On Convergence Probability of Direct Acyclic Graph-Based Ledgers in Forking Blockchain Systems

Zhilan Xie, *Student Member, IEEE*, Shuping Dang, *Member, IEEE*, Zhenrong Zhang, *Member, IEEE*

***Abstract*—Direct Acyclic Graph (DAG)-based ledger is a promising technology for the Internet of Things (IoT). Compared with a single-chain topology, DAG and forking blockchain topology can solve some problems in IoT, such as high resource consumption, high transaction fee, low transaction throughput, and long confirmation delay. We propose the convergence probability to aid further analysis of the performance and security of DAG-based ledgers. Under unsteady load regimes, the convergence probability is the probability of each possible cumulative weight of the observed transaction when it is approved by all new arrival transactions. In this paper, we derive a closed-form expression and an approximate expression of the convergence probability under the high-to-low regime (H2LR). Also, we verify the accuracy of the derived expressions through Markov chain Monte Carlo (MCMC) simulations. Numerical results shows that the simulation results match well with its analytical results, which indicates the accuracy of the exact expression and the approximate expression of the convergence probability.**

***Index Terms*—Convergence probability, blockchain, forking topology, direct acyclic graph, Markov chain Monte Carlo algorithm.**

I. Introduction

IN RECENT years, the Internet of Things (IoT) has aroused great interest in academia and industry [1]. The IoT will change our lives in many aspects, such as intelligent transportation, smart home, and health related applications. However, there are still some challenges remaining. First, security is the most critical issue [2], as the IoT systems are vulnerable to cyber-attacks, and data may be tampered by malicious parties. Second, users are also concerned about privacy [3], since their information is shared over the IoT. Third, with the limited capability of devices in the IoT [4], scalability is also one of the urgent issues to be addressed. The distributed ledger in blockchain system is tamper-proof and provides a higher level of encryption to ensure the data security of the IoT. In addition, blockchain has the characteristics of transparency and trust-building, which can quickly process transactions and reduce transaction costs. However, the speed and capacity of blocks generated by blockchain are limited, resulting in throughput limitation and long confirmation delay. Also, the participation of miners increases the transaction fees of blockchain. Therefore, the power-intensive and low-throughput nature of blockchain makes it unsuitable for power-constrained IoT devices. Consensus algorithms that are widely used in blockchain, such as Proof-of-Stake (PoS) and Proof-of-Work (PoW), are based on single-chain structures, which could lead to problems, e.g., long confirmation delay, high transaction fee, and low transaction throughput.

To overcome the above PoS and PoW related problems, a distributed ledger based on directed acyclic graph (DAG) was proposed. DAG consensus algorithm was proposed for the first time in [5]. DAG is a new technology with great potential for the development of the IoT. DAG technology is used in many fields, including improving the reliability and security of Internet of Vehicles (IoV) data [6] and federated learning frameworks [7], and solve the latency problem of mobile edge computing [8]. DAG technology gets rid of blocks and chains, and thus gets rid of the main pain points or limitations of blockchains, such as fees, scalability, and throughput. Therefore, DAG-based ledgers are more suitable for IoT systems that mainly composed of micro-payment transactions. It contains several consensus algorithms, such as Tangle [5], Byteball [9], and HashGraph [10]. In DAG, each transaction is validated concurrently, resulting in a large amount of verification for transactions in parallel. Theoretically, there is no upper limit to the transaction throughput [11]. DAG is an accumulation-oriented distributed ledger that does not require professional miners. Therefore, the resource consumption and transaction costs per node can be reduced to a low level. In [12], Li et al. leveraged the Markov chain Monte Carlo (MCMC) algorithm[1] to analyze the performance and security of Tangle with four different network loads, including high load regime (HR), low load regime (LR), high-to-low load regime (H2LR), and low-to-high load regime (L2HR). In H2LR, when the number of tips decrease from the initial to two, the observed transaction must be verified by the newly arrived transaction. Therefore, the tip selection algorithm plays a crucial role in terms of system security.

The convergence probability represents the probability of each possible cumulative weight of an observed transaction when it is indirectly validated by all new transactions in H2LR. Among the four network load regimes, H2LR has the slowest cumulative weight growth rate and the longest confirmation delay, resulting in the highest probability of successful attacks by the attackers under the same conditions. Therefore, we focus on deriving the convergence probability of H2LR, which is helpful to gain insight into the performance and security of DAG-based ledger under H2LR. In H2LR, by comparing the cumulative weight reaching the convergence condition with the confirmation threshold, we can evaluate the confirmation delay and the growth rate of the cumulative weight of the system. To the best of the authors' knowledge, existing literature did not give an analytical expression for the convergence probability. In this regard, we derive a closed-form expression and an approximate expression of convergence probability under H2LR in this paper. Through the derived results,

This work was supported in part by Guangdong Guangxi Joint Science Key Foundation under grant 2021GXNSFDA076001 and in part by Guangxi Major Projects of Science and Technology under grants 2020AA21077007 and 2020AA24002AA.

Zhilan Xie is with the School of Computer, Electronics and Information, Guangxi University, Nanning 530004, China (e-mail: 2013391128@st.gxu.edu.cn).

Shuping Dang is with Department of Electrical and Electronic Engineering, University of Bristol, Bristol BS8 1UB, UK (e-mail: shuping.dang@bristol.ac.uk).

Zhenrong Zhang is with the Guangxi Key Laboratory of Multimedia Communications and Network Technology, School of Computer Electronics and Information, Guangxi University, Nanning 530004, China (e-mail: zzr76@gxu.edu.cn).

[1]The MCMC algorithm can effectively solve the parasite chain attack. The MCMC algorithm needs to combine other mechanisms to effectively deal with other types of double-spending attacks [10] [13].

we can also reveal and quantify the impacts of a number of parameters on the convergence probability of DAG-based ledgers in forking blockchain systems. This helps to understand and improve the performance and security so that DAG-based ledgers can better serve IoT. The derived expressions for convergence probability complements the H2LR security analysis in [12].

II. Basics of Direct Acyclic Graph-Based Ledgers

The core of the DAG consensus is attaching new transactions to the topology. A DAG requires that a new transaction must select one or more previous transactions to verify. Transactions are not bundled into blocks in a DAG; as a result, each transaction can be viewed as a node in the context of graph theory; directed lines representing verification relations between two transactions can be regarded as edges of a graph. The history of transactions forms a DAG. Compared to classical single-chain structures, the DAG applies the heaviest chain consensus instead of the longest chain consensus. Each incoming unit in the DAG validates its parent unit(s) and those unit(s) directly or indirectly linked to its parent unit(s), until it reaches the genesis transaction and obtains the hash of the genesis transaction. By verifying transactions, the DAG can significantly reduce transaction fees, which is a breakthrough for frequent and micro-payment transactions in the IoT.

There are three main consensus algorithms that can efficiently generate DAGs. Among them, the entire network in Tangle participates in the verification of transaction legitimacy and can be used on a large scale without transaction fees. And the offline asynchronous processing capability of the Tangle is particularly important in IoT applications. Therefore, Tangle is a more suitable consensus algorithm for IoT [5]. Transactions in Tangle are issued by nodes. When a new transaction arrives, it would typically approve two previous transactions that have not been validated before. These unverified transactions are known as tips. In particular, Tangle uses the MCMC algorithm to select tips for verification. These approvals between transactions are represented by directed edges. When there is only one directed edge between two transactions, it is called direct approval; on the other hand, an indirect approval occurs when there is more than one directed edges between two transactions. Each transaction has its own weight, which is proportional to the amount of work allocated by the issuer. Without losing generality, we configure the average weight of each transaction to unity. And the cumulative weight of a transaction is equal to its own weight plus the own weights of all transactions that directly or indirectly validate the transaction. When the cumulative weight of a transaction reaches the confirmation threshold, the transaction is confirmed by Tangle. The greater the cumulative weight is, the higher the confirmation level will be.

Li et al. in [12] analyzed the cumulative weights, confirmation delays, and security under four different network regimes. HR and LR are two steady regimes. Let h_r be the duration time of transaction revealing, and let the transaction arrival rate be λ. If the average inter-arrival time between two transactions $h = 1/\lambda \leq h_r$, the network load regime is called HR. On the contrary, it is called LR, instead. In HR, the transaction arrival rate is high, which is denoted as λ_h; And the number of tips in HR is $L(t) = 2\lambda_h h_r$. In LR, the transaction arrival rate is low, which is similarly denoted as λ_l; And the number of tips in LR is $L(t) = 2\lambda_l h_r = 1$. H2LR and L2HR are two transitional regimes. H2LR is the mode by which the network load changes from HR to LR. Therefore its transaction arrival rate change from λ_h to λ_l, and the number of tips decrease from $L(t) = 2\lambda_h h_r$ to $L(t) = 1$. L2HR is the converse to H2LR, representing the transitional state from a lower amount of load to a higher amount of load.

When a new transaction arrives, it selects two tips to approve. The tip identity will be nullified once selected. Instead, the new transaction becomes a new tip. Therefore, when a new transaction arrives, the number of tips for H2LR will be reduced by one. During the tip selection process, the cumulative weight of observed transaction would increase by one either by direct or indirect approval. When the number of tips changes to two, the cumulative weight of the observed transaction will present different values depending on its verified times. Let $\{i, j\}$ be the state of a transaction, where i and j represent the cumulative weight of the transaction and the existing number of tips. Based on the above description, the one-step transition probability is given by [12]

$$\begin{cases} P\{i+1, j-1|i, j\} = 2/j, \\ \qquad i = 1, 2, \cdots, L_h - 1;\ j = 2, 3, \cdots, L_h, \\ P\{i, j-1|i, j\} = 1 - 2/j, \\ \qquad i = 1, 2, \cdots, L_h - 1;\ j = 2, 3, \cdots, L_h, \\ P\{i+1, 1|i, j\} = 1,\ i = 1, 2, \cdots, \infty;\ j = 1, \end{cases} \tag{1}$$

where L_h is the number of tips in HR. $P\{i+1, j-1|i, j\}$ in (1) represents the probability that the observed transaction has been verified by a new transaction. Thus, its cumulative weight increases by one, and the number of tips decreases by one. $P\{i, j-1|i, j\}$ stands for the probability that the observed transaction has not been approved. Therefore, its cumulative weight remains the same, and the number of tips decreases by one. $P\{i+1, 1|i, j\}$ indicates that H2LR has been transformed into LR, and thus, the number of tips is one. As a result of LR, each new transaction would choose the last tip for verification, and hence the probability that the observed transaction is indirectly verified by the incoming transaction becomes one.

The convergence probability, denoted as $P_{\{i,2\}}$, is defined as the probability that when the number of tips is two, the cumulative weight of an observed transaction is i. In short, $P_{\{i,2\}}$ is the probability that the observed transaction has been directly or indirectly verified for $i-1$ times when H2LR has completed and transferred to LR. Clearly, $P_{\{i,2\}}$ is a multi-step transition probability from state $\{1, L_h\}$ to state $\{i, 2\}$. The convergence probability is directly related to the probability of successful attack of parasite chain in H2LR and of paramount importance for reliability and security analysis of forking blockchain systems.

III. Derivation of the Convergence Probability

In this section, we derive the closed-form expression and approximate expression of the convergence probability as follows.

Theorem 1: Let a_1, a_2,..., a_N denote the N cases of selecting $i-1$ numbers from 1 to $L_h - 2$, where $N = \binom{L_h-2}{i-1}$. Denote $\mathcal{L}_1(c) \in \{a_1, a_2, \ldots, a_N\}$, where c is the serial number of a. As a result, the convergence probability can be written as

$$P_{\{i,2\}} = \frac{2^i}{L_h(L_h-1)} \sum_{c=1}^{\binom{L_h-2}{i-1}} \left(\prod_{k \in \mathcal{L}_1(c)} k \right)^{-1}. \tag{2}$$

Proof: The H2LR model is a Markov process, and because its time and state are discrete it is also a Markov chain. In order to figure out the multi-step transition probability of a Markov chain, we resort to the Chapman-Kolmogorov (C-K) equation. Let P_{ij} be the one-step transition probability from state i to state j, and we denote P_{ij}^n as the n-step transition probability from state i to state j, which can be explicitly expressed as $P_{ij}^n = P\{X_{n+k} = j|X_k = i\} = P\{X_{n+k} = j|X_k = i, X_{k-1}, \ldots, X_1\}$, given $n \geq 0$, $i, j \geq 1$ and $j \geq i$. The C-K equation provides a calculation method for multi-step transition probability: $P_{ij}^{n+m} = P\{X_{n+m} = j|X_1 = i\} = \sum_{k=1}^{\infty} P\{X_{n+m} = j, X_n = k|X_1 = i\} = \sum_{k=1}^{\infty} P\{X_{n+m} = j|X_n = k, X_1 = i\}P\{X_n = k|X_1 = i\} = \sum_{k=1}^{\infty} P_{kj}^m P_{ik}^n$. The expended multi-step transition probability by the C-K equation breaks down the probability of an intact process into the probabilities of the process from initial state i to intermediate state k and the process from intermediate state k to final state j. As k is not unique, summing over all intermediate states yields the probability that the process will be in state j after $n+m$ transitions. The C-K equation can be written in a matrix form: $\mathbf{P}^{(n+m)} = \mathbf{P}^{(n)}\mathbf{P}^{(m)}$, where $\mathbf{P}^{(n)}$ represents the matrix of the n-step transition probability $P_{ij}^n, \forall\, i, j$.

According to the nature of a generic multi-step transition, its probability matrix can be obtained by multiplying all the previous one-step transition probability matrices. Through (1), we can easily figure out the one-step transition probability of every step. There are L_h-2 steps from initial state $\{1, L_h\}$ to state $\{i, 2\}$, and therefore, $P_{\{i,2\}}$ is in essence the (L_h-2)-step transition probability from state $\{1, L_h\}$ to state $\{i, 2\}$. As a result, the (L_h-2)-step transition probability matrix can be determined by multiplying the previous (L_h-2) one-step transition probability matrices. The one-step transition probability matrix is an $(L_h-2)\times(L_h-1)$ matrix. In order to facilitate the following calculation, we add a row of zero elements to make the one-step transition probability matrix become an $(L_h-1)\times(L_h-1)$ augmented matrix.

Because there are only two paths in the first transition, the cumulative weight remains unchanged or changes from 1 to 2. Accordingly, the (L_h-2)-step transition probability matrix is obtained by multiplying the first one-step transition probability matrix to the (L_h-2)th one-step transition probability matrix, which gives

$$\begin{array}{c} \\ 1 \\ 2 \\ \vdots \\ L_h-2 \\ L_h-1 \end{array} \begin{array}{c} \begin{array}{ccccc} 1 & \cdots & i & \cdots & L_h-1 \end{array} \\ \begin{pmatrix} \alpha_{11} & \cdots & \alpha_{1i} & \cdots & \alpha_{1(L_h-1)} \\ 0 & \cdots & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & \cdots & 0 & \cdots & 0 \\ 0 & \cdots & 0 & \cdots & 0 \end{pmatrix} \end{array}, \tag{3}$$

where $\alpha_{11} = P_{\{1,2\}} = (1-\frac{2}{L_h})(1-\frac{2}{L_h-1})\times\cdots\times(1-\frac{2}{3})$, $\alpha_{1(L_h-1)} = P_{\{L_h-1,2\}} = (\frac{2}{L_h})(\frac{2}{L_h-1}\times\cdots\times(\frac{2}{3})$. Hence $\alpha_{1i} = P_{\{i,2\}}$ can be written as

$$P_{\{i,2\}} = \frac{2^{i-1}}{L_h(L_h-1)\times\cdots\times 3} \sum_{c=1}^{\binom{L_h-2}{L_h-2-(i-1)}} \left(\prod_{k\in\mathcal{L}_2(c)} k\right), \tag{4}$$

where $b_1, b_2, \ldots, b_M$ represent the M cases of selecting $L_h-2-(i-1)$ numbers from 1 to L_h-2. Denote $\mathcal{L}_2(c) \in \{b_1, b_2, \ldots, b_N\}$, and c is the serial number of b. It is apparent that $M = \binom{L_h-2}{L_h-2-(i-1)}$ and $|\mathcal{L}_2(c)| = L_h-2-(i-1) = L_h-i-1$. By simple mathematical manipulations, (4) can be reduced to (2), which proves Theorem 1. ∎

As what we derived is *de facto* a probability distribution $P_{\{i,2\}}$ for $i = 1, 2, \ldots, L_h-1$, we can mathematically examine the correctness of (2) by analyzing its fundamental attributes as a probability distribution. We hereby propose and prove *Proposition 1* as follows:

Proposition 1: Non-negativity and normativity are two necessary attributes of a legitimate probability distribution of discrete random variables. For the derived probability distribution $\{P_{\{i,2\}}\}_{i=1}^{L_h-1}$, both non-negativity and normativity are held.

Proof: The initial state of the transaction has its own weight, leading to cumulative weight $i \geq 1$. It is clear that before reaching state $\{i, 2\}$, $L_h \geq 3$ for the transaction of interest is always valid. Therefore, according to (2), the value of $P_{\{i,2\}}$ is always greater than or equal to zero, $\forall\, i \in \{1, 2, \ldots, L_h-1\}$, which proves the non-negativity of the derived probability distribution. In terms of normativity, we can prove it by the following derivation:

$$\sum_{i=1}^{L_h-1} P_{\{i,2\}} = \sum_{i=1}^{L_h-1} \frac{2^i}{L_h(L_h-1)} \sum_{c=1}^{\binom{L_h-2}{i-1}} \left(\prod_{k\in\mathcal{L}_1(c)} k\right)^{-1} = 1. \tag{5}$$

∎

The derived exact expression of the convergence probability is of high analytical complexity, albeit in closed form. As it is common in practical IoT networks that $L_h \gg i$, we can further simplify the derived relation given in (2) by leveraging $L_h \gg i$ and obtain an approximate expression of the convergence probability for the special case. We hereby propose and prove *Proposition 2*:

Proposition 2: When $L_h \gg i$, we can approximate $P_{\{i,2\}}$ by the following relation:

$$P_{\{i,2\}} = \frac{2^i\binom{L_h-2}{i-1}}{L_h(L_h-1)}\mathbb{E}\left\{\frac{1}{A}\right\} \approx \frac{2^i\binom{L_h-2}{i-1}}{L_h(L_h-1)}\left(\frac{1}{L_h-2}\sum_{m=1}^{L_h-2}\frac{1}{m}\right)^{i-1}. \tag{6}$$

Proof: Let $A = m_1m_2\times\ldots\times m_{i-1}$, where m_1, m_2, …, m_{i-1} are the sequence numbers of the verified steps. When $L_h \gg i$, m_1, m_2, …, m_{i-1} can be regarded as mutually independent and equivalent. Consequently, the expectation of the reciprocal of A can be approximated as $\mathbb{E}\{\frac{1}{A}\} = \mathbb{E}\left\{\frac{1}{m_1m_2\times\ldots\times m_{i-1}}\right\} \approx \mathbb{E}\left\{\frac{1}{m_1}\right\}\mathbb{E}\left\{\frac{1}{m_2}\right\}\times\ldots\times\mathbb{E}\left\{\frac{1}{m_{i-1}}\right\} = \left(\mathbb{E}\left\{\frac{1}{m}\right\}\right)^{i-1} = \left(\frac{1}{L_h-2}\sum_{m=1}^{L_h-2}\frac{1}{m}\right)^{i-1}$, where $\mathbb{E}\{\cdot\}$ returns the expected value of the enclosed. This derived approximation directly validates the final equality presented in (6). ∎

IV. Numerical Results

To corroborate the proposed exact expression and approximate expression of the convergence probability under H2LR, we present and discuss the simulation results in this section. All the numerical results are obtained using MATLAB.

We first vary target parameter L_h to substantiate the analytical results regarding the convergence probability derived in (2) and (6). To reveal the impact of i, the cumulative weight of the observed transaction when the number of tips is 2, we set i=2, 3 and 4 to make a comparison. The simulation results are illustrated in

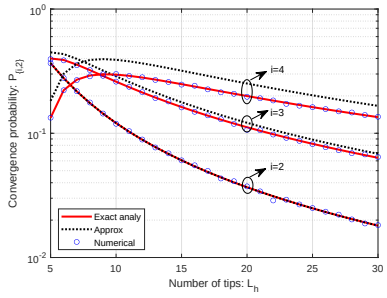

Fig. 1. Convergence probability vs. the number of tips, given $i = 2, 3, 4$.

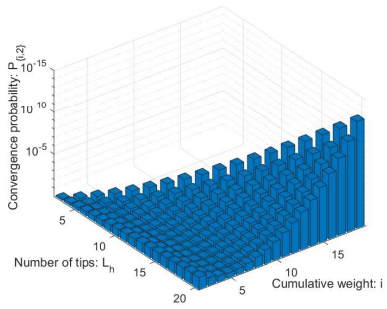

Fig. 2. Convergence probability vs. the number of tips and the cumulative weight.

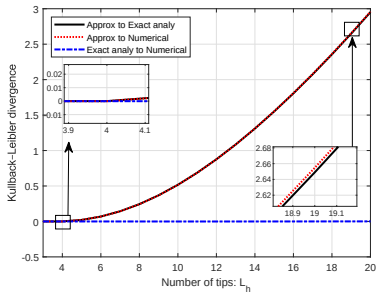

Fig. 3. K-L divergence vs. the number of tips between different sets of data.

Fig. 1. By observing Fig. 1, it is clear to see that the simulation results match well with its analytical results, which indicates the accuracy of (2). Also, the approximate expression curves and the exact expression curves have roughly the same change trend, which verifies (6). Meanwhile, when L_h remains unchanged, with an increasing i, the gaps between the approximate expression curves and the exact expression curves become larger. On the contrary, when i remains unchanged, increasing L_h will lead to a smaller gap. The reason is that when $L_h \gg i$, the tip selection process can be approximately regarded as independent. Concerning the effects of i and L_h, further investigations are worthwhile. The relation between convergence probability and i as well as L_h is shown in Fig. 2. From this figure, it is evident that when i is equal to unity, increasing L_h will result in a lower convergence probability. On the contrary, when $i \geq 2$, the convergence probability gradually increases with an increasing L_h.

The Kullback-Leibler (K-L) divergence can be used to measure the distance between two distributions, and the convergence probability is actually a probability distribution. Therefore, we analyze the K-L divergence among exact expression, approximate expression and simulation results. We assume that $G(\varphi)$ and $Q(\varphi)$ are two different distributions, and the expression of the $G(\varphi)$ to $Q(\varphi)$ K-L divergence can be written as $\mathsf{KL}(G \parallel Q) = \sum_{\varphi} G(\varphi) \log\left(\frac{G(\varphi)}{Q(\varphi)}\right)$. The results in Fig. 3 show that the K-L divergence of the exact analysis to the numerical simulation is approximately equal to zero and not affected by L_h, which means that the analytical results almost perfectly match the simulation results, and the exact expression has good robustness. Because the analytical results and the simulation results are closely matched, the K-L divergence curve of the approximate expression to the exact expression is with limited difference from the K-L divergence curve of the approximate expression to the simulation results. Meanwhile, due to the inaccuracy of the approximate expression, its K-L divergences increase rapidly with an increasing L_h and are much greater than zero.

V. Conclusion

In this paper, by studying the H2LR model of the Tangle and its state transition rules, we derived the closed-form expression of the convergence probability. In addition, by assuming that the tip selection process is independent when L_h is much greater than i, we obtained the approximate expression of the convergence probability in a much simplified form. Furthermore, we performed a series of numerical simulations and verified the accuracy of the derived expressions for the convergence probability. Through the numerical results, we have obtained more insights into the DAG-based ledger and the impacts of parameters on its performance in forking blockchain systems. The convergence probability can be used as a performance criterion for transactions that satisfy convergence conditions and facilitates security analysis under H2LR. It is anticipated that the expressions derived can expedite the research progress of forking blockchain systems.

References

[1] J. K. Reena and R. Parameswari, "A smart health care monitor system in IoT based human activities of daily living: A review," in *Proc. IEEE COMITCon*, Faridabad, India, 2019, pp. 446–448.

[2] W. Iqbal *et al.*, "An in-depth analysis of IoT security requirements, challenges, and their countermeasures via software-defined security," *IEEE IoT J.*, vol. 7, no. 10, pp. 10 250–10 276, Oct. 2020.

[3] C. Li and B. Palanisamy, "Privacy in Internet of Things: From principles to technologies," *IEEE IoT J.*, vol. 6, no. 1, pp. 488–505, Feb. 2019.

[4] M. N. Khan, A. Rao, and S. Camtepe, "Lightweight cryptographic protocols for IoT-constrained devices: A survey," *IEEE IoT J.*, vol. 8, no. 6, pp. 4132–4156, Mar. 2021.

[5] S. Popov, "The tangle," *White paper*, vol. 1, no. 3, 2018.

[6] Y. Lu, X. Huang, K. Zhang, S. Maharjan, and Y. Zhang, "Blockchain empowered asynchronous federated learning for secure data sharing in internet of vehicles," *IEEE Trans. Veh. Technol.*, vol. 69, no. 4, pp. 4298–4311, 2020.

[7] ——, "Low-latency federated learning and blockchain for edge association in digital twin empowered 6G networks," *IEEE Trans. Industr. Inform.*, vol. 17, no. 7, pp. 5098–5107, 2020.

[8] X. Huang, S. Leng, S. Maharjan, and Y. Zhang, "Multi-agent deep reinforcement learning for computation offloading and interference coordination in small cell networks," *IEEE Transactions on Vehicular Technology*, vol. 70, no. 9, pp. 9282–9293, 2021.

[9] A. Churyumov, "Byteball: A decentralized system for storage and transfer of value," *URL https://byteball. org/Byteball. pdf*, 2016.

[10] L. Baird, "The swirlds hashgraph consensus algorithm: Fair, fast, Byzantine fault tolerance," *Swirlds Tech Reports*, 2016.

[11] B. Cao, Z. Zhang, D. Feng, S. Zhang, L. Zhang, M. Peng, and Y. Li, "Performance analysis and comparison of pow, pos and dag based blockchains," *Digital Communications and Networks*, vol. 6, no. 4, pp. 480–485, 2020.

[12] Y. Li *et al.*, "Direct acyclic graph-based ledger for Internet of Things: Performance and security analysis," *IEEE/ACM Trans. on Net.*, vol. 28, no. 4, pp. 1643–1656, Aug. 2020.

[13] M. Rosenfeld, "Analysis of hashrate-based double spending," *arXiv preprint arXiv:1402.2009*, 2014.