drones

MDPI

Article

LECast: A Low-Energy-Consumption Broadcast Protocol for UAV Blockchain Networks

Haoxiang Luo [1,*], Shiyuan Liu [2], Shizhong Xu [1] and Jian Luo [3]

[1] Key Laboratory of Optical Fiber Sensing and Communications (Ministry of Education), University of Electronic Science and Technology of China, Chengdu 611731, China
[2] Department of Electrical Engineering, Stanford University, Palo Alto, CA 94305, USA
[3] Meishan Branch, State Grid Corporation of China, Meishan 620010, China
* Correspondence: lhx991115@163.com

Abstract: With the continuous development of communication technology, drones are playing an important role in many fields, such as power transmission line inspection and agricultural pesticide spraying. In order to protect the data privacy and communication security of drones, many experts are considering blockchain as its enabling technology. However, due to their small size and limited power storage, drones cannot support energy-intensive blockchain applications. In addition, the future 6G communications need to implement an important key performance indicator, namely extremely low-power communications (ELPCs). As a consequence, research into green blockchain is becoming more and more popular. The broadcast of the blockchain is one of the most energy-intensive parts because it entails flooding and there are a lot of unnecessary communication processes. Therefore, in order to make blockchain more suitable for ELPC requirements in 6G communications and unmanned aerial vehicle (UAV) networks, we took the blockchain broadcast as an improvement candidate and designed LECast, a low-energy-consumption protocol. LECast first analyzes the energy consumption model of the communication between two drones and constructs the shortest-path broadcast tree (SPB Tree) for the UAV networks to minimize energy consumption. Meanwhile, to make the sending drone address the receiving drone in a more convenient way, we proposed an extended Huffman coding (EHC) scheme to name the drones. Furthermore, the other issues with the broadcast tree are reliability and security. When a channel fails, subsequent drones cannot smoothly receive the transaction or block data. As a result, we introduced multichannel transmission with splitting data (MTSD); that is, the transaction or block data are divided into segments and transmitted in parallel multiple times over multiple channels. Finally, through the analysis and simulation of LECast in terms of energy consumption, latency, throughput, reliability, security, and coverage rate, the advantages of LECast were confirmed, which could meet the requirements of ELPCs and be well applied to UAV networks.

Keywords: unmanned aerial vehicle (UAV) networks; blockchain; broadcast tree; green communication; energy consumption

Citation: Luo, H.; Liu, S.; Xu, S.; Luo, J. LECast: A Low-Energy-Consumption Broadcast Protocol for UAV Blockchain Networks. *Drones* **2023**, *7*, 76. https://doi.org/10.3390/drones7020076

Academic Editor: Vishal Sharma

Received: 30 December 2022
Revised: 12 January 2023
Accepted: 18 January 2023
Published: 20 January 2023

1. Introduction

Currently, the blockchain is playing an important role in various applications of UAVs due to its security characteristics (Figure 1), such as power transmission line inspection and agricultural pesticide spraying. However, due to the small size of drones, the capacity of their batteries is limited, making it difficult for them to carry the energy-intensive blockchain. Furthermore, after decades of rapid development, various information technologies have gradually encountered the bottleneck of energy consumption, and the rapidly increasing energy consumption has evolved into one of the factors of global warming. According to a report by China Mobile Communications Corporation LTD, the mobile communication network operated by the company consumed 24.7 billion kilowatt-hours of

electricity in 2018. The energy consumption of 5G is 3–4 times that of 4G; thus, the total energy consumption of the network is still rising. In the future, as the Information age moves toward 6G communication, it will be necessary to use a large amount of computing power to support artificial intelligence applications, which will require deploying a large number of additional edge servers and will significantly increase energy consumption [1].

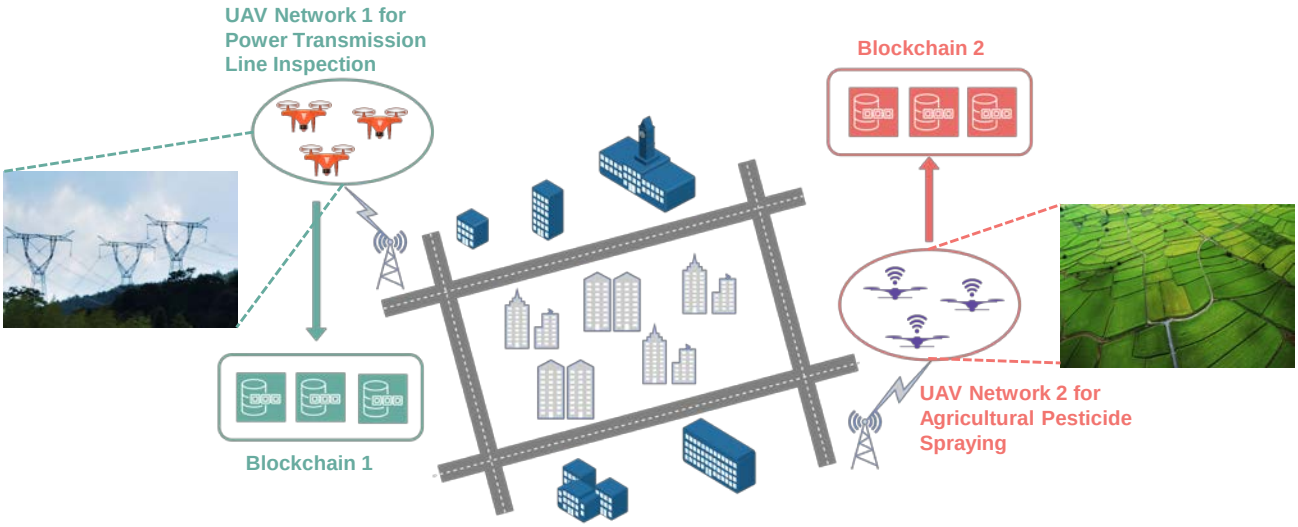


Figure 1. UAV Blockchain Networks.

As is known to all, energy crises and environmental pollution have become two major bottlenecks restricting the sustainable development of human society. Meanwhile, regarding 6G communications, Ying-Dar Lin proposed an important key performance indicator, ELPC [2]. Guangyi Liu pointed out that the energy consumption of 6G communications should be one-thousandth to one-tenth of that of 5G communications [3]. Therefore, it is urgent to research and break through theory and technology that can greatly reduce the energy consumption of whole networks while improving the capacity of mobile communication systems so as to achieve green communication.

1.1. Research Motivation

Recently, blockchain technology has been considered as an important means to effectively solve network and information security problems as it is distributed, decentralized, cannot be tampered with, is auditable, and has other characteristics. It has been widely used in the Internet of Things (IoT) [4], Internet of Vehicles (IoV) [5], Internet of Energy (IoE) [6], and Internet of Medical Things (IoMT) [7]. Many scholars regard it as an indispensable potential technology and a new paradigm in 6G communications, and it can enable 6G networks to achieve native security [2,3,8–12]. In the above applications, the blockchain has shown its great potential and possibilities in UAV networks.

However, the complex workflow (Figure 2) of the blockchain consumes a lot of electrical energy when it is actually deployed, which is contrary to the goal of ELPC in 6G communications and cannot easily meet the needs of low-energy UAV applications. For example, in Bitcoin, a typical blockchain application, energy consumption will soon reach 7.67 GW per year, which is very close to the total annual energy consumption of Austria (8.2 GW) [13]. Similarly, Ethereum also has the problem of consuming too much energy [14]. Therefore, we need to further improve blockchain technology to reduce its energy consumption and allow it to meet the ELPC goals in the 6G era and be well-applied in UAV networks. Faced with this problem, researchers have been trying to develop a more advanced and energy-efficient blockchain technology from the aspects of the ledger structure, performance evaluation framework, consensus algorithm, and so on. Ansh Riyal et al. [15] designed a hierarchical ledger structure called a blockchain tree, which can effectively save energy. Kai Qian et al. [16] provided a performance-testing framework for

the IoT-oriented blockchain, which can optimize energy consumption in a targeted manner. Xiaoqiong Xu et al. [4] proposed an energy-efficient Practical Byzantine Fault-Tolerance (PBFT) consensus algorithm, security-guaranteed PBFT (S-PBFT), for energy-constrained IoT applications.

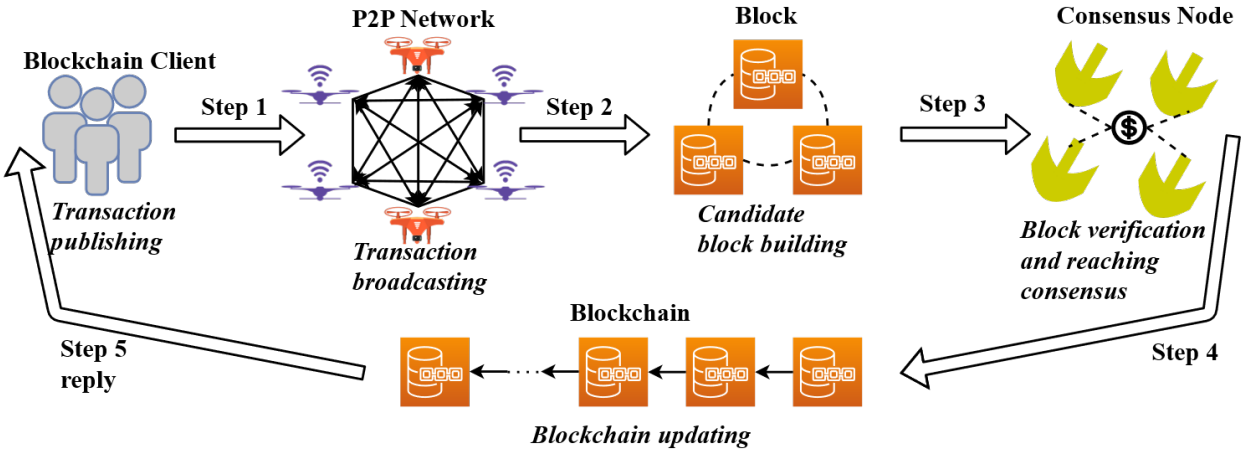


Figure 2. Blockchain Workflow.

Further, as an indispensable process in the blockchain network, broadcast (Step 2 in Figure 2) plays an important role in diffusing transactions or block data to the whole network. Many scholars often pay attention to its broadcast latency and throughput. For example, [17] proposed a broadcast protocol named Urocissa for blockchain networks with low latency. However, few scholars have paid attention to its energy consumption. The flooding method is often used to spread transactions and block data, so there are many unnecessary communication processes leading to a large amount of energy consumption. As a result, the broadcast protocol in blockchain networks is a very important part for reducing the energy consumption of blockchain systems.

To sum up, for the current green blockchain technology, there are still the following deficiencies:

- There is almost no research on green and energy-saving blockchains from the perspective of the broadcast protocol in blockchain networks;
- The low-energy-consumption broadcast protocol will inevitably avoid repetitive communication, but it will also affect the reliability and security of the broadcast. Therefore, how to balance these performances is also an important issue.

1.2. Our Contributions

- First, considering the limited capacity of a drone's battery, the concept of a green blockchain network is proposed. The paper investigates the blockchain workflow, choosing the broadcasting process for energy consumption improvement. Then, a complete LECast broadcast protocol is presented in detail. To the best of our knowledge, this is the first study to improve blockchain energy consumption from the broadcast protocol perspective;
- Second, we model the energy consumption of communication between two drones, and build a broadcast tree, named the SPB Tree, for UAV blockchain networks, which can minimize the energy overhead. Meanwhile, the SPB Tree is not necessarily a binary tree, and the traditional binary naming rule cannot name the nodes (drones) within it. Therefore, we further transform the naming problem into an information source coding problem, proposing the EHC scheme, a node-naming rule for multiway trees;
- Furthermore, when a channel fails, subsequent nodes in the broadcast tree cannot receive the transaction or block data. Therefore, we propose MTSD, which splits the transaction or block data into multiple sub-data, and they are transmitted in parallel over multiple channels multiple times to ensure the security and reliability of the

SPB Tree. To our best knowledge, this is the first study to combine multi-channel transmission and data-splitting in the blockchain network;
- Finally, we also analyze and simulate the broadcast latency, throughput, and broadcast coverage of the LECast protocol constructed from the SPB tree. The results show that LECast can not only effectively solve the high energy consumption problem in broadcasting, but also maintain superior performance in other aspects.

1.3. Structure of This Paper

The remaining contents of this paper are arranged as follows. Section 2 reviews the related work. Section 3 introduces LECast, including the construction method of the SPB Tree, the node-naming rule EHC, and how MTSD ensures the security and reliability of LECast in detail. Then, in Section 4, we analyze the security, reliability, broadcast coverage, broadcast latency, throughput, and energy consumption of the LECast protocol. Then, these performances are verified by simulation in Section 5. Finally, Section 6 is the conclusion of this work.

2. Related Work

This section is divided into two parts to review the relevant work, namely that related to green blockchain technology and the broadcast protocol in blockchain networks.

2.1. Green Blockchain Technology

More and more researchers are realizing that the consensus algorithm, data storage, etc. in blockchain will lead to high energy consumption. There have been some improvements in and optimizations of the above parameters.

Firstly, there are many researchers working on an energy-efficient consensus to reduce the energy costs in the blockchain. In [4], the authors studied the energy-consumption model for node communication in the IoT and proposed a node-selection scheme based on residual energy. Further, the authors designed a consensus S-PBFT with high energy efficiency, and these nodes participated in the consensus as master nodes to reduce energy consumption. In addition, in [17], the authors improved the PBFT consensus based on synchronous transmission and proposed a new, real-time, lightweight Byzantine consensus (RELI) for the IoT. When an IoT system has 45 nodes, the operation speed can be increased by 80% and the radio connection time can be reduced by 78%. The simulation results showed that their works could be well-applied to low-power IoT systems. Moreover, H. S. Saeed et al. [18] studied the energy consumption performance of Proof-of-Work (PoW) in the UAV network. The results showed that, although Byzantine UAVs linearly increase the consensus energy cost, energy consumption can be reduced by increasing the total number of UAVs.

Second, there are also researchers improving the blockchain storage mode to save energy. As we all know, each node in the blockchain needs to back up the entire system's data so that the data cannot be tampered with. However, each node is involved in storing all of the data, which will not only need a lot of storage resources, but also cause energy consumption in the storage process. In [15], the authors improved the blockchain storage structure and proposed a Blockchain Tree with hierarchical nodes. Then, they deleted the data stored on underlying nodes periodically to save storage resources and energy. S. Iqra et al. [19] proposed a blockchain-based green big-data-visualization (BGbV) scheme. It uses Hyperledger Sawtooth to optimize resource utilization, which can guarantee security at a lower storage cost. Additionally, considering that it uses less energy, this is also an environmentally friendly solution. Moreover, sharding can not only limit the storage cost in a certain range, but also save energy by reducing communication times, as mentioned in [20–22].

Third, due to the lack of testing tools for blockchain performance, some scholars have built performance-testing tools to quantitatively describe the energy consumption of blockchain in order to improve its performance accordingly. In [23], the first blockchain-

testing tool "Blockbench" was designed. It integrates many contracts to evaluate the basic performance of the blockchain in a variety of scenarios. In [16], the authors proposed a novel framework to achieve a green IoT-oriented blockchain in 6G communications, named "fine-grained benchmarking and targeted optimization".

By reviewing the above technologies, we found that there have been few studies on improving blockchain energy consumption from the broadcast protocol perspective. Therefore, we chose to model the energy consumption of broadcasting in blockchain networks in order to design an improved green broadcast protocol.

2.2. Broadcast Protocol in Blockchain Networks

The earliest blockchain broadcast protocols were from Gossip, a flooding method in Bitcoin [24]. Flooding is an unstructured overlay network, which is relatively robust but has high data redundancy and communication overhead. This approach has been proven by [25] to have a serious negative impact on the scalability of blockchain networks. Ethereum has since improved this flaw by using a structured broadcast tree called Kademlia [26]. This is a binary tree that can quickly broadcast to the whole network. On this basis, Elias Rohrer et al. [27] further proposed a broadcast protocol called Kadcast, which can reliably broadcast to the whole network. For more information on the Bitcoin and Ethereum network layer, we refer the reader to [28].

Additionally, there has been a lot of work to continuously improve the performance of the broadcast protocol in blockchain networks. In [17], the method of constructing a broadcast tree based on the forwarding hop count could greatly save broadcast latency. In [29], an improved Gossip protocol, GossipSub, was designed. This method divides the network into multiple meshes, and then broadcasts successively. Through detailed verification, GossipSub was found to resist a variety of attacks and have high security. In [30], the authors designed a broadcast protocol named Graphene. This is a novel combination of the Bloom filter and the Invertible Bloom Lookup Table (IBLT) that has improved reliability and efficiency. Graphene is a complex protocol, and, fortunately, M. A. Imtiaz et al. [31] tested it in a real scenario and proved that it has superior communication performance.

After reviewing the broadcast protocol in blockchain networks, we found that these studies mainly focused on broadcast latency, security, and so on. Almost no research has focused on energy consumption, an important indicator. Therefore, in order to make the blockchain technology better for enabling 6G communications and meeting the ELPC needs as soon as possible, we chose to improve the energy consumption of the broadcast protocol in blockchain networks.

3. LECast Protocol for UAV Blockchain Networks

In this section, we show the LECast protocol in detail, including the SPB Tree construction, the naming rule EHC for the tree's nodes, and the multi-channel transmission method MTSD. Based on these approaches, the workflow of the LECast can be summarized as in Figure 3.

3.1. SPB Tree

In the broadcast tree, drones are seen as nodes. To construct a broadcast tree with targeted energy cost, we first need to study the energy-consumption model between two nodes. According to [32], this model can be described as in Figure 4, and the energy consumption can be presented as in (1).

$$E = E_{tx} + E_{rx} \tag{1}$$

where E_{tx} is the energy cost of the transmitting node and E_{rx} is the energy consumption by the receiving node. Further, we have

$$E_{tx} = E_{elec} \times k + E_{amp} \times k \tag{2}$$

$$E_{rx} = E_{elec} \times k \tag{3}$$

where k is the data length and E_{elec} is the energy required to run basic functions in the transmitting node or the receiving node. E_{amp} represents the energy needed to amplify the signal, often related to distance.

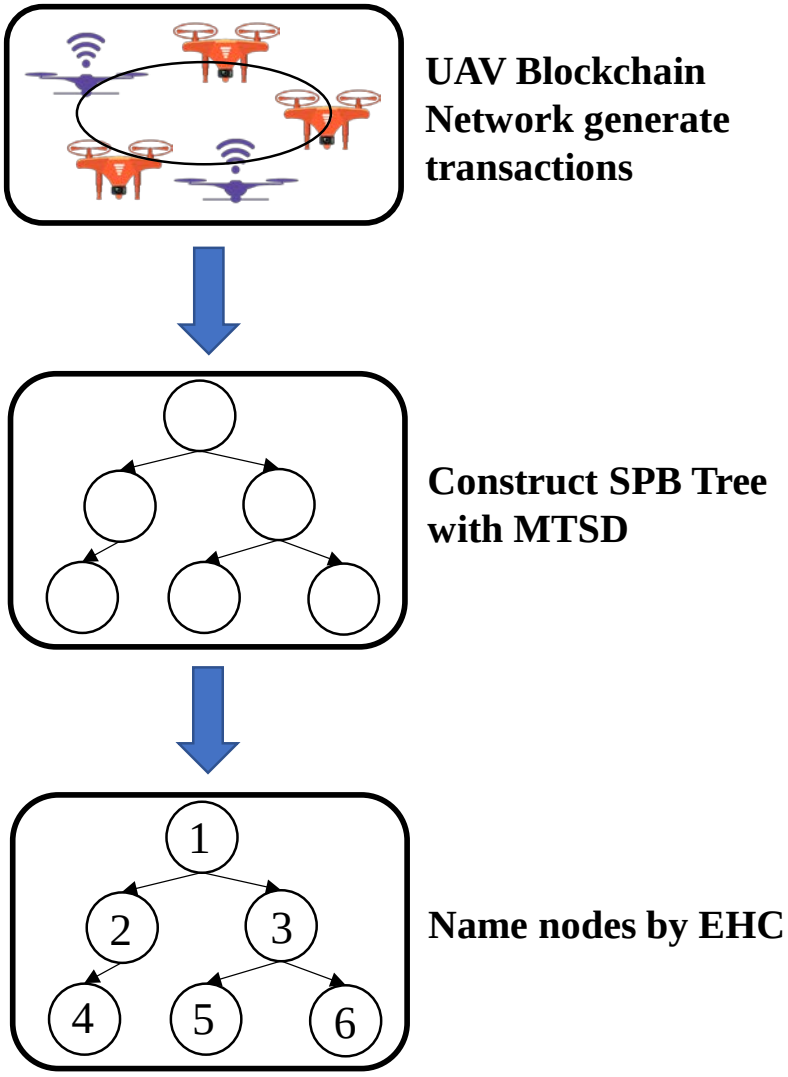


Figure 3. The Workflow of LECast.

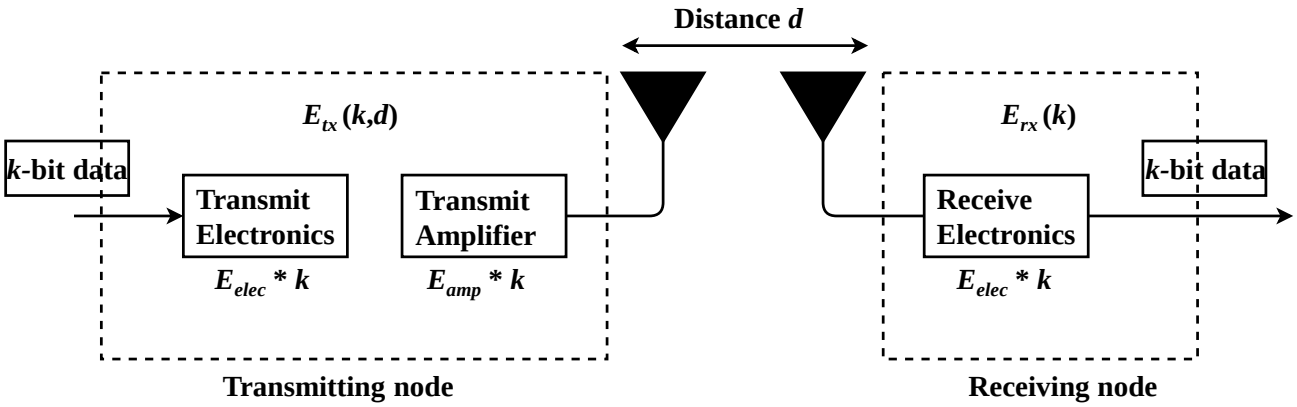


Figure 4. Energy consumption model of communication between two nodes.

According to [33,34], we use a distance threshold d_0. When the distance d between two nodes is less than d_0, E_{amp} uses the free space (FS)-propagation model with the parameter η_{fs}. When d is greater than d_0, E_{amp} uses a two-ray (TR) ground-reflection model with parameter η_{tr}. Then, E_{amp} can be presented as

$$E_{amp} = \begin{cases} \eta_{fs} \times d^2, & d < d_0 \\ \eta_{tr} \times d^4, & d \geq d_0 \end{cases} \tag{4}$$

where η_{fs} = 10 pJ/bit/m^2 and η_{tr} = 0.0013 pJ/bit/m^2; thus, d_0 = 87.7 m. It can be easily found that E_{elec} is a monotonically increasing function about d.

Combining (1), (2), (3) and (4), when a transmitting node sends a message and is received by the receiving node, the energy consumption of this process is

$$E = \begin{cases} k \times \left(2E_{elec} + \eta_{fs} \times d^2\right), & d < d_0 \\ k \times \left(2E_{elec} + \eta_{tr} \times d^4\right), & d \geq d_0 \end{cases} \tag{5}$$

According to the conclusion of (5), in order to reduce the energy consumption of broadcasting in a blockchain network, on one hand, the amount of data transmission should be reduced to avoid excess E_{elec}; on the other hand, the transmission distance should be reduced to minimize E_{amp}.

As a result, we can change the traditional flooding in Bitcoin into a broadcast tree, reducing the excess E_{elec} by unnecessary transmissions, as shown in Figure 5. In addition, although the rule of (5) targets two nodes, for a multi-node broadcast tree, we only need to minimize the sum of the communication distances between all nodes to minimize E_{am}. Therefore, according to the GPS positioning of each drone, we used the shortest path algorithm [35] to construct an SPB Tree with the minimum distance.

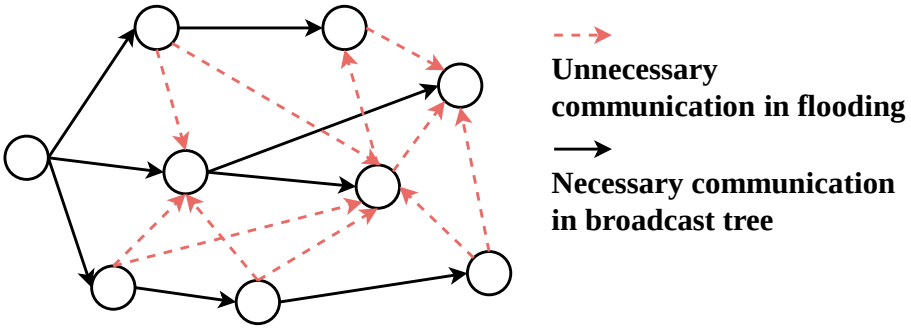


Figure 5. Example of Communication links in Broadcast Tree and Flooding.

3.2. EHC Scheme

After building the SPB Tree using the shortest path algorithm, we also need to name the nodes on the tree. This is to make it easier for the transmitting node to address the receiving node. The premise of addressing is to provide each node with a unique name.

However, traditional naming methods usually use binary; in this way, only the nodes on the binary tree can be named. Obviously, our SPB Tree does not have to be a binary tree—it may be a trinomial tree or even a multiway tree. Recently, regarding the multiway tree node-naming method, on the one hand, the multiway tree can be transformed into a binary tree, which can be named in binary. Then, the named binary tree is transformed back into the original multiway tree. On the other hand, multiway-tree nodes can also be named through number field operations, but this method often has high computational complexity, such as Galois Fields [36].

In this part, we transform the naming problem into the information source coding problem. Taking the multiway tree shown in Figure 6 as an example, we used the EHC to name the nodes on it, and the results are shown in Table 1.

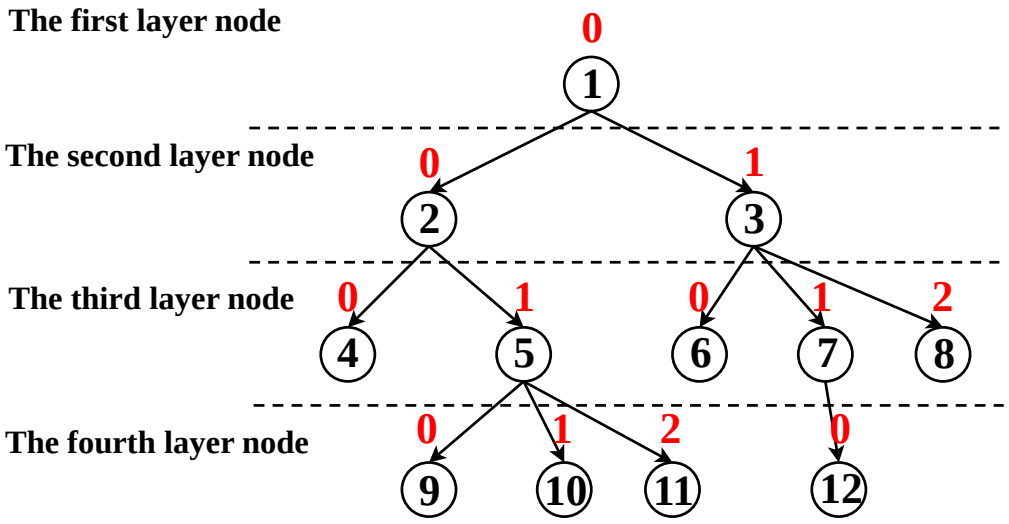


Figure 6. Example of a Multiway Tree.

Table 1. EHC Naming for the Multiway Tree.

Node Number	Initial Coding	Wrong Extension	Correct Extension
1	0	0	000
2	00	00	0000000
3	01	01	000001
4	000	000	000000000
5	001	001	000000001
6	010	010	000001000
7	011	011	000001001
8	012	**0110**	000001010
9	0010	0010	000000001000
10	0011	0011	000000001001
11	0012	00110	000000001010
12	0110	**0110**	000001001000

EHC is an improvement of Huffman coding in [37]. Based on Table 1, we summarize the naming process of EHC as follows:

(1) *Initial coding:* First, each node is preliminarily encoded according to the Huffman code. However, the traditional Huffman code is binary, thus, we should make an improvement on the Huffman code. That is, the ary of the Huffman code is the same as the number of ways of the broadcast tree (n-ary corresponds to n-way tree);

(2) *Extension:* Then, as computer systems can only recognize and manipulate binary numbers, we need to convert the results of the first step into binary. Here, we chose the extension method to convert *these* numbers. It should be noted that each digit needs to be expanded. If we only expand the digits that are not binary, it will cause their names to be duplicated with other nodes, such as the bold digits in Table 1 (node 8 in the third layer and node 12 in the fourth layer); Meanwhile, the specific extension method is to expand each digit into a binary number with n bits, where n is the same as the n in Step 1, such as the "Correct extension" results in Table 1. These results are the final naming for nodes on the n-way tree;

(3) *Addressing:* The *transmitting* node splits the naming result into a group with n-bits and reads it in turn while addressing the receiving node, as shown in Table 2.

In general, the EHC naming rule for multiway trees has the following two advantages:

- Two or more nodes do not have the same name;
- Nodes are very simple to name and read.

Table 2. Reading Method of the EHC Naming Results.

Node Number	Naming Results	Reading Method
1	000	(000)
2	000000	(000) (000)
3	000001	(000) (001)
4	000000000	(000) (000) (000)
5	000000001	(000) (000) (001)
6	000001000	(000) (001) (000)
7	000001001	(000) (001) (001)
8	000001010	(000) (001) (010)
9	000000001000	(000) (000) (001) (000)
10	000000001001	(000) (000) (001) (001)
11	000000001010	(000) (000) (001) (010)
12	000001001000	(000) (001) (001) (000)

3.3. MTSD Method

The SPB Tree can reduce energy consumption to the greatest extent, and it has advantages common to broadcast trees, such as a fast broadcast speed and lower data redundancy. However, it cannot avoid the disadvantage of low reliability because the tree structure has the problem that a node or channel failure will cause subsequent nodes to fail to receive transaction or block data.

Moreover, according to the theory of (6) from [38,39], transmission latency has an opposite relationship with reliability; that is, low latency and ultra-reliability cannot be realized in the same transmission channel. Therefore, we consider the multi-channel transmission method to ensure low latency and improve the reliability of the SPB Tree.

$$p_f = f_Q\left(\frac{NTBC - NTBR + \frac{\log NTB}{2}}{(\log e)\sqrt{NTB}}\right) \tag{6}$$

In (6), p_f is the probability of a channel transmission failure, and the larger this value is, the worse the reliability will be. f_Q is the Q function and T is the latency. N represents the number of subcarriers and B represents the bandwidth. R and C represent the transmission speed and channel capacity, respectively.

Multi-channel transmission can simultaneously transmit data over multiple channels in parallel to improve the successful transmission rate and avoid an unreachable problem due to noise or interference in the case of a single channel. However, the risk is that repeated transmission affords the attacker the opportunity to listen to the data many times, which increases the probability of successful interception and reduces security [40,41]. After obtaining the transmitted data in the blockchain network, the attacker may carry out man-in-the-middle attacks and identity forgery attacks. Therefore, it is necessary to improve reliability and ensure security at the same time.

We introduce the concept of MTSD, which splits the data into s sub-data and transmits them multiple times in s channels. In this case, s is also called the data redundancy parameter. In this way, reliability can be ensured through a multipath approach, and splitting data also increases the attack cost and difficulty, improving security. This is because the attacker must obtain all the sub-data to recover the full transmission data.

Furthermore, in the same channel, the transmission latency is only related to the data length; thus, we transmit each sub-datum s times to keep the total length of the data unchanged so that the transmission latency and throughput will not be affected. In addition, after receiving sub-data from s channels, the receiving node deletes redundant data and then splices and verifies the remaining data according to the splitting scheme, which cannot increase the validation latency of redundant data.

Figure 7 shows the models of single-channel transmission (ST), conventional multi-channel transmission (CMT), and MTSD. Readers can intuitively compare the differences between the three through this figure.

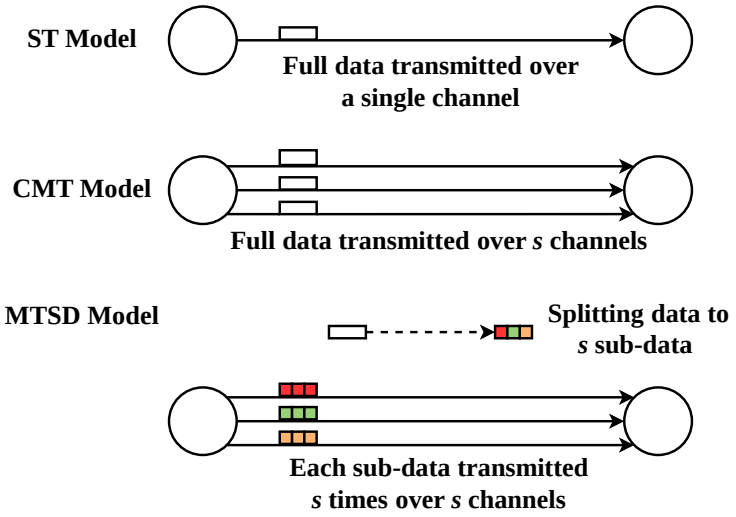


Figure 7. Models of ST, CMT, and MTSD.

4. Performance Analysis

In this section, we successively analyze the security, reliability, broadcast coverage, broadcast latency, throughput, and energy consumption of LECast for blockchain networks. These performances basically cover all aspects of the broadcast protocol.

4.1. Security

In this part, we compare the security between MTSD and CMT. We set the probability of an attacker successfully obtaining data from a channel to be p_A. Then, for CMT, when full data are sent over s channels in parallel, the attacker obtains data from one channel, which is regarded as a successful attack, and the probability is

$$P_{S\_CMT} = \sum_{i=1}^{s} C_s^i {p_A}^i (1 - p_A)^{(s-i)} \tag{7}$$

Additionally, for MTSD, the attacker needs to capture all the sub-data from s channels to recover the original data, and the probability can be expressed as

$$P_{S\_MTSD} = \left(\sum_{i=1}^{s} C_s^i {p_A}^i (1 - p_A)^{(s-i)} \right)^s \tag{8}$$

It can easily be found that (7) is a number less than 1, and (8) is (7) to the s power, so (8) is obviously smaller than (7). This indicates that MTSD is more secure than CMT, which is proven in Section 5.

4.2. Reliability

In this part, we further compare the reliability between MTSD and ST. For ST, the probability of the receiving node successfully receiving a datum from the sending node is $1 - p_f$; thus, the reliability of ST is also $1 - p_f$, which can be expressed as

$$P_{R\_ST} = 1 - p_f \tag{9}$$

For MTSD, the receiving node only needs to obtain a segment of sub-data arbitrarily in each channel to recover the original data based on the data-splitting scheme, so its reliability can be expressed as

$$P_{R\_MTSD} = \left(\sum_{i=1}^{s} C_s^i (1 - p_f)^i {p_f}^{(s-i)} \right)^s \tag{10}$$

4.3. Broadcast Coverage

In this part, we measure the broadcast coverage of the SPB Tree by reliability. When the first-layer node broadcasts data to other nodes in the broadcast tree, the probability that the second-layer nodes successfully receive data is the reliability (10). Then, the reliability of the third-layer nodes receiving data is the square of (10). Further, for nodes in the *zth* layer, the reliability is

$$P_{R\_Z} = {P_{R\_MTSD}}^{z-1} \tag{11}$$

If n_i represents the number of nodes in the *ith* layer (n_1 = 1), the number of nodes that successfully receive data in the SPB Tree is

$$n_{re} = \sum_{i=1}^{z} {P_{R\_MTSD}}^{i-1} n_i \tag{12}$$

The total number of nodes is

$$n_{total} = \sum_{i=1}^{z} n_i \tag{13}$$

Therefore, the broadcast coverage is

$$m = \frac{\sum_{i=1}^{z} {P_{R\_MTSD}}^{i-1} n_i}{\sum_{i=1}^{z} n_i} \tag{14}$$

4.4. Broadcast Latency

In a computer network, the latency is divided into transmission latency, propagation latency, processing latency, and queuing latency. In a blockchain network, the processing delay is usually the validation latency of the node to verify transaction or block data. In addition, there is no queuing latency in Bitcoin's blockchain network. The reason is that, in Bitcoin, a block needs to be broadcast every 10 min, at which time the last block has already been broadcast; for transaction data, each block contains 500 transactions; thus, each transaction needs to be broadcast within 1.2 s on average, which is also long enough to broadcast.

Therefore, the broadcast latency between two nodes in the blockchain network can be expressed as

$$T = T_t + T_p + T_v \tag{15}$$

where T_t, T_p, and T_v represent the transmission latency, propagation latency, and validation latency, respectively. These three latencies can be expressed as

$$T_t = \frac{k}{R} \tag{16}$$

$$T_p = \frac{L}{c} \tag{17}$$

$$T_v = \frac{kU}{f} \tag{18}$$

where L represents the distance between two nodes and $c = 3 \times 10^8$ m/s is the light speed. U represents the number of CPU cycles required per bit and f is the CPU frequency of one node.

Therefore, (15) can be further expressed as

$$T = \frac{k}{R} + \frac{L}{c} + \frac{kU}{f} \tag{19}$$

If the transmission speed R of each channel, and the processing capacity f of each node are fixed values, then the broadcast latency of the whole blockchain network depends on the longest routing path. The distance of this longest path is denoted as d_L and the number of connected nodes is z; then, the total latency can be denoted as

$$T_{total} = \sum_{i=0}^{z-1} T_i = (z-1)\left(\frac{k}{R} + \frac{kU}{f}\right) + \frac{d_L}{c} \tag{20}$$

4.5. Throughput

In blockchain networks, throughput refers to the number of transactions or blocks broadcast per second.

However, in most blockchain systems, the throughput is fixed, such as 7 transactions per second (TPS) for Bitcoin and around 20–30 TPS for Ethereum [42]. Therefore, in order to study the throughput of the SPB Tree, we do not consider the actual situation of the blockchain system generating transactions and blocks. After the first node validates the data and sends it to the next node, it can process the next data. As a consequence, the maximum throughput can be expressed as the reciprocal of the validation latency, as shown in (21).

$$TPS = \frac{1}{T_v} = \frac{f}{kU} \tag{21}$$

4.6. Energy Consumption

The overall energy consumption of blockchain network broadcasting is related to the distance sum in the SPB Tree, the total number of nodes, and the system redundancy s, which can be expressed as

$$E_{total} = \Big((n_{total} - 1)(2E_{elec} \times k) + E_{amp\_total} \times k\Big)s \tag{22}$$

The above function can be understood as the energy required by nodes to send and receive in $(n_{total} - 1)$ transmissions, plus the energy required by all amplifiers in the network.

Further, E_{elec} is equal to power multiplied by time, and power equals current I multiplied by voltage U; thus, it can be represented as

$$E_{elec} = IUT_v \tag{23}$$

Meanwhile, E_{amp_total} can be expressed as the sum of E_{amp_fs} the energy consumption in the FS model, and E_{amp_tr} the energy consumption in the TR model, namely

$$E_{amp\_total} = E_{amp\_fs} + E_{amp\_tr} \tag{24}$$

$$E_{amp\_fs} = \eta_{fs} \times \sum_{j=1}^{x} {d_j}^2, \;\; d_j < d_0 \tag{25}$$

$$E_{amp\_tr} = \eta_{tr} \times \sum_{k=1}^{y} {d_k}^4, \;\; d_k \geq d_0 \tag{26}$$

$$x + y = n_{total} - 1 \tag{27}$$

5. Performance Simulations

In this section, the performance analyzed in Section 4 is simulated. In terms of broadcast coverage, broadcast latency, and energy consumption, Kadcast in [26] was selected as the comparison protocol. Kadcast is a low-latency-oriented broadcast protocol in blockchain networks, and has a similar setting to LECast.

For **security**, first, we took the probability of channel transmission failure caused by an attack as the abscissa and compared the successful attack rates of MTSD and CMT with different s values. The simulation is shown in Figure 8. We found that MTSD consistently had a lower successful attack rate, regardless of the value of the channel failure probability. Meanwhile, with the increase in the s value, the security gap between the two further widened; namely, when the s value is large, MTSD showed higher security. Second, we took the value of s as the abscissa to compare the successful attack rate of MTSD and CMT with different p_A values; the simulation is shown in Figure 9. We found that the successful attack rate of MTSD was always lower than that of CMT, indicating that MTSD had higher security. In addition, when s increased, the successful attack rate of CMT also increased, while the change in MTSD was not obvious and was in a fluctuating state.

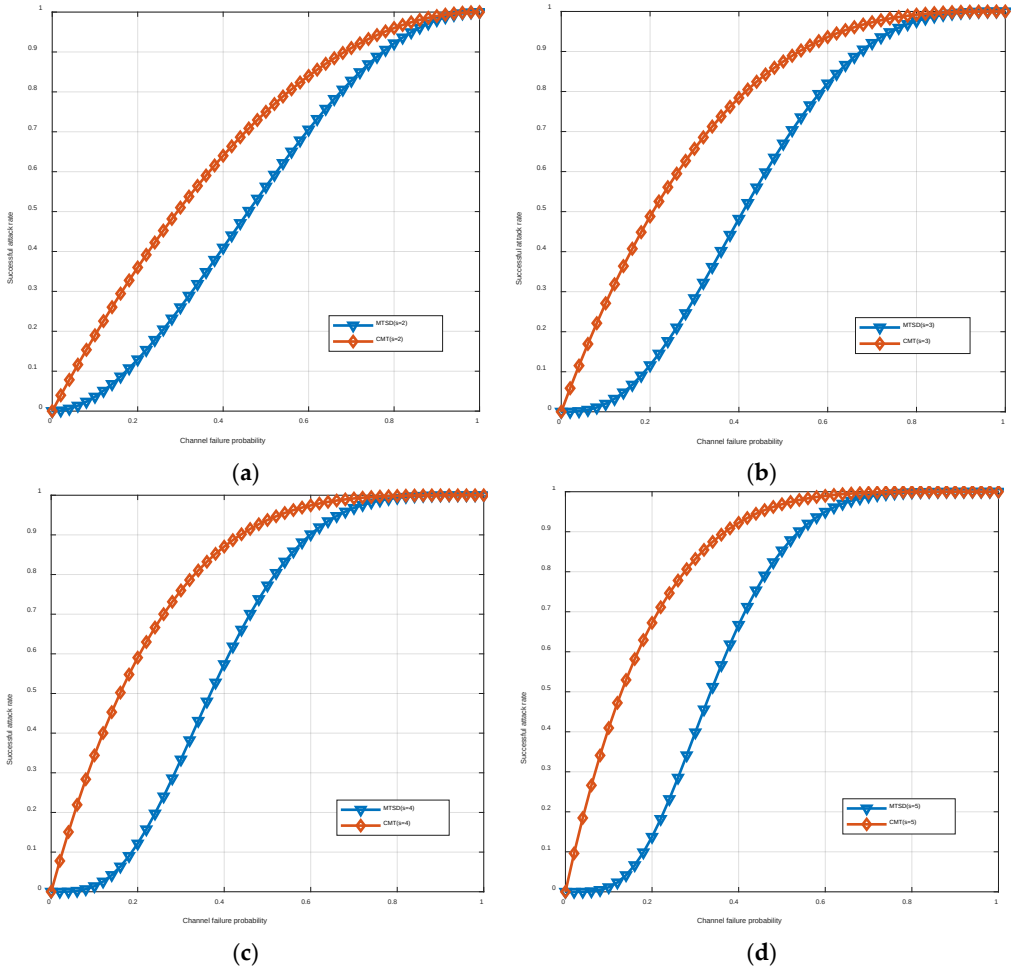


Figure 8. (**a**) Security comparison between MTSD and CMT with s = 2; (**b**) Security comparison between MTSD and CMT with s = 3; (**c**) Security comparison between MTSD and CMT with s = 4; (**d**) Security comparison between MTSD and CMT with s = 5.

For **reliability**, first, we took the unreliability of a channel, namely p_f, as the abscissa and compared the reliability of MTSD and ST with different s values; the simulation is shown in Figure 10a. We found that MTSD was more reliable than ST for lower p_f values, regardless of the value of s. ST was more reliable only when p_f was close to 1. However, in real communication, if there is no interference from external attackers, the value of p_f should be very low, just larger than 0. Second, we took the value of s as the abscissa to compare the reliability of MTSD with different p_f values; the simulation is shown in Figure 10b. According to this result, the reliability increased with an increase in the s value, indicating that MTSD could enhance reliability by increasing the system redundancy.

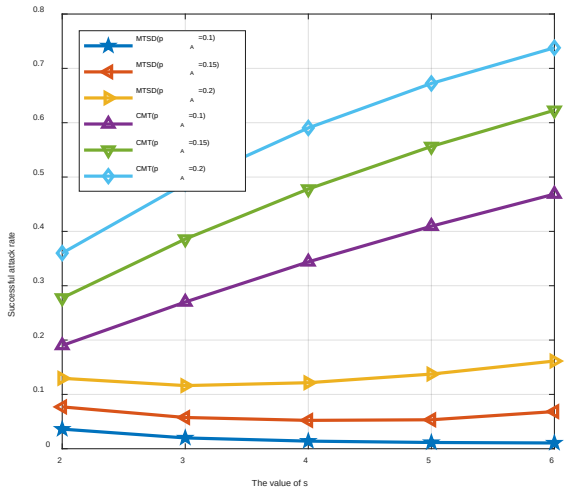
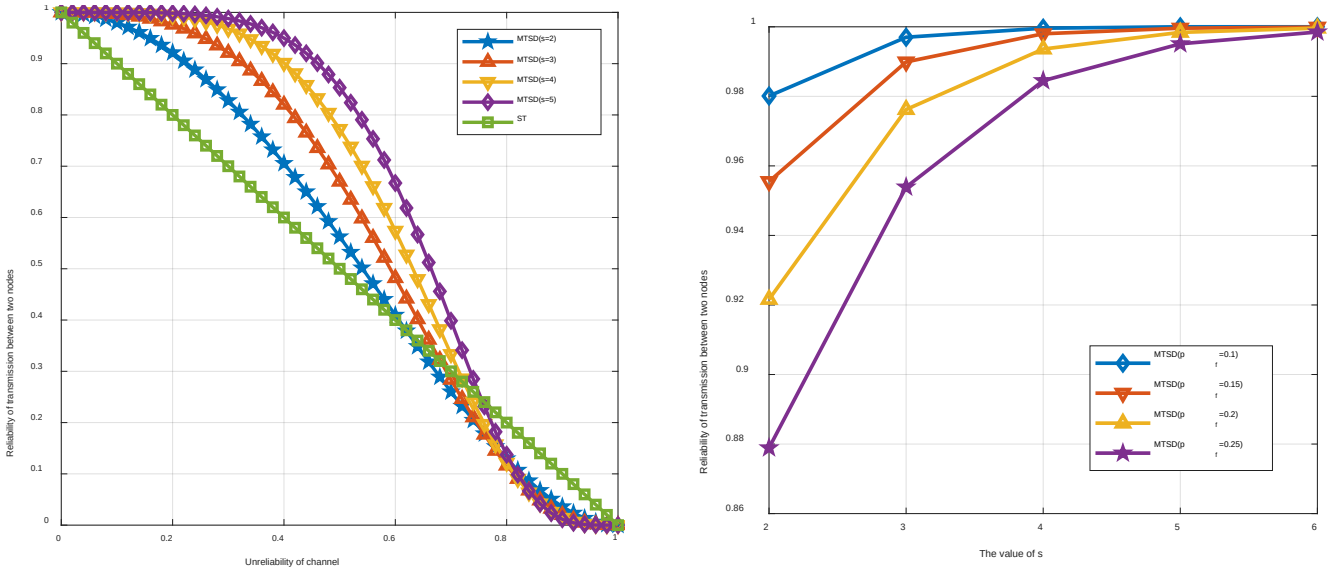


Figure 9. Security comparison between MTSD and CMT with different p_A values.

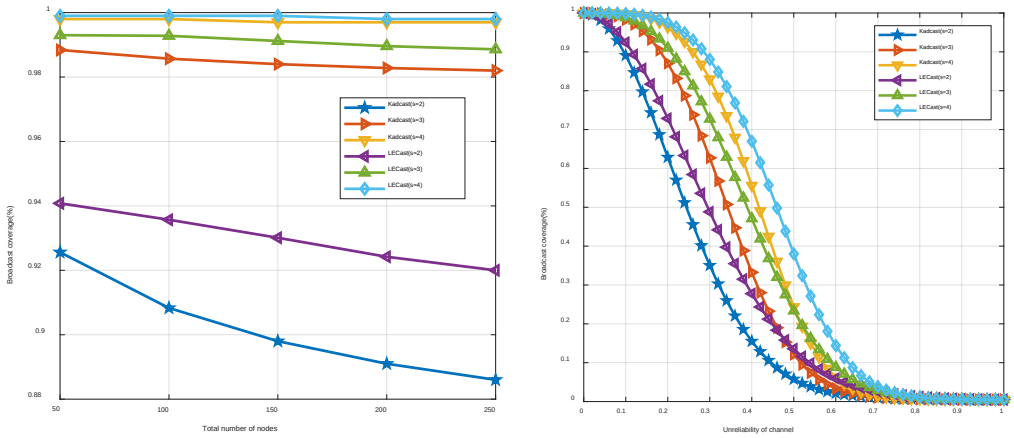


(a) **(b)**

Figure 10. (**a**) Reliability comparison between MTSD and ST with different s values; (**b**) reliability comparison between MTSD and ST with different s values.

For **broadcast coverage**, we set an area with a radius of 1 km in OMNet++. The drones in the blockchain obeyed the two-dimensional Poisson distribution with respect to the center of the circle in this region. Then, we built the SPB Tree with the center of the circle as the root node, which is the first-layer node in Figure 6. Further, in this environment, we compared the broadcast coverage between LECast and Kadcast. First, we fixed $p_f = 0.1$ and took the total number of nodes n_{total} as the abscissa to compare the difference between them with different values of s, which is shown in Figure 11a. We found that, as the number of nodes increased, the coverage decreased. The reason is the higher the number of nodes is, the deeper the layer depth of the broadcast tree will be. According to (11), nodes in deeper layers have a lower probability to receive data. Meanwhile, the broadcast coverage of LECast was always higher than that of Kadcast with the same value of s. This indicates that Poisson distribution caused the SPB Tree to connect more nodes in each layer and reduced the number of layers in the SPB Tree. Additionally, an increase in s could also improve the broadcast coverage, which is consistent with reliability. Second, we fixed $n_{total} = 200$ and

took p_f as the abscissa to compare the coverage of the above two broadcast protocols, as shown in Figure 11b. The simulation also showed that LECast had significant advantages.



(a) **(b)**

Figure 11. (**a**) Broadcast coverage comparison between LECast and Kadcast with $p_f = 0.1$; (**b**) broadcast coverage comparison between LECast and Kadcast with $n_{total} = 200$.

For **broadcast latency**, based on the simulation of broadcast coverage, we set the parameters as shown in Table 3 to compare the broadcast latency of Kadcast and LECast. When the transaction or block data were broadcast to the whole network by the root node, the latency of the two were determined, as shown in Figure 12. The simulation results show that our proposed LECast protocol had more advantages in terms of broadcast delay. The reason is that the SPB Tree had fewer layers, saving the data transmission and validation times, facilitating fast data broadcast.

Table 3. Simulation parameters.

Parameters	Values
R	100 Mbps
f	1000 Hz
U	1/64
k	1 MB

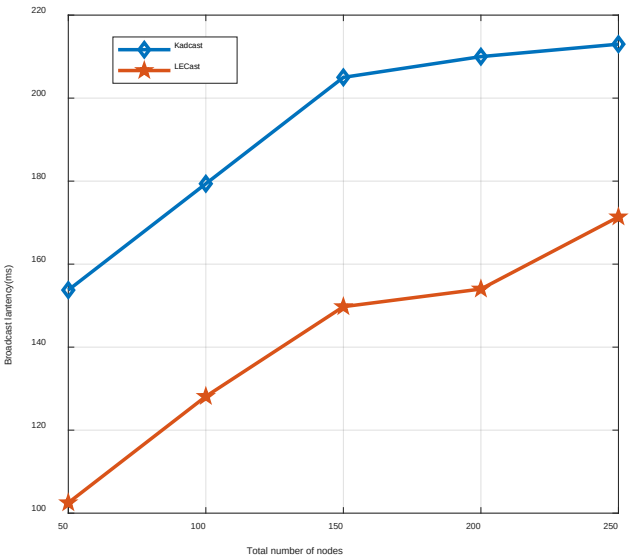


Figure 12. Broadcast latency comparison between LECast and Kadcast.

For **throughput**, in the blockchain system, a block is generated every 10 min, and each block contains 500 transactions. In other words, each transaction is evaluated within 1.2 s. This throughput obviously does not reach the limit of the blockchain network; thus, this simulation was conducted to explore the throughput upper limit of LECast. We simulated the throughput of LECast for different transaction-generation rates, and the result is shown in Figure 13. We set the computing capacities of nodes to be 1000 Hz and 2000 Hz. When f = 1000 Hz, the upper limit of throughput was 64; when f = 2000 Hz, the upper limit of throughput was 128. This result is consistent with the theoretical analysis in Section 4. The transmission performance in blockchain networks did not restrict the throughput, but the node's computing power had an impact on it. As a consequence, we could consider raising the node's computing capacity to improve throughput in real deployment.

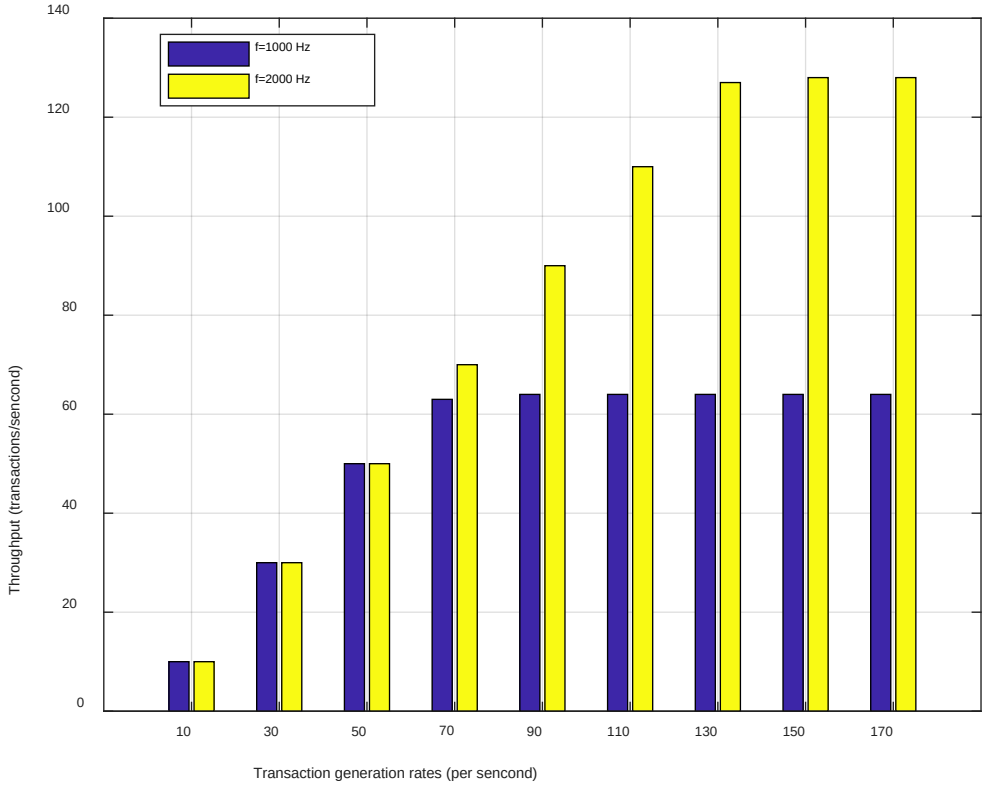
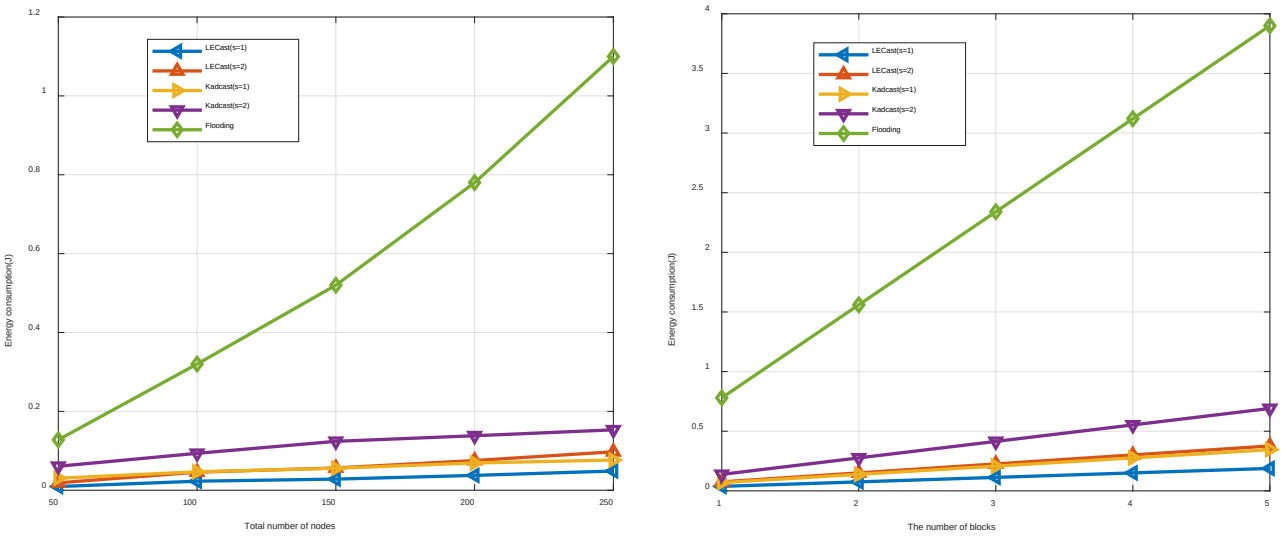


Figure 13. Throughput of LECast with different f values.

For **energy consumption**, according to (23), we set $E_{elec} = 5 \times 10^{-8}$ J. First, when the blockchain network broadcasted one block, we simulated the energy consumption of LECast, Kadcast, and Flooding and observed their relationship with the total number of nodes $n_{total.}$ The result is shown in Figure 14a. Here, the Flooding connectivity was set to 5; that is, each node could connect to five neighbor nodes. The simulation results showed that LECast had a significant advantage over Flooding in energy consumption, being 1/20 to 1/10 of that of Flooding. It was also better than Kadcast, being 1/3 to 1/2 that of Kadcast. This progress proves that LECast basically conformed to the perspective of 6G communications in [2,3]. Moreover, as the connectivity of Flooding increased, the superiority in the energy consumption of LECast could be further highlighted. Second, we fixed $n_{total} = 200$ to compare the impact of the block generation number on energy consumption, as shown in Figure 14b. This simulation result revealed the relationship between the number of generation blocks and the energy consumption, and also demonstrated the advantages of LECast in energy consumption.



(a) **(b)**

Figure 14. (**a**) Energy consumption with different n_{total} values; (**b**) energy consumption with different numbers of blocks.

6. Conclusions

In our paper, we designed a broadcast protocol called LECast for UAV blockchain networks. This protocol was designed to reduce the energy consumption by blockchain network broadcasting in order to better fit UAV networks and meet the ELPC objectives of 6G communication. First, LECast modeled the energy consumption of the communication between two drones and identified the factors affecting the energy consumption. Then, the SPB Tree with the lowest energy consumption was constructed. Meanwhile, LECast designed a simple naming rule, the EHC scheme, for nodes on the SPB Tree. Additionally, in order to solve the reliability and security problems of the broadcast tree, LECast introduced multipath transmission and proposed the MTSD method.

Through rigorous theoretical analysis and simulation, it was proven that LECast had advantages in many aspects. MTSD was superior to CMT and ST in terms of security and reliability, respectively. SPB Tree was superior to Kadcast in broadcast coverage and broadcast latency, and also had high throughput. Most importantly, LECast showed significant advantages in energy consumption, which was superior to those of Flooding and Kadcast, and basically reached the ELPC target for 6G communications.

In general, LECast not only fulfills our original intention of reducing the energy consumed by blockchain network broadcasting, but also has other advantages that make it well-suited for UAV networks. However, the proper operation of LECast depends on the geographic location of each drone to build the SPB Tree. When a drone is underground or indoors, it cannot receive GPS signals, and it is difficult to locate accurately. Therefore, future research is needed to design blockchain-based energy-efficient broadcast protocols without the geographic location of drones. In addition, we will also improve energy consumption in other areas of the blockchain.

Author Contributions: Conceptualization, H.L. and S.L.; methodology, H.L. and S.L.; software, H.L.; validation, H.L., S.L. and S.X.; formal analysis, H.L.; investigation, H.L.; resources, H.L. and S.L.; data curation, H.L.; writing—original draft preparation, H.L. and S.L; writing—review and editing, S.X. and J.L; visualization, S.X.; supervision, S.X.; project administration, S.X. and J.L.; funding acquisition, S.X. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the National Key Research and Development Program of China under Grant 2019YFB1802800; in part by the Natural Science Foundation of Sichuan Province

under Grant 2022NSFSC0913; and in part by the PCL Future Greater-Bay Area Network Facilities for Large-Scale Experiments and Applications under Grant PCL2018KP001.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Niu, Z.; Zhou, S.; Sun, Y. Green communication and networking for Carbon-peaking and Carbon-neutrality: Challenges and solutions. *J. Commun.* **2022**, *43*, 1–14.
2. Nguyen, V.-L.; Lin, P.-C.; Cheng, B.-C.; Hwang, R.H.; Lin, Y.D. Security and Privacy for 6G: A Survey on Prospective Technologies and Challenges. *IEEE Commun. Surv. Tutor.* **2021**, *23*, 2384–2428. [CrossRef]
3. Liu, G.; Huang, Y.; Li, N.; Dong, J.; Jin, J.; Wang, Q.; Li, N. Vision, requirements and network architecture of 6G mobile network beyond 2030. *China Commun.* **2020**, *17*, 92–104. [CrossRef]
4. Xu, X.; Sun, G.; Yu, H. An Efficient Blockchain PBFT Consensus Protocol in Energy Constrained IoT Applications. In Proceedings of the 2021 International Conference on UK-China Emerging Technologies (UCET), Chengdu, China, 4–6 November 2021; pp. 152–157.
5. Luo, L.; Feng, J.; Yu, H.; Sun, G. Blockchain-Enabled Two-Way Auction Mechanism for Electricity Trading in Internet of Electric Vehicles. *IEEE Internet Things J.* **2022**, *9*, 8105–8118. [CrossRef]
6. Ping, J.; Yan, Z.; Chen, S. A privacy-preserving blockchain-based method to optimize energy trading. *IEEE Trans. Smart Grid* 2022, *early access*. [CrossRef]
7. Chen, Y.; Luo, H.; Bian, Q. A Privacy Protection Method Based on Key Encapsulation Mechanism in Medical Blockchain. In Proceedings of the IEEE 21st International Conference on Communication Technology (ICCT), Tianjin, China, 13–16 October 2021; pp. 295–300.
8. Porambage, P.P.; Gür, G.; Osorio, D.P.M.; Liyanage, M.; Gurtov, A.; Ylianttila, M. The Roadmap to 6G Security and Privacy. *IEEE Open J. Commun. Soc.* **2021**, *2*, 1094–1122. [CrossRef]
9. Jiang, W.; Han, B.; Habibi, M.A.; Schotten, H.D. The Road Towards 6G: A Comprehensive Survey. *IEEE Open J. Commun. Soc.* **2021**, *2*, 334–366. [CrossRef]
10. Wang, M.; Zhu, T.; Zhang, T.; Zhang, J.; Yu, S.; Zhou, W. Security and privacy in 6G networks: New areas and new challenges. *Digit. Commun. Netw.* **2020**, *6*, 281–291. [CrossRef]
11. Li, M.; Yu, F.R.; Si, P.; Zhang, Y.; Qian, Y. Intelligent Resource Optimization for Blockchain-Enabled IoT in 6G via Collective Reinforcement Learning. *IEEE Netw.* **2022**, *36*, 175–182. [CrossRef]
12. Xu, H.; Klaine, P.V.; Onireti, O.; Cao, B.; Imran, M.; Zhang, L. Blockchain-enabled resource management and sharing for 6G communications. *Digital Commun. Netw.* **2020**, *6*, 261–269. [CrossRef]
13. Nair, R.; Gupta, S.; Soni, M.; Shukla, P.K.; Dhiman, G. An approach to minimize the energy consumption during blockchain transaction. *Mater. Today Proc.* **2020**, 1–6. [CrossRef]
14. Monrat, A.A.; Schelén, O.; Andersson, K. A Survey of Blockchain From the Perspectives of Applications, Challenges, and Opportunities. *IEEE Access* **2019**, *7*, 117134–117151. [CrossRef]
15. Riyal, A.; Kumar, G.; Sharma, D.K.; Gupta, K.D.; Srivastava, G. Blockchain Tree Powered Green Communication for Efficient and Sustainable Connected Autonomous Vehicles. *IEEE Trans. Green Commun. Netw.* **2022**, *6*, 1428–1437. [CrossRef]
16. Qian, K.; Liu, Y.; Shu, C.; Sun, Y.; Wang, K. Fine-grained Benchmarking and Targeted Optimization: Enabling Green IoT-oriented Blockchain in the 6G Era. *IEEE Trans. Green Commun. Netw.* 2022, *early access*. [CrossRef]
17. Goyal, H.; Kausik, H.M.; Saha, S. ReLI: Real-Time Lightweight Byzantine Consensus in Low-Power IoT-Systems. In Proceedings of the 2022 18th International Conference on Network and Service Management (CNSM), Thessaloniki, Greece, 31 October–4 November 2022; pp. 275–281.
18. Alsamhi, S.H.; Shvetsov, A.V.; Shvetsova, S.V.; Hawbani, A.; Guizan, M.; Alhartomi, M.A.; Ma, O. Blockchain-Empowered Security and Energy Efficiency of Drone Swarm Consensus for Environment Exploration. *IEEE Trans. Green Commun. Netw.* 2022, *early access*. [CrossRef]
19. Shahzad, I.; Maqbool, A.; Rana, T.; Mirza, A.; Khan, W.Z.; Kim, S.W.; Zikria, Y.B.; Din, S. Blockchain-based green big data visualization: BGbV. *Complex Intell. Syst.* **2021**, *8*, 3707–3718. [CrossRef]
20. Li, X.; Luo, H.; Duan, J. Security Analysis of Sharding in Blockchain with PBFT Consensus. In Proceedings of the 4th International Conference on Blockchain Technology (ICBCT'22), Shanghai, China, 25–27 March 2022; pp. 9–14.
21. Hong, Z.; Guo, S.; Li, P.; Chen, W. Pyramid: A Layered Sharding Blockchain System. In Proceedings of the 2021 IEEE Conference on Computer Communications, Vancouver, BC, Canada, 10–13 May 2021; pp. 1–10.
22. Li, W.; Feng, C.; Zhang, L.; Xu, H.; Cao, B.; Imran, M.A. A Scalable Multi-Layer PBFT Consensus for Blockchain. *IEEE Trans. Parallel Distrib. Syst.* **2021**, *32*, 1146–1160. [CrossRef]
23. Dinh, T.; Wang, J.; Chen, G.; Liu, R.; Ooi, B.C.; Tan, K.L. Blockbench: A framework for analyzing private blockchains. In Proceedings of the 2017 ACM International Conference on Management of Data (SIGMOD'17), Chicago, IL, USA, 14–19 May 2017; pp. 1085–1100.

24. Karp, R.; Schindelhauer, C.; Shenker, S.; Vocking, B. Randomized rumor spreading. In Proceedings of the 41st Annual Symposium on Foundations of Computer Science, Washington, DC, USA, 12–14 November 2000; pp. 565–574.
25. Chawathe, Y.; Ratnasamy, S.; Breslau, L.; Lanham, N.; Shenker, S. Making gnutella-like p2p systems scalable. In Proceedings of the 2003 ACM International Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications (SIGCOMM'03), Karlsruhe, Germany, 25–19 August 2003; pp. 407–418.
26. Maymounkov, P.; Mazires, D. Kademlia: A peer-to-peer information system based on the XOR metric. In Proceedings of the 1st International Workshop on Peer-to-Peer System (IPTPS'02), Cambridge, MA, USA, 7–8 March 2002; pp. 53–65.
27. Rohrer, E.; Tschorsch, F. Kadcast: A structured approach to broadcast in blockchain networks. In Proceedings of the 1st ACM Conference on Advances in Financial Technologies (AFT'19), Zurich, Switzerland, 21–23 October 2019; pp. 199–213.
28. Neudecker, T.; Hartenstein, H. Network Layer Aspects of Permissionless Blockchains. *IEEE Commun. Surv. Tutor.* **2019**, *21*, 838–857. [CrossRef]
29. Vyzovitis, D.; Napora, Y.; McCormick, D.; Dias, D.; Psaras, Y. GossipSub: Attack-resilient message propagation in the Filecoin and ETH2.0 networks. *arXiv* **2020**, arXiv:2007.02754.
30. Ozisik, A.P.; Andresen, G.; Levine, B.N.; Tapp, D.; Bissias, G.; Katkuri, S. Graphene: Efficient interactive set reconciliation applied to blockchain propagation. In Proceedings of the 2019 ACM Special Interest Group on Data Communication, Beijing, China, 19–23 August 2019; pp. 303–317.
31. Imtiaz, M.A.; Starobinski, D.; Trachtenberg, A. Empirical Comparison of Block Relay Protocols. *IEEE Trans. Netw. Serv. Manag.* 2022, *early access*. [CrossRef]
32. Bagula, A.; Abidoye, A.P.; Zodi, G.A.L. Service-aware clustering: An energy-efficient model for the internet-of-things. *Sensors* **2015**, *16*, 9. [CrossRef]
33. Heinzelman, W.R.; Chandrakasan, A.; Balakrishnan, H. Energy-efficient communication protocol for wireless microsensor networks. In Proceedings of the 33rd Annual Hawaii International Conference on System Sciences, Maui, HI, USA, 4–7 January 2000; pp. 1–10.
34. Heinzelman, W.B.; Chandrakasan, A.P.; Balakrishnan, H. An application-specific protocol architecture for wireless microsensor networks. *IEEE Trans. Wirel. Commun.* **2002**, *1*, 660–670. [CrossRef]
35. Bondy, J.A.; Murty, U.S.R. *Graph Theory with Applications*; Elsevier: Amsterdam, The Netherlands, 1976.
36. Westall, J.; Martin, J. *An Introduction to Galois Fields and Reed-Solomon Coding*; Clemson University: Clemson, SC, USA, 2020.
37. Huffman, D.A. A Method for the Construction of Minimum-Redundancy Codes. *Proc. IRE* **1952**, *40*, 1098–1101. [CrossRef]
38. Yang, X.; Luo, H.; Duan, J.; Yu, H. Ultra Reliable and Low Latency Authentication Scheme for Internet of Vehicles Based on Blockchain. In Proceedings of the 2022 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), Virtual, 2–5 May 2022; pp. 1–5.
39. Chang, B.; Zhang, L.; Li, L.; Zhao, G.; Chen, Z. Optimizing Resource Allocation in URLLC for Real-Time Wireless Control Systems. *IEEE Trans. Veh. Technol.* **2019**, *68*, 8916–8927. [CrossRef]
40. Luo, H.; Chen, W.; Chen, C.; Yang, Y.; Zhang, Y.; Wu, Y. Analysis of a Multichannel Lightweight Identity Authentication Method. In Proceedings of the IEEE 19th International Conference on Communication Technology (ICCT), Xi'an, China, 16–19 October 2019; pp. 1285–1290.
41. Deng, Y.; Luo, H. A Distributed Identity Authentication Scheme for Differential Fault Attack. In Proceedings of the IEEE 21st International Conference on Communication Technology (ICCT), Tianjin, China, 13–16 October 2021; pp. 731–735.
42. Cao, B.; Li, Y.; Zhang, L.; Zhang, L.; Mumtaz, S.; Zhou, Z.; Peng, M. When Internet of Things Meets Blockchain: Challenges in Distributed Consensus. *IEEE Netw.* **2019**, *33*, 133–139. [CrossRef]