Blockchain-Enhanced Time-Variant Mean Field-Optimized Dynamic Computation Sharing in Mobile Network

Fenhua Bai, Tao Shen, *Member, IEEE,* Zhuo Yu, Jian Song, Bei Gong, *Member, IEEE,* Muhammad Waqas, *Senior Member, IEEE,* Hisham Alasmary, *Member, IEEE*

***Abstract*—Although 5G and beyond communication technology empower a large number of edge heterogeneous devices and applications, the stringent security remains a major concern when dealing with the millions of edge computing tasks in the highly dynamic heterogeneous networks (HDHNs). Blockchains contribute significantly to addressing security challenges by guaranteeing the reliability of data and information. Since the node's mobility, there are risks of exiting the network and leaving the remaining tasks noncomputed. Therefore, we model the cost function of offloaded computing tasks as a dynamic stochastic game. To reduce the computational complexity, the Time-Variant Mean-Field term (TVMF) is adopted to solve the cost-optimized problem. What's more, we design an Adaptivity-Aware Practical byzantine fault tolerance consensus Protocol (AAPP) to dynamically formulate domains, execute leader node selection with regard to task completion and quickly verify computational results. In addition, a Dynamic Multi-domain Fractional Repetition uncoded repair storage (DMFR) scheme with variant redundancy is proposed to reduce the storage pressure and repair overhead. The simulation is implemented to demonstrate our scheme outperforms the benchmarks in terms of cost and time overhead.**

***Index Terms*—Dynamic Networks; Tasks offload; Blockchains; Mean-field Game; Fractional Repetition Code**

I. Introduction

As the state-of-the-art 5G and beyond wireless network communication techniques become approximately omnipresent [1], it enables the considerable amounts of edge heterogeneous devices and applications, which accelerates the construction development of the Industrial Internet of Things (IIoT) [2]. The stringent security remains a significant concern when dealing with the millions of mobile edge computing (MEC) tasks [3], [4]. Nevertheless, facing the highly dynamic heterogeneous network (HDHNs) consisting of massive IoT mobile devices and multiple network links (such as cellular network and wireless fidelity (WiFi)), the topology of the HDHNs varies dynamically and unpredictably because of nodes joining and removing. Therefore, reliable connections and secure computations are the main challenges to be addressed in dynamic networks [5]. To solve this issue, some technologies in terms of NOMA [6] [11], IRS [9] [10] [11], blockchain [7] [8], etc., are implemented to provide a secure environment for MEC networks. In [11], NOMA is adopted through the cooperation interference to confuse the decoding process so as to guarantee security. Nevertheless, the current NOMA work is exploited in static environment, which the participants' sharing information maintain constant and result in inadaptability to the dynamic networks [6]. Moreover, IRS [9] [10] [11] is utilized to adapt to the highly dynamic networks and guarantee offloading computation security. However, for the IRS deployed in MEC-enabled IIoT networks, the issue with regard to channel estimation and complicated reflection optimization remains to be tackled. Different from the above methods of physical layer security, as a distributed network communication consistency technology, the arrival of blockchain provides a secure offloading computation environment for its immutability and traceability by consensus algorithm and underlying cryptography [7] [8] [14] [15] [16].

To our best knowledge, consensus protocols are the core component of blockchain technology [17]–[19] for the computational transactions verification. Under the highly dynamic circumstance, on the one hand, network partitions may happen frequently and the communications between nodes might also be unreliable, [20] designs a stable PoW consensus protocol solving the privacy and security challenges in mobile ad-hoc network environment. It is also demonstrated that the original PoW protocol as well as its variants may result in high expenditure of energy and is not suitable for resource-limited mobile devices. On the other hand, the successive blocks need a fixed interval to generate and cannot work effectively in dynamic networks. Furthermore, [22] shows the effectiveness and resilience of the suggested clustering protocol. The authors of [23] present the concept of dynamic consensus (i.e., emergent dynamics) and robust stability for practical asymptotic synchronization of heterogeneous networked systems. For permission blockchain, [24] develops a CSMA/CA request access modification in practical Byzantine fault tolerance (PBFT) subscription service applicable

Fenhua Bai, Tao Shen and Jian Song are with the Faculty of Information Engineering and Automation, Kunming University of Science and Technology of China, Kunming, China (e-mail: bofenhua@stu.kust.edu.cn; shentao@kust.edu.cn; songjian@kust.edu.cn).

Zhuo Yu is with the Department of Research and Development, State Grid Information and Telecommunication Co., Ltd. (e-mail: yuzhuo@sgitg.sgcc.com.cn).

Bei Gong is with the Department of Computing, Beijing University of Technology (e-mail: gongbei@bjut.edu.cn).

M. Waqas is with the Computer Engineering Department, College of Information Technology, University of Bahrain, 32038, Bahrain, and School of Engineering, Edith Cowan University, Perth WA, 6027, Australia. (e-mail: engr.waqas2079@gmail.com).

H. Alasmary is with the Department of Computer Science, King Khalid University, Abha 61421, Saudi Arabia (e-mail: alasmary@kku.edu.sa).

Corresponding author: Tao Shen, (shentao@kust.edu.cn).

to mantling specific geographic regions through connected with agents gathered data from IoT domains. However, the existing PBFT consensus mechanism related to the completion degree of computing tasks has not been researched in HDHNs. Meanwhile, the conventional PBFT-based consensus is not adaptive to the dynamic environment since the requirement of more than 2/3 of honest nodes may suffer the scalability problem [20], [21].

In addition, the storage strategy of block generated after improved PBFT consensus is especially important for ensuring the reliability of computing results and traceability [46] [48]. For example, distributed hash table (DHT), interplanetary file system (IPFS) and cloud-based off-chain storage scheme [46]. Except for this, the collaborative on-chain storage scheme of coding-based [48], such as Reed-Solomon code and simple regenerating code, increases the communication bandwidth and repair cost due to the encoding and decoding process. Particularly, for the isomorphic fractional repetitive coding, the fixed parameters construction is not suitable for dynamic condition when the number of members is changeable in the network [55]. Accordingly, it should be considered that the scalable fractional repetitive coding storage of calculation results verified by consensus algorithm when the storage node is faulty.

After improving the blockchain consensus and storage scalability to adapt to dynamic networks and then applying it to MEC secure computing sharing, offloading optimization needs to be considered due to the large number of resource constrained edge devices that are difficult to support the computing capabilities of blockchain network nodes [25] [26]. [27] and [28] address the issue for mobile devices with limited resources by offloading computational tasks to edge servers or fog nodes with rich resource in the networks. Except for the above research, game theory is a forceful means to solve security problems. In [29], Mean-field games (MFG) can realize a higher accuracy analysis and privacy preservation with a large population of participants. [1] provides an MFG-empowered method for making distributed safety defense decisions among a malicious node and a large number of legitimate nodes in mobile ad hoc networks. Although blockchain and MFG are employed to construct a good and secure ecosystem for resource sharing, due to the nodes are highly dynamic and their mobility is unpredictable, there is a risk that the nodes cannot achieve all the computing tasks if it is moving out of the network. Accordingly, the punishment cost should be considered to guarantee all the published computing tasks be finished in dynamic network. However, there is little relevant literature to analyze this issue.

Therefore, aiming at tackling the involved issues above, we first construct an architecture blockchain-based that adapts to dynamic computation resources sharing in the highly dynamic heterogeneous network. Then, considering the node's mobility in HDHNs, the approach of the mean-field game is adopted to partially offload the tasks to others to guarantee the computing tasks' achievement in the current network. Lastly, we dynamically construct domains and devise an adaptivity-aware consensus algorithm and storage scheme to realize effective and efficient computing results onto the blockchain. The contributions of this paper are stated as follows:

1) We construct a novel blockchain-based architecture to provide dynamic, secure computing resources sharing services in HDHNs. Mainly, the blockchain is applied to conveniently manage the majority of 5G-empowered IoT mobile devices joining and leaving the network. The IoT nodes joining the blockchain are classified as three types of nodes (i.e., computing nodes, consensus nodes and storage nodes).
2) To avoid the computing node leaving the remaining tasks non-computed and exiting the network before the deadline, the punishment cost blockchain-based is put forward and can be automatically executed. This is the first work that utilized blockchain and MFG to solve the optimal offloading computing power varied with the channel's dynamics. Meanwhile, the existing literature only researches the offloading calculation strategies but does not consider the reliable storage of calculation results and its scalability and data repair when the storage node is faulty.
3) We dynamically model the cost function of offloading computing nodes in HDHNs as a dynamic stochastic game considering the offloading computational power and corresponding price. However, as the number of players increases, classical solutions cannot be practical due to computational complexities. To reduce the cost function optimized solving complexity with the large population of IoT mobile devices, we exploit the Time-Variant Mean-Field term approach (TVMF) to estimate the optimal offloading strategy at the next timeslot of HDHNs.
4) Since the node's mobility is highly dynamic and the network topology unpredictable, based on the completed status of computing tasks, we design an Adaptivity-Aware PBFT consensus Protocol (AAPP) among consensus nodes to dynamically formulate domains, execute leader selection and quickly verify computational results. At an arbitrary timeslot, the number of nodes satisfies the correctness of Byzantine failures.
5) In terms of block storage strategy, it is laborious to save a full copy of the whole blockchain ledger due to the limited memory space of IoT mobile devices. To tackle this problem and reduce the repair bandwidth of the fault node, a Dynamic Multi-domain Fractional Repetition uncoded repair storage scheme (DMFR) with variant redundancy among the storage nodes is proposed. When the formal block is generated, it is partitioned as fragments and stored by storage nodes in the dynamically formulated domains.

The rest of this article is organized as follows: In Section II, related works are presented. The problem statement, including the system model and cost function TVMF-optimized, is given in Section III. The suggested AAPP and dynamic storage scheme DMFR are described in Section IV and Section V, respectively. Experimental studies and performance evaluations are gathered in Section VI. In the end, the article in Section VII can be summarized.

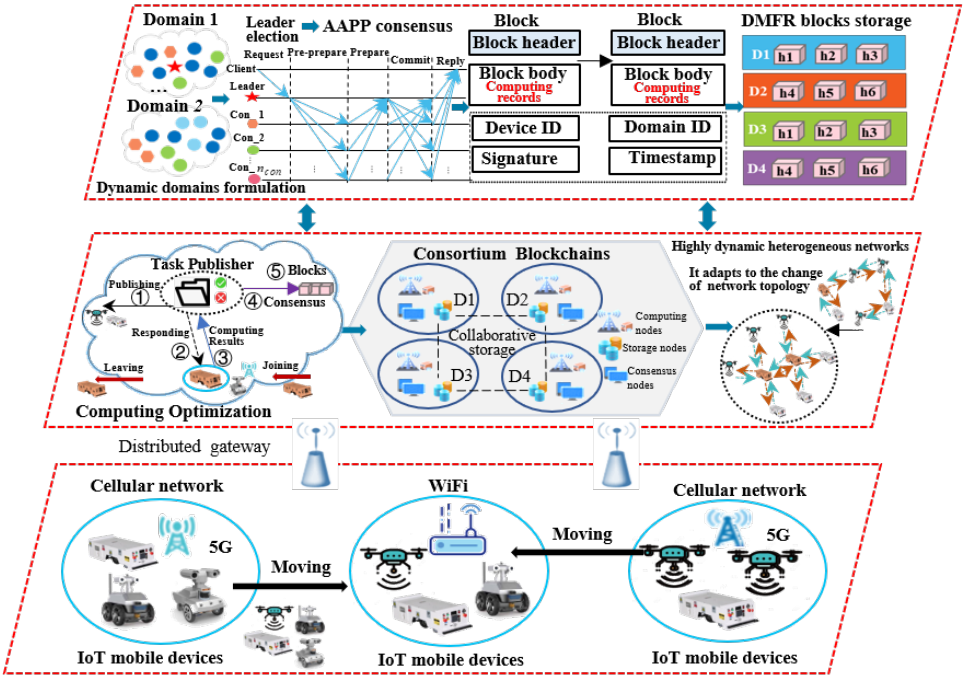

Fig. 1. Architecture of the proposed system

II. Related Works

There are many literatures focus on the edge computing in mobile networks. For example, in [12], an SDN-based architecture is constructed to offer computing services, which can achieve time-varying computing resource optimization. Although AI/ML tools [13] requires a large amount of computation resources, by exploiting this technology, it can obtain beneficial information and make decision in edge computing offloading for realizing low-latency services. However, these works have not considered secure computing sharing under the participation of large-scale game players.

Therefore, reliable connections and secure computations are the main challenges to be addressed in dynamic networks [5]. To solve this issue, the blockchain provides a secure offloading computation environment for its immutability and traceability by consensus algorithm and underlying cryptography [7] [8] [14] [15] [16].

A. Mean Field Game Blockchain-enabled

By applying blockchain technology to MEC secure computing sharing, MEC resource optimization needs to be considered due to the large number of resource constrained edge devices that are difficult to support the computing capabilities of blockchain network nodes [25]. Therefore, a variety of approaches and mathematical theories have been taken into account for assisting the performance analysis [30], security enhancements [1] and economic profits strategies [31]. Game theory is suitable for analyzing the competitive mining behavior of a large number of miners [29]. Moreover, it has also been applied to analyze edge computing offloading decisions and actions of participants in the blockchain-enabled network [32] [33]. Nevertheless, as the amounts of participants grow more extensive, it is impractical for information exchange with each other because of their privacy and time delay. The conventional game theoretical approach cannot be effective since a lack of information exchange or high computing complexities. The arrival of MFG is employed as a powerful way to address such problems. As the number of participants becomes greater, the impact of all participants can be encapsulated as a term named the mean field (MF) term [29], [34].

Considering the channel dynamics, authors in [35] model the joint optimization problem as a multi-user non-cooperative dynamic stochastic game, then propose an MFG-based algorithm to solve the issue that joint offloading decisions become prohibitively complex in a dynamic wireless environment. To deal with the overload issue for massive heterogeneous devices with different computing capabilities, [36] leverages the MF term by information exchange among neighbor devices. Considering energy efficiency performances, [37] presents a MFG theoretical framework with the interference mean-field approximation, which is mainly used in cellular networks. In addition, [38] combines with the DRL tool for offloading strategy. However, the existing works have not consider data recovery and task computing automatic penalty. The detailed comparison of different schemes is shown in Table I.

B. Blockchain PBFT consensus

A blockchain is a distributed system reaching an agreement that all nodes decide on a common result. Herein, the members collectively make a decision on whether to approve or abandon a block related to their transactions [17]. The Byzantine Fault Tolerance (BFT) algorithm derived from the Byzantine general problem is created by discovering nodes of distributed systems tend to be faulty and may lose the characteristics of liveliness and security [39]. The PBFT agreement algorithm can endure

TABLE I
COMPARISON OF DIFFERENT MEAN FIELD GAME SCHEMES

Schemes	Security	Privacy	Reliable storage	Result validation	Immutability	Method	Data recovery	Automatic penalty
[28]	✗	✓	✗	✗	✗	MFTG	✗	✗
[29]	✓	✓	✓	✓	✓	MFG	✗	✗
[35]	✗	✓	✗	✗	✗	MFG	✗	✗
[36]	✗	✓	✗	✗	✗	MFT	✗	✗
[37]	✗	✓	✗	✗	✗	MFG	✗	✗
[38]	✗	✓	✗	✗	✗	MFG-DRL	✗	✗
Ours	✓	✓	✓	✓	✓	TVMF	✓	✓

failure nodes to a certain extent and improve the applicability and solidity by utilizing state machine duplicated services [40]. This consensus is accomplished by most of ballot theory and message passing mechanism in an asynchronous communication condition. What's more, PBFT has been deployed in several systems. Kotla et al. introduce an optimistic linear path into PBFT, which is utilized in Byzcoin [41] while the leader replacement protocol remains $O(n^3)$. Combined with a digital signature to authenticate all messages, consensus can bring enhanced safety for the system from comprehensive directions/boost protocol security [24]. Through adopting the encrypted approach, PBFT can resist spooling as well as replay attacks and further identify damaged information. In the work [42], a new private key encryption mechanism combined with secret sharing is used to guarantee the security of information and its hash values. Xu et al. [43] design an ABC-GSPBFT consensus with a grouping score mechanism and employ artificial bee colony-optimized to improve the reliability of flight data sharing. Based on the novel fault model, [44] puts forward a distributed protocol with higher efficiency to reach an $(a,b)-$majority consensus within $O(n)$.

Aiming at solving the problem of frequent inter-node communications and scalability, Li et al. [40] present a double-layer PBFT, which significantly reduced communication complexity. Then the scheme is stretched to arbitrary-level systems and carries out corresponding analysis in terms of communication complexity as well as security. As the first distributed protocol that can tolerate Byzantine failures, PBFT is applied to enhance the resilience and fault tolerance of mobile networks. Nevertheless, the complexity of communication is $O(n^2)$ with the increasement of consensus nodes [17]. In addition, its waiting time is significantly raised in an asynchronous transmission condition, which becomes a significant obstacle to the performance of permissioned blockchains.

C. Block Storage Model

As a distributed ledger, blockchain is required to store computing results for long periods with very high reliability to protect the data from being lost when nodes fail. Blockchain node as a complete node stores complete data to ensure data security through the high redundancy storage mechanism. However, these characteristics may lead to the problem of storage scalability. For the storage-limited IoT device, it cannot well store a full duplicate of the whole blockchain ledger. To overcome the problem of storage scalability, a DHT, IPFS and cloud-based off-chain storage scheme are stated [46]. The data in the block body is transferred from the original block body to the stored system offline, and only the pointer to the data is stored in the block body. However, there are two obvious challenges. On the one hand, it is required to consider how to select nodes with sufficient storage capacity while maintaining the storage system under the chain, and also ensure that these nodes are not malicious and control the data authenticity of the blockchain. On another hand, how to determine the proportion between blockchain nodes and non-blockchain nodes in the distributed storage system needs to be considered to ensure the security of the storage system.

Focusing on the on-chain cooperative storage scheme, in [48], the received multiple packets are encoded and fused based on the cooperative coding storage, which improves the network performance but increases the communication bandwidth and repair cost due to the encoding and decoding process. The concrete comparison of different storage schemes is presented in Table II. To guarantee the constructed codes have low complexity in HDHNs [49], it is required that the block stored needs satisfy what calls the uncoded repair property. Constructing exact MBR codes with an uncoded repair is resilient to multiple failures [49].

Although the existing fractional repetitive coding is mainly used in the public chain [42] [50] and the fixed storage parameters are not suitable for dynamic storage, there is a lack of work in the alliance chain and conduct corresponding research under the dynamic heterogeneous condition when the number of members is changeable in the network.

III. SYSTEM MODEL AND PROBLEM FORMULATION

A. System Model

The system architecture designed is shown in Fig. 1. There are diverse network link ways including cellular networks. The distributed gateway provides computing data reliable aggregation into the blockchain. Since the network topology may change continually and unpredictably, any new identity entering the network is managed by blockchain. Moreover, the dynamic computing results can be stored on the blockchain after finishing the consensus. Assuming that the computation task can be partitioned randomly and processed in parallel [51]. Particularly, the computing tasks can be partially offloaded from tasks responder to other computation-intensive mobile device to avoid overload and redundancy. The node's mobility can be regarded as moving from one communication link (cellular network) to another (WiFi), vice versa.

TABLE II
DIFFERENT STORAGE SCHEME COMPARISON

Schemes			Data security	Domain form	Storage content	Storage size	Consensus nodes	Compression ratio	TPS
Full node			-	S	Cl	Cb	N	0	1×
Off-chain	DHT-based [45]		Kademlia	-	-	-	N_{on}	-	1×
	IPFS-based		IPFS protocol	-	-	-	N_{on}	91.83%	1×
	Cloud-based		Cloud storage	-	-	-	N_{on}	-	1×
On-chain	Collab-orative	Code-based	-	D	Sb	block subset	N_{con}	-	1×
		Ours	-	D	Sb	block subset	N_{con}	-	$Domain$×
		Cluster-based	-	S	Cl	block set	N_{con}	-	1×
		Sharding-based [46]	-	D	Dl	Dl	N_{con}/Seg	$1-1/Seg$	Seg×
	Compression [47]		-	D	Cl	Almost 20%	N_{con}	Up to 80%	1×
	Light node		-	S	Cl	$\frac{block\ header}{block\ size}$	N_{con}	Block header	1×

Note: S means static; D means dynamic; Cl represents complete ledger; Sb denotes single block; Cb denotes complete block; Dl is the domain ledger; N_{on} means the nodes on-chain; N_{con} denotes consensus nodes; × means times.

Given that the computing tasks offload occurs in $t \in [0,\ T]$. T is the longest time for a round of blockchain consensus since the computing results need to store on-chain for immutability. Denote $g_i(t)$ the channel gain between IoT mobile device i and the offload computation responder/undertaker. According to Ito's lemma [35], the channel dynamics can be modeled as

$$d\mathrm{g}_i(t) = \vartheta_i(t, \mathrm{g}_i(t))dt + \sigma_i(t)dW(t) \tag{1}$$

here $\vartheta_i(t, \mathrm{g}_i(t))$ represents a deterministic smooth function and denotes the evolution of path loss with time because of device mobility. The initial channel gain $\mathrm{g}_i(0)$ is obtained. Stochastic term $\sigma_i(t)dW(t)$ denotes a Brownian motion, which accounting for unpredictable channel variations and satisfying $\mathcal{N}(0,\ \sigma_i(t)dt)$.

Let the set of players participate in offloading computing at time t of HDHNs is $N_c = \{1, 2, \ldots i \ldots, n_c\}$. To avoid the node has not completed the promised calculation tasks and leave the network because of its mobility, the computational tasks of an IoT mobile device i needs to transfer partially to the MEC servers or proceed to other computation-intensive devices due to MEC server overload at timeslot t. The remaining data volume of offloading tasks $X_i(t)$ is related to the transmitted rate $r_i(t)$ as well as the computational power [35]. Each IoT mobile device i should make a decision on its offloading computational power $w_i(t)$, which is equal to the difference between input and output transmitted rate divided by the time steps Δt. Under the assumption above, the dynamics of network state can be characterized as follows:

$$dX_i(t) = -r_i(t)dt = w_i(t)\Delta t dt \tag{2}$$

In light with the Shannon formula, the transmission rate $r_i(t)$ can be expressed as

$$r_i(t) = B\log_2(1 + p_i(t)g_i(t)/\sigma_0^2) \tag{3}$$

where B, $p_i(t) \in [0, p_i^{\max}]$, σ_0^2 represent the channel bandwidth, transmission power and background noise, respectively.

The global flow containing three major stages and can be presented in Fig. 2:

Stage I : Computing

Task publisher: The IoT mobile device posts computing tasks in a broadcast manner and aggregates the computing results.

Task responder: In HDHNs, the participants randomly respond to the published computing tasks. The responder (e.g., MEC server) performs the calculation locally and evaluates whether it can finish all tasks or not in the current network. If it cannot finish the promised tasks due to overload or redundancy, it must partially transfer the tasks to other computation-intensive mobile devices to achieve the calculation.

Task partially offload undertaker: It undertakes the partially offloading tasks from the task responder and feedbacks the computational results to the task publisher.

Stage II : Consensus. After computing, all the offloading computing results served as transactions are packed in the form of a candidate block by the leader, which launches rounds of consensus to vote for the computing outcomes and finally generates the formal block. To improve the adaptivity of nodes that participates in consensus in HDHNs, the leader is selected through our proposed AAPP method elaborated in subsequent section IV.

Stage III : Storage. The generated formal blocks by consensus are stored on-chain and through the storage scheme put forward in section IV. The fractional repetition uncoded repair method aims to reduce the bandwidth of data repair and decrease the storage pressure on-chain.

B. Cost Function

In the blockchain-based HDHNs, according to the features of current IoT mobile devices such as computing frequency, communicational bandwidth and memory space, the nodes N are classified as three types (i.e., computing nodes N_c, consensus nodes N_{con} and storage nodes N_s). They are responsible for dealing with the tasks of edge computing, computing results voting and computing results storing, respectively.

As previously described, the set of blockchain computing nodes in HDHNs is $N_c = \{1, 2, ..., n_c\}$. The cost of the IoT mobile device i is proportional to the computational power and mainly consists of computation, transmission and penalty cost.

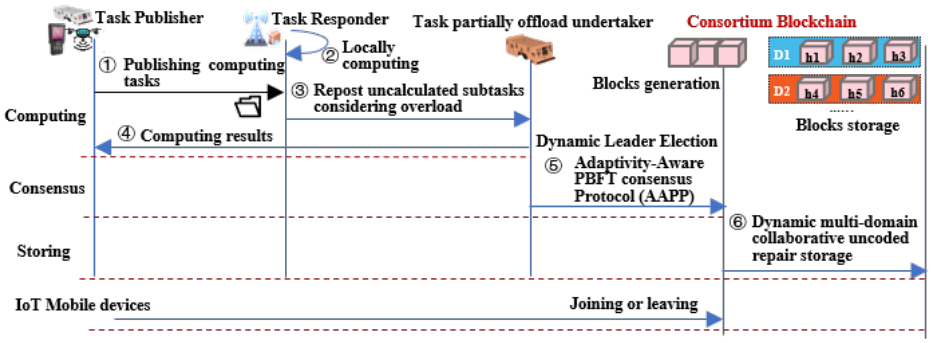

Fig. 2. The global flow of our architecture.

1) The computing cost of IoT mobile device i at t time

By referring to [35], the computing cost of IoT mobile device i equals to the computational power multiply unit computing price. Therefore, it can be denoted as

$$J_i^c(t) = v_i(t) w_i(t) \tag{4}$$

here, $v_i(t)$ means the unit price per unit of computational power.

In [35], the computational price is determined according to the offloading computational power. When the number of the player is greater than 2, computation price is coupled as shown in Eq. (5), and v_s is the single pricing. μ is the convert coefficient between the computing power and the unit price.

$$v_i(t) = \begin{cases} v_s, \ N_c = 1 \\ v_s + \dfrac{\mu}{N_c - 1} \displaystyle\sum_{j=1, j\neq i} w_j(t), \ N_c \geq 2 \end{cases} \tag{5}$$

where $w_{N_c} = (w_1, ...w_i..., w_{n_c})$. The computing nodes can offload tasks continuously due to overload and redundancy.

2) The transmission cost of IoT mobile devices i when offloading computation tasks is

$$J_i^t(t) = \xi p_i(t) = \xi (2^{\frac{r_i(t)}{B}} - 1) \frac{\sigma_0^2}{g_i(t)} = \xi (2^{\frac{-w_i(t)\Delta t}{B}} - 1) \frac{\sigma_0^2}{g_i(t)} \tag{6}$$

where ξ means the convert coefficient from unit transmission power to price.

3) Punishment cost at T timeslot of HDHNs

Since the goal of local computing and computation offloading is to finish all the published tasks in the concerned blockchain consensus time T, it can be penalized for the number of computation tasks that remains at time T. The penalized function (i.e., remaining cost) [35] for device i is stated as the following equation:

$$\varphi(X_i(T)) = \frac{\lambda}{1 + e^{-\varsigma X_i(T)}} - \frac{\lambda}{2} \tag{7}$$

ς and λ represent the steepness and maximum value of the penalty function, respectively.

Therefore, the total cost function of IoT mobile device i with respect to t is described as

$$J_i(t) = J_i^c(t) + J_i^t(t) \tag{8}$$

The penalized function $\varphi(X_i(T))$ is utilized to relax C4 constraint of the offloading computing tasks completed condition in the following Eq. (9). Here, if $X_i(T) = 0$, then $\varphi(X_i(T)) = 0$. When $X_i(T) > 0$, $\varphi(X_i(T))$ maintains a relatively large value to penalize the device, which is responsible for offloaded computing tasks but not complete tasks in time T.

Through the comprehensive cost analysis of the device i, the optimal control problem on computational power $w_i^*(t)$ that suitable for undertaking offloaded computation tasks can be obtained and modeled as

$$\begin{aligned} & w_i^*(t) = \arg\min_{w_i(t)} \left[\int_0^T J_i(t) dt + \varphi(X_i(T)) \right] \\ & s.t. \quad C1: \ dg_i(t) = \vartheta_i(t, g_i(t)) dt + \sigma_i(t) dW(t) \\ & \qquad C2: \ dX_i(t) = -r_i(t) dt = w_i(t) \Delta t dt \\ & \qquad C3: \ X_i(0) = X_0 \\ & \qquad C4: \ X_i(T) = 0 \end{aligned} \tag{9}$$

here X_0 means the initial data size to be computed. Provided that the parameters $g_i(0)$ and $(\vartheta_i(t), \ \sigma_i(t))$ (written as $(\vartheta(t), \ \sigma_b)$) can be obtained for $t = 0$ timeslot. The optimal solution of Eq. (9) in continuous time $[0, T]$ is induced through a Bellman function construction at time duration $[t, T]$ [35]. Then according to inverted time order, the equation can be solved. Let the state $S_i(t) = [X_i(T), g_i(t)]$. Therefore, the Bellman function (i.e., running cost function $c_i(t, S_i(t))$) for IoT mobile device i ($i \in N_c$) is described as:

$$c_i(t, X_i(T), g_i(t)) = \min_{w_i(t)} \left[\int_{q=t}^T J_i(q) dq + \varphi(X_i(T)) \right] \tag{10}$$

Definition 1: Existing a computational power strategy in terms of calculation offloading $w^*(t) = (w_1^*(t), ..., w_i^*(t), ..., w_{N_c}^*(t))$ serve as a Nash equilibrium for dynamic stochastic game of Eq. (9) only if $w_i^*(t)$ is the optimal solution for Eq. (9), namely,

$$w_i^*(t) = \arg\min_{w_i^*(t)} \left[\int_{q=t}^T J_i(q, w_i(q), w_{-i}^*) dq + \varphi(X_i(T)) \right] \tag{11}$$

here w_{-i}^* means the all IoT mobile devices' computation offloading strategies except for IoT mobile device i. According to the Nash equilibrium definition, no IoT mobile device can reduce its cost through changing their current computation offloading strategy.

Nash Equilibrium: Refer to [52] and by means of Taylor's expansion with respect to $c_i(t, S_i(t))$, existing N_c mutual

dependence solutions $c_i(t,S_i(t))$ for the N_c Hamilton-Jacobi-Bellman (HJB) equations relevant to the optimized issue of Eq. (9) can serve as an adequate condition for existing a Nash equilibrium, i.e.,

$$\min_{w_i(t)} [J_i(t)+w_i(t)\Delta t\partial_{X_i}c_i^*(t,S_i(t))+\vartheta(t)\partial_{g_i}c_i^*(t,S_i(t)) + \tfrac{1}{2}\sigma_b^2\partial_{g_ig_i}c_i^*(t,S_i(t))]+\partial_t c_i^*(t,S_i(t))=0 \tag{12}$$

Theorem 1: The optimal running cost $c_i^*(t,S_i(t))$ serve as the proof of existence of uniqueness of solutions of the HJB Eq. (12). Hence, the optimum offloading computational power can be calculated:

$$w_i^*(t,S)=Blog_2\left[\frac{Bg_i(t)}{\xi\sigma_0^2\Delta t^2\ln 2}(\Delta t\partial_{X_i}c_i^*(t,S)+v_i(t))\right] \tag{13}$$

Proof: According to the HJB Eq. (12), the Hamiltonian is sleek [53] and then

$$H(w_i(t),S_i(t),\nabla c_i(t,S_i(t)))=\min_{w_i(t)}\ [(2^{\frac{-w_i(t)\Delta t}{B}}-1)\frac{\xi\sigma_0^2}{g_i(t)} + v_i(t)w_i(t)\ +w_i(t)\Delta t\partial_{X_i}c_i^*(t,S_i(t)) + \vartheta(t)\partial_{g_i}c_i^*(t,S_i(t))+\tfrac{1}{2}\sigma_b^2\partial_{g_ig_i}c_i^*(t,S_i(t))] \tag{14}$$

Through the differentiation of the infima with regard to $w_i(t)$ and let it equals to 0, i.e.,

$$\frac{-\xi\sigma_0^2\Delta t\ln 2}{Bg_i(t)}2^{\frac{-w_i(t)\Delta t}{B}}+v_i(t)+\Delta t\partial_{X_i}c_i^*(t,S_i(t))=0 \tag{15}$$

consequently, the optimum $w_i^*(t,S_i(t))$ of IoT mobile device i can be obtained in Eq. (13). However, the computational price $v_i(t)$ is relevant to the computational power, which depends on all participants' computing power $\sum_{j\in N_c} w_j(t)$ when the player cost existing minimum value. Since the optimized problem (9) is to find a Nash equilibrium for the N-user non-cooperative dynamic stochastic game requires to solve N coupled HJB equations (12) for each IoT mobile device, namely, the computational complexity is $O(N^2)$. That means the N_c coupled HJB equations (12) needs to be decoupled by novel approach for each IoT mobile device.

C. Time-Variant Mean Field Game Approach

To reduce the computational complexity, the Time-Variant Mean-Field term (TVMF) is adopted to solve the cost-optimized problem. By the TVMF, we can transform the N-player dynamic stochastic game into a 2-player MFG, the computational complexity is $O(1)$ [35]. Therefore, the computational complexity can be reduced. The two players here consist of IoT mobile device offloading the computing tasks, and a mean field continuum including the large mass of mobile devices competing against the IoT mobile device that offloading the computing tasks. The iterative process will continue until a Nash equilibrium is achieved. It is obvious that the converge time of MFG, as a 2-player game. Also, it will converge fast [35] and not cause high time overhead, as the iterative algorithm includes only two game players instead of N players if without the mean field game.

Although the solving process in Eq. (12) is prohibitively complex with N_c becomes large, the estimated time-variant aggregative term (i.e., mean field (MF)) can replace the full

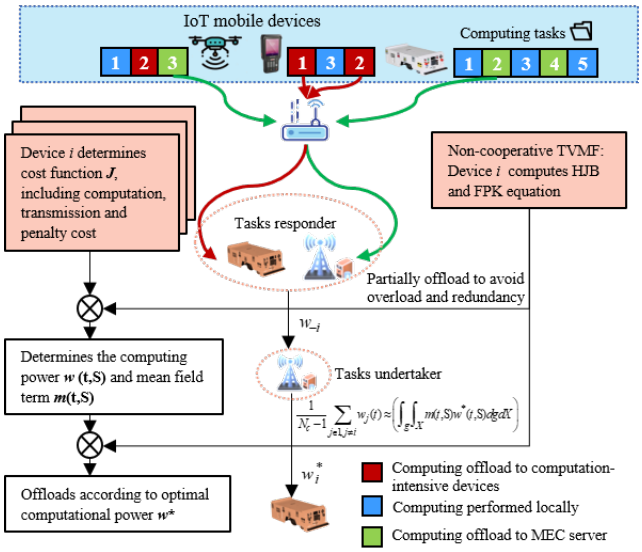

Fig. 3. The proposed TVMF computation offloading

value so that the individual player can obtain the optimal computing power and determine its strategy of offloading control. For the optimized problem of Eq. (9), the equivalent MF can be defined as:

Definition 2: Define the MF as a statistics distribution of the state space $S(t)=[X(t),g(t)]$ at t timeslot [35] [37], then denote the density of probability as

$$m(t,S)=\lim_{N_c\to\infty}M(t,S)=\lim_{N_c\to\infty}\frac{1}{N_c}\sum_{i=1}^{N_c}\mathbf{1}_{S_i(t)=S} \tag{16}$$

here $M(t,S)$ represents the ratio of IoT mobile devices at state space S in timeslot $t\in T$. If satisfy a given condition, the directive function $\mathbf{1}$ will return 1. If not satisfy, return 0. When the number of computing nodes N_c grows infinite, $M(t,S)$ can realize convergence to $m(t,S)$ characterized the state evolution of the IoT mobile devices with time [35] and satisfies

$$\int_g\int_X m(t,\mathrm{S})\,dgdX=1 \tag{17}$$

Therefore, the above expression needs to be redescribed based on the convergence of $M(t,\mathrm{S})$. since $S(t)=(X(t),g(t))$, then computational price is denoted as

$$v_i(t)=v_s+\frac{\mu}{N_c-1}\sum_{j\in 1,j\neq i}w_j(t) =v_s+\mu[\frac{N_c}{N_c-1}\int_g\int_X M(t,\mathrm{S})w^*(t,\mathrm{S})dgdX-\frac{1}{N_c-1}w_i(t,\mathrm{S})] \tag{18}$$

Then the MF term of computational price $v(t)$ can be derived with N_c tends to infinite

$$v(t)=\lim_{N_c\to\infty}v_i(t)=\lim_{N_c\to\infty}v_s +\mu[\frac{N_c}{N_c-1}\int_g\int_X M(t,\mathrm{S})w^*(t,\mathrm{S})dgdX-\frac{1}{N_c-1}w_i(t,\mathrm{S})] \approx v_s+\mu\left(\int_g\int_X m(t,\mathrm{S})w^*(t,\mathrm{S})dgdX\right) \tag{19}$$

1) Optimized problem MFG-based and solution: For a large population of computing nodes in HDHNs, the computational complexity is highly improved. Therefore, it is impractical to solve the point of Nash Equilibrium of N_c players for game through solving each player's computational power in Eq.(9). According to the definition of the MFG [37], the estimated aggregative value over time is named the time-variant mean field term (TVMF) irrespective of the other participants' information. To some extent, TVMF can preserve the player's privacy because it needn't all other players' computing power information. By transforming the optimized problem of Eq. (9) into an equivalent MF term form, each participant can calculate its optimal computation offloading control strategy (as shown in Fig. 3) based on the following rewritten HJB equation

$$\begin{aligned}&\min_{w(t)}\ [J(t,S)+w(t,S)\Delta t\partial_X c^*(t,S)+\\&\vartheta(t)\partial_g c^*(t,S)+\tfrac{1}{2}\sigma_b^2\partial_{gg}c^*(t,S)]+\partial_t c^*(t,S)=0\end{aligned}\tag{20}$$

Corresponding to Eq. (20), $w^*(t,S)$ can be obtained based on computational price $v(t)$

$$w^*(t,S)=Blog_2\left[\frac{Bg(t)}{\xi\sigma_0^2\Delta t^2\ln 2}(\Delta t\partial_X c^*(t,S)+v(t))\right]\tag{21}$$

Furthermore, through the FPK equation [35], the evolution of MF $m(t,S)$ is derived as

$$\begin{aligned}&m(t,S)+\partial_g(\vartheta(t)m(t,S))\\&+\partial_X(\Delta t w(t,S)m(t,S))-\tfrac{1}{2}\sigma_b^2\partial_{gg}m(t,S)=0\end{aligned}\tag{22}$$

According to [35], finite difference approach can be adopted to find the solution for the coupled HJB and FPK equations. Based on this method, discretizing the time interval of computation offloading $[0,\ T]$, the residual computation assignment volume $[0,X_0]$ as well as the channel state $[g_{min},\ g_{max}]$ over Δt, ΔX *and* Δg steps, respectively. Correspondingly denote indices α,β,γ to discrete time, residual task as well as the channel state, namely, $t=\alpha\Delta t,\ X=\beta\Delta X$ *and* $g=\gamma\Delta g$. Therefore, the finite difference equations of MFG are obtained:

$$\begin{aligned}\frac{\partial c^*(t,X,g)}{\partial t}&\approx\frac{c^*(\alpha+1,\beta,\gamma)-c^*(\alpha,\beta,\gamma)}{\Delta t}\\\frac{\partial c^*(t,X,g)}{\partial X}&\approx\frac{c^*(\alpha,\beta,\gamma)-c^*(\alpha,\beta-1,\gamma)}{\Delta X}\\\frac{\partial c^*(t,X,g)}{\partial g}&\approx\frac{c^*(\alpha,\beta,\gamma+1)-c^*(\alpha,\beta,\gamma)}{\Delta g}\\\frac{\partial^2 c^*(t,X,g)}{\partial g^2}&\approx\frac{c^*(\alpha,\beta,\gamma+1)-2c^*(\alpha,\beta,\gamma)+c^*(\alpha,\beta,\gamma-1)}{\Delta g^2}\end{aligned}\tag{23}$$

Through substituting the above equation into Eq. (20), then another expression can be described in backward reasoning when $\alpha=T,\ c^*(\alpha+1,\ \beta,\ \gamma)=\varphi(X(T))$,

$$\begin{aligned}&c^*(\alpha,\beta,\gamma)=[c^*(\alpha+1,\beta,\gamma)-(\tfrac{w(\alpha)\Delta t^2}{\Delta X})c^*(\alpha,\beta-1,\gamma)\\&+\tfrac{\sigma_b^2\Delta t}{2\Delta g^2}c^*(\alpha,\beta,\gamma-1)\ +\Delta t(\tfrac{\vartheta(\alpha)}{\Delta g}+\tfrac{\sigma_b^2}{2\Delta g^2})c^*(\alpha,\beta,\gamma+1)\\&+\Delta t J(\alpha)][1+\Delta t(\tfrac{-w(\alpha)\Delta t}{\Delta X}+\tfrac{\vartheta(\alpha)}{\Delta g}+\tfrac{\sigma_b^2}{\Delta g^2})]^{-1}\end{aligned}\tag{24}$$

Algorithm 1 Iterative algorithm for the TVMF updated strategy

1. **Initialize**: $m_0,c_0^*,w_0^*,v_0,k=0,k_{\max},\varepsilon$
2. **For** $(k<k_{\max};\ k++)$
3. **A**ccording to Eq. (24) and w_{k-1}^* to update c_k^*
4. **A**ccording to Eq. (21), c_k^* and v_{k-1} to update w_k^*
5. **C**alculating $\left|w_k^*(\alpha,\beta+1,\gamma)-w_{k-1}^*(\alpha,\beta+1,\gamma)\right|$
6. **If** $\left|w_k^*(\alpha,\beta+1,\gamma)-w_{k-1}^*(\alpha,\beta+1,\gamma)\right|>\varepsilon$
7. **R**epeating step 3—step 5
8. **else**
9. **end If**
10. **A**ccording to Eq. (25) and w_k^* to update m_k^*
11. **A**ccording to Eq. (19), m_k and w_k^* to update v_k
12. **end For**

Moreover, the FPK equation (22) are denoted by another form as

$$\begin{aligned}&m(\alpha+1,\beta,\gamma)\\&=\tfrac{1}{2}[m(\alpha,\beta+1,\gamma)+m(\alpha,\beta-1,\gamma)+m(\alpha,\beta,\gamma+1)+m(\alpha,\beta,\gamma-1)]\\&-\tfrac{\Delta t}{2g}[\vartheta(\alpha,\beta,\gamma+1)m(\alpha,\beta,\gamma+1)-\vartheta(\alpha,\beta,\gamma-1)m(\alpha,\beta,\gamma-1)]\\&-\tfrac{\Delta t^2}{2\Delta X}[w(\alpha,\beta+1,\gamma)m(\alpha,\beta+1,\gamma)+w(\alpha,\beta-1,\gamma)m(\alpha,\beta-1,\gamma)]\\&+\tfrac{\sigma_b^2\Delta t}{2(\Delta g)^2}[m(\alpha,\beta,\gamma+1)-2m(\alpha,\beta,\gamma)+m(\alpha,\beta,\gamma-1)]\end{aligned}\tag{25}$$

2) Iterative algorithm for TVMF updated strategy: The iterative algorithm is adopted to obtain the estimation value of the offloaded decision for computing power. The pseudocode of the iterative process for the TVMF updated strategy is shown in Algorithm 1. Firstly, initialize c_0^*,w_0^*,v_0 as zero, and the evolution density m_0 conforms to a normal distribution. Let ε be the converged threshold condition and is enough small. k_{max} means the maximum number of iterations. Then, the algorithm iteratively solves Eq. (24) and Eq. (25) to update the parameters m,c^*,w^*,v until satisfying threshold $\left|w_k^*(\alpha,\beta+1,\gamma)-w_{k-1}^*(\alpha,\beta+1,\gamma)\right|<\varepsilon$ or $k<k_{\max}$.

After updating the TVMF by algorithm 1, the optimal computational power at the next timeslot can be estimated, then the thread of consensus will be open to verify the computing results.

IV. Adaptive-aware PBFT Consensus in HDHNs

In HDHNs, nodes may join/leave the network due to their high mobility. The traditional PBFT consensus protocol is inefficient for the variant number of nodes and not suitable for the highly dynamic network. Accordingly, we devise a new AAPP consensus algorithm to dynamically manage the node's join and out of the network and guarantee the security and efficiency of offloading computing results in the dynamic network.

A. Nodes Dynamically Join/Exit and Domains Formulation

According to the definition of scalability [43], a consensus node can safely enter/exit the blockchain system and the performance of current system is not affected. In a consortium blockchain, it is assumed that the number of participants is preset before consensus and will not allow nodes freely enter or bow out of the system halfway. In addition, with

the increase of the number of nodes, communication overhead is also very expensive. Therefore, it is expected to realize a reliable and low bandwidth cost distributed computing and adapts to HDHNs. The size of the network should change in a polynomial way due to the addition or deletion of nodes [22].

As mentioned before, a dynamic and synchronous HDHN with regard to discretized time t is supposed. N represents the maximal nodes in blockchain system in timeslot t. $\#D$ means the amount of domains, $|D_k|$ denotes the size of domain D_k. The network size dynamically changes with the nodes entering and exiting. Based on [22], a domain of logarithmic size including exceed two-thirds of correct nodes is chosen. Then, the selective domain is used to split network as $\#D$ domains, $\{D_1, D_2, ..., D_{|D|}\}$, and each domain size is $slogN$ (choosing a security parameter s meet the requirements: the more reasonable the selection of s, the fewer opportunity the attacker has ability of controlling exceed a third of members in a domain). Herein, the security parameter s is determined according to the adaptivity calculation of nodes introduced in the subsequent section.

The key element of determining protocol security lies in each domain including exceed two-thirds of the correct members. The sufficiency is proved in [22] to guarantee the validity of the PBFT algorithm that every domain consists of the majority of correct nodes when facing node's joining/leaving behavior under high dynamics. A representative domain of logarithmic size including exceed two-thirds of correct nodes is chosen. Then it is used to realize network partition and obtains multiple domains. Every domain size is $slogN$ (as shown in Fig. 4). Additionally, to hold the correctness of protocol when facing node's joining/leaving behavior under high dynamics, the network's shuffle (i.e., the nodes in one domain are exchanged with nodes randomly selected in other domains) is very crucial. It involves four phases and each phase with $poly\log N$ communication overhead:

Join: The cluster head node of the network is responsible for getting in contact with a new node i entering the network. Its detailed function demonstrated in next part. Furthermore, the cluster head node chooses a domain using a random walk. Then the selected domain handles by adding the node i and applying interchange method to its nodes for one timeslot.

Segregate: When domain's size is greater than $ls\log N$ (here, l is a constant parameter larger than $\sqrt{2}$), the domain is partitioned into two domains. The old domain keeps the original and neighboring information, whereas the new domain is added into $\hat{G}$ by adding $O(\log N)$ random edges to connect to the graph [22].

Leave: When a node leaves the network or other nodes in the same domain detect its absence, the domain's cluster head node removes this node from node list, and the domain applies the algorithm exchange. Moreover, a domain receiving nodes from this domain also executes an exchange for all its nodes.

Amalgamate: When the size of a domain D is less than $slogN/l$. In such a case, a random domain D' is selected randomly and then lets the nodes of D into D'.

B. Consensus Process

In the proposed AAPP, four different roles exist: network cluster head node, Ordinary consensus nodes, Leader and Storage nodes.

The duties of **network cluster head node** includes: **Scanning** timely the state of the heartbeat of other nodes based on the node's heartbeat mechanism. **Receiving** the information sent by the newly registered nodes and verify its legitimacy, determine whether to approve the nodes' joining blockchain system, distribute the node IP, add this new entrant to node list then broadcast the entrant's information in HDHNs. **Selecting** dynamically the primary node according to the adaptivity calculation in the following subsection. **Validating** synchronization: Judging the node's block whether up to the maximum block height or not. If not, synchronize it.

Ordinary consensus nodes: As before analyzed, PBFT adopting most ballot of greater than 2/3 honest members, $N_{con} \geq 3f+1$ members serve as the total number of consensus nodes that can put up with the maximal f nodes appeared Byzantine conducts. In our AAPP, the consensus nodes are chosen by the node's adaptivity evaluation which tops forty percent apart from the primary node in one domain.

Leader: The cluster head node sorts the node's adaptivity and randomly chooses one of the top ten percent as the master node, which is responsible for packing the calculation data, generating the preparation block quickly and collecting the voting information. Breakdown of the leader results in the election of a new master node (often means view change), the election principles will be elaborated in the next subsection.

The consensus protocol in one domain consists of three phases (as shown in Fig .4):

Pre-prepare: The master node sends a batch of transactions (offloading computing results, i.e., pri-block) to all selected ordinary consensus nodes. The pri-block in the form of $< priblock, Sig_p >$ (Sig_p is the signature of the primary peer).

Prepare: Consensus nodes check the pri-block and feedback their agreement/disagreement to the leader. On the basis of vote information, the leader generates the regular block $< pri\_hash, t, m\_root, Sig_{con} >$. (where pre_hash means the prior block hash, t is the timestamp, m_root represents the Merkel root [27], which comprises of the hashed transaction, Sig_{con} denotes the signature of consensus peer) and broadcast in the network.

Commit: Once the leader gathers $2f+1$ validated messages, it conducts the solicited update and notifies the client consequently.

Based on the above three phases, it can be calculated that the number of communications required to complete a AAPP consensus process in one domain is $(N_{con}-1)+(N_{con}-1)+(N_{con}-1) = 3(N_{con}-1)$. Namely, the communication complexity is $O(N_{con})$. It has significantly reduce compared to the original PBFT $O(N^2)$ [40]. In addition, since AAPP needs more communication signals than original PBFT to control the joining/leaving process of devices, the communications cost between consensus domain is $logN$ when considering the domain formulation. Hence, the total number of communications required to complete a AAPP process is $logN \times n_{con}$. Therefore, this AAPP algorithm is employed to

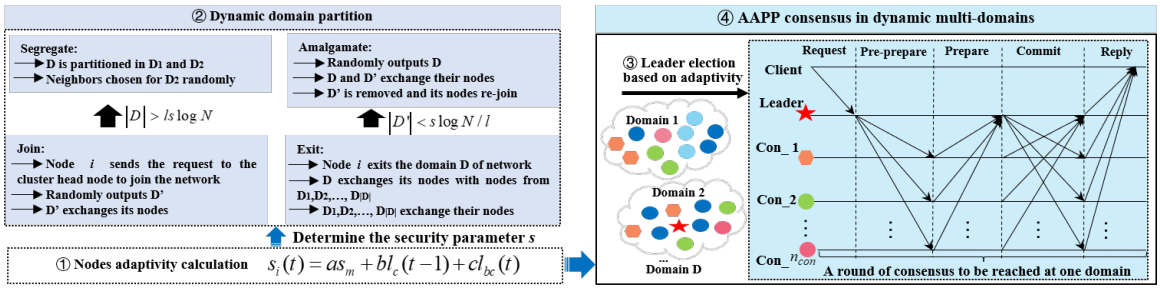

Fig. 4. AAPP Blockchains Consensus Process in HDHNs.

execute process of offloading results in HDHNs and only have message complexity of $O(logN \times n_{con})$.

C. Adaptivity-aware leader election of domains

After executing the computing tasks, the offloading computing results need to go through a round of consensus before on-chain to ensure the truthfulness of the results. The conventional PBFT consensus protocol elects the leader randomly according to v mod n [54], which is not adaptive for the dynamic network because the total number of nodes is constant. Our proposed adaptivity-aware leader election method comprehensively considers network dynamics, node's historical performance and offloading computing tasks completion in the HDHNs. Then dynamically calculate the node's adaptivity of the consensus nodes and choose the most adaptive node in the top 10 percent to undertake the role of a primary node in the current round of consensus. The adaptivity of node i denoted by $s_i(t)$ and dynamically update. The election principle at the t timeslot is as follows:

$$s_i(t) = as_m + bl_c(t-1) + cl_{bc}(t) \tag{26}$$

where s_m represents the node's mean adaptivity during past historical time in terms of its performance, whether successfully participating in consensus or not, and a is the ratio of the historical performance. $l_c(t-1)$, b mean the $i-th$ blockchain node's computing tasks completed degree at $t-1$ time slot and its weight, respectively, $l_{bc}(t)$, c, represent the amount of calculation tasks to be completed at t time slot and its weight. All weights satisfy $a+b+c=1$.

D. Security Analysis

1) Safety: After performing a polynomial operations of enter and exit (including segregating and amalgamating of domains), each domain has greater than two-thirds of honest nodes if the proportion τ of Byzantine nodes dominated by an antagonist is smaller than $1/3-\varepsilon$ (for some constant $\varepsilon > 0$, which is irrelevant to n).

-State of a domain after exchange

In the highly dynamic network, suppose that the nodes either enter or exit at each timeslot. As a result, the partition or amalgamation operation of domains can occur. When a domain executes an amalgamate operation, its nodes re-enter the network in a subsequent time slot causing normal enter operations. Denote a domain D, f_t^D means the ratio of Byzantine nodes in D at time slot t.

Lemma 1: (2/3 of correct nodes in a domain). If a domain D has interchanged its entire nodes at timeslot t, then $P(f_t^D > \tau(1+\varepsilon)) \le n^{-\zeta}$ holds for arbitrary positive constant ζ once the security parameter s is chosen rationally.

Proof: Once a domain D exchanges one of its nodes with other domain, this domain is chosen randomly in accordance with the probability distribution $(|D|_t/n)$, and then one node is selected randomly and uniformly from the domain. Under the condition, the possibility of executing one exchange operation with a Byzantine node is τ. Through standard Chernoff bound mathematical theory [22], the conclusion with regard to the number X (a binary random variable $X=0$ or $X=1$) of Byzantine nodes among domain $|D|_t$ can be induced: $P(X > (1+\varepsilon)\tau|D|_t) \le e^{-\varepsilon^2\tau|D|_t/3}$. Therefore, when s is choosen rationally for some constant ζ, $|D|_t \ge (s\ log\ N)/l$, $P(X > \tau|D|_t(1+\varepsilon)) \le N^{-\zeta}$ holds.

The above lemma is an inference of the Chernoff bound mathematical theory. It is indicated that the sufficient condition obtaining greater than two-thirds of honest nodes in a domain is $\tau+\varepsilon < 1/3$.

-Advancement of the divergence

It can be summarized that one domain exchanges entire nodes at each time when satisfy $\tau(1+\varepsilon) < 1/3$, greater than two-thirds of correct nodes can be obtained in the formulated domain. Moreover, it can be further proven that this attribute also applies in two exchanges. To verify the applicability, a series of operations such as joining and exiting are considered by analyzing a domain D. Firstly, if a domain has fewer than $\tau(1+\varepsilon/2)$ Byzantine nodes, then after exchanging $O(\log N)$ nodes in the domain, there will be no greater than $\tau(1+\varepsilon)$ Byzantine nodes in the domain. Subsequently, it can be proven that if the domain has Byzantine nodes in a ratio between $\tau(1+\varepsilon/2)$ and $\tau(1+\varepsilon)$, then after exchanging $O(\log N)$ nodes in the domain, its proportion of owning Byzantine nodes is lower than $\tau(1+\varepsilon/2)$.

Lemma 2: Given a domain D has fewer than $\tau(1+\varepsilon/2)|D|$ Byzantine nodes, after performing $O(\log N)$ exchanges with randomly chosen nodes, the domain contains no more than $\tau(1+\varepsilon)|D|$ Byzantine nodes.

Proof: A domain D with a Byzantine node ratio of p has a maximum possibility of having $p(1-\tau)$, which reduces this ratio by $1/|D|$, and at least a possibility of having $(1-p)\tau$, which increases it by the same amount, i.e. $1/|D|$. If this ratio is at most $\tau(1+\varepsilon/2)$, it can be proven that it increases ε with possibility $1/N^{\zeta}$, because ζ is arbitrarily large, depending

upon the selection value of s.

The proportion of Byzantine nodes in one domain is determined by the control line with an initial status of $\tau(1+\varepsilon/2)$, which improves or reduces $1/|D|$ with possibility τ. This control line will not greater than $\tau(1+\varepsilon)$ after the $O(\log N)$ operations (as mentioned earlier, $s\log N/l\,|D| \le sl\log N$).

Let s is chosen rationally and $T^{exchange}$ means the number of exchanges. The complexity of $T^{exchange}$ is $O(\log N)$ and then there is a constant M that satisfies $T \le M\log N$. According to Azuma Hoeffding's inequality [22], it can be concluded that

$$\begin{aligned} &Prob(p^C > \tau(1+\varepsilon/2)) < e^{-\varepsilon^2/4\sum_{i=1}^{exchange} 1/|D|^2} \\ &\le e^{-\varepsilon(s/l)^2\log^2 N/4(M\log N)} \\ &= e^{-\varepsilon(s/l)^2\log(N)/4M} = n^{-\zeta} \end{aligned} \tag{27}$$

Likewise, if the proportion of Byzantine nodes in one domain exceeds $\tau(1+\varepsilon/2)$, after the $O(\log N)$ operations, the proportion of Byzantine nodes in the domain is lower than $\tau(1+\varepsilon)$.

Lemma 3: Provided domain D, where the proportion of Byzantine nodes is between $\tau(1+\varepsilon/2)$ and $\tau(1+\varepsilon)$ $(\varepsilon > 0)$, then after exchanging $O(\log N)$ with selected nodes randomly, the proportion of Byzantine nodes of this domain is lower than $\tau(1+\varepsilon/2)$.

Proof: Here, the proportion of Byzantine nodes will reduce by $1/|D|$ with a possibility more than or equals to $\tau(1+\varepsilon/2)$ and improve by $1/|D|$ with a possibility of τ. Consequently, when starting from a ratio of up to $\tau(1+\varepsilon)$, after $O(\log N)$ operations, the proportion of Byzantine nodes in the domain is lower than $\tau(1+\varepsilon/2)$.

Employing Lemma 2 and Lemma 3 in the partitioned domains, we notice that there are always greater than two-thirds of the honest node in each domain with an adequate s for a sequence γ whose length is polynomial.

Theorem 2: After polynomial operations in N at each time, all domains contain greater than two-thirds of correct nodes.

Proof: Note that to employ the above lemmas, it is necessary to guarantee that the nodes selected randomly and uniformly substitute for the interchanged nodes. This requirement is met by the proposed enter and exit operations. Nevertheless, if a domain D with which D' has interchanged nodes, then the possibility that D' obtains a Byzantine node is equal to the ratio of Byzantine nodes in D. It is the reason why interchange all the nodes in D. Next, regarding a time sequence $t_1,...,t_k,...$, at t_k, the proportion of nodes dominated by the antagonist in D is lower than $\tau(1+\varepsilon/2)$, and the proportion becomes larger or equal to $\tau(1+\varepsilon/2)$. Then at t_{k+1}, it becomes greater or equal to $\tau(1+\varepsilon/2)$ and lower than $\tau(1+\varepsilon)$. Lemma 3 guarantees that time slot t_{k+2} comes within $O(\log N)$ steps, and Lemma 2 guarantees that between t_{k+1} and t_{k+2}, the antagonist never dominates that exceed a $\tau(1+\varepsilon)$ proportion of nodes in a domain. Through a union bound over all domains, the announced security can be obtained.

2) Liveness: The view-change algorithm guarantees liveness by enabling the blockchain system to make progress in the event of the primary node becomes invalid [17]. To prevent indefinite waiting, a replica initiates a timer upon accepting a request and ceases the timer when it is no more waiting to perform the demand.

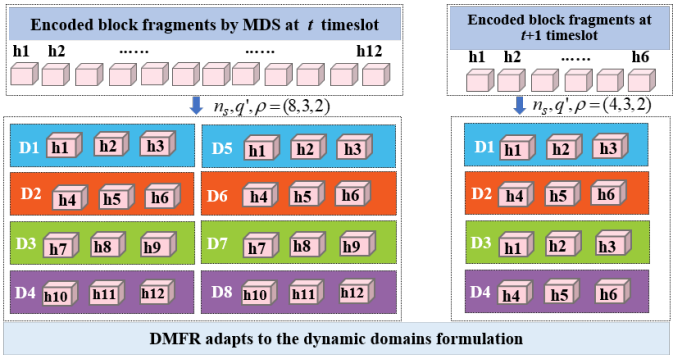

Fig. 5. DMFR adapts to the dynamic domains formulation.

The backup broadcasts a view-change information for view $v+1$ [17] when its timer expires, and ceases operation in the current view v. The view-change message includes the highest-committed sequence number and other prepared messages. The new view will be assigned view number $v+1$. Once the new leader receives $2f$ correct view-change message for the next view $v+1$ from other replicas, it broadcast a new-view message to all other replicas. This new-view message contains all the view-change messages and all prepared messages that have not been committed. The attached view-change messages serve as proofs of correctness for other replicas. Upon accepting the new-view messages, a replica enters the next view, updates its local state accordingly and recomputes. The view-change protocol ensures that the same sequence number cannot be assigned to different demands in different views. Therefore, AAPP always guarantee the safety property by distributing a unique sequence number to each request.

V. Dynamic Multi-Domain FR uncoded repair storage (DMFR)

With the IoT mobile devices registered as blockchain nodes, it encounters the problem of node failure in HDHNs. Therefore, efficiently and reliably recovering data of the failed node is essential. We propose the DMFR to reduce the repair overhead. After the AAPP consensus is completed, the storage nodes are responsible for storing the formal block. By dividing all the storage nodes into domains, the nodes in one domain just save the partial blocks and their header based on the DMFR with variant redundancy strategy to decrease the storage burden, reduce the repair bandwidth and adapt to the dynamic network condition. The relevant notation and its meaning are shown in Table III.

In our storage scheme, the formal block voting by the AAPP consensus process is divided into h fragments based on Maximum Distance Separable (MDS) (n,k) [50]. The internal repetition code is constructed by the shadow method. The concrete steps are as follows:

The generated formal blocks $B = \{b_1, b_2, ..., b_k\}$ which encoded $\{1,...,h\}$ fragments by MDS are repeatedly stored in $N_s = 1,2,\ldots,n_s$ nodes lies in dynamically formulated domains (shown in Fig. 5), each node saves q' encoded fragments and

TABLE III
NOTATION AND MEANING

Notation	Meaning
$B=\{b_1,b_2,\ldots,b_k\}$	The formal block generated
$h=\{h_1,h_2,...\}$	The number of fragments that the formal block is divided into
(n,k)	MDS encoding parameters
Z	A set containing k elements
ψ	The set contains y subsets
A	The shadow sub-incidence matrix
$q^{'}$	The number of encoding fragments saved by each storage node
ρ	Redundancy of segmented block fragments stored
$\partial\theta$	The shadow set
$\partial\theta_\chi$	The sub-shadow set
$\partial\psi^{'}$	The set after deleting a subset of each sub-shadow set $\partial\theta_\chi$
A'	The new incidence matrix after exchanging rows and columns of the shadow sub-incidence matrix A corresponding to $\partial\psi^{'}$

TABLE IV
THE DIFFERENT STORAGE CODES SCHEME COMPARISON

	Node repair	Reed-Solomon	Simple regenerating	FR Steiner ternary	FR Hadamard matrix	FR Shadow
RBO	Single node	B	$(f+1)B/k$	$3B/k$	$3B/k$	$3B/k$
	Two nodes	B	$2(f+1)B/k$ or B	$6B/k$	$3B/k$ or $6B/k$	$4B/k$ or $6B/k$
RL	Single node	k	$2f$	3	3	2
	Two nodes	k	k	6	3 or 6	2 or 4

every encoded fragment must belong to ρ nodes. According to the principle of FR code, when $\rho=2$ based on a regular graph, the parameter above introduced should conform to the equation [49]:

$$h\rho = n_s q' \tag{28}$$

According to the DMFR, the storage overhead of blockchains decreases h/q' compared with the entire save while the number of nodes is constant. However, it is not adaptive to changeable block storage under the dynamic network condition since the redundancy of FR is fixed.

Therefore, for the achievement of the variant redundancy FR code, the constructed process of the shadow-based method can be described as follows:

Step 1: Let Z is a set containing k elements, and there must be a $(\rho+1)$ -meta set ψ, $((\rho+1)<n)$, and the set ψ meets the following two conditions: a) The set ψ contains y subsets, each subset contains $\rho+1$ elements, and the set ψ has n different elements of set Z. b) There are no identical elements in the subset.

Step 2: Get its shadow set $\partial\theta$ from the set ψ and the set $\partial\theta$ contains y sub-shadow sets $\partial\theta_\chi(0<\chi<y)$. Delete a subset of each sub-shadow set $\partial\theta_\chi$, including ρ subset. Then, the shadow set formed is $\partial\psi^{'}$. According to the method of constructing partial repeat codes based on the shadow in [55], heterogeneous FR codes with the same storage capacity and repetition degree of ρ or ρ–1 can be constructed from the set $\partial\psi^{'}$. Meanwhile, based on the system's storage capacity, the shadow set $\partial\theta_\chi$ can be deleted to meet the repeatability requirements.

Step 3: Exchange rows and columns of the shadow sub incidence matrix A corresponding to the shadow set $\partial\psi^{'}$ to obtain a new incidence matrix A'.

Step 4: Each row in matrix A' represents a storage node, and the $i-th$ row in matrix A' represents the $i-th$ storage nodes. The FR code is constructed by $N_s=\{j:a_{ij}=1\}, j=1,2,\ldots,n$, i represents the $i-th$ FR node, and a_{ij} stands for the value of row i and column j of matrix A'. N_s denotes the storage node of FR code. The data block in N_s is characterized by the number of columns corresponding to all 1 in row i of matrix A'. Extract the number of columns to obtain the data block stored in a node. Heterogeneous FR codes with storage capacity ρ or ρ–1 and repeatability ρ can be constructed for each node. The comparison of different code schemes in terms of repair bandwidth overhead (RBO) and repair locality (RL) is shown in Table IV. It is shown the advantages of the heterogeneous FR codes constructed based on the shadow.

VI. SIMULATION RESULTS AND ANALYSIS

Herein, we validate the offloading computation power and cost through conducting several sets of parameters comparison in MATLAB 2019b. The experimental environment is deployed on the OpenStack cloud platform and multiple virtual machines are created as blockchain nodes. All the blockchain nodes are configured develop software JDK 1.8 and MySQL 5.7. Considering the mobility of IoT mobile devices and maintain the correctness of domains, we partition the network into different domains in HDHNs. Especially, we split 2, 4, and 8 domains and the nodes in each domain are 4 or 5. When the domains dynamically formulate, the AAPP consensus is running. After generating a formal block, the DMFR strategy is conducted for computing results storage.

A. Offloading computing strategy TVMF-based

Herein, numerical simulations are conducted to analyze the computation offloading strategy in terms of computational power and cost in the HDHNs. Refer to [35], the detailed system parameters setting are shown in Table V. Moreover, we model the channel variation [35] as

$$g(t)=g(0)+A\sin(f_0 t) \tag{29}$$

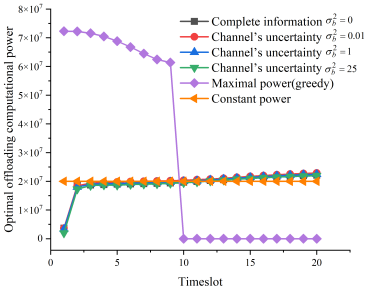

Fig. 6 The offloading computational power under different strategies

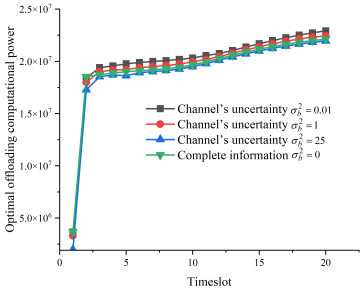

Fig. 7 The computational power affected by channel dynamics

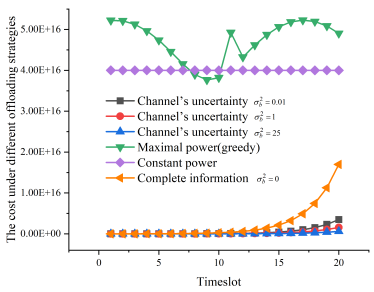

Fig. 8 The cost under different offloading strategies

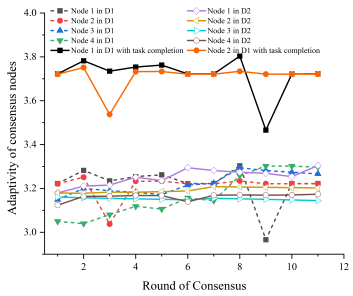

Fig. 9 The node's adaptivity evaluation

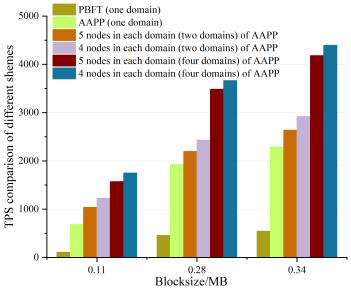

Fig. 10 The performance of AAPP

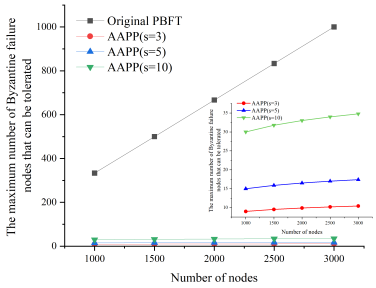

Fig. 11 The maximum number of Byzantine failure nodes tolerated

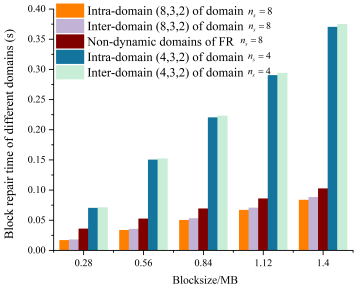

Fig. 12 Block repair time overhead $\rho=2$

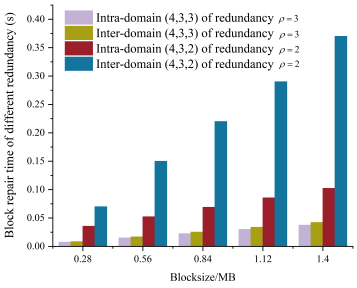

Fig. 13 Block repair time overhead $n_s=4$

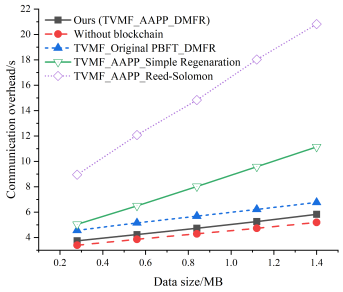

Fig. 14 Comparison of communication overhead under different schemes

here, $g(0)=2\times10^3$, $A=10^3$ and $f_0=0.4$. Consequently, channel's dynamic $\vartheta(t)$ can be defined as $\vartheta(t)=Af_0\cos(f_0t)$.

To depict the channel's uncertainty, we set the stochastic term $\sigma_i(t)dW(t)$ of channel model with variance $\sigma_b^2=0.01$, $\sigma_b^2=1$ and $\sigma_b^2=25$ [56]. There are three offloading schemes, i.e., maximal computational power (similar to greedy), constant computational power and complete information [35] are selected as a benchmark to analyze the performance of the proposed TVMF scheme. For the complete information scheme, that means the channel information is known so the time-varying channels is predictable and the uncertainty equals to zero, i.e., $\sigma_b=0$.

Fig. 6 shows the variation of computational power with the timeslot. For the greedy offloading scheme, the maximal

TABLE V
DETAILED SYSTEM PARAMETERS

Parameters	Value
Number of time slots	20
Time step Δt	0.5ms
Initial data size X_0	$2.5\times10^5 bits$
Data offloading step ΔX	$5\times10^3 bits$
Channel grid resolution Δg	$3\times10^8 bits$
Channel bandwidth B	$5MHz$
Maximal power p^{max}	$23dBm$
Noise power σ_0^2	$50dBm$
Channel's uncertainty σ_b	$(0,10]$
Convert coefficient ξ	10^{-2}
Single pricing of computing node v_s	10^{-8}
Initial mean field $m(0,X)$	$\mathcal{N}(8\times10^6,\ 2.5\times10^{11})$

computational power is adopted to offloads all the computing tasks as quickly as possible and then avoid the punishment at deadline. Under this condition, once the calculation tasks are fully offloaded, the computational power is decreased to zero. It doesn't consider the load balance and approximate to the greedy scheme. Although the constant computational power scheme is relative stable, there is a lack of consideration of requirement of time-varying offload computing. Different from the other two offloading strategies, the proposed TVMF scheme not only take the time-varying offload requirement into account but can adapt to the computation price variations. In particular, as the channel uncertainty decrease from 25 to 0.01 (as shown in Fig. 7), our scheme increases the computational power and accomplish the calculation tasks at a faster speed before the deadline T.

It can be demonstrated from the Fig. 8 that the cost changes under different offloading strategies. The proposed TVMF scheme realizes the lower running/cumulated cost compared with maximal and constant offloading schemes when conducting the offloading calculation tasks. It demonstrates that the effectiveness of the proposed solution. Especially, under the uncertainty channels of $\sigma_b^2 = 25$, the lowest cumulated cost can be achieved. Compared to the complete information $\sigma_b^2 = 0$, it has the higher cost due to it needs all players' accurate information of subsequent channels to make its offloading-optimized decisions. In another word, the lower uncertainty of channels, the higher cost of the offloading strategies.

B. Dynamic Consensus Performance analysis in HDHNs

Since the node's joining and leaving in HDHNs, it is critical to ensure the high efficiency to meet the requirement of dynamic computing results consensus onto the blockchain. Through running the AAPP blockchain consensus mechanism on the afore-stated environment configuration, the adaptivity of node in the top ten percent is elected as a leader on each round of consensus. When the mean adaptivity s_m is 10 and its weight a=0.3, taking the example of two domains, the node's adaptivity is shown in Fig. 9. According to the evaluation results, we can see that the adaptivity of nodes varies with each round of consensus. In particular, for both node 1 in D_1 at round 9 and node 2 in D_1 at round 3, the value of adaptivity is relatively smaller than others. That may result from these two nodes without successfully participating in consensus. For node 3 and node 4 in the domain D_1 as well as node 2 in the domain D_2, their adaptivity is gradually improved after executing 11 rounds of consensus. It is shown that these nodes are more likely to be selected as leader and adapt to process the computing results in HDHNs. Through considering the computing tasks completion $l_c(t-1)$=0.2, b=0.1, $l_{bc}(t)$=0.8, c=0.6, it has been shown that the possibility of these two nodes becoming leader can be significantly improved at the next time slot compared with the existing works.

Moreover, we analyze the transaction process times of 2 and 4 domains at t timeslot of dynamic network for the proposed AAPP solution compared to the AAPP one domain and original PBFT, respectively. The processing computing transactions per second (TPS) performance of each domain with 4 and 5 nodes is considered. All domains satisfy the correctness of exceeding two-thirds of normal nodes. From Fig. 10, TPS of the AAPP consensus algorithm is far higher than the original PBFT for one domain since the communication complexity of PBFT is $O(N^2)$ [40] whereas the AAPP is $O(logN \times N_{con})$. Focusing on the two domains and four domains formulation, the TPS of both sides has a significant ascend with the rise of block size from 0.11MB to 0.34MB. It is noteworthy that different number of nodes results in unequal TPS performance in the same domain. The more nodes are implied (the more interactions) the lower TPS. For example, the speed of processing transactions of 5 nodes is inferior to the 4 nodes. Except for this, we can obtain that the number of dynamic domain formulations becomes significant. The TPS is gradually improved because multi-domains can process computing outcomes in parallel.

C. Comparative Analysis of Safety Performance for Consensus

The consensus mechanism can provide failure tolerance and resist system errors and attacks, ensuring the system with strong security. As analyzed before, security parameter s is relevant and notable to the safety of the blockchain systems. Moreover, the security parameter s is determined according to the adaptivity calculation of nodes. Therefore, choosing suitable security parameter based on the node's adaptivity ($s = 3$). Furthermore, $s = 5$ and $s = 10$ are selected to make a comparison. As shown in Fig. 11, it demonstrates the variation of maximal number of byzantine failure nodes tolerated with the increasement of node numbers. Provided that the total amount of blockchain nodes is 3000, for original PBFT mechanism, it requires the maximum number of nodes with failure tolerance reached to 1000. In contrast with the original PBFT, the proposed AAPP needs tolerant far smaller fault nodes, it can decrease the number of the attacker has ability of controlling exceed a third of the nodes in a domain and further improve the security of the blockchain-based computing sharing systems.

D. DMFR Repair Evaluation

For the DMFR scheme analysis, the generated formal blocks are encoded in $h = 4$, $h = 6$, or $h = 12$ fragments, respectively. When $\rho - 1$ nodes are offline, the FR code can maintain the uncoded repair feature. According to the principle of constructing heterogeneous FR encoding parameters shadow-based, we adopt the parameters $n_s, q', \rho = (8,3,2)$ of FR comparing with $n_s, q', \rho = (4,3,2)$ and the condition of $\rho = 3$, which $n_s, q', \rho = (4,3,3)$. Fig. 12 illustrates that the block repair time continuously increases with the increase of block size. When the number of fragments stored in each node is 3 and the redundancy equals to 2, the block repair time of intra-domain $n_s, q', \rho = (8,3,2)$ of domain $n_s = 8$ is reduced by 77% compared to intra-domain $n_s, q', \rho = (4,3,2)$ of domain $n_s = 4$. This indicates that the more domains are divided, the shorter the repair time. Furthermore, compared to the scheme of non-dynamic domains with FR (8,3,2), the average repair overhead of DMFR is decreased by 49%, indicating

the advantages of the proposed DMFR scheme. When a block fragment of a node in one domain is lost, a request needs to be sent from within the domain (i.e., intra-domain) or other domains (i.e., inter-domain) for repair due to the segmented block fragments being stored in the formed multiple domains. Therefore, the block repair time in terms of inter-domain and intra-domain is analyzed. As for $n_s, q', \rho = (8,3,2)$, both the block repair time of inter-domain and intra-domain are rising when the formal block size increasing from 0.28MB to 1.4MB. Similarly, the block repair time of $n_s, q', \rho = (4,3,2)$ in inter-domain is as quickly as intra-domain because the stored formal block is partitioned into smaller fragments. For once repair, the time of block fragments send to the requester in the domain is almost the same as the cost in the inter-domain thanks to the high-speed transmission of 5G communication. Under the same domains $n_s = 4$, the different redundancy $\rho = 2$ and $\rho = 3$ also leads to distinct block repair time overhead (as shown in Fig. 13). Specifically, when the number of each block fragment stored in the domain is more, i.e., $n_s, q', \rho = (4,3,3)$, the block repair time is reduced by 89% compared to $n_s, q', \rho = (4,3,2)$. The advantage of the proposed DMFR lies in ensuring loss data reliably recover from fault node and greatly reducing the repair time compared to other storage strategies without considering dynamic multi-domain fractional repetition storage [50].

E. Communication Overhead Performance Comparison

From the Fig. 14, it can be observed that the time overhead of all schemes is increased with the processing data size become large. Although the addition of the blockchain make total time overhead rise, the magnitude of increasement is only 9% compared to the scheme without blockchain. Moreover, the total communication overhead of the proposed scheme is also lower than other three schemes [40] [55] (i.e., computing offloading decision adopt TVMF, but consensus algorithm and storage strategy are different). Adding additional blockchain stages will not significantly affect computing efficiency and can guarantee the security and reliability of computing. That means communication overhead performance of our additional blockchain stage outweighs the importance of faster computing/storage. By comprehensively consider the balance between security and communication efficiency, our scheme is suitable for the shared dynamic computing environment.

VII. Conclusion and Future works

Focusing on IoT mobile devices' mobility and the high dynamics of heterogeneous network topology, blockchain technology emerges as a powerful tool to establish trusted computing resources sharing environments. However, whenever devices respond to the published computing tasks in HDHNs, there is a risk that it exits the network and leaves the remaining tasks noncomputed. Therefore, we model the offloading computing tasks of computing nodes in HDHNs as a dynamic stochastic game. To tackle the coupled cost function with a large population of IoT mobile devices, the TVMF is exploited to solve the offloading computing control strategy and can significantly reduce the complexity. Meanwhile, an AAPP consensus mechanism is designed to dynamically formulate domains for network correctness, perform leader election and quickly verify computational results. Furthermore, aiming to reduce the storage pressure and repair bandwidth of blockchain, we propose a DMFR scheme with variant redundancy. The proposed solution turns out to be effective and feasible for experimental results. In future research, we plan to take into the choice of domain not only logical grouping of nodes but geographical distance consideration, then optimize inter-domain communication latency.

Acknowledgments

The authors thank Yunnan Fundamental Research Projects (No. 202301AV070003), the Major Scientific and Technological Projects in Yunnan Province (No. 202002AB080001-8). The authors also extend their appreciation to King Khalid University for funding this work through Large Group Project under grant number RGP.2/312/44.

References

[1] Y. Wang, F. R. Yu, H. Tang, and M. Huang, "A mean field game theoretic approach for security enhancements in mobile ad hoc networks," *IEEE Transactions on Wireless Communications*, vol. 13, no. 3, pp. 1616–1627, 2014.

[2] Y. Feng, W. Zhang, X. Luo, and B. Zhang, "A consortium blockchain-based access control framework with dynamic orderer node selection for 5g-enabled industrial iot," *IEEE Transactions on Industrial Informatics*, pp. 1–1, 2021.

[3] S. Garg, K. Kaur, G. Kaddoum, P. Garigipati, and G. S. Aujla, "Security in iot-driven mobile edge computing: New paradigms, challenges, and opportunities," *IEEE Network*, vol. 35, no. 5, pp. 298–305, 2021.

[4] M. Haus, M. Waqas, A. Y. Ding, Y. Li, S. Tarkoma and J. Ott, "Security and Privacy in Device-to-Device (D2D) Communication: A Review," in IEEE Communications Surveys & Tutorials, vol. 19, no. 2, pp. 1054-1079, Secondquarter 2017.

[5] F. Song, L. Li, I. You, S. Yu, and H. Zhang, "Optimizing high-speed mobile networks with smart collaborative theory," *IEEE Wireless Communications*, vol. 29, no. 3, pp. 48–54, 2022.

[6] Y. Ju, Z. Cao, Y. Chen, L. Liu, Q. Pei, S. Mumtaz, M. Dong, and M. Guizani, "Noma-assisted secure offloading for vehicular edge computing networks with asynchronous deep reinforcement learning," *IEEE Transactions on Intelligent Transportation Systems*, 2023.

[7] Y. Liu, Z. Su, and Y. Wang, "Energy-efficient and physical-layer secure computation offloading in blockchain-empowered internet of things," *IEEE Internet of Things Journal*, vol. 10, no. 8, pp. 6598–6610, 2022.

[8] Z. Zhang, K. Zeng, and Y. Yi, "Blockchain-empowered secure aerial edge computing for aiot devices," *IEEE Internet of Things Journal*, 2023.

[9] Z. Wang, Y. Wei, Z. Feng, F. R. Yu, and Z. Han, "Resource management and reflection optimization for intelligent reflecting surface assisted multi-access edge computing using deep reinforcement learning," *IEEE Transactions on Wireless Communications*, vol. 22, no. 2, pp. 1175–1186, 2022.

[10] J. Xu, A. Xu, L. Chen, Y. Chen, X. Liang, and B. Ai, "Deep reinforcement learning for ris-aided secure mobile edge computing in industrial internet of things," *IEEE Transactions on Industrial Informatics*, 2023.

[11] B. Li, W. Wu, W. Zhao, and H. Zhang, "Security enhancement with a hybrid cooperative noma scheme for mec system," *IEEE Transactions on Vehicular Technology*, vol. 70, no. 3, pp. 2635–2648, 2021.

[12] J. Du, C. Jiang, A. Benslimane, S. Guo, and Y. Ren, "Sdn-based resource allocation in edge and cloud computing systems: An evolutionary stackelberg differential game approach," *IEEE/ACM Transactions on Networking*, vol. 30, no. 4, pp. 1613–1628, 2022.

[13] J. Du, C. Jiang, J. Wang, Y. Ren, and M. Debbah, "Machine learning for 6g wireless networks: Carrying forward enhanced bandwidth, massive access, and ultrareliable/low-latency service," *IEEE Vehicular Technology Magazine*, vol. 15, no. 4, pp. 122–134, 2020.

[14] J. Sengupta, S. Ruj, and S. D. Bit, "Fairshare: Blockchain enabled fair, accountable and secure data sharing for industrial iot," *IEEE Transactions on Network and Service Management*, 2023.

[15] C. Zhang, T. Shen, and F. Bai, "Toward secure data sharing for the iot devices with limited resources: A smart contract–based quality-driven incentive mechanism," *IEEE Internet of Things Journal*, 2022.

[16] J. Sengupta, S. Ruj, and S. Dasbit, "A comprehensive survey on attacks, security issues and blockchain solutions for iot and iiot," *Journal of Network and Computer Applications*, 2019.

[17] J. Xu, C. Wang, and X. Jia, "A survey of blockchain consensus protocols," *ACM Computing Surveys*, 2023.

[18] A. Badshah, M. Waqas, F. Muhammad, G. Abbas, Z. H. Abbas, S. A. Chaudhry and S. Chen, "AAKE-BIVT: Anonymous Authenticated Key Exchange Scheme for Blockchain-Enabled Internet of Vehicles in Smart Transportation," in *IEEE Transactions on Intelligent Transportation Systems*, vol. 24, no. 2, pp. 1739-1755, Feb. 2023.

[19] S. Tu, A. Badshah, H. Alasmary and M. Waqas, "EAKE-WC: Efficient and Anonymous Authenticated Key Exchange Scheme for Wearable Computing," in *IEEE Transactions on Mobile Computing*, Jul. 2023.

[20] Z. Jiao, B. Zhang, L. Zhang, M. Liu, W. Gong, and C. Li, "A blockchain-based computing architecture for mobile ad hoc cloud," *IEEE Network*, vol. 34, no. 4, pp. 140–149, 2020.

[21] S. Tu, H. Yu, A. Badshah, M. Waqas, Z. Halim and I. Ahmad, "Secure Internet of Vehicles (IoV) With Decentralized Consensus Blockchain Mechanism," in *IEEE Transactions on Vehicular Technology*, 2023.

[22] R. Guerraoui, F. Huc, and A.-M. Kermarrec, "Highly dynamic distributed computing with byzantine failures," in *Proceedings of the 2013 ACM symposium on Principles of distributed computing*, 2013, pp. 176–183.

[23] E. Panteley and A. Loria, "Synchronization and dynamic consensus of heterogeneous networked systems," *IEEE Transactions on Automatic Control*, pp. 1–1, 2018.

[24] J. Mišić, V. B. Mišić, X. Chang, and H. Qushtom, "Adapting pbft for use with blockchain-enabled iot systems," *IEEE Transactions on Vehicular Technology*, vol. 70, no. 1, pp. 33–48, 2021.

[25] 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 Communications Surveys & Tutorials*, vol. 21, no. 2, pp. 1508–1532, 2019.

[26] J. Feng, F. R. Yu, Q. Pei, J. Du, and L. Zhu, "Joint optimization of radio and computational resources allocation in blockchain-enabled mobile edge computing systems," *IEEE Transactions on Wireless Communications*, vol. 19, no. 6, pp. 4321–4334, 2020.

[27] F. Bai, T. Shen, Z. Yu, K. Zeng, and B. Gong, "Trustworthy blockchain-empowered collaborative edge computing-as-a-service scheduling and data sharing in the iioe," *IEEE Internet of Things Journal*, vol. 9, no. 16, pp. 14 752–14 766, 2022.

[28] R. A. Banez, H. Tembine, L. Li, C. Yang, and H. V. Poor, "Mean-field-type game based computation offloading in multi-access edge computing networks," *IEEE Transactions on Wireless Communications*, vol. PP, no. 99, pp. 1–1, 2020.

[29] A. Taghizadeh, H. Kebriaei, and D. Niyato, "Mean field game for equilibrium analysis of mining computational power in blockchains," *IEEE Internet of Things Journal*, vol. 7, no. 8, pp. 7625–7635, 2020.

[30] Y. Sun, L. Zhang, G. Feng, B. Yang, B. Cao, and M. A. Imran, "Blockchain-enabled wireless internet of things: Performance analysis and optimal communication node deployment," *IEEE Internet of Things Journal*, vol. 6, no. 3, pp. 5791–5802, 2019.

[31] J. Li, T. Liu, D. Niyato, P. Wang, J. Li, and Z. Han, "Contract-theoretic pricing for security deposits in sharded blockchain with internet of things (iot)," *IEEE Internet of Things Journal*, vol. 8, no. 12, pp. 10 052–10 070, 2021.

[32] Q. Xu, S. Zhou, Q. Zheng, M. Luo, and K. Zhang, "Game theoretical secure caching scheme in multihoming edge computing-enabled heterogeneous networks," *IEEE Internet of Things Journal*, vol. PP, no. 99, pp. 1–1, 2018.

[33] Y. Bai, L. Chen, L. Song, and J. Xu, "Risk-aware edge computation offloading using bayesian stackelberg game," *IEEE Transactions on Network and Service Management*, vol. 17, no. 2, pp. 1000–1012, 2020.

[34] X. Wang, Z. Ning, L. Guo, S. Guo, X. Gao, and G. Wang, "Mean-field learning for edge computing in mobile blockchain networks," *IEEE Transactions on Mobile Computing*, 2022.

[35] R. Zheng, H. Wang, M. De Mari, M. Cui, X. Chu, and T. Q. Quek, "Dynamic computation offloading in ultra-dense networks based on mean field games," *IEEE Transactions on Wireless Communications*, pp. 1–1, 2021.

[36] D. Wang, W. Wang, Z. Han, and Z. Zhang, "Delay optimal random access with heterogeneous device capabilities in energy harvesting networks using mean field game," *IEEE Transactions on Wireless Communications*, vol. PP, no. 99, pp. 1–1, 2021.

[37] C. Yang, J. Li, P. Semasinghe, E. Hossain, S. M. Perlaza, and Z. Han, "Distributed interference and energy-aware power control for ultra-dense d2d networks: A mean field game," *IEEE Transactions on Wireless Communications*, vol. 16, no. 2, pp. 1205–1217, 2017.

[38] D. Shi, H. Gao, L. Wang, M. Pan, Z. Han, and H. V. Poor, "Mean field game guided deep reinforcement learning for task placement in cooperative multiaccess edge computing," *IEEE Internet of Things Journal*, vol. 7, no. 10, pp. 9330–9340, 2020.

[39] A. Momose and L. Ren, "Multi-threshold byzantine fault tolerance," in *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security*, 2021, pp. 1686–1699.

[40] W. Li, C. Feng, L. Zhang, H. Xu, B. Cao, and M. A. Imran, "A scalable multi-layer pbft consensus for blockchain," *IEEE Transactions on Parallel and Distributed Systems*, vol. 32, no. 5, pp. 1146–1160, 2020.

[41] I. E. et.al, "Bitcoin-ng: A scalable blockchain protocol," https://arxiv.org/pdf/1510.02037.pdf.

[42] R. K. Raman and L. R. Varshney, "Coding for scalable blockchains via dynamic distributed storage," *IEEE/ACM Transactions on Networking*, vol. 29, no. 6, pp. 2588–2601, 2021.

[43] J. Xu, Y. Zhao, H. Chen, and W. Deng, "Abc-gspbft: Pbft with grouping score mechanism and optimized consensus process for flight operation data-sharing," *Information Sciences*, vol. 624, pp. 110–127, 2023.

[44] G. Jing, Y. Zou, D. Yu, C. Luo, and X. Cheng, "Efficient fault-tolerant consensus for collaborative services in edge computing," *IEEE Transactions on Computers*, 2023.

[45] R. Li, T. Song, B. Mei, H. Li, X. Cheng, and L. Sun, "Blockchain for large-scale internet of things data storage and protection," *IEEE Transactions on Services Computing*, vol. 12, no. 5, pp. 762–771, 2018.

[46] S. Z. et al., "Research progress of blockchain storage scalability," *Journal of Software*, vol. 32, no. 1, p. 20, 2021.

[47] Z. Guo, Z. Gao, Q. Liu, C. Chakraborty, Q. Hua, K. Yu, and S. Wan, "Rns-based adaptive compression scheme for the block data in the blockchain for iiot," *IEEE Transactions on Industrial Informatics*, vol. 18, no. 12, pp. 9239–9249, 2022.

[48] M. Dai, S. Zhang, H. Wang, and S. Jin, "A low storage room requirement framework for distributed ledger in blockchain," *IEEE Access*, pp. 22 970–22 975, 2018.

[49] Z. bin et al., "Research on partial repeat codes based on grouping design," *Journal on Communications*, vol. 36, no. 2, p. 8, 2015.

[50] B. Qu, L. E. Wang, P. Liu, Z. Shi, and X. X. Li, "Gcblock: A grouping and coding based storage scheme for blockchain system," *IEEE Access*, vol. PP, no. 99, pp. 1–1, 2020.

[51] M. Chen and Y. Hao, "Task offloading for mobile edge computing in software defined ultra-dense network," *IEEE Journal on Selected Areas in Communications*, vol. 36, no. 3, pp. 587–597, 2018.

[52] Y. Jiang, Y. Hu, M. Bennis, F.-C. Zheng, and X. You, "A mean field game-based distributed edge caching in fog radio access networks," *IEEE Transactions on Communications*, vol. 68, no. 3, pp. 1567–1580, 2019.

[53] P. Semasinghe and E. Hossain, "Downlink power control in self-organizing dense small cells underlaying macrocells: A mean field game," *IEEE Transactions on Mobile Computing*, vol. 15, no. 2, pp. 350–363, 2015.

[54] G. Xu, H. Bai, J. Xing, T. Luo, N. N. Xiong, X. Cheng, S. Liu, and X. Zheng, "Sg-pbft: A secure and highly efficient distributed blockchain pbft consensus algorithm for intelligent internet of vehicles," *Journal of Parallel and Distributed Computing*, vol. 164, pp. 1–11, 2022.

[55] S. Wei, "Fr code construction for efficient repair of faulty nodes," Master's thesis, Chang'an University, 2021.

[56] M. De Mari, E. C. Strinati, M. Debbah, and T. Q. Quek, "Joint stochastic geometry and mean field game optimization for energy-efficient proactive scheduling in ultra dense networks," *IEEE Transactions on Cognitive Communications and Networking*, vol. 3, no. 4, pp. 766–781, 2017.