14, no. 4, pp. 352–375, 2018.

[10] A. P. Joshi, M. Han, and Y. Wang, "A survey on security and privacy issues of blockchain technology," *Math. Found. Comput.*, vol. 1, no. 2, pp. 121–147, May 2018.

[11] Y. Xiao, N. Zhang, W. Lou, and Y. T. Hou. (2019). *A Survey of Distributed Consensus Protocols for Blockchain Networks*. [Online]. Available: https://arxiv.org/abs/1904.04098

[12] E. Kokoris-Kogias, P. Jovanovic, L. Gasser, N. Gailly, E. Syta, and B. Ford, "OmniLedger: A secure, scale-out, decentralized ledger via sharding," in *Proc. IEEE Security Privacy*, 2018, pp. 583–598.

[13] H. Yoo, J. Yim, and S. Kim, "The blockchain for domain based static sharding," in *Proc. IEEE TrustCom/BigDataSE*, 2018, pp. 1689–1692.

[14] S. Forestier, D. Vodenicarevic, and A. Laversanne-Finot. (2018). *Blockclique: Scaling Blockchains Through Transaction Sharding in a Multithreaded Block Graph*. [Online]. Available: https://arxiv.org/abs/1803.09029

[15] M. H. Manshaei, M. Jadliwala, A. Maiti, and M. Fooladgar, "A game-theoretic analysis of shard-based permissionless blockchains," *IEEE Access*, vol. 6, pp. 78100–78112, 2018.

[16] S. Li, M. Yu, C.-S. Yang, A. S. Avestimehr, S. Kannan, and P. Viswanath. (Sep. 2018.). *Polyshard: Coded Sharding Achieves Linearly Scaling Efficiency and Security Simultaneously*. [Online]. Available: https://arxiv.org/abs/1809.10361

[17] R. Yang, F. R. Yu, P. Si, Z. Yang, and Y. Zhang, "Integrated blockchain and edge computing systems: A survey, some research issues and challenges," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 2, pp. 1508–1532, 2nd Quart., 2019.

[18] B. Omoniwa, R. Hussain, M. A. Javed, S. H. Bouk, and S. A. Malik, "Fog/edge computing-based IoT (FECIoT): Architecture, applications, and research issues," *IEEE Internet Things J.*, vol. 6, no. 3, pp. 4118–4149, Jun. 2019.

[19] Z. Yang, K. Yang, L. Lei, K. Zheng, and V. C. M. Leung, "Blockchain-based decentralized trust management in vehicular networks," *IEEE Internet Things J.*, vol. 6, no. 2, pp. 1495–1505, May 2018.

[20] V. Hassija, V. Chamola, S. Garg, D. N. G. Krishna, G. Kaddoum, and D. N. K. Jayakody, "A blockchain-based framework for lightweight data sharing and energy trading in V2G network," *IEEE Trans. Veh. Technol.*, vol. 69, no. 6, pp. 5799–5812, Jun. 2020.

[21] Y. Liu, K. Wang, Y. Lin, and W. Xu, "A lightweight blockchain system for industrial Internet of Things," *IEEE Trans. Ind. Informat.*, vol. 15, no. 6, pp. 3571–3581, Mar. 2019.

[22] K. Dorri, S. S. Kanhere, R. Jurdak, and P. Gauravaram, "LSB: A lightweight scalable blockchain for IoT security and anonymity," *J. Parallel Distrib. Comput.*, vol. 134, pp. 180–197, Dec. 2019.

[23] R. Blum and T. Bocek, "Superlight–A permissionless, light-client only blockchain with self-contained proofs and BLS signatures," in *Proc. IFIP/ IEEE Symp. Integr. Netw. Service Manag. (IM)*, Apr. 2019, pp. 36–41.

[24] L. Li *et al.*, "CreditCoin: A privacy-preserving blockchain-based incentive announcement network for communications of smart vehicles," *IEEE Trans. Intell. Transp. Syst.*, vol. 19, no. 7, pp. 2204–2220, Jan. 2018.

[25] R. A. Michelin *et al.*, "SpeedyChain: A framework for decoupling data from blockchain for smart cities," in *Proc. 15th EAI Int. Conf. Mobile Ubiquitous Syst. Comput. Netw. Services*, Nov. 2018, pp. 145–154.

[26] N. Lasla, M. Younis, W. Znaidi, and D. B. Arbia, "Efficient distributed admission and revocation using blockchain for cooperative its," in *Proc 9th IFIP Int. Conf. New Technol. Mobility Security (NTMS)*, 2018, pp. 1–5.

[27] L. Xu, L. Chen, Z. Gao, S. Xu, and W. Shi, "EPBC: Efficient public blockchain client for lightweight users," in *Proc. 1st Workshop Scalable Resilient Infrastruct. Distrib. Ledgers*, Dec. 2017, pp. 1–6.

[28] C. Ehmke, F. Wessling, and C. M. Friedrich, "Proof-of-property: A lightweight and scalable blockchain protocol," in *Proc. 1st Int. Workshop Emerg. Trends Softw. Eng. Blockchain*, May 2018, pp. 48–51.

[29] Z. Liu, S. Tang, S. S. M. Chow, Z. Liu, and Y. Long, "Fork-free hybrid consensus with flexible proof-of-activity," *Future Gener. Comput. Syst.*, vol. 96, pp. 515–524, Jul. 2019.

[30] A. C. An, P. T. X. Diem, L. T. T. Lan, T. V. Toi, and L. D. Q. Binh, "Building a product origins tracking system based on blockchain and PoA consensus protocol," in *Proc. IEEE Int. Conf. Adv. Comput. Appl. (ACOMP)*, Nov. 2019, pp. 27–33.

[31] Binance Academy. *Proof of Authority Explained*. Accessed: Apr. 2020. [Online]. Available: https://www. binance.vision/blockchain/proof-of-authority-explained

[32] H. Wei, W. Feng, C. Zhang, Y. Chen, Y. Fang, and N. Ge. (Jan. 2020). *Creating Efficient Blockchains for the Internet of Things by Coordinated Satellite-Terrestrial Networks*. [Online]. Available: https://arxiv.org/pdf/2001.01358.pdf

[33] Q. Zhou, H. Huang, Z. Zheng, and J. Bian, "Solutions to scalability of blockchain: A survey," *IEEE Access*, vol. 8, pp. 16440–16455, 2020.

[34] N. Chawla, H. W. Behrens, D. Tapp, D. Boscovic, and K. S. Candan, "Velocity: Scalability improvements in block propagation through rateless erasure coding," in *Proc. IEEE Int. Conf. Blockchain Cryptocurrency (ICBC)*, May 2019, pp. 447–454.

[35] R. Yasaweerasinghelage, M. Staples, and I. Weber, "Predicting latency of blockchain-based systems using architectural modelling and simulation," in *Proc. IEEE Int. Conf. Softw. Archit. (ICSA)*, Apr. 2017, pp. 253–256.

[36] C. Decker and R. Wattenhofer, "Information propagation in the bitcoin network," in *Proc. IEEE P2P*, Sep. 2013, pp. 1–10.

[37] W. Chen, Z. Zhang, Z. Hong, C. Chen, J. Wu, and S. Mahar, "Cooperative and distributed computation offloading for blockchain-empowered industrial Internet of Things," *IEEE Internet Things J.*, vol. 6, no. 5, pp. 8433–8446, Oct. 2019.

[38] M. Liu, F. R. Yu, Y. Teng, V. C. M. Leung, and M. Song, "Distributed resource allocation in blockchain-based video streaming systems with mobile edge computing," *IEEE Trans. Wireless Commun.*, vol. 18, no. 1, pp. 695–708, Jan. 2019.

[39] M. Liu, F. R. Yu, Y. Teng, V. C. M. Leung, and M. Song, "Computation offloading and content caching in wireless blockchain networks with mobile edge computing," *IEEE Trans. Veh. Technol.*, vol. 67, no. 11, pp. 11008–11021, Nov. 2018.

[40] S. Sen, I. Goswami, and S. Airiau, "Expertise and trust-based formation of effective coalitions: An evaluation of the art testbed," in *Proc. AAMAS*, 2006, pp. 71–78.

[41] Y. Liu, Q. Li, and J. Zhang, "Coalition formation game based reputation system," in *Proc. ACM WIT-EC*, 2012, pp. 1–12.

[42] L. Mashayekhy and D. Grosu, "A reputation-based mechanism for dynamic virtual organization formation in grids," in *Proc. IEEE ICPP*, 2012, pp. 108–117.

[43] L. F. Bilecki, A. Fiorese, and F. Matos, "A trust reputation architecture for virtual organization integration in cloud computing environment," in *Proc. ICEIS*, 2017, pp. 695–702.

[44] D. Xu and Y. Tian, "A comprehensive survey of clustering algorithms," *Ann. Data Sci.*, vol. 2, no. 2, pp. 165–193, Jun. 2015.

[45] S. Shukri, H. Faris, I. Aljarah, S. Mirjalili, and A. Abraham, "Evolutionary static and dynamic clustering algorithms based on multi-verse optimizer," *Eng. Appl. Artif. Intell.*, vol. 72, pp. 54–66, Jun. 2018.

[46] J. Chen, H. Sun, D. Woodruff, and Q. Zhang, "Communication-optimal distributed clustering," in *Proc. Adv. Neural Inf. Process. Syst.*, 2016, pp. 3727–3735.

[47] L. Bing, M. Zhuang, Z. Yanshuo, and C. Juliang, "Distributed dynamic clustering in wireless networks with capacity constraint and concealed data," in *Proc. IEEE CCET*, 2018, pp. 86–90.

[48] A. Asheralieva and D. Niyato, "Hierarchical game-theoretic and reinforcement learning framework for computational offloading in UAV-enabled mobile edge computing networks with multiple service providers," *IEEE Internet Things J.*, vol. 6, no. 5, pp. 8753–8769, Oct. 2019.

[49] 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 Internet Things J.*, vol. 7, no. 3, pp. 1974–1993, Mar. 2020.

[50] A. Asheralieva, T. Q. S. Quek, and D. Niyato, "An asymmetric evolutionary Bayesian coalition formation game for distributed resource sharing in a multi-cell device-to-device enabled cellular network," *IEEE Trans. Wireless Commun.*, vol. 17, no. 6, pp. 3752–3767, Jun. 2018.

[51] A. Asheralieva, "Bayesian reinforcement learning-based coalition formation for distributed resource sharing by device-to-device users in heterogeneous cellular networks," *IEEE Trans. Wireless Commun.*, vol. 16, no. 8, pp. 5016–5032, Aug. 2017.

[52] G. Q. Pérez, J. A. Hernández, and D. L. López, "Delay analysis of fronthaul traffic in 5G transport networks," in *Proc. IEEE ICUWB*, Sep. 2017, pp. 1–5.

[53] R. O. Afolabi, A. Dadlani, and K. Kim, "Multicast scheduling and resource allocation algorithms for OFDMA-based systems: A survey," *IEEE Commun. Surveys Tuts.*, vol. 15, no. 1, pp. 240–254, 1st Quart., 2013.

[54] A. Asheralieva and D. Niyato, "Learning-based mobile edge computing resource management to support public blockchain networks," *IEEE Trans. Mobile Comput.*, early access, Dec. 16, 2019, doi: 10.1109/TMC.2019.2959772.

[55] *OPNET Network Simulator*. Accessed: Apr. 2020. [Online]. Available: http://opnetprojects.com/opnet-network-simulator/

[56] *Omnet Disrete Event Simulator*. Accessed: Apr. 2020. [Online]. Available: https://omnetpp.org/

[57] *The Network Simulator—NS-2*. Accessed: Apr. 2020. [Online]. Available: https://www.isi.edu/nsnam/ns/

[58] M. Chen, Y. Miao, and I. Humar, "Introduction to OPNET network simulation," in *OPNET IoT Simulations*. Singapore: Springer, 2019, pp. 77–153.

[59] *Evolved Universal Terrestrial Radio Access (E-UTRA) and Evolved Universal Terrestrial Radio Access Network (E-UTRAN); Overall Description; Stage 2, Release 13*, 3GPP Standard TS 36.300, 2016.

[60] "Characteristics of a bending-loss insensitive single-mode optical fibre and cable for the access network," ITU-T, Geneva, Switzerland, Recommendation G.657, Dec. 2017.

[61] P. Sethi and S. R. Sarangi, "Internet of Things: Architectures, protocols, and applications," *J. Elect. Comput. Eng.*, vol. 2017, Jan. 2017, Art. no. 9324035.

[62] Z. Xiong, S. Feng, W. Wang, D. Niyato, P. Wang, and Z. Han, "Cloud/fog computing resource management and pricing for blockchain networks," *IEEE Internet Things J.*, vol. 6, no. 3, pp. 4585–4600, Jun. 2019.

[63] 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.

[64] *Intel® Xeon® Processor 7150N*. Accessed: Apr. 2020. [Online]. Available: https://ark.intel.com/content/www/us/en/ark/products/28029/intel-xeon-processor-7150n-16m-cache-3-50-ghz-667-mhz-fsb.html

[65] K. P. Apt and A. Witzel, "A generic approach to coalition formation," *Int. Game Theory Review*, vol. 11, no. 3, pp. 347–367, Sep. 2009.

Alia Asheralieva received the B.S. degree from Kyrgyz Technical University, Bishkek, Kyrgyzstan, in 2004, the M.E. degree from the Asian Institute of Technology, Khlong Luang, Thailand, in 2007, and the Ph.D. degree from the University of Newcastle, Callaghan, NSW, Australia, in 2015.

From 2015 to 2016, she was a Research Assistant Professor with the Graduate School of Information Science and Technology, Hokkaido University, Sapporo, Japan. In 2017, she was a Postdoctoral Research Fellow with the Information Systems Technology and Design Pillar, Singapore University of Technology and Design, Singapore. She is currently an Assistant Professor with the Department of Computer Science and Engineering, Southern University of Science and Technology, Shenzhen, China. Her main research interests span many areas of communications and networking, including cognitive radio networks, heterogeneous networks, D2D and IoT communications, cloud/edge/fog computing, mobile blockchains, cross-layer resource allocation and optimization, congestion control and routing, game theory, computational and artificial intelligence for wireless networks, as well as queuing theory, simulation and network modeling, QoS, and performance evaluation.

Dusit Niyato (Fellow, IEEE) received the B.Eng. degree from the King Mongkut's Institute of Technology Ladkrabang, Bangkok, Thailand, in 1999, and the Ph.D. degree in electrical and computer engineering from the University of Manitoba, Winnipeg, MB, Canada, in 2008.

He is currently a Professor with the School of Computer Science and Engineering, Nanyang Technological University, Singapore. His research interests include energy harvesting for wireless communication, Internet of Things, and sensor networks.