Age-Critical and Secure Blockchain Sharding Scheme for Satellite-based Internet of Things

Bingzheng Wang, Jian Jiao, *Member, IEEE,* Shaohua Wu, *Member, IEEE,* Rongxing Lu, *Fellow, IEEE* and Qinyu Zhang, *Senior Member, IEEE*

***Abstract*—It is witnessed that blockchain technology has been widely studied in Internet of Things (IoT) applications due to its decentralized tamper-resistance. Meanwhile, satellite-based IoT (S-IoT) becomes popular and has been regarded as a potential solution of the scalability due to its ubiquitous coverage inherited from satellites. Nevertheless, the large-scale blockchain network enabled S-IoT (BNS-IoT) would be limited by timely performing consensus. In this paper, we propose an age-critical blockchain sharding (ABS) scheme with the metric of information timeliness, i.e., age of information (AoI) to realize timely consensus in BNS-IoT. Specifically, we propose a forking-waiting-retransmission (FR) mechanism for the ABS scheme to deal with forking events, and realize a secure consensus. Then, we derive the closed-form expressions of average AoI (AAoI), throughput and security performance of the FR mechanism in ABS scheme, respectively, and compare with the n-block confirmation and select the longest-chain (n-LC) mechanism. Simulation results show that our ABS scheme can realize the linear expansion of throughput with the increasing number of shards, and our FR mechanism can greatly improve the security by sacrificing minor AAoI compared with the n-LC mechanism. Furthermore, our ABS scheme can outperform the conventional random sharding (RS) scheme in terms of AAoI and throughout.**

***Index Terms*—Blockchain enabled Satellite-based Internet of Things, scalability, sharding, age of information, security**

I. Introduction

The Internet of Things (IoT) has played a significant role in connecting the physical industrial environment and cyber-space with the rapid development of information and communication technology [1]. Especially in the fifth generation (5G) communication technology featured with decentralization, diversity, heterogeneity and complex network, the massive access of IoT devices and explosive growth of data need an efficient and secure network [2]. The integration of blockchain and IoT, which supports a secure network by means of distributed storage and verification, makes it possible to realize these demands. However, the existing blockchain IoT system based on terrestrial network [3]–[5] cannot solve the limitation of geographical environment. Recently, considering the ubiquitous coverage inherited from high throughput satellite [6]–[8], satellite-based IoT (S-IoT) has been regarded as a potential solution of the scalability of blockchain in remote regions.

Blockchain is a distributed ledger technology (DLT) to enable autonomously and securely transactions without control from central institutions [9]. The transactions are mined into blocks by miners following consensus protocols [10], such as Proof of Work (PoW) in Bitcoin, Proof of Stake (PoS) in Peercoin, Practical Byzantine fault tolerance (PBFT) in Hyperledger Fabric, etc. Note that the consensus procedure are completed in DLT nodes, instead of the lightweight IoT devices with limited resources. However, the large-scale blockchain network enabled S-IoT (BNS-IoT) would also be limited to timely commit the consensus due to the peer-to-peer communications in such a ubiquitous blockchain. For example, the transmission latency of the network with n DLT nodes based on PBFT consensus protocol increases at the rate of $O(n^2)$ [11], and it also consumes large storage resources. This weak scalability puts pressures on the expansion and coverage of BNS-IoT [12].

Therefore, constructing a scalable blockchain system for BNS-IoT becomes a critical issue. Moreover, considering that a transaction is committed when a block is accepted and verified in blockchain, the knowledge of elapsed time since the generated transaction is successfully stored in the distributed ledgers, i.e., the age of information (AoI) for a consensus, is a more proper indicator than the consensus latency to quantify the timeliness of transactions in BNS-IoT [13]–[15].

To address the scalability challenges, the sharding scheme is proposed in Elastico protocol by L. Luu [16], which divides all DLT nodes into multiple disjoint subsets according to certain rules, and each subset manages a sub-chain separately, and completes the transmission and verification of transactions in a small area or small committee (i.e., shard). Since each shard processes transactions independently, the throughout, i.e., the transaction per second (TPS), can significantly increase with the number of shards, which realizes the upgrading of scalability to blockchain. Luu's sharding scheme can avoid communication overhead and transaction replications in the whole blockchain, which is considered as the most effective way to improve scalability along with the increasing size of blockchain [17]. Therefore, several improved versions of

Manuscript received xxx, 2022. This work was supported in part by the National Natural Sciences Foundation of China (NSFC) under Grant 62071141, Grant 61871147, Grant 61831008, and Grant 62027802, in part by the Natural Science Foundation of Guangdong Province under Grant 2020A1515010505, in part by the Guangdong Science and Technology Planning Project under Grant 2018B030322004, in part by the Shenzhen Science and Technology Program under Grant ZDSYS20210623091808025 and Grant GXWD20201230155427003-20200822165138001, and in part by the Major Key Project of PCL under Grant PCL2021A03-1. (*Corresponding author: Jian Jiao.*)

B. Wang, J. Jiao, S. Wu, and Q. Zhang are with the Communication Engineering Research Centre, Harbin Institute of Technology (Shenzhen), Shenzhen 518055, China, and also with Peng Cheng Laboratory, Shenzhen 518055, China (e-mail: 20s152056@stu.hit.edu.cn; jiaojian@hit.edu.cn; hitwush@hit.edu.cn; zqy@hit.edu.cn).

R. Lu is with the Faculty of Computer Science, University of New Brunswick, Fredericton, NB E3B 5A3, Canada, (e-mail: rlu1@unb.ca).

sharding schemes were proposed then, such as Byzcoin [18], OmniLedger [19], RapidChain [20], etc, which have improved the throughput to a certain extent. However, the formation of shards and the selection of internal DLT nodes are random, which leads to higher transmission latency since the nodes in the same shard may be far away from each other. Therefore, some sharding mechanisms [21]–[23] are divided DLT nodes according to their positions to eliminate the problem caused by random sharding (RS) scheme. Considering that the satellites are rendezvous points to realize the cross-shard-transactions, we propose an age-critical blockchain sharding (ABS) scheme according to geographic domain for BNS-IoT.

Furthermore, the consensus protocols within above sharding schemes need to elect a leader node in each shard based on Byzantine fault tolerance (BFT), which makes the shards vulnerable by the failure or malicious attacks to the leader node. Therefore, the existing sharding mechanisms face two security priorities [24]. The first one is intra-consensus-safety: How to measure and improve the anti-attack ability of sub-chain inside a shard? For example, the blockchain network based on BFT protocol can only tolerate up to 33% of attackers in each committee. Compared with controlling the whole network, it is easier to dominate the shard with fewer nodes. Therefore, the sharding mechanism improves the throughput at the expense of security. The second one is cross-shard-atomicity: How to support the cross-shard verification and guarantee the atomicity? i.e., the transactions between two parties are either fully executed or not executed at all, and it is not secure to only record cross-shard-transaction message in one party, which means that the communication and transaction information must be synchronously maintained in these two shards.

Note that our ABS scheme is sharding according to geographic domain in BNS-IoT to improve the AAoI and TPS performances than the existing RS and non-sharding schemes. However, the forking events in the sharding scheme may decrease the security, such as *Double Spend* and *Timeout* attacks [17], compare to the conventional non-sharding scheme. Therefore, we propose a novel mechanism in ABS scheme to relief the *Double Spend* and *Timeout* attacks from the forking events, which can significantly decrease the attack success probability (ASP).

The main contributions of the paper are as follows.

- First, we propose an ABS scheme for BNS-IoT to improve the scalability, where the shards are divided according to their geographical domains. In each shard, the intra-shard-transactions are through the base stations (BS), and the cross-shard-transactions utilize its ground stations (GS) through the satellite. Then, to address the above mentioned two security issues in sharding mechanism, we propose a novel Forking-waiting-retransmission (FR) mechanism to solve forking event and improve intra-consensus-safety in the intra-shard-transactions, and a two-phase-confirmation (2PC) mechanism to solve cross-shard-atomicity in the cross-shard-transactions. To the best of our knowledge, our ABS scheme and its FR and 2PC mechanisms is the first sharding scheme for BNS-IoT, which can eliminate *Double Spend* and relief *Timeout* attack by utilizing both the communication and computing capabilities.
- Second, we analyze the probability of forking events in the intra-shard-transactions, and the latency of PoW consensus process. Then, we introduce the FR and the n-block confirmation and select the longest-chain (n-LC) mechanisms in our ABS scheme, and derive the closed-form expressions of average AoI (AAoI) and TPS for both FR and n-LC mechanisms in the ABS scheme, respectively. Furthermore, we model the attack process and derive the expressions of ASP for both FR and n-LC mechanisms to analyze the key parameters in security performance, respectively.
- Third, our 2PC mechanism can record transaction information synchronously in two sub-chains without affecting other shards, which satisfies the atomicity at the minor cost of TPS. Moreover, theoretical analysis prove that our ABS scheme can realize the linear expansion of TPS with the increasing number of shards. Simulation results validate the accuracy of our theoretical derivations, and show that our FR mechanism can reduce AAoI and greatly improve security at the same time. Moreover, our FR mechanism can eliminate the *Double Spend* attack by sacrificing minor timeliness compared with the n-LC mechanism, and our ABS scheme outperforms RS scheme in terms of AAoI and TPS.

The rest of this paper is organized as follows. In Section II, we propose the system model of our ABS scheme. In Section III, we derive the close-form expressions of AAoI and throughout for FR and n-LC mechanisms in our ABS scheme, respectively. In Section IV, we model the malicious attack process and derive the security performance of two mechanisms. Simulation results are presented in Section V, and the conclusion is in Section VI.

II. System Model and ABS Scheme

In this section, we propose an ABS scheme for BNS-IoT, and introduce the detail process of intra-shard-transactions and cross-shard-transactions in ABS scheme. Then, we propose the FR mechanism for the forking events. Note that the massive DLT nodes may join in the BNS-IoT at anytime, we utilize the PoW consensus protocol in our ABS scheme, which is suitable for this kind of public network that needs security and decentralization simultaneously [25].

A. Age-Critical Blockchain Sharding Scheme for BNS-IoT

As illustrated in Fig. 1, the ABS scheme divides the whole blockchain into M sub-chains according to the independent domains, where launch $N_i (i = 1, 2, ..., M)$ DLT nodes separately, and one shard maps one domain, such as islands, mountains, remote districts, etc. The DLT nodes in each shard are responsible for completing the consensus of two transaction types: Intra-shard-transactions and cross-shard-transactions. The satellite is employed as rendezvous point since the ground facilities are difficult to cover, and each shard has a GS to exchange cross-shard-transactions information with other shards through the satellite. Note that

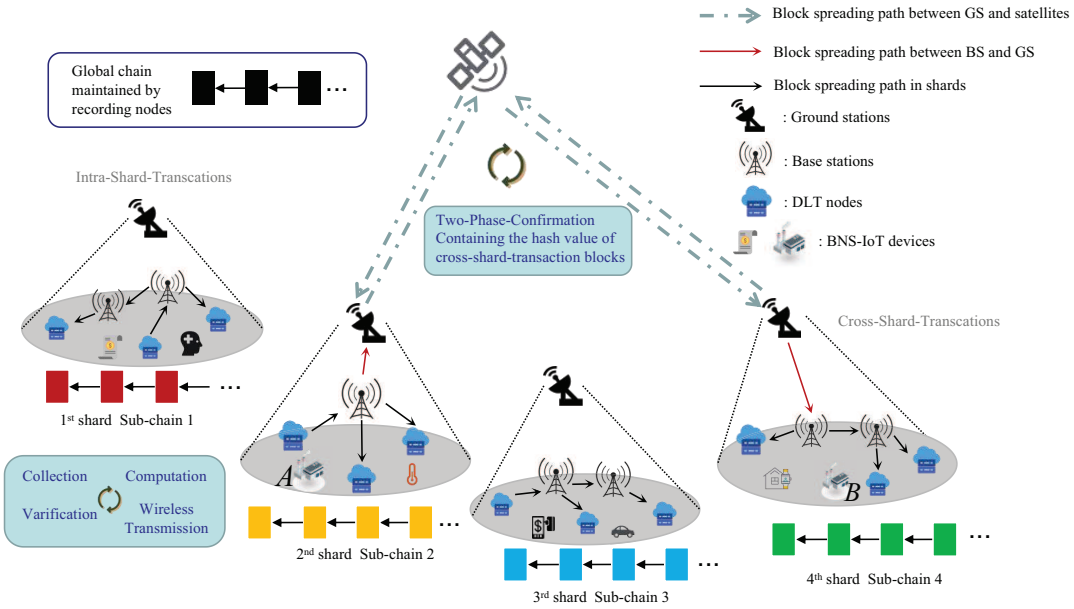

Fig. 1. The ABS scheme system model. Blocks marked with different colors represent different sub-chains, which are belong to disjoint GS service areas. For the 1st shard and 3rd shard, PoW-based intra-shard-transactions are executed. Assuming that there are cross-shard-transactions between sub-chain 2 and sub-chain 4, as the rendezvous point, satellite communication between the two sub-chains is required to start the 2PC mechanism.

in order to record the information of different sub-chains such as calculation difficulty, version synchronization, etc, and ensure the fully operational between the sub-chains, a global chain should be maintained by special recording nodes, which records the status information of each sub-chain at regular intervals. In our model, we do not consider this process since it does not affect the task of DLT nodes, thus we mainly focus on the consensus process of the shards.

In the intra-shard-transactions, the shards maintain a non-intersecting shard transaction record as a single-chain consensus system by ground BS. Moreover, in the intra-shard-transactions, BNS-IoT devices should only deliver the latest generated information to their surrounding DLT nodes to accomplish the consensus task due to their limited storage and computing power. For the intra-shard-transactions in shard-1 and shard-3 as shown in Fig. 1, there are 4 steps in the consensus process based on PoW: 1) Collection, each BNS-IoT device constantly generates transaction data and delivers to the surrounding DLT nodes. When enough transaction information is collected by DLT nodes, it will be packaged into a block; 2) Computation, in PoW consensus, each DLT node in the sub-chain calculates a hash value based on the data block, the nonce, and the hash value of the previous block; 3) Competition, the completed DLT nodes deliver the computed blocks to their surrounding BS, and each BS broadcasts the block to other DLT nodes. The first computed block should be the first arriving to all other DLT nodes, otherwise a forking event occurs. 4) Verification, the uncompleted DLT nodes which receiving the block would stop their computation immediately and verify the correctness of the block. After committed by them, the transactions in the block can take effect.

In the cross-shard-transactions, i.e., if two parties of a transaction belong to two sub-chains, we design a two-phase-confirmation (2PC) mechanism to complete the communication and synchronization between the two sub-chains. For example, the BNS-IoT devices A and B which belong to shard-2 and shard-4 preform a fund transfer as shown in Fig. 1, these two shards would complete two rounds of consensus. First, a round of intra-shard-consensus wound be completed in shard-2. After verification, shard-2 sends an acknowledgment (ACK) packet to the satellite by its GS, which contains the confirmed payment record of device A, and the corresponding valid block hash value. Then, the satellite delivers it to shard-4, where the DLT nodes would record this hash value to local sub-chain to realize synchronization. This cross-shard-transactions wound not be completed until shard-4 completes another round of intra-shard-consensus, which commits that B accomplishes a deposit successfully, and returns an ACK packet to shard-2. The attacker who wants to change the fund record of A or B needs to attack both 2nd shard and 4th shard to erase the payment record. Thus, the 2PC mechanism can ensure that the transaction cannot be tampered privately by any party after two synchronization and confirmations at the cost of throughout, since we divide a transaction into two parts. Note that the satellite only forwards ACK packets in the 2PC mechanism, and the forwarding latency is much shorter than the line-of-sight (LoS) duration between GS and any satellite, we can ignore the mobility of satellite.

B. Analysis and Solution of Forking Event in Blockchain System

The forking event in the blockchain means that for a parent block, two or more child blocks are connected to it, resulting in multiple branches. In the blockchain system, the first DLT node who completes the computation would send its block to all other DLT nodes as soon as possible. If any DLT node

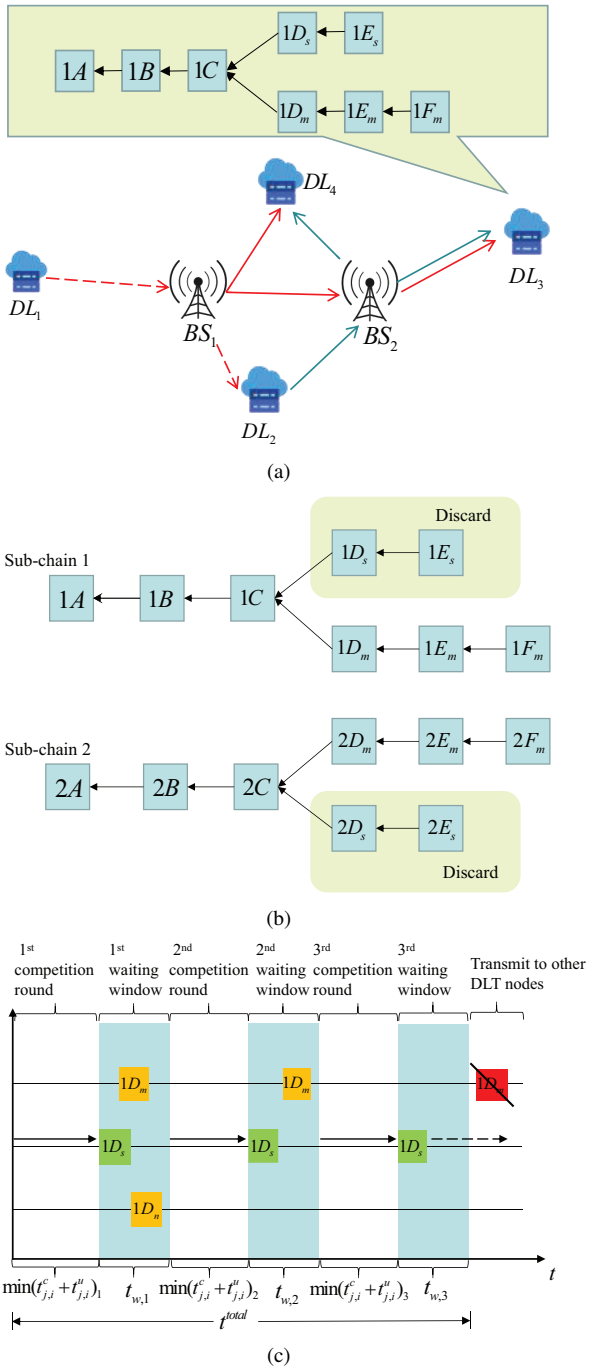

Fig. 2. Forking events and two protection mechanisms. (a) The forking event occurs to the intra-shard-transactions. The node DL_1 with the shortest computation latency fails to be the first to update its computing result to other nodes. (b) The n-LC mechanism, it allows multiple blocks spreading in the network in each consensus round, then it selects the longest branch as the main chain. (c) The FR mechanism, when there is one and only one block $1D_s$ in the waiting window $t_{w,3}$, it would be forwarded to ensure that only one block is spreading in the network.

cannot timely receive this block, it would continue to calculate and publish its own block, which leads to a forking event.

For example, as shown in Fig. 2(a), $1D$ will be connected to $1C$ after the competition. However, after DL_1 first completes the computation of $1D_s$, DL_2 also completes the computation of $1D_m$ since it does not timely receive $1D_s$ from DL_1 via BS_1, and it transmits $1D_m$ to DL_3 and DL_4 by BS_2. The red path represents the spreading of $1D_s$, and the green lines represent the spreading path of $1D_m$ as shown in Fig. 2(a). Therefore, in this consensus round there are two valid blocks spreading in the sub-chain 1, DL_3 would record two valid blocks into its ledger, resulting in that two branches are connecting to the main chain. In blockchain, forking event is only allowed to occur temporarily and then resolved by the forking protection mechanism, and the system only maintains one main chain.

1) n-LC Mechanism: S. Nakamoto proposed the n-LC mechanism to solve the forking events [9], i.e., the system accumulates n_c confirmation blocks after a forking event occurs, then selects the longest branch chain as the main chain, and discards the rest branches. As shown in Fig. 2(b), the forking event occurs in two sub-chains at block D. After generating $n_c = 3$ confirmed blocks, the branch chains $1D_s \leftarrow 1E_s$ and $2D_s \leftarrow 2E_s$ are discarded, while the longer chain $1D_m \leftarrow 1E_m \leftarrow 1F_m$ and $2D_m \leftarrow 2E_m \leftarrow 2F_m$ are retained.

However, in the n-LC mechanism, the attacker can create blocks that containing fraud transaction messages on the branch, and generate more fraud blocks than confirmation blocks to replace the correct information. Furthermore, in the cross-chard-transactions, if the correct transaction input $1D_s$ and output $2D_s$ in two parties are retained or discarded at the same time, the atomicity of the transaction is not affected. However, if the input is $1D_s$ and the output is $2D_m$, and $1D_m$ and $2D_m$ are retained in sub-chain 1 and sub-chain 2, respectively, the fund records is wrong. Therefore, we perform the 2PC mechanism, where the sub-chain 1 and sub-chain 2 need to confirm their main chain via n-LC mechanism successively in cross-chard-transactions to ensure the cross-shard-atomicity. Obviously, we can select a small n_c to reduce the confirmation latency in the n-LC mechanism with tradeoff the security, which is analyzed in Section III and Section IV.

2) FR Mechanism: In this paper, we propose a novel FR mechanism to the forking event in each sub-chain: Assuming that all BS are synchronized in each receiving round with time interval $\min(t^c_{j,i} + t^u_{j,i}) + t_w$ as shown in Fig. 2(c), where t_w is the period of waiting window at the BS, $t^c_{j,i}$ and $t^u_{j,i}$ represent the computing latency and transmission latency from the j-th DLT node to a BS in the sub-chain i. After collecting the original transaction data, the DLT nodes start to computing in $t^c_{j,i}$, then they deliver their blocks to surrounding BS with $t^u_{j,i}$, thus the competitive latency of the DLT node j in the i-th sub-chain is $t^c_{j,i} + t^u_{j,i}$, and the first arrive (FA) block would reach the BS in this competition round with the latency $\min(t^c_{j,i} + t^u_{j,i})$. Then, all BS can receive blocks from DLT nodes in t_w. If only one block is received by a BS in t_w, the BS forwards the block and reject to receive blocks after t_w, thus this competition round is finished. Otherwise if more than one block is received by BS in t_w, a new competition round is performed.

Specifically, as shown in Fig. 2(c), after finishing the first competition round , the first arriving (FA) block $1D_s$ arrives at the BS_2 with the latency of $\min(t^c_{j,i} + t^u_{j,i})_1$, and all the BS begin the waiting window $t_{w,1}$. Then, another two blocks

TABLE I
PARAMETERS DESCRIPTIONS

Notation	Definition
N_i	The number of DLT nodes in the i-th shard
M	The number of shards
λ_0	The complexity of computing hash value
λ_{int}	The average latency of collecting one transaction
l_d	The size of one transaction
N_t	The number of transactions in a block
D	The size of a block
l_{ah}	The size of the ACK packet
P_i^c, P_i^u	The power of computation and transmission of the DLT node in the i-th shard (sub-chain i)
B^d,B^g,B^b,B^s	The bandwidth of DLT nodes, GS, BS and satellite
γ_i^d, γ_i^g, γ^b, γ^s	The SNR of $DL_{j,i}$ nodes, GS_i, BS and satellite
γ_{th}	The threshold of receiving SNR
n_c	The number of confirmation blocks in n-LC
n_w	The maximum number of retransmission in FR
t_w	The waiting window of BS in FR
μ	The scale factor of t_w
L_i	The distance between the satellite and GS_i
f	The system transmitting frequency
g^d, g^s	Channel gain between DLT nodes to BS and GS to satellite links, respectively
α	Proportion of computing power of attacker
c	The proportion of cross-shard-transactions

$1D_m$ and $1D_n$ reach BS_k and BS_1, respectively, and a forking event occurs and all the three blocks cannot be forwarded in the first competition round. In the second competition round, there are still two blocks $1D_s$ and $1D_m$ are received by BS_2 and BS_k in $t_{w,2}$, respectively, and they have to begin the third competition round. Finally, only $1D_s$ is received by BS_2 in $t_{w,3}$, and $1D_m$ is rejected due to it arrives later than $t_{w,3}$, and the BS would forward $1D_s$ and finish this consensus.

Thus, there is no branch in our FR mechanism, which fundamentally avoids the occurrence of forking event. Moreover, we can reduce $t^{total} = \sum_f \min(t_{j,i}^c + t_{j,i}^u)_f + t_{w,f}$ by setting a shorter t_w. Shorter t_w may decrease the AAoI, and makes it difficult for attacker to deliver its fraud blocks in the shard and improve the security performance. The theoretical analysis is provided in Section III and Section IV.

III. ANALYSIS OF AOI AND THROUGHOUT IN ABS SCHEME

In our ABS scheme, each DLT node are divided into three parts to calculate the latency respectively: 1) Cache, storages the newly collected transactions information; 2) Computation, calculates and verifies the hash value; 3) Wireless communication, transmits the block to other nodes, and completes the FA block with Computation jointly. Some important parameter notations in our system are clarified in Table I.

A. Channel Model

1) DLT Nodes to BS and Intra-Shard-Transactions: In the i-th shard, there are N_i DLT nodes $DL_{j,i}$ $(j = 1, 2, ...N_i)$ maintaining a sub-chain i together. The BNS-IoT devices in the same domain would send new transaction information to the DLT nodes at regular intervals. Assume that both the time interval between two transactions generated by BNS-IoT devices, and the time interval when transaction data arrive at DLT nodes are following exponential distribution with expected value $1/\lambda_{g,i}$ and $1/\lambda_{r,i}$, respectively [14]. Then we can derive the average latency for collecting one block in the sub-chain i as follows:

$$E_i^a = E(t_{j,i}^a) = N_t(\frac{1}{\lambda_{g,i}} + \frac{1}{\lambda_{r,i}}) = N_t\lambda_{int,i}, \quad i = 1, 2, ..., M, \tag{1}$$

where N_t is the number of transactions in one block, and $\lambda_{int,i}$ is the average latency of collecting one transaction from BNS-IoT devices. Assume that the average size of each transaction is l_d bits, hence the size of a block is $D = N_t l_d$.

Then, the DLT nodes start the computation of the hash value, which is iterated continuously to satisfy a certain threshold requirement [5]. Without loss of generality, we assume that the computation power of the DLT nodes in the sub-chain i is P_i^c, and the period of a DLT node accomplishing this task can be formulated as an exponential random variable $t_{j,i}^c$ with distribution $f_{T_c}(t) = \lambda_i^c e^{-\lambda_i^c t}$, where $\lambda_i^c = \lambda_0 P_i^c$, and λ_0 is a constant as the computation complexity coefficient in all shards [5]. The average computation latency of the first computed (FC) block E_i^{fc} is given as follows:

$$\begin{aligned} E_i^{fc} &= E\left[\min_{1\le j\le N_i}(\mathrm{t}_{j,i}^c)\right] \\ &= \int_0^\infty \Pr\left[\min_{1\le j\le N_i}(T_{j,i}^c) > t\right] dt \\ &= \int_0^\infty e^{-\lambda_o t \sum_{j=1}^{N_i} P_i^c} dt = \frac{1}{\lambda_0 N_i P_i^c}. \end{aligned} \tag{2}$$

All the computed blocks are sent to a BS and broadcasted to all other DLT nodes. We consider the fading channel in the intra-shard-transactions, i.e., between the DLT nodes and BS, are the Rayleigh fading channel with mean 1, thus the small-scaling channel fading gain is a random variable h^d with distribution $f_{H^d}(h^d) = e^{-h^d}$ [26], [27]. Without loss of generality, we suppose that the channel gains between all $DL_{j,i}$ and the associated BS are the same, denoted as g_i^d, and the additive white Gaussian noise (AWGN) power is σ_d^2 W. Similarly, assume that the transmission power of the DLT nodes in sub-chain i is the same, which is denoted as P_i^u. Let signal-to-noise ratio (SNR) of the DLT nodes is $\gamma_i^d = P_i^u/\sigma_d^2$, thus the SNR at the BS is $\gamma_{re}^d = \frac{P_i^u g_i^d h^d}{\sigma_d^2} = \gamma_i^d g_i^d h^d$ with the probability density function (PDF) as follows:

$$f_{\gamma_{re}^d}(\gamma_{re}^d) = \frac{1}{\gamma_i^d g_i^d} e^{-\frac{1}{\gamma_i^d g_i^d}\gamma_{re}^d}. \tag{3}$$

Denote the transmission latency to send a block with D bits from $DL_{j,i}$ to a BS is $t_{j,i}^u = D/\left[B^d \log(1+\gamma_{re}^d)\right]$. Then, the PDF of $t_{j,i}^u$ in Rayleigh fading channel is:

$$f_{T_u}^d(t_{j,i}^u, \gamma_i^d) = \frac{\ln 2}{\gamma_i^d g_i^d} \exp\left(-\frac{2^{\frac{D}{B^d t_{j,i}^u}} - 1}{\gamma_i^d g_i^d}\right) \frac{D}{B^d (t_{j,i}^u)^2} 2^{\frac{D}{B^d t_{j,i}^u}}, \tag{4}$$

where B^d is the transmission signal bandwidth of the DLT nodes.

Assume that γ_{th} is the outage threshold of SNR, which is corresponding to the timeout threshold of achievable transmission latency $\bar{t} = D/\left[B^d \log(1+\gamma_{th})\right]$. Thus, the average

transmission latency of the block from DLT nodes to the associated BS is given by:

$$E^d_{T_u}(\gamma^d_i) = \frac{D}{E\left[B^d \log_2(1+\gamma^d_{re})\right]}. \tag{5}$$

By retaining the second order of Taylor expansion of Eq. (5), we can derive an approximation as follows,

$$\begin{aligned} E^d_{T_u}(\gamma^d_i) &= \frac{D}{E\left[B^d \log_2(1+\gamma^d_{re})\right]} \\ &\approx \frac{D}{A^d}\left\{\ln\left[1+E(\gamma^d_{re})\right] - \frac{E\left(\gamma^d_{re}\right)^2 - E^2(\gamma^d_{re})}{2[1+E(\gamma^d_{re})]^2}\right\}^{-1} \\ &= \frac{D}{A^d}\left\{\ln\left[1+\theta^d_i \Gamma(2, \frac{\gamma_{th}}{\theta^d_i})\right] - \frac{\Gamma(3, \frac{\gamma_{th}}{\theta^d_i}) - \Gamma^2(2, \frac{\gamma_{th}}{\theta^d_i})}{2\left[\frac{1}{\theta^d_i} + \Gamma(2, \frac{\gamma_{th}}{\theta^d_i})\right]}\right\}^{-1}, \end{aligned} \tag{6}$$

where $\theta^d_i = \gamma^d_i g^d_i$, $A^d = B^d \log_2(e)$ and $\Gamma(a,b)$ is the upper incomplete Gamma function.

Then, in the BNS-IoT with PoW consensus protocol, the verification latency $t^v_{j,i}$ of $DL_{j,i}$ follows an exponential distribution, and its expected value is [15]:

$$E^v_i = E(t^v_{j,i}) = \frac{1}{\lambda_{v,i}} = \frac{N_t \lambda_0}{N_i P^c_i}. \tag{7}$$

Finally, we have the service latency $T^{In}_{se,i}$ for the intra-shard-transactions block in the sub-chain i as follows,

$$T^{In}_{se,i} = E(t^{total}_i) + E^v_i + (N_i - 1)E^d_{T_u}(\gamma^b), \tag{8}$$

where γ^b is the SNR of BS. Let $E(t^{total}_i)$ denote the average processing latency for the forking event in the sub-chain i, and we derive the expressions in the FR and n-LC mechanisms, respectively, in the following Section III.B.

2) GS to Satellite and Cross-Shard-Transactions: In the cross-shard-transactions, assume GS_i in the i-th shard needs to deliver the ACK packet to the satellite and forward to GS_n in the n-th shard, where the BS-to-GS link is assumed as error-free fiber link and the transmission latency can be ignored, and the GS-to-satellite link is the widely-used Lognormal rain attenuation channel [28], [29], and the PDF of channel gain h^s is as follows [30]:

$$f_{H^s}(h^s) \doteq \frac{\varepsilon_p^{m_p}}{\Gamma(m_p)}(h^s)^{m_p-1}\exp(-\varepsilon_p h^s), \tag{9}$$

where $\Gamma(\cdot)$ is the Gamma function, $m_p = \frac{1}{\exp(\sigma_p)-1}$, $\Omega_p = q_p\sqrt{\frac{(m_p+1)}{m_p}}$, $\varepsilon_p = m_p/\Omega_p$, q_p is the constant given by $q_p = e^{\mu_p}$, μ_p and σ_p represent the Lognormal location and scale parameters, respectively.

For the GS_i with transmission power P^g_i and bandwidth B^g, let the SNR of GS_i is $\gamma^g_i = P^g_i/\sigma^2_s$ and assume that the antenna gain G_a of all GS is the same. Let $g^s_i = G_a\zeta_i$, where ζ_i is the free path loss model:

$$\zeta_i = 92.44 + 20\log_{10}(L_i) + 20\log_{10}(f), \tag{10}$$

where L_i and f are the distance between GS_i and satellite, and system operating frequency, respectively. Thus, the SNR of receiving signal is $\gamma^s_{re} = g^s_i h^s \gamma^g_i$ with the PDF of:

$$f_{\gamma^s_{re}}(\gamma^s_{re}) \doteq \frac{1}{g^s_i\gamma^g_i}\frac{\varepsilon_p^{m_p}}{\Gamma(m_p)}(\frac{\gamma^s_{re}}{g^s_i\gamma^g_i})^{m_p-1}\exp(-\frac{\gamma^s_{re}\varepsilon_p}{g^s_i\gamma^g_i}). \tag{11}$$

Similarly with Eq (6), by retaining the second order of Taylor expansion, the approximation of average transmission latency to send the ACK packet with the size of l_{ah} from GS_i to the satellite in the Lognormal rain attenuation channel is:

$$\begin{aligned} E^s_{T_u}(\gamma^g_i) &= \frac{l_{ah}}{E\left[B^g \log_2(1+\gamma^s_{re})\right]} \\ &\approx \frac{l_{ah}}{A^g}\left\{\ln\left[1+\frac{\theta^g_i\Gamma(m_p+1, \frac{\gamma_{th}}{\theta^g_i})}{\Gamma(m_p)}\right]\right. \\ &\quad \left. - \frac{\Gamma(m_p+2, \frac{\gamma_{th}}{\theta^g_i}) - \Gamma^2(m_p+1, \frac{\gamma_{th}}{\theta^g_i})}{2\left[\frac{1}{\theta^g_i} + \Gamma(m_p+1, \frac{\gamma_{th}}{\theta^g_i})\right]^2}\right\}^{-1}, \end{aligned} \tag{12}$$

where $\theta^g_i = \frac{\gamma^g_i g^g_i}{\varepsilon_p}$, and $A^g = B^g \log_2(e)$.

Note that in the cross-shard-transactions, the ACK packet may need retransmission due to the high bit error rate (BER) in the GS-to-satellite link. We derive the probability that a packet contains at least one error bit as the packet error rate (PER) in the following Theorem 1.

Theorem 1: Given the size of ACK packet is l_{ah} bits, and the SNR of transmit signal is γ, the average number of retransmissions $n_{ar}(\gamma)$ and PER $\rho(\gamma)$ in Lognormal rain fading channel are given by,

$$n_{ar}(\gamma) = \left\lceil \frac{1}{\left(1 - \frac{1}{\sqrt{2\pi}} \cdot \frac{\varepsilon_p^{m_p}}{(\varepsilon_p+1)^{m_p} g^s \gamma}\right)^{l_{ah}}} \right\rceil, \tag{13}$$

and

$$\rho(\gamma) = 1 - \left[1 - \frac{1}{\sqrt{2\pi}} \cdot \frac{\varepsilon_p^{m_p}}{(\varepsilon_p+1)^{m_p} g^s \gamma}\right]^{l_{ah}}. \tag{14}$$

Proof: Please see Appendix A. ∎

Assume that the SNR of the satellite is γ^s. In this paper, we mainly consider the retransmission $n_{ar}(\gamma)$ between satellite and GS. The average latency to transmit an ACK packet from GS_i to the satellite and then to GS_n is given by:

$$E(T^{acr}_i) = n_{ar}(\gamma^g_i)E^s_{t_u}(\gamma^g_i) + n_{ar}(\gamma^s)E^s_{t_u}(\gamma^s). \tag{15}$$

Finally, we have the service latency $T^{Cr}_{se,i,n}$ of the cross-shard-transactions between the sub-chain i and sub-chain n is as follows,

$$T^{Cr}_{se,i,n} = T^{In}_{se,i} + E(T^{acr}_i) + T^{In}_{se,n} + E(T^{acr}_n). \tag{16}$$

B. Derivation of AAoI

As illustrated in Fig. 3, the red marked line above each trapezoid represents the evolution process of AoI of sub-chain 1, t^a_1, t^a_2, t^a_3 are the generation time of the block $1A$, $1B$ and $1C$, respectively, and UT_1, UT_2 and UT_3 are the time when the blocks are verified by all DLT nodes, i.e., updated in sub-chain 1. The evolution process of AoI in $0 < t < UT_1$ continues to grow to $\Delta t_{p,1}$ since there is no block is updated

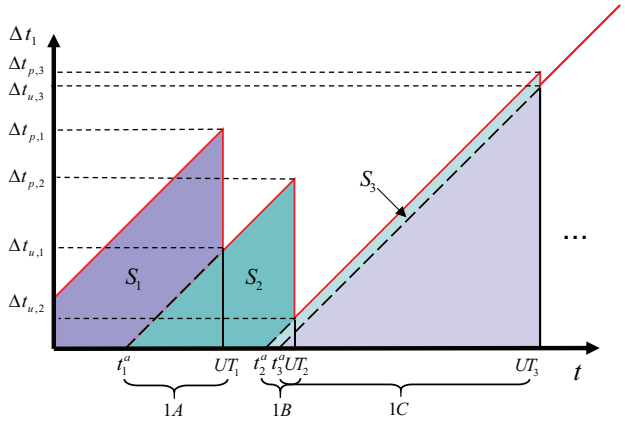

Fig. 3. The red line above each trapezoid represents the evolution process of AoI of the sub-chain 1.

in sub-chain 1. Then, at time UT_1, the intra-shard-transactions block $1A$ completes its consensus process and the sub-chain 1 is updated, thus the AoI reduces to $\Delta t_{u,1} = UT_1 - t_1^a = T_{se,1,1}^{In}$ at UT_1. Similarly, if $1C$ performs cross-shard-transaction, the AoI of sub-chain 1 would increase to $\Delta t_{p,3}$ until the 2PC mechanism between sub-chain 1 and sub-chain n is finished, i.e., $\Delta t_{u,3} = UT_3 - t_3^a = T_{se,1,n,3}^{Cr}$.

Therefore, compare the AoI with the service latency $T_{se,1}^{In}$ or $T_{se,1,n}^{Cr}$ of a block, the instant AoI $\Delta t_{p,i} > UT_i - t_i^a$ before the latest block is updated into the sub-chain, which can better illustrate the update frequency of the sub-chain. This because the generation interval between blocks do not affect the service latency $UT_i - t_i^a$, while the instant AoI $\Delta t_{p,i} = UT_i - t_{i-1}^a$. For example, $UT_2 - t_2^a \ll UT_2 - t_1^a$ as shown in Fig. 3, and the AAoI equals to the trapezoidal area of S_2, which is larger than the service latency $UT_2 - t_2^a$.

The expression of AAoI for sub-chain i is given by [31]:

$$\begin{aligned}\bar{\Delta} t_i &= \lim_{T\to\infty} \frac{1}{T}\int_0^T \Delta t dt = \lim_{n\to\infty} \frac{\sum_{k=1}^{n} S_k}{UT_n} \\ &= \frac{E(S_k)}{E_i^a(t_{a,i})} = \frac{E_i^a(t_{a,i}^2)}{2E_i^a(t_{a,i})} + E(UT_i - t_i^a),\end{aligned} \tag{17}$$

where S_k is the trapezoidal area of each different color in Fig. 3.

1) AAoI of FR Mechanism: The block service latency of the FR mechanism includes the latency of multiple competition rounds caused by forking event. In our FR mechanism, a forking event is solved when there is no other block arrives at the BS within the waiting window after the FA block, and the probability of forking event is solved in this waiting window, i.e., the non-forking probability $P_{nf,i}^{FR}$ is given in the following Theorem 2.

Theorem 2: Given the computation power is P_i^c, the SNR is γ_i^d, and the number of DLT nodes is N_i in i-th shard, the non-forking probability of FR mechanism $P_{nf,i}^{FR}$ is given by Eq. (18) at the bottom of this page, where $T^* = t_{j^*,i}^c + t_{j^*,i}^d + t_w$.

Proof: Please see Appendix B. ■

Therefore, the competition rounds X caused by forking event in the sub-chain i follows a geometric distribution, i.e., $P(X_i = k) = (1 - P_{nf,i}^{FR})^{k-1} P_{nf,i}^{FR}$, and the average number of retransmission is $1/P_{nf,i}^{FR}$. In the FR mechanism, the total latency from $DL_{j,i}$ to BS is $t^{total} = \sum_f \min(t_{j,i}^c + t_{j,i}^u)_f + t_{w,f}$, where $t_{w,f}$ is a constant to affect the security and AAoI, and we will discuss in Section V. Now, we derive the average latency of FA block E_i^{fa} by the following Theorem 3.

Theorem 3: Given the size of block is D bits, the SNR of transmit signal is γ_i^d, and the computing power is P_i^c, the average competition latency of FA block E_i^{fa} in the i-th shard can be expressed as follows:

$$E_i^{fa} = E\left[\min_{1\le j\le N_i}(t_{j,i}^c + t_{j,i}^u)\right] = \int_{E_i^{fc}}^{E_i^a + \bar{t}} F_i^{fa}(t)dt, \tag{19}$$

where

$$F_i^{fa}(t) = \left[1 - \exp\left(\frac{1 - 2^{\frac{D}{B^d t}}}{\gamma_i^d g_i^d}\right) + e^{-\lambda_i^c t}\int_0^t e^{\lambda_i^c x} f_{T_u}^d(x)dx\right]^N. \tag{20}$$

Proof: Please see Appendix C. ■

Since t_w should decrease with both $t_{j,i}^c$ and $t_{j,i}^u$ due to the lower $t_{j,i}^c$ and $t_{j,i}^u$ would lead to higher forking probability, we define the duration of waiting window is $t_w = \mu E_i^{fa} = \mu E\left[\min_{1\le j\le N_i}(t_{j,i}^c + t_{j,i}^u)\right]$, where μ is a scalar coefficient. Thus, the average processing latency $E(t_i^{total})$ for a forking event in the FR mechanism is given by:

$$E_{FR}(t_i^{total}) = \frac{1}{P_{nf,i}^{FR}}\left(E_i^{fa} + t_w\right) = \frac{1+\mu}{P_{nf,i}^{FR}} E_i^{fa}. \tag{21}$$

By substituting Eq. (21) into Eq. (8), we can obtain the service latency of the intra-shard-transactions block of i-th shard in our FR mechanism:

$$T_{FRse,i}^{In} = \frac{1+\mu}{P_{nf,i}^{FR}} E_i^{fa} + (N_i - 1)E_{T_u,d}(\gamma^b) + \frac{N_t \lambda_0}{N_i P_i^c}. \tag{22}$$

In the cross-shard-transactions, each block would complete two rounds of consensus in two sub-chains according to our 2PC mechanism. Meanwhile, each ACK packet would be transmitted twice from GS_i to the satellite and then from satellite to GS_n. Therefore, the service latency of cross-shard-transactions block is given by:

$$T_{FRse,i,n}^{Cr} = E(T_i^{acr}) + T_{FRse,i}^{In} + E(T_n^{acr}) + T_{FRse,n}^{In}. \tag{23}$$

Assume that the proportion of cross-shard-transactions in all transactions is c, combining Eq. (17), Eq. (22), and Eq.

$$P_{nf,i}^{FR} = \int_0^{\bar{t}}\int_0^{E_i^a}\left\{1 - \exp\left(\frac{1 - 2^{\frac{D}{B^d T^*}}}{\gamma_i^d g_i^d}\right) + \exp(-\lambda_i^c T^*)\int_0^{T^*} e^{\lambda_i^c t_{j,i}^d} f_{T_u}^d(t_{j,i}^u)dt_{j,i}^u\right\}^{N_i - 1} f_{T_c}(t_{j^*,i}^c)dt_{j^*,i}^c f_{T_u}^d(t_{j^*,i}^u)dt_{j^*,i}^u. \tag{18}$$

(23), the AAoI of our FR scheme in ABS scheme are given by:

$$\bar{\Delta}t_{FR} = \frac{1}{M}\sum_{i=1}^{M}\left[(1-c)T_{FRse,i}^{In} + \frac{N_t\lambda_{int,i}}{2}\right] + \frac{2c}{M(M-1)}\sum_{i=1}^{M}\sum_{n=i+1}^{M} T_{FRse,i,n}^{Cr}, \tag{24}$$

where $\frac{2c}{M(M-1)}\sum_{i=1}^{M}\sum_{n=i+1}^{M} T_{FRse,i,n}^{Cr}$ is the average service latency of cross-shard-transactions between all pair of shards.

2) AAoI of n-LC Mechanism: In the n-LC mechanism, we simplify the judgment condition of non-forking event in a sub-chain by ignoring the transmission latency between the BS to DLT nodes: Before the FC block reaches the BS, no other block arrives. Therefore, with the help of [32], the non-forking probability in n-LC mechanism is given by Eq. (25) at the bottom of this page.

Once a forking occurs, the DLT nodes would require to wait for extra n_c confirmation blocks. The average processing latency for a forking event of the i-th shard in n-LC mechanism is given by:

$$E_{nLC}(t_i^{total}) = (1-P_{nf,i}^{nLC})(n_c+1)\mathcal{K}_i, \tag{26}$$

where $\mathcal{K}_i$ is defined as the service latency without any forking event, and we have:

$$\mathcal{K}_i = E_i^{fc} + E_{T_u}^{d}(\gamma_i^d) + E_i^v + (N_i-1)E_{T_u}^{d}(\gamma^b). \tag{27}$$

The service latency for intra-shard-transactions in the i-th shard is given by:

$$\begin{aligned} T_{nLCse,i}^{In} &= P_{nf,i}^{nLC}\mathcal{K}_i + (1-P_{nf,i}^{nLC})(n_c+1)\mathcal{K}_i \\ &= (n_c+1-n_cP_{nf,i}^{nLC})\mathcal{K}_i, \end{aligned} \tag{28}$$

and the service latency for cross-shard-transactions is given by:

$$T_{nLCse,i,n}^{Cr} = E(T_i^{acr}) + T_{nLCse,i}^{In} + E(T_n^{acr}) + T_{nLCse,n}^{In}. \tag{29}$$

Similarly, the AAoI of n-LC mechanism is given by combining the Eq. (17), Eq. (28), and Eq. (29) as follows,

$$\bar{\Delta}t_{nLC} = \frac{1}{M}\sum_{i=1}^{M}\left[(1-c)T_{nLCse,i}^{In} + \frac{N_t\lambda_{int,i}}{2}\right] + \frac{2c}{M(M-1)}\sum_{i=1}^{M}\sum_{n=i+1}^{M} T_{nLCse,i,n}^{Cr}. \tag{30}$$

Note that both Eq. (24) and Eq. (30) are focusing on the update frequency of ledger without considering the influence of successful attack under our n-LC and FR mechanisms.

C. Derivation of TPS

1) TPS of FR Mechanism: TPS is an indicator of throughput in the blockchain. For the TPS of a sub-chain, it can be expressed as: $\mathrm{T} = N_t/T_{se}$. Thus, the TPS of FR mechanism in our ABS scheme is given by:

$$\mathrm{T}_{FR} = \sum_{i=1}^{M}\left[\frac{(1-c)N_t}{T_{FRse,i}^{In}}\right] + \frac{2}{M-1}\sum_{i=1}^{M}\sum_{n=i+1}^{M}\frac{cN_t}{T_{FRse,i,n}^{Cr}}. \tag{31}$$

2) TPS of n-LC Mechanism: Similarly, the TPS in n-LC mechanism is given by:

$$\mathrm{T}_{nLC} = \sum_{i=1}^{M}\left[\frac{(1-c)N_t}{T_{nLCse,i}^{In}}\right] + \frac{2}{M-1}\sum_{i=1}^{M}\sum_{n=i+1}^{M}\frac{cN_t}{T_{nLCse,i,n}^{Cr}}. \tag{32}$$

IV. Analysis of Security in ABS Scheme

In this section, we model the attack process of the malicious node and derive the security expressions of n-LC and FR mechanisms in the ABS scheme, respectively. Although the attack types in the two mechanisms are different, the successful attack of the malicious node would both lead to transaction failure. Therefore, we take the transaction failure probability, i.e., ASP, to measure secure performance of two mechanisms. The sharding scheme divides a whole blockchain into M sub-chains, it weakens the anti-attack ability of each sub-chain. Assume that the computing power of the whole blockchain is P^{ct}, and the computing power in each sub-chain is P^{ct}/M.

A. Security of n-LC Mechanism

In the n-LC mechanism based on PoW protocol, the security performance is the ASP for an attacker with computing power $\frac{\alpha\cdot\mathrm{P}^{ct}}{M}$ in a sub-chain. Note that if $\alpha \geq 0.5$, ASP = 1, and if $\alpha = 0$, ASP = 0 [33]. If there is an attacker with higher α to generate fraudulent information in branch, it may cover the block of the honest nodes to realize the attack, such as *Double Spend*. Moreover, to decrease the ASP in n-LC mechanism, we set that any branch needs at least 2 blocks longer than other branch as shown in Fig. 4, where the conventional n-LC mechanism only need one block advanced [33].

Therefore, to launch an attack against a normal transaction in this n-LC mechanism, the attacker must generate a forking event deliberately: 1) The attacker generates a regular transaction information, for example, pay for a commodity A with fund Ω, and broadcasts it to other DLT nodes, then this transaction is recorded in the block $1C_h$ after consensus. 2) The attacker privately fabricates a fraud block $1C_m$, which is conflicting to $1C_h$, for example, pay for another commodity B with the same fund Ω. At this time, a forking event occurs and the honest nodes start to accumulate n_c confirmation blocks,

$$P_{nf,i}^{nLC} = \left(1-e^{-\lambda_i^c E_a^i}\right)\int_0^{\bar{t}}\left[\int_0^{T_{j^*,i}^u} e^{-\lambda_i^c(T_{j^*,i}^u-t)} f_{T_u}^d(t)dt + \exp\left(\frac{1-2^{\frac{D}{B^d\bar{t}}}}{\gamma_i^d g_i^d}\right) - \exp\left(\frac{1-2^{\frac{D}{B^d T_{j^*,i}^u}}}{\gamma_i^d g_i^d}\right)\right]^{N_i-1} f_{T_u}^d(T_{j^*,i}^u)dT_{j^*,i}^u. \tag{25}$$

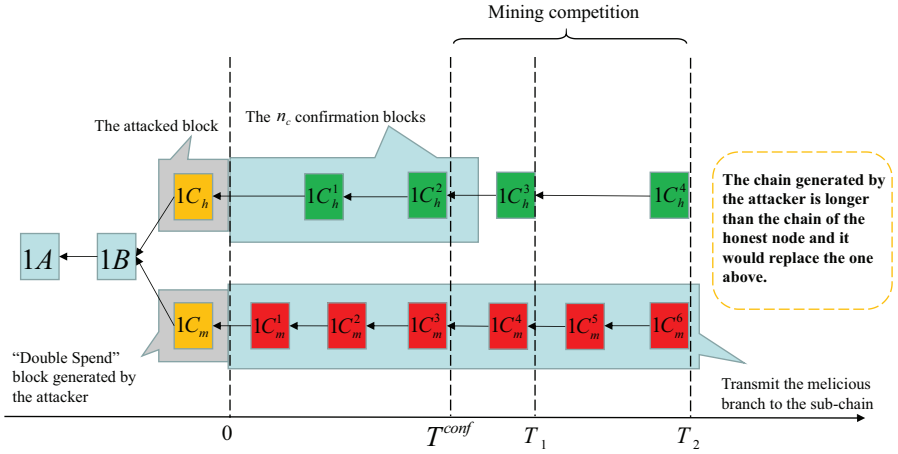

Fig. 4. The process of *Double Spend* attack. The branch generated by the attacker (red marked blocks) wants to replace the branch generated by honest nodes (green marked blocks). But this malicious branch would not publish until it is 2 blocks longer than the one above at time T_2. This attack is successful since the sub-chain would choose the longer branch.

while the attacker starts mining secretly. 3) When there are enough confirmation blocks, i.e., $1C_h^1$ and $1C_h^2$ in the honest branch if $n_c = 2$ at time T^{conf}, the sub-chain would select $1C_h^1 \leftarrow 1C_h^2$ as the main chain, and the record of purchasing A would take effect. Meanwhile, the attacker would not release the branch $1C_m^1 \leftarrow 1C_m^2 \leftarrow 1C_m^3$. 4) Then, to realize *Double Spend*, a mining competition begin at the time T^{conf}. If the attacker generates a longer branch than the honest branch, it would release its malicious branch, and the sub-chain would discard the honest branch, thus the record of purchasing B would take effect, i.e., the attacker realizes two payments with the same fund Ω.

Assume that an attacker in a sub-chain has $\frac{\alpha \cdot \mathrm{P}^{ct}}{M}$ computing power, and the honest nodes have $\frac{\beta \cdot \mathrm{P}^{ct}}{M} = (1-\alpha) \cdot \frac{\mathrm{P}^{ct}}{M}$ computing power. At a certain time, the branch length of the honest node and attacker are l_h and l_a, respectively. Let $n = l_h - l_a$ denote the number of blocks that the honest chain minus that of the attacker. Obviously, n would increase by 1 with probability β, and n would decrease by 1 with probability α.

Denote $p_{a,n}$ as the ASP that the malicious branch has n blocks less than the honest branch, and $p_{a,n} = 1$ when $n \leq -2$. For any $n \geq -1$, if the attacker gets the next block, the malicious branch is $n-1$ blocks shorter than the honest branch, and the ASP becomes $p_{a,n-1}$, else if the honest nodes find the next block, the ASP becomes $p_{a,n+1}$. Therefore, we have $p_{a,n}$ as follows:

$$p_{a,n} = \begin{cases} \left(\frac{\alpha}{\beta}\right)^{n+2}, & n \geq -1, \\ 1, & n \leq -2. \end{cases} \tag{33}$$

Assume that during the period from the beginning of the forking event to the sub-chain gets n_c confirmation blocks, the attacker has generated Y blocks, which follows a negative binomial distribution with the probability:

$$P_{an}(Y = m) = \binom{m + n_c - 1}{m} \alpha^m \beta^{n_c}. \tag{34}$$

Suppose that the malicious branch is $n_c - m$ blocks less than the honest branch at a certain moment. Combine Eq. (33) and Eq. (34), we can obtain the ASP as follows:

$$\begin{aligned} P_{as,nLC}(n_c, \alpha) &= \sum_{m=0}^{\infty} P_{an}(Y = m) p_{a,n_c - m} \\ &= \sum_{m=0}^{\infty} \binom{m + n_c - 1}{m} \alpha^m \beta^{n_c} \left(\frac{\alpha}{\beta}\right)^{n_c - m + 2} \\ &= 1 - \sum_{m=0}^{n_c} \binom{m + n_c - 1}{m} (\alpha^{m+1} \beta^{n_c - 1} - \alpha^{n_c + 2} \beta^{m-2}). \end{aligned} \tag{35}$$

B. Security of FR Mechanism

In our FR mechanism, the BS in each shard either does not forward blocks, or only forwards one block in a competition round. Therefore, in terms of security issues around fraud prevention, our FR mechanism eliminates *Double Spend* attacks since there is no branch. However, the *Timeout* attack caused by the retransmissions, i.e., multiple competition rounds, still exists in our FR mechanism, and we set that if the competition rounds X caused by the forking events exceeds a preselected threshold n_w, the block is discarded.

Thus, we define the ASP in the FR mechanism is the success probability that the number of competition rounds has reached the threshold n_w. When an attack satisfies two conditions, it leads to a new competition round: 1) The attacker already held a fraud block secretly before the FC block; 2) The attacker sends its block to the BS within t_w after the FA block, causing the sub-chain to restart a new competition round. We denote the realization probabilities of condition 1 and condition 2 as $p_{a,FR1}$ and $p_{a,FR2}$, respectively, and we have

$$\begin{aligned} p_{a,FR1} &= \int_0^{\infty} \int_0^{y} \lambda_0 \alpha e^{-\lambda_0 \alpha x} P[\min(t_c) = y] dx dy \\ &= \int_0^{\infty} \int_0^{y} \lambda_0 \alpha e^{-\lambda_0 \alpha x} \lambda_0 N_i \beta e^{-\lambda_0 N_i \beta y} dx dy \\ &= \alpha, \end{aligned} \tag{36}$$

TABLE II
SIMULATION PARAMETERS

Notation	Setting
λ_0	1×10^{-3}
λ_{int}	100
l_d, l_{ah}	0.5 kb, 5 kb
D	1 Mb
μ_p, σ_p	-2.6 dB, 1.6 dB
B^d, B^b, B^g, B^s	1, 100, 200, 500 (MHz)
γ^b, γ^s, γ_{th}	110 dB, 140 dB, 0 dB
n_w	6
L	600 km
f	28 GHz

and

$$p_{a,FR2} = \int_0^{t_w} f_{T_u}^d(t, \gamma^a)dt = \exp(\frac{1 - 2^{\frac{D}{B^d t_w}}}{\gamma^a g^d}), \quad (37)$$

where γ^a is the SNR of the attacker.

To analyze the ASP with n_w in FR mechanism, we assume that the number of retransmissions, i.e., the number of competition rounds caused by attacker is m, and the corresponding probability is $p_{FR}(M = m)$, hence the number of competition rounds caused by honest nodes is $n_w - m$. Therefore, the ASP can be obtained as follows:

$$\begin{aligned} P_{as,FR}(t_w, \alpha) &= \sum_{m=0}^{n_w} p_{FR}(M = m) \\ &= \sum_{m=0}^{n_w} \left(\begin{array}{c} n_w \\ m \end{array} \right) (p_{a,FR})^m (1 - P_{nf,i}^{FR})^{n_w - m}, \end{aligned} \quad (38)$$

where $p_{a,FR}$ is the attack simultaneously satisfies the above two conditions, and we have

$$p_{a,FR} = p_{a,FR1} \cdot p_{a,FR2} = \alpha \exp(\frac{1 - 2^{\frac{D}{B^d t_w}}}{\gamma^a g^d}). \quad (39)$$

V. SIMULATION AND ANALYSIS

In this section, the derived average number of retransmission n_{ar} in Eq. (13), the non-forking probability formula of FR and n-LC mechanisms given in Eq. (18) and Eq. (25), AAoI expressions given in Eq. (24) and Eq. (30), and ASP expression given in Eq. (35) and Eq. (38) are verified by Monte Carlo simulations, respectively. Moreover, we analyze the impact of t_w on the FR mechanism and compare the performance of FR and n-LC schemes under the same parameters [5], [6]. Finally, we analyze the relationship between sub-chain communication conditions and security, and compare the performance of our ABS scheme and RS scheme [19]. We assume that the parameter settings of each shard are the same for simplicity, and the main parameters settings are given in Table II [30], [34].

A. PER and Forking Probability for FR and n-LC Mechanism

In Fig. 5, the simulation results shows the correctness of our PER analytic expression Eq. (13) in Theorem 1 for the cross-shard-transactions. Obviously, with the increasing of packet size, n_{ar} and PER $\rho(\gamma)$ are increasing under the same transmission SNR. When $l_{ah} = 5$ kb, the retransmissions

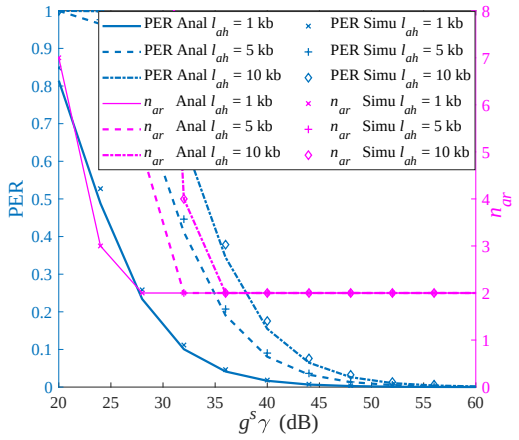

Fig. 5. Simulation of PER and retransmissions numbers n_{ar} with different packet size l_{ah} in cross-shard-transactions.

numbers n_{ar} is reduced at $g^s\gamma^g > 32$ dB. Thus, we set $g^s\gamma^g \approx 35$ dB in the following simulations, which corresponds to $n_{ar} = 2$.

Then, the number of DLT nodes with different computing power P^c versus the forking probability is shown in Fig. 6, where the simulation results of forking probability agree well with the theoretical expressions Eq. (18) and Eq. (25). Note that higher P^c leads to faster generating blocks, which shorten the average competition latency, and leads to the increasing of forking probability as shown in Fig. 6(a) and Fig. 6(b), thus we set $P^c = 300$ W in the following simulations. Moreover, μ is a scalar coefficient of t_w, and larger t_w leads to higher forking probability due to the higher chance to receive more blocks as shown in Fig 6, and n_c in n-LC mechanism does not affect the forking probability according to Eq. (25). Note that t_w can improve the AAoI and security in the FR mechanism, and we set $\mu \in \{0.4, 0.6, 0.8\}$ in the following simulations.

B. AAoI and Throughput Performance of FR and n-LC Mechanism

Fig. 7 compares the AAoI performance of FR and n-LC mechanisms in our ABS and the non-sharding schemes [31], respectively, where the simulation results agree well with the theoretical expressions Eq. (24) and Eq. (30). We can observe that AAoI is increasing with N in both FR and n-LC mechanisms, since more DLT nodes lead to larger forking probability, and the higher latency from BS to all nodes according to Eq. (8). Note that $n_c \in \{4, 6, 8\}$ in n-LC mechanism have the similar consensus latency t^{total} with that of $\mu \in \{0.4, 0.6, 0.8\}$ in the FR mechanism. In the comparison of the three groups, the performance of AAoI in FR mechanism is about 3% higher than that of n-LC mechanism when $N < 60$, while 12% higher than that of n-LC mechanism when $N = 100$. Furthermore, the AAoI in our ABS decreases 46.4% and 45.2% compared with non-sharding scheme with μ = 0.4 and n_c = 4 when N = 60, respectively. This is because all DLT nodes need to deliver the block to each other instead of less peer-to-peer (P2P) transmission in the shards, and long distance and multi-hop

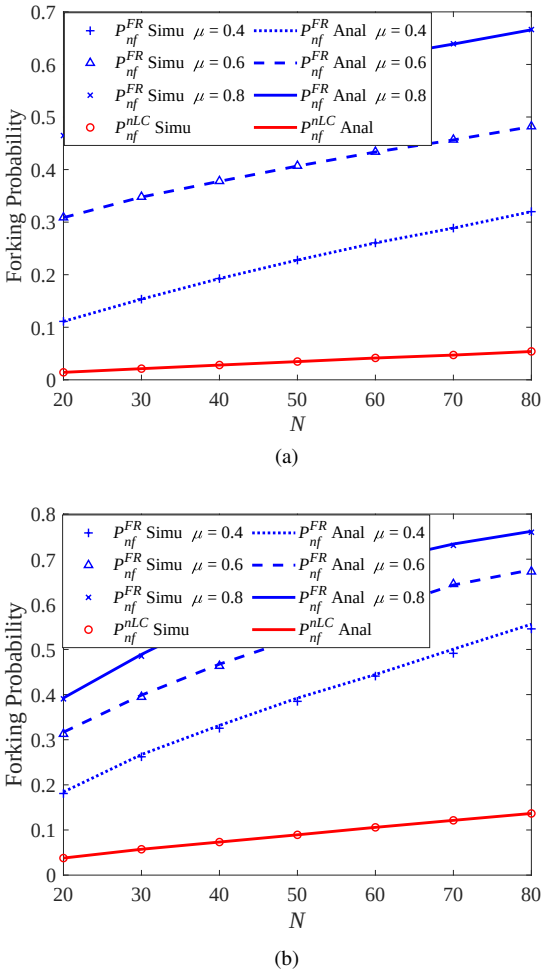

Fig. 6. The forking probability versus number of DLT nodes in FR and n-LC mechanisms, where μ is a scalar coefficient of t_w and larger μ leads to higher forking probability in FR mechanism, but not effect on the n-LC mechanism. (a) The forking probability versus number of DLT nodes when $P^c = 300$ W. (b) The forking probability versus number of DLT nodes when $P^c = 500$ W.

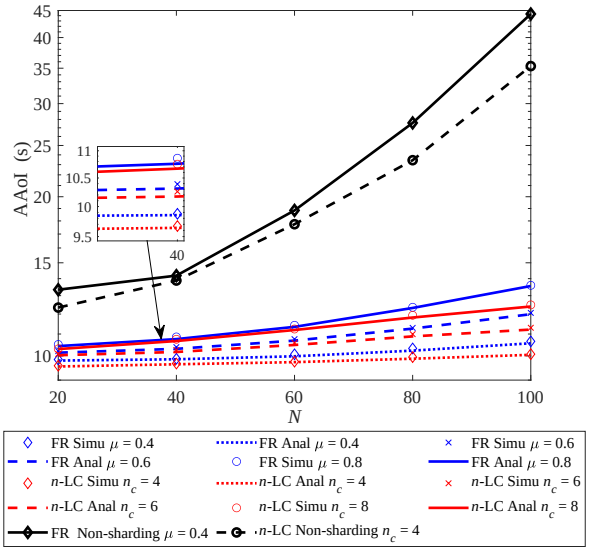

Fig. 7. Comparison of AAoI performance between FR and n-LC mechanisms when $c = 0.6$ and $\gamma^d = 40$ dB.

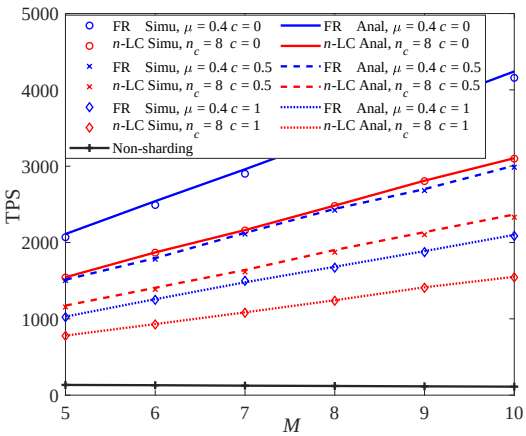

Fig. 8. Comparison of TPS performance versus number of shards M between FR and n-LC mechanisms when $N = 60$.

transmission in the terrestrial-satellite network leads to higher transmission latency. Meanwhile, the greater t_w leads to higher AAoI. Thus, we can set a lower t_w to obtain better AAoI and TPS performance.

The TPS comparison of FR and n-LC mechanisms in the ABS scheme is shown in Fig. 8, our TPS theoretical expressions of FR and n-LC mechanisms are matched with simulation results. Note that our ABS scheme has significant improved the TPS than the non-shading scheme, and with the increasing the number of shards M, our ABS scheme can further improve the TPS. Moreover, the 2PC mechanism in cross-shard-transactions divides one transaction into two parts, thus, the larger proportion c of cross-shard-transactions leads to higher loss of TPS. If $c = 1$, i.e., all transactions are divided into two parts, the 2PC mechanism would loss 50% TPS than $c = 0$.

C. Security Performance of FR and n-LC Mechanism

The security performance is the most significant advantage of our FR mechanism compare with n-LC mechanism, and the simulation results agree well with our analytical expressions of ASP as shown in Fig. 9. We assume that $B^a = B^d$ and $\gamma^a = \gamma^d$, the simulation results show that when $\alpha > 0.15$, the ASP of FR mechanism is much lower than that of n-LC mechanism, since the computing power is the only factor limiting the ASP in n-LC scheme according to Eq. (35), while the ASP is determined by both computing and communication power in FR mechanism according to Eq. (38). For example, when $\mu = 0.6$, $\alpha = 0.35$ and $n_c = 6$, the security is improved 210 times compared with n-LC mechanism. This is because with the increasing of α of attacker in n-LC mechanism, the ASP of *Double Spend* attack is rapidly increasing. When $\alpha < 0.5$, larger n_c can decrease the ASP with the cost of AAoI, and if $\alpha > 0.5$, the attacker would always be able to complete the attack as $ASP = 1$ regardless of n_c. However, in

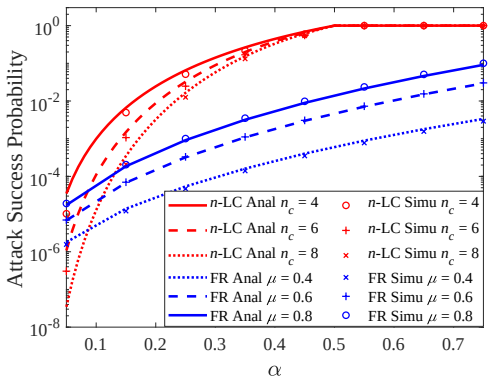

Fig. 9. Comparison of ASP in n-LC and FR mechanisms for different α when $n_w = 6$.

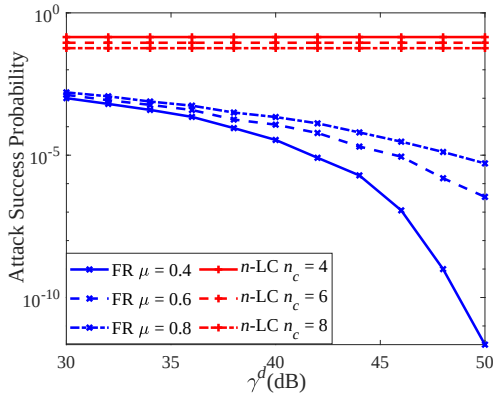

Fig. 10. Comparison of ASP in n-LC and FR mechanisms for different γ^d when $n_w = 6$ and $\alpha = 0.3$.

our FR mechanism, smaller t_w can significantly decrease the ASP even with higher α as shown in Fig. 9, since it is more difficult for the attacker to insert its block in a short t_w. That is, we improve the security by requiring more communication resource from the attacker, such as higher bandwidth B^a and γ^a to deliver its malicious block to BS in a shorter t_w. Therefore, the transmission latency $t^u_{j,i}$ becomes the main bottleneck for the attacker in the FR mechanism, while P^c is the bottleneck for the attacker in the n-LC mechanism.

Then, we compare the ASP with different γ^d in the honest DLT nodes in n-LC and FR mechanisms as shown in Fig. 10, where the SNR of attacker $\gamma^a = 40$ dB and $N = 60$. In the n-LC mechanism, the attacker is not affected by γ^d of the honest DLT nodes and the ASP is unchanged. On the other hand, greater γ^d leads to lower t_w at the BS when μ is a constant in our FR mechanism, and shorter $t^u_{j,i}$, which decreases the ASP of the attacker.

D. Comparison with Random Sharding Scheme

Finally, we compare the AAoI, TPS, and security performance in RS scheme [19] and our ABS scheme with n-LC and FR mechanisms as shown in Fig. 11. First, our FR mechanism in ABS scheme has 2.98% higher AAoI than that of n-LC mechanism in ABS scheme, but our ABS scheme outperforms

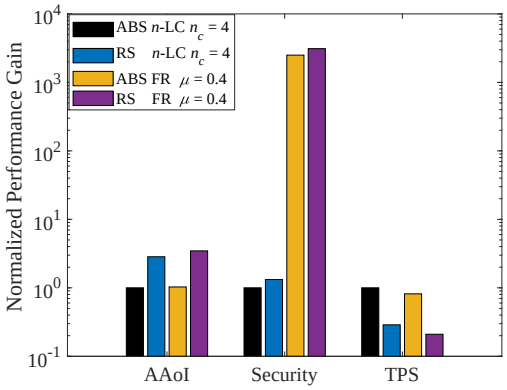

Fig. 11. The performance comparisons of n-LC and FR mechanisms in ABS and RS schemes when $M = 6$ and $N = 60$.

RS scheme in AAoI by reducing almost 64.5% and 70.2% in the n-LC and FR mechanisms, respectively, since that we alleviate the multi-hop and long distance transmission latency between the DLT nodes in ABS scheme. Second, the RS scheme can decrease about 32% ASP than our ABS scheme with both n-LC and FR mechanisms at the cost of higher AAoI and lower TPS, but our FR mechanism improves about 2500 times of APS with $\mu = 0.4$ than that of n-LC mechanism with $n_c = 4$ in both RS and ABS schemes. Last, our ABS scheme realizes 307% and 345% higher TPS in the n-LC and FR mechanisms than that of RS scheme, respectively. Comprehensively, in a sub-chain which needs both security and information timeliness, employing the FR mechanism into ABS scheme is a better compromission. Furthermore, we can utilize the deep Q-learning framework to achieve a further security optimization in our ABS scheme in the future [35], [36].

VI. Conclusion

In this paper, we focused on the AAoI and security issues in the blockchain and designed an ABS scheme for BNS-IoT, where a novel FR mechanism was proposed for the intra-consensus-safety, and a 2PC mechanism was proposed to ensure the cross-shard-atomicity. Specifically, we derived the theoretical expressions of AAoI, TPS and ASP of our FR and n-LC mechanisms in ABS scheme, and validated the accuracy by the Monte Carlo simulations, respectively. The most remarkable innovation was that our FR mechanism can achieve at least 10^2 times improvement of security by utilizing both the communication and computing capabilities of DLT nodes, and realized the lower ASP in a more efficient communication environment. Furthermore, our ABS scheme realized linear increase in TPS under larger N compared with the non-sharding scheme, and achieved the AAoI reduction of nearly 65% and improved TPS by 307% compared with the RS scheme. The theoretical derivations could provide meaningful insights and guidelines to optimize our ABS scheme via the deep Q-learning framework in future work.

APPENDIX A
PROOF OF THEOREM 1

The BER of BPSK in AWGN channel is as follows,

$$P_e = Q(\sqrt{\frac{2E_b}{N_0}}) = \frac{1}{2}erfc(\sqrt{\frac{E_b}{N_0}}). \tag{40}$$

After the signal spreads through the Lognormal rain attenuation channel with channel gain h^s, $g^s = G_a\zeta$, and the BER can be expressed as:

$$P_e(\gamma) = \frac{1}{2}erfc(\sqrt{\frac{h^s g^s E_b}{N_0}}) = \frac{1}{2}erfc(\sqrt{h_s g^s \gamma}), \tag{41}$$

where

$$\text{erfc}(x) = \frac{2}{\sqrt{\pi}}\int_x^\infty e^{-y^2}dy, \tag{42}$$

and γ is the SNR of transmit signal. Let $\gamma_{re} = g^s h^s \gamma$, which is a random variable that follows the PDF $f_{\gamma_{re}}(\gamma_{re})$:

$$f_{\gamma_{re}}(\gamma_{re}) = \frac{\varepsilon_p^{m_p}}{\Gamma(m_p)g^s\gamma}\gamma_{re}^{m_p-1}e^{-\varepsilon_p\gamma_{re}}. \tag{43}$$

We can derive the BER at the receiver as follows:

$$\begin{aligned} P_e(\gamma) &= \frac{1}{2}\int_0^\infty erfc(\sqrt{\gamma_{re}})f_{\gamma_{re}}(\gamma_{re})d\gamma_{re} \\ &= \frac{\varepsilon_p^{m_p}}{\sqrt{\pi}\Gamma(m_p)g^s\gamma}\int_0^\infty\int_{\sqrt{\gamma_{re}}}^\infty e^{-y^2-\varepsilon_p\gamma_{re}}\gamma_{re}^{m_p-1}dyd\gamma_{re} \\ &= \frac{1}{\sqrt{2\pi}}\cdot\frac{\varepsilon_p^{m_p}}{(\varepsilon_p+1)^{m_p}g^s\gamma}. \end{aligned} \tag{44}$$

For a size l_{ah} ACK packet, the packet is error if there is at least one error bit, thus the PER is given by:

$$\rho(\gamma) = 1 - [1 - P_e(\gamma)]^{l_{ah}}. \tag{45}$$

The packet would be retransmitted if $\rho(\gamma) > 0$. Thus, the number of retransmissions x follows a geometric distribution with the excepted value $\frac{1}{1-\rho(\gamma)}$. We can derive Eq. (13) since x is an integer.

APPENDIX B
PROOF OF THEOREM 2

Suppose that the first DLT node accomplishes to transmit its block to the BS is denoted by j^*, whose computation latency and transmission latency are $t^c_{j^*,i}$ and $t^u_{j^*,i}$, respectively. Non-forking event means that no other block arriving at the BS in the period $T^* = t^c_{j^*,i} + t^u_{j^*,i} + t_w$. Therefore, the non-forking probability is given by:

$$P^{FR}_{nf,i} = \int_0^{\bar{t}}\int_0^{E_i^a}\prod_{j=1,j\neq j^*}^{N_i}\varphi f_{T_c}(t^c_{j^*,i})dt^c_{j^*,i}f^d_{t_u}(t^u_{j^*,i})dt^u_{j^*,i}, \tag{46}$$

where $j^* = \arg\min_{1\le j\le N_i}(t^c_{j,i} + t^u_{j,i})$, and φ is given by:

$$\begin{aligned} \varphi &= \Pr(t^c_{j^*,i} + t^u_{j^*,i} + t_w < t^c_{j,i} + t^u_{j,i} | t^c_{j^*,i} = T^c_{j^*,i}, t^u_{j^*,i} = T^u_{j^*,i}) \\ &= 1 - \int_0^{T^*}\int_0^{T^*-t^u_{j,i}} f_{T_c}(t^c_{j,i})dt^c_{j,i}f^d_{T_u}(t^u_{j,i})dt^u_{j,i} \\ &= \exp(-\lambda_i^c T^*)\int_0^{T^*} e^{\lambda_i^c t^u_{j,i}}f^d_{T_u}(t^u_{j,i})dt^u_{j,i} + 1 - \exp(\frac{1-2^{\frac{D}{B^d T^*}}}{\gamma_i^d g_i^d}). \end{aligned} \tag{47}$$

By combining Eq. (46) and Eq. (47), we can obtain the non-forking probability in Eq. (19) since $t^c_{j,i}$ and $t^u_{j,i}$ of different DLT nodes have identical distributions.

APPENDIX C
PROOF OF THEOREM 3

First, the complementary cumulative distribution function (CCDF) $F_i^{fa}(t)$ of the FA block is derived as follows:

$$\begin{aligned} F_i^{fa}(t) &= \Pr[\min_{1\le j\le N_i}(t^c_{j,i} + t^u_{j,i}) \ge t] \\ &\overset{(a)}{=} \left[\Pr(t^c_{j,i} + t^u_{j,i} \ge t)\right]^N \\ &= \left[1 - \int_0^t\int_0^{t-x} f_{T_c}(t^c)dt^c f^d_{T_u}(x)dx\right]^N \\ &= \left[1 - \exp(\frac{1-2^{\frac{D}{B^d t}}}{\gamma_i^d g_i^d}) + e^{-\lambda_i^c t}\int_0^t e^{\lambda_i^c x}f^d_{T_u}(x)dx\right]^N, \end{aligned} \tag{48}$$

where the second equation (a) holds since each DLT node has identical distribution for $t^c_{j,i}$ and $t^u_{j,i}$. Then, the average competition latency E_i^{fa} in one competition round is as follows,

$$E_i^{fa} = E\left[\min_{1\le j\le N_i}(t^c_{j,i} + t^u_{j,i})\right] = \int_{E_i^{fc}}^{E_i^a+\bar{t}} F_i^{fa}(t)dt, \tag{49}$$

where the upper limit of the integral is to limit the computing latency less than the average collecting latency E_i^a to avoid data overflow, and the lower limit of the integral is given by $E\left[\min(t^c_{j,i} + t^u_{j,i})\right] \ge E\left[\min(t^c_{j,i})\right] = E_i^{fc}$.

REFERENCES

[1] P. Lade, R. Ghosh, and S. Srinivasan, "Manufacturing analytics and industrial Internet of Things," *IEEE Intelligent Systems*, vol. 32, no. 3, pp. 74–79, May-June 2017.

[2] H. -N. Dai, Z. Zheng, and Y. Zhang, "Blockchain for Internet of Things: A survey," *IEEE Internet of Things Journal*, vol. 6, no. 5, pp. 8076–8094, Oct. 2019.

[3] Z. Yang, K. Yang, L. Lei, K. Zheng, and V. C. M. Leung, "Blockchain-based decentralized trust management in vehicular networks," *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 1495–1505, April 2019.

[4] P. Danzi, A. E. Kalor, C. Stefanovic, and P. Popovski, "Analysis of the communication traffic for blockchain synchronization of IoT devices," in *2018 IEEE International Conference on Communications (ICC)*, 2018, pp. 1–7.

[5] L. D. Nguyen, I. Leyva-Mayorga, A. N. Lewis, and P. Popovski, "Modeling and analysis of data trading on blockchain-based market in IoT networks," *IEEE Internet of Things Journal*, vol. 8, no. 8, pp. 6487–6497, 2021.

[6] H. Wei, W. Feng, C. Zhang, Y. Chen, Y. Fang, and N. Ge,"Creating efficient blockchains for the Internet of Things by coordinated satellite-terrestrial networks," *IEEE Wireless Communications*, vol. 27, no. 3, pp. 104–110, June 2020.

[7] S. Fu, J. Gao, and L. Zhao,"Integrated resource management for terrestrial-satellite systems," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 3, pp. 3256–3266, March 2020

[8] Y. Zhang and X. F. Liu, "Satellite broadcasting enabled blockchain protocol: A preliminary study," in *2020 Information Communication Technologies Conference (ICTC)*, 2020, pp. 118–124.

[9] S. Nakamoto, et al., "Bitcoin: a peer-to-peer electronic cash system," [Online] Available: https://bitcoin.org/en/bitcoin-paper. 2008.

[10] N. Chaudhry and M. M. Yousaf, "Consensus algorithms in blockchain: Comparative analysis, challenges and opportunities," in *2018 12th International Conference on Open Source Systems and Technologies (ICOSST)*, pp. 54–63, 2018.

[11] C. Cachin, "Yet another visit to Paxos," *IBM Res.*, Zrich, Switzerland, Tech. Rep. RZ3754, 2009.

[12] Reyna, A., et al. "On blockchain and its integration with IoT: Challenges and opportunities," *Future Generation Computer Systems* vol. 88, pp. 173–190, Nov., 2018.

[13] Y. Sun, E. Uysal-Biyikoglu, R. Yates, C. E. Koksal, and N. B. Shroff, "Update or wait: How to keep your data fresh," in *IEEE INFOCOM 2016 - The 35th Annual IEEE International Conference on Computer Communications*, pp. 1–9, 2016.

[14] S. Lee, M. Kim, J. Lee, R. H. Hsu, and T. Q. S. Quek,"Is blockchain suitable for data freshness? An age-of-information perspective," *IEEE Network*, vol. 35, no. 2, pp. 96–103, March/April 2021.

[15] A. Sugranes and A. Razi, "Optimizing the age of information for blockchain technology with applications to IoT sensors," *IEEE Communications Letters*, vol. 24, no. 1, pp. 183–187, 2020

[16] L. Luu, et al, "A secure sharding protocol for open blockchains," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, pp. 17–30, 2016.

[17] G. Yu, et al, "Survey: Sharding in blockchains," *IEEE Access*, vol. 8, pp. 14155–14181, 2020.

[18] E. K. Kogias, et al, "Enhancing bitcoin security and performance with strong consistency via collective signing," in *Proc. 25th USENIX Security Symp.*, pp. 279–296, 2016.

[19] E. Kokoris-Kogias, et al, "OmniLedger: A secure, scale-out, decentralized ledger via sharding," in *Proc. IEEE Symp. Secur. Privacy*, pp. 583–598, 2018.

[20] Mahdi Zamani, Mahnush Movahedi, and Mariana Raykova. "RapidChain: Scaling blockchain via full sharding," in *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security (CCS '18)*, New York, USA, pp. 931–948, 2018.

[21] H. Yoo, J. Yim, and S. Kim, "The blockchain for domain based static sharding," in *2018 17th IEEE International Conference On Trust, Security And Privacy In Computing And Communications/ 12th IEEE International Conference On Big Data Science And Engineering (TrustCom/BigDataSE)*, 2018, pp. 1689–1692.

[22] W. Tong, X. Dong, Y. Shen, and X. Jiang, "A hierarchical sharding protocol for multi-domain IoT blockchains," in *ICC 2019 - 2019 IEEE International Conference on Communications (ICC)*, 2019, pp. 1–6.

[23] J. -Y. Kwak, J. Yim, N.-S. Ko and S. -M. Kim, "The design of hierarchical consensus mechanism based on service-zone sharding," *IEEE Transactions on Engineering Management*, vol. 67, no. 4, pp. 1387–1403, Nov. 2020.

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

[25] T. Ali Syed, et al, "A Comparative Analysis of Blockchain Architecture and its Applications: Problems and Recommendations," *IEEE Access*, vol. 7, pp. 176838-176869, 2019.

[26] X. Liang, J. Jiao, S. Wu, and Q. Zhang, "Outage analysis of multirelay multiuser hybrid satellite-terrestrial millimeter-wave networks," *IEEE Wireless Communications Letters*, vol. 7, no. 6, pp. 1046–1049, Dec. 2018.

[27] M. K. Samimi and T. S. Rappaport, "3-D millimeter-wave statistical channel model for 5G wireless system design," *IEEE Transactions on Microwave Theory and Techniques*, vol. 64, no. 7, pp. 2207–2225, July 2016.

[28] A. D. Panagopoulos, P. M. Arapoglou, and P. G. Cottis, "Satellite communications at KU, KA, and V bands: Propagation impairments and mitigation techniques" *IEEE Communications Surveys & Tutorials*, vol. 6, no. 3, pp. 2–14.

[29] Kostic, I. M. , "Analytical approach to performance analysis for channel subject to shadowing and fading," *IEE Proceedings - Communications* vol. 152, no. 6, pp. 821–827, Dec. 2005.

[30] K. An, et al, "Performance limits of cognitive-uplink FSS and terrestrial FS for Ka-band," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 55, no. 5, pp. 2604–2611, 2019,

[31] B. Wang, J. Jiao, W. Wu, S. Wu, and Q. Zhang, "Age-critical blockchain resource allocation over satellite-based Internet of Things," in *2021 IEEE 94th Vehicular Technology Conference (VTC2021-Fall)*, 2021.

[32] G. Lee, J. Park, W. Saad, and M. Bennis, "Performance analysis of blockchain systems with wireless mobile miners," *IEEE Networking Letters*, vol. 2, no. 3, pp. 111–115, Sept. 2020.

[33] M. Rosenfeld, "Analysis of hashrate-based double spending," 2014, arXiv:1402.2009. [Online]. Available: http://arxiv.org/abs/1402.2009.

[34] FCC, "Application to Launch and Operate on a Satellite Space Stations filin," [Online]. Available: https://fcc.report/IBFS/SAT-LOA-20200526-00055/2378869, 2020.

[35] J. Liao, T. Tsai, C. He, and C. Tien, "SoliAudit: Smart contract vulnerability assessment based on machine learning and fuzz testing," in *2019 Sixth International Conference on Internet of Things: Systems, Management and Security (IOTSMS)*, 2019, pp. 458-465.

[36] S. R. Pokhrel, "Blockchain brings trust to collaborative drones and LEO satellites: An intelligent decentralized learning in the space," *IEEE Sensors Journal*, pp. 1-1, 2021.