TSINGHUA SCIENCE AND TECHNOLOGY
ISSN 1007-0214 05/09 pp817–831
DOI: 10.26599/TST.2021.9010065
Volume 27, Number 5, October 2022

Distributed Consensus for Blockchains in Internet-of-Things Networks

Li Yang, Yifei Zou*, Minghui Xu, Yicheng Xu, Dongxiao Yu, and Xiuzhen Cheng

Abstract: In recent years, due to the wide implementation of mobile agents, the Internet-of-Things (IoT) networks have been applied in several real-life scenarios, servicing applications in the areas of public safety, proximity-based services, and fog computing. Meanwhile, when more complex tasks are processed in IoT networks, demands on identity authentication, certifiable traceability, and privacy protection for services in IoT networks increase. Building a blockchain system in IoT networks can greatly satisfy such demands. However, the blockchain building in IoT brings about new challenges compared with that in the traditional full-blown Internet with reliable transmissions, especially in terms of achieving consensus on each block in complex wireless environments, which directly motivates our work. In this study, we fully considered the challenges of achieving a consensus in a blockchain system in IoT networks, including the negative impacts caused by contention and interference in wireless channel, and the lack of reliable transmissions and prior network organizations. By proposing a distributed consensus algorithm for blockchains on multi-hop IoT networks, we showed that it is possible to directly reach a consensus for blockchains in IoT networks, without relying on any additional network layers or protocols to provide reliable and ordered communications. In our theoretical analysis, we showed that our consensus algorithm is asymptotically optimal on time complexity and is energy saving. The extensive simulation results also validate our conclusions in the theoretical analysis.

Key words: distributed algorithm; consensus in blockchain; Internet-of-Things (IoT); SINR model

1 Introduction

The Internet-of-Things (IoT) network is an emerging Internet-based information architecture that can be configured to exchange information between agents and provide services for users. Recently, due to the extensive use of mobile agents, IoT networks and relative technologies have been recognized as an integral part of the real world by providing several applications in smart city systems[1]. In these applications, blockchain technology is envisioned as one of the most anticipated technologies to support the decentralization and security in IoT[2, 3] and has been gaining vast attentions from the broad academic research and industries in recent years.

Generally, the implementation of a blockchain system consists of the following fundamental technologies: Cryptographic hash, digital signature, and distributed consensus protocol[4]. The cryptographic hash technique is used to construct Merkle trees and engineer Proof-of-Work (PoW) puzzles. A digital signature ensures that the identity of an agent is unique while sending a message. The consistency of distributed ledgers is guaranteed under the distributed consensus protocol, with which each agent in the blockchain network complies to exchange messages and make decisions. As a result, all agents in the blockchain network can maintain the same transaction ledgers in common. From the above technologies, the distributed consensus protocol plays

• Li Yang, Yifei Zou, Minghui Xu, Dongxiao Yu, and Xiuzhen Cheng are with the School of Computer Science and Technology, Shandong University, Qingdao 266237, China. E-mail: 202020632@mail.sdu.edu.cn; yfzou@sdu.edu.cn; mhxu@sdu.edu.cn; dxyu@sdu.edu.cn; xzcheng@sdu.edu.cn.

• Yicheng Xu is with the Shenzhen Institute of Advanced Technology, Chinese Academy of Sciences, Shenzhen 518055, China. E-mail: yc.xu@siat.ac.cn.

∗ To whom correspondence should be addressed.

Manuscript received: 2021-06-14; revised: 2021-07-25; accepted: 2021-08-18

a prominent role in decentralizing blockchains, which would remarkably affect the performance and properties of a blockchain system, such as its fault tolerance, throughput, and scalability. To achieve high-quality transmissions and high-confidence requirements in IoT networks, an effective and efficient consensus protocol is the main focus. For example, the performance of many works[5–14] may benefit from a new effective and efficient consensus protocol.

Different from the conventional consensus protocol deployed on the full-blown Internet, the challenges for designing a distributed consensus protocol for blockchain systems in an IoT network mainly rely on the time complexity and energy consumption of reaching a consensus among agents in a network. Time complexity is the first issue considered by a large fraction of researchers. The more rapidly a consensus can be obtained in a blockchain system, the more real-time services can be provided by the system. However, in a traditional blockchain system, it often takes more than one hour to achieve a consensus, which is not suitable for shared economy applications in IoT networks. Providing more energy to support strong computation and transmission activities is a possible way to facilitate the process of a consensus. However, this method is not suitable for the consensus protocol in IoT networks, because a large fraction of devices in IoT networks are powered by a weak battery or small solar cell. Thus, energy consumption is another important challenge faced by devices in IoT networks. Neither the weak battery nor the solar cell can cover an energy-consuming consensus. The larger the energy consumed by the computation and transmission activities, the shorter the time that the devices in IoT will survive. Thus, it is much of importance to reach a good balance on the time complexity and energy consumption of a consensus protocol for blockchains in IoT networks using a carefully designed algorithm.

In recent years, the time complexity and energy consumption challenges in blockchain consensus have been considered in several studies. For example, the earliest and most widely used consensus protocol in the public blockchain, i.e., PoW is also famous for its wastage on electricity consumption[15]. To solve the energy consumption problem in PoW, the Proof-of-Stake (PoS) scheme in Ref. [16], and the Proof-of-Capacity scheme in Ref. [17] have been proposed, which require less energy to achieve a consensus. However, they are not specifically designed for a wireless network context. The Proof-of-Communication scheme in Ref. [4] seeks for a fast consensus in single-hop wireless networks. Experimental results showed that the Proof-of-Communication scheme solves the consensus problem with asymptotically optimal time complexity. However, the multi-hop scenario was not taken into account in Ref. [4]. If we directly adopt the Proof-of-Communication scheme from Ref. [4] in a multi-hop wireless network, the power consumption would exponentially increase when the diameter of the network gets larger. To the best of our knowledge, there are only a few works that simultaneously consider the time complexity and energy consumption problem in an IoT network background. Thus, it is significant to design a consensus protocol for blockchain systems in IoT networks[†], which has non-trivial performances on the time complexity and energy-consuming issues.

In this paper, we proposed the first distributed consensus protocol for blockchain systems in multi-hop wireless networks that has an asymptotically optimal time complexity and is energy saving to a large extent. The main contributions of our work are summarized as follows:

- We presented the first consensus protocol in the context of a multi-hop wireless blockchain network. Our consensus protocol enabled the implementation of the blockchain system in wireless networks realistically.
- We proposed a distributed algorithm to achieve a consensus among all agents in a network within $O(\log \mathit{\Gamma})$[‡] time steps, where $\mathit{\Gamma}$ denotes the maximum distance between any two miners (agents) in the blockchain network. Compared with the well-known lower bound $O(\log n)$ in Ref. [18] for a successful transmission in wireless networks, our algorithm has asymptotically optimal time complexity.
- In our algorithm, when miners seek for a consensus, a network organization termed as spanner[19] is obtained. Miners in spanner efficiently communicate with one another with an approximate transmission power, i.e., when nodes are close with one another, they will transmit with small transmission power. Otherwise, relatively large transmission power will be used. With this adaptive scheme on transmission power, our consensus algorithm has an energy-saving property. Essentially, the scheme we adopted to construct a spanner in this study is also original and has faster running time than the state-of-the-art spanners[19].

[†] Which are in usual the multi-hop wireless networks.

[‡] All $\log \mathit{\Gamma}$ and $\log n$ in this paper have constant bases larger than 1.

Extensive simulations were conducted to estimate the running time and energy expenditure of our algorithms, which also well corroborate our theoretical analysis.

Roadmap. The remainder of this paper is organized as follows: Section 2 presents related works. Section 3 presents the necessary models and problem definition. Sections 4 and 5 discuss algorithm description and analyses, respectively. Section 6 describes a simulation implemented to estimate the performance of our algorithm. Finally, Section 7 concludes our work.

2 Related Work

In general, two kinds of consensus protocols have been widely adopted in blockchain systems in recent years. The first one can be termed as Proof-of-X (PoX) consensus protocols, including PoW[20], PoS[16], Proof-of-Activity[21], and Proof-of-Space[17]. As usual, the PoW consensus protocol is recognized as the earliest and most used consensus protocol in public blockchains. Taking advantage of the PoW consensus protocol, Bitcoin[22], the first digital currency system, is applied to the decentralized peer-to-peer network. The main idea of PoW is to guarantee the consistency of the data and knowledge of safety via nodes' hashrate competition. Meanwhile, several significant drawbacks exist in the PoW consensus protocol. For instance, the huge electricity consumption[23] and the severe state fork problem degrade the energy efficiency and scalability of PoW consensus protocol. Accordingly, King and Nadal[15] presented the PoS concept with Peercoin, which can avoid the brute-force hashing competition and improve the energy-efficiency used for yielding blocks due to resources-free of block miners[24]. However, fairness and security would not be well guaranteed in the system owing to the weak randomness[25] and centralization risk[26]. The second one is byzantine fault tolerance (BFT) based consensus protocols, such as HotStuff[27] and Practical BFT (PBFT)[28]. For the PBFT consensus protocol, while nodes in the blockchain network vote for the final decision, they require $n \times n$ broadcasts in three crucial phases to tolerate faults. Obviously, $n \times n$ broadcasts are not an elegant operation for the time complexity and energy consumption in consensus algorithms, even though it strengthens the safety of consensus algorithms. However, none of the above consensus protocols is specifically designed in a wireless network, and most of them require stable transmissions between miners. Thus, the performances of these consensus protocols in a wireless IoT network are unknown.

To the best of our knowledge, the consensus protocol most relevant to our works is the one presentd in Ref. [4], in which an efficient and fair distributed Proof-of-Communication consensus protocol in wireless blockchain systems was proposed. Currently, it is one of the few known distributed consensus protocols for blockchains under a realistic physical interference model in the wireless network. It was proven that the consensus algorithm in Ref. [4] could achieve consistency in a single-hop wireless network within $O(\log n)$ time steps. However, a complex multi-hop wireless network was not considered, and no discussion for the energy consumption was given in Ref. [4]. Achieving the consensus in multi-hop wireless networks based on the physical interference model and saving the energy using an adaptive scheme on transmission power are the main differences of our work as compared with those in Ref. [4]. References [29–34] indicated some topics that our work can be applied to.

3 Models and Problem Definition

We modeled a multi-hop wireless blockchain network with a set V of n devices deployed arbitrarily in a two-dimensional Euclidean space. The devices in the IoT network are also termed the miners in the blockchain system. The Euclidean distance between two miners u and v is denoted by $d(u, v)$. The time in our algorithm implementation is divided into synchronized slots, each of which is the minimum time for message transmission. Within each slot, a miner v can decide to transmit or listen at will. A round is an interval that comprises a constant number of slots. For the benefit of better applicability to half-duplex and full-duplex transceiver equipped networks, we assumed that each miner is equipped with a half-duplex transceiver; i.e., each miner can transmit or listen in each slot but cannot do both. For simplicity, we assumed that the minimum distance between any pair of miners is normalized to 1 and denote the maximum distance between any two miners in the blockchain network by $\varGamma$.

3.1 Communication model

In this work, we adopt a typical physical interference model, termed as Signal-to-Interference-plus-Noise (SINR) model, to depict the message reception, interference, and contention generated by simultaneous transmissions among miners. The SINR model has been widely considered in Refs. [35–38] due to its closer reflection of reality than graph-based models. For any miner v, let Signal(v) be the strength of the signal received by v, which is given by the SINR Eq. (1). Then, let SINR(u,v) be the SINR rate of miner v for the transmission from u, the detail of which is illustrated in SINR Eq. (2).

$$\mathrm{Signal}(v) = \sum_{w \in S} P_w \cdot d(w,v)^{-\alpha} + N \tag{1}$$

$$\mathrm{SINR}(u,v) = \frac{P_u \cdot d(u,v)^{-\alpha}}{\sum\limits_{w \in S \setminus \{u\}} P_w \cdot d(w,v)^{-\alpha} + N} \tag{2}$$

where $S \subseteq V$ denotes the set of miners simultaneously transmitting with u, P_u (P_w) is the transmission power of miner u (w), α is the path-loss exponent whose value is normally between 2 and 6, and N is the ambient noise. In particular, when SINR$(u,v) \geqslant \beta$, miner v can decode the message from u, where β is not only related to the hardware, but also related to the size of the message. In our algorithm, because the miners in various layers transmit the same size of messages, we can set the same beta for all nodes. In this study, we assumed that $\alpha \in (2,6)$ and $\beta \geqslant 1$. Table 1 is used to show the notations and parameters in this work.

We assumed that each miner can determine its transmission power for each transmission at will and the maximum transmission power $P_{\max}$ is slightly larger than $\beta \cdot N \cdot \Gamma^{\alpha}$, which means that it is possible for the two miners with maximum distance in a network to transmit with each other, but the weak battery cannot support such transmission as regular transmissions.

Table 1 Notations and parameters used in this work.

Parameter	Value
V	Set of miners in IoT networks
n	Number of miners in IoT networks
u, v, w	Miners u, v, w
$d(u,v)$	Euclidean distance between miners u and v
Γ	Maximum distance between any two miners in the blockchain network
Signal(v)	Strength of the signal received by miner v
P_u	Transmission power of miner u
R	Transmission range, $R \geqslant d(u,v), \forall u, v \in V$
α, β, N	Parameters in SINR model
c	Positive constant in Algorithm 1
state$_v$ = $\{\mathcal{A}, \mathcal{S}, \mathcal{D}, \mathcal{R}\}$	States of nodes, corresponds to active, silent, dominator, and leader state, respectively
Count$_1$	Number of disagree miners
Count$_2$	Total number of miners
B_u	Block proposed by the elected leader u
ζ	Constant threshold

3.2 Definition of a consensus for blockchain systems in an IoT network

The goal of the classical consensus problem in a distributed system is to have all the nodes within a system agree on some common data values. Typically, it must have the following properties: Agreement, i.e., all nodes decide for the same value; termination, i.e., all nodes terminate in finite time; and validity, i.e., the decision value must be the input value of a node[39]. For the blockchain system, the miners record the history of network transactions into blocks, and these blocks are linked by a chain. Similar to the classical distributed consensus, the blockchain consensus protocol aims at making all participating miners in a blockchain agree on the common network transaction history. In our multi-hop wireless blockchain network, the following fundamental principles should be satisfied for a blockchain consensus when no fault occurs in a blockchain system.

- **Agreement**. All miners will have the same decision on whether a new block should be accepted or discarded. The local blockchain of each miner should have the same sequence on blocks.
- **Termination**. For a new block, each miner eventually decides to discard or write the new block into its local blockchain in finite time.
- **Validity**. For a new block that is going to be updated to the local blockchain of each miner, the transactions in the new block should be the same as what had happened in the history of the blockchain system, which is stronger than the definition of validity in a classical consensus.

In this study, we considered the problem of achieving the consensus in a blockchain system in a multi-hop wireless network. Specifically, we considered how to design an efficient and energy-saving distributed algorithm, which can be executed by all miners to achieve the consensus in a multi-hop wireless blockchain system.

Knowledge and capability of miners. We assumed that each miner has the values of R and N, the SINR parameters α and β, and their location information, which can be provided using the GPS service. Moreover, some auxiliary instruments, e.g., laser radar ranging technology and wireless local area network based location mechanism, can be used to increase the accuracy of devices' locations. Physical carrier sensing is also needed, which can monitor the signal in the channel when miners listen in the channel.

4 Algorithm

In this section, we first presented the overall framework of our consensus protocol. Then, we showed how we construct a network organization called a *spanner* to help miners exchange messages. Finally, we presented in detail how our algorithm works in multi-hop wireless blockchain networks.

4.1 Framework of the consensus protocol

Similar to the framework in Ref. [4], we also used four phases for achieving a consensus in a multi-hop wireless blockchain network: Spanner construction, block proposal, block verification, and chain update. The difference between Ref. [4] and our work is that the information scheduling in our phases is more complex because Ref. [4] only considered the consensus in a single-hop wireless network, in which a node can communicate with others by a simple leader election and broadcast operations. The input of our consensus algorithm is transaction data, which are generated and verified by all miners. The output is the encapsulated blocks and updated blockchain. Specifically, the blockchain system will update a new block to the blockchain in finite rounds via the following four-phase consensus:

- **Spanner construction phase (SCP)**. Because miners are not within a single hop wireless network, we construct a spanner to connect all miners and schedule the information among them. Then, a miner in spanner will be elected as the root to generate a new block.
- **Block proposal phase (BPP)**. After SCP, the elected root will pose a new block, which contains the trade records and corresponding information of the previous block, and will disseminate the new block to other miners in the spanner.
- **Block verification phase (BVP)**. Our spanner construction helps to efficiently disseminate the new block to each miner. After receiving the new block, a miner will check the validation of the block. The aggregation of the validation report for the new block from other miners to the root miner will also be facilitated through our spanner construction.
- **Chain update phase (CUP)**. When all the validation reports are aggregated to the root, the root will count the fraction of miners who agreed on the block. If the fraction is larger than a threshold, then the root will make an acceptance decision and broadcast it to other miners, to let them also update the block to their local chain. Otherwise, the new block will not be added to the chain. The threshold can be set by the administrator of the blockchain system when our protocol is implemented in reality.

4.2 Spanner construction

Because the constructed spanner plays an important role in the BPP and BVP, we first showed how the spanner is constructed in this subsection with the help of a dominating set election algorithm. As mentioned above, the minimum distance between two miners is normalized to 1, and the maximum one is denoted by Γ. We hierarchically construct a spanning tree with $\log\Gamma + 1$ levels by repeatedly executing Algorithm 1 for $\log\Gamma$ times. Algorithm 1 is a distributed algorithm designed in this work. Its input and output are all sets of miners, and the output miners constitute a dominating set of the input miners. The definition of a dominating set is given in the following:

Definition 1: A set of nodes D' is a dominating set of a set D if for each node v, either $v \in D$, or v has a neighbor $u \in D$. Nodes in set D' are called dominators, and nodes in set $D \setminus D'$ are called dominatees.

Let V_i be the set of nodes in level i. Initially, all miners belong to level 0, and are the input of our Algorithm 1. The output of Algorithm 1 will be the set V_1. Evidently, V_1 is the dominating set of V_0 and each dominatee in V_0 knows its dominator in V_1 if we can prove the correctness of Algorithm 1. Then, let set V_1 be the input of Algorithm 1 again, and we get set V_2. By repeatedly executing Algorithm 1 for $\log\Gamma$ times, we divide nodes into set $\{V_0, V_1, ..., V_{\log\Gamma}\}$. By connecting all dominatees in set V_i with their dominator in set V_{i+1}, we constitute our spanner.

In the following, we will introduce how Algorithm 1 elects a dominating set V_{i+1} from the set V_i in detail with $i \in \{0, 1, \ldots, \log\Gamma - 1\}$. At the beginning of Algorithm 1, each node v is active and knows which cell and windows it belongs to according to its location. We assume that (v_x, v_y) is the location of node v in the

Algorithm 1 Dominating set election at Layer *i*

```
Initialization: $\text{state}_v = A$, slot $= 0$;
Each miner $v$ in cell $(c_x, c_y)$ does:
for $c \cdot v$ slots do
    if $\text{state}_o = \mathcal{A}$ and $j = c \cdot (c_x \bmod c) + c_y$ then
        transmit a message with power $P_i = 2N \cdot r_i^{\alpha}$;
        if receive a message from miners in same window then
            $\text{state}_v = \mathcal{S}$;
    slot $++$;
        if $\text{state}_v = \mathcal{A}$ then
        $\text{state}_v = \mathcal{D}$;
```

coordination of our network; cell(v) denotes the cell with a length of $\frac{r_i}{\sqrt{2}}$ and also the corresponding ID of the cell in which node v is located in; (c_x, c_y) is the coordination of cell(v); window(v) denotes the window with a length of $\frac{2r_i}{\sqrt{2}}$ and also the corresponding ID of the window in which node v is located in; and (w_x, w_y) is the coordination of window(v). Then, we have

$$c_x = \left\lfloor \frac{\sqrt{2}v_x}{r^i} \right\rfloor,$$

$$c_y = \left\lfloor \frac{\sqrt{2}v_y}{r^i} \right\rfloor,$$

$$w_x = \left\lfloor \frac{c_x}{2} \right\rfloor,$$

$$w_y = \left\lfloor \frac{c_y}{2} \right\rfloor,$$

in which $r_i = 2^{i+1}$.

Let constant

$$c = 2\left\lceil \left[\left(32\frac{\alpha-1}{\alpha-2}+4\right)\cdot 2\beta\cdot 2^{\alpha/2}\right]^{\frac{1}{\alpha}} \right\rceil + 2.$$

In the following $c \cdot c$ slots, for any active node v and Slot j, if $j = c \cdot (c_x \bmod c) + c_y$, then v will transmit with power $P_i = 2N\beta \cdot r_i^{\alpha}$. Otherwise, v listens, and if it receives a message from miners in the same window, it becomes inactive. The inactive miners in set V_i do nothing until the current dominating set election algorithm ends. At the end of the $c \cdot c$ slots, active nodes become the dominator and constitute the output set V_{i+1}.

For a clear description, Fig. 1 illustrates the changes in cells and windows in different layers when i increases. And a simple spanner construction example is shown in Fig. 2, whose construction in reality may be complex due to the distribution of nodes.

When electing the dominator V_{i+1} from the set V_i, Algorithm 1 ensures that: (1) any two nodes within a distance of $(c-1)\cdot r_i$ will not simultaneously transmit in the same slot, which helps us handle the contention and interference in wireless transmissions; (2) each cell contains at most one miner in set V_i; and (3) there are four cells and at most four miners in a window, where the miner who first transmits will become the dominator of the other miners in the window. Point (1) makes sure that any two miners in the same window will not transmit in the same slot. These conclusions will be proven in our analysis.

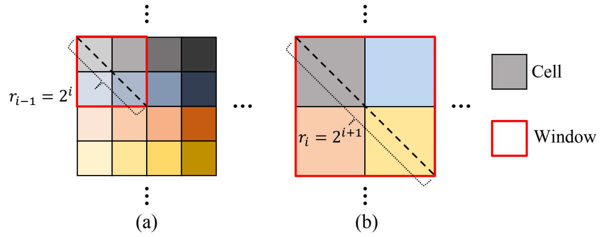


Fig. 1 Size of a window increasing when *i* gets larger in different layers. (a) Window setting at Layer *i*−1; (b) window setting at Layer *i*.

Thus, by repeating Algorithm 1 for $\log\Gamma$ times, the sets V_0, V_1, ..., $V_{\log\Gamma}$ would be elected sequentially. Moreover, for a dominator in set V_{i+1}, it has at most three dominatees in set V_i, which is sparse enough for our message transmission. When Algorithm 1 was used to elect the dominator from set $V_{\log\Gamma-1}$, there will only be one miner left in set $V_{\log\Gamma}$, and it becomes the root of the spanner. The correctness and efficiency of the spanner construction will be proven in the next section.

4.3 Algorithm description for consensus

In this subsection, we will show how the SCP, BPP, BVP, and CUP are implemented individually in our consensus algorithm. In our framework, there are four states for miners, namely $\mathcal{A}$, $\mathcal{S}$, $\mathcal{D}$, and $\mathcal{R}$. Miners in different states have various operations. Specifically, State $\mathcal{A}$ means that the miner is active in spanner construction. State $\mathcal{S}$ means that the miner has finished its spanner

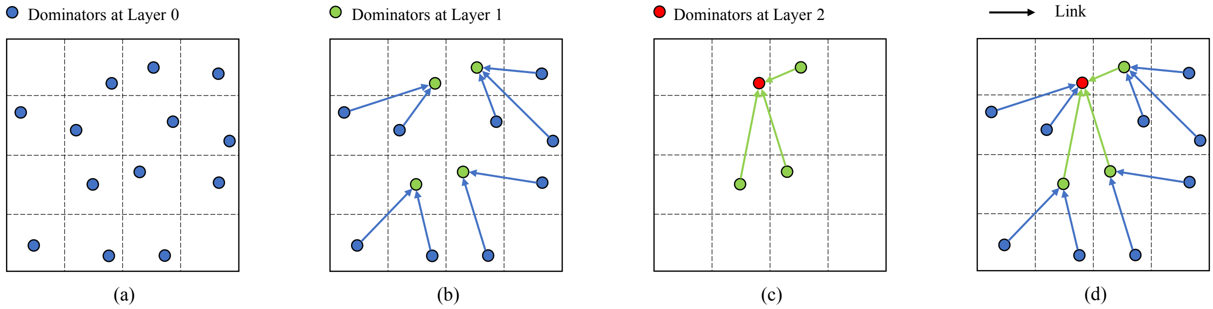


Fig. 2 Example of how a three-layer spanner was constructed. (a) Initial state, i.e., miners in set V_0; (b) dominating set construction from miners in layer V_0 to miners in layer V_1; (c) dominating set construction from miners in layer V_1 to miners in layer V_2; (d) final constructed spanner.

construction and keeps silent in SCP. State $\mathcal{D}$ means that the miner has become a dominator in the current dominating set election algorithm. State $\mathcal{R}$ means that the miner is the root of the spanner at the end of the SCP and will propose a new block for all miners to achieve a consensus.

4.3.1 SCP

Initially, as introduced before, all miners are in set V_0 of a spanner. Letting V_i with $i \in \{0, 1, 2, \cdots, \log \Gamma - 1\}$ as the input, Algorithm 1 will output set V_{i+1} as the dominators of miners in set V_i. By executing Algorithm 1 for $\log \Gamma$ times, we obtain the set $V_{\log \Gamma}$, which will become the root of our spanner. The correctness and efficiency of the dominating set election algorithm directly determine the performance of our SCP, which will be analyzed in the next section.

4.3.2 BPP

After the spanner construction, the root of the spanner will propose a block and broadcasts such a block with transmission power $P_{\max}$. The objective of the BPP is to let all miners know about the proposed block.

4.3.3 BVP

After receiving the block proposed by the root of the spanner, all the other miners will validate the block to examine whether the content of the block is the same as the transactions that occurred in the history of the blockchain network. However, it is not easy to rapidly aggregate the verification results from $n - 1$ miners. In this phase, we completed the aggregation within $O(\log \Gamma)$ rounds, which is an efficient result. Specifically, we gradually aggregated the verification results from dominatees in V_i to their dominators V_{i+1} when i increases from 0 to $\log \Gamma - 1$. The pseudocodes for dominator and dominatees are given in Algorithms 2 and 3, respectively. In the next stage, we will use the message aggregation from sets V_i to V_{i+1} as an example.

Without loss of generality (W.l.o.g. for short), we assume that v_a is the dominator in V_{i+1}, and v_b, v_c, and v_d are the dominatees of v_a in set V_i. In our spanner, there are at most three dominatees in set V_i for a dominator in set V_{i+1}. Before aggregating the verification results from dominatees v_b, v_c, and v_d in set V_i to dominator v_a in set V_{i+1}, we assume that the verification results are aggregated from V_0 to V_1, V_1 to V_2, $\ldots$, V_{i-1} to V_i. Each miner v_a has known the number of disagreeing miners and the total number of miners on its subtree, and recorded them using parameters Count_1 and Count_2, respectively. Then, the dominator transmits a message with power $P_i = 2N\beta \cdot r_i^{\alpha}$ in Slot 1, and listens in Slot 2, and the dominatee listens in Slots 1 and 2. The strengths of the channel sensed by dominatee v in Slots 1 and 2 are recorded by parameters $N_2(v)$ and $N_1(v)$, respectively. Then, as proven in our analysis, if the dominatee v transmits with a transmission power $(\text{Count}_1 \cdot U \cdot P_i)/(N_2(v) - N_1(v))$, then the strength of the signal will be $\text{Count}_1 \cdot U$ when the signal arrives at the location of the dominator. U is a sufficiently small number. Thus, when dominatees v_b, v_c, and v_d simultaneously transmit in Slot 3, the accumulated signal at the dominator v_a, i.e., $N_2(v_a)$ would be $(\text{Count}_1(v_b) + \text{Count}_1(v_c) + \text{Count}_1(v_d)) \cdot U + N_1(v_a)$. Let $X = (N_2(v_a) - N_1(v_a))/U$, the dominator v_a knows the value of $\text{Count}_1(v_b) + \text{Count}_1(v_c) + \text{Count}_1(v_d)$, i.e., the number of disagree miners in the subtree of dominator v_a. By letting each dominatee v transmit with power $(\text{Count}_2 \cdot U \cdot P_i)/(N_2(v) - N_1(v))$, the dominator v_a knows the total number of dominatees in its subtree. Because v_a can be any miner in set V_{i+1}, and v_b, v_c, and v_d are the dominatees of v_a in set V_i, the messages are aggregated from dominatees in set V_i to dominators in set V_{i+1}, when Algorithms 2 and 3 are executed in the i-th time. Thus, when Algorithms 2 and 3 are executed for $\log \Gamma$ times, the root knows the number of disagreeing miners and the total number of miners in the network.

Algorithm 2 Counting algorithm for dominator *u* at Layer *i*

```
Initialization: $N_1(u) = N_2(u) = 0$;
Slot 1: Transmit a message with power $P_i = 2N \cdot r_i^{\alpha}$;
Slot 2: Listen; $N_1(u) = \text{Signal}(u)$;
Slot 3: Listen; $N_2(u) = \text{Signal}(u)$; $X = \frac{N_2(u) - N_1(u)}{U}$;
        $\text{Count}_1 = \text{Count}_1 + \lfloor X \rfloor$;
Slot 4: Listen; $N_2(u) = \text{Signal}(u)$; $X = \frac{N_2(u) - N_2(u)}{U}$;
        $\text{Count}_2 = \text{Count}_2 + \lfloor X \rfloor + 1$;
```

Algorithm 3 Counting algorithm for dominatee *v* at Layer *i*

```
Initialization: $\text{state}_v = \mathcal{S}, N_1(v) = N_2(v) = v$;
Slot 1: Listen, $N_2(v) = \text{Signal}(v)$;
Slot 2: Listen, and $N_1(v) = \text{Signal}(v)$;
Slot 3: Confirm the leader and verify the block;
  if the block is invalid
    $\text{Count}_1 = \text{Count}_1 + 1$;
    Transmit with transmission power $P_C = \frac{\text{Count}_1 \cdot U \cdot P_t}{N_2(v) - N_1(v)}$;
Slot 4: Transmit with transmission power $P_C = \frac{\text{Count}_2 \cdot U \cdot P_i}{N_2(v) - N_1(v)}$;
```

4.3.4 CUP

In CUP, the root knows the number of disagreeing miners and the total number of miners, recorded by Count_1

and Count_2, respectively. As is mentioned above, the administrator can set a threshold ζ for the blockchain system. If the fraction of disagreeing miners in all miners is larger than ζ, the blockchain system will not accept the newly proposed block. Thus, when $\text{Count}_1/\text{Count}_2 > \zeta$ for the root, the root broadcasts a message with transmission power $P_{\max}$ to let all the miners discard the newly proposed block. Otherwise, the root keeps silent, and all miners will add the new block into their local chains.

5 Analysis

In this section, we analyzed the correctness and efficiency of our proposed spanner and consensus algorithms.

Lemmas 1 and 2, and Theorem 1 are given to analyze our spanner construction.

Lemma 1. Given the initial set V_0 as the input, by executing Algorithm 1 once, the output set V_1 is a dominating set for set V_0 with respect to distance 2^i with $i = 1$.

Proof. Initially, all miners are in set V_0, and the minimum distance between any pair of miners is normalized to 1. In Algorithm 1, each miner v with the coordination (v_x, v_y) belongs to cell(v) with length $r_0/\sqrt{2}$, $r_0 = 2^0$, and window(v) with length $2r_0/\sqrt{2}$. Thus, the following claims can be derived.

Claim 1. There is at most one miner in each cell.

Claim 2. There are four cells in each window and at most four miners in each window.

In Algorithm 1, only node v with $c \cdot (c_x \bmod c) + c_y = j$ will transmit on Slot j. Thus, the miners in a window will likely transmit in a same slot. Without loss of generality, we assume that a window w contains four miners v_a, v_b, v_c, and v_d, where v_a is the first one transmitting with power P_0 on Slot s. In the following, we will prove that miners v_b, v_c, and v_d will receive the message from v_a.

For miner v_b, we divide the whole blockchain network region into annuluses $\{C_b : b \geqslant 1\}$, with each C_b having the distance from v_b between $(b-1)(c-1) \cdot r_i/\sqrt{2}$ and $b(c-1) \cdot r_i/\sqrt{2}$. D_b denotes the set of dominators that simultaneously transmit in slot s and are located in C_b for $b \geqslant 2$. Since there is at most one miner transmitting in each window (Claim 2), and our transmitting scheme in Algorithm 1 makes sure that any two transmitting miners in Slot s are separated by a distance of at least $(c-1) \cdot r_i/\sqrt{2}$. Thus, disks centered at the miners in D_b with radius $(c-1) \cdot (\sqrt{2}/4)r_i$ are disjoint. In addition, these disks are in the annulus with the distance from v_b between $(b-3/2)(c-1) \cdot (r_i/\sqrt{2})$ and $(b+1/2)(c-1) \cdot (r_i/\sqrt{2})$. Then, the number of dominators transmitting simultaneously with v_a is upper bounded as below:

$$\frac{\pi\left(\dfrac{r_i}{\sqrt{2}}\right)^2\left[\left(b+\dfrac{1}{2}\right)^2(c-1)^2-\left(b-\dfrac{3}{2}\right)^2(c-1)^2\right]}{\pi\left[(c-1)\dfrac{\sqrt{2}}{4}r_i\right]^2} \leqslant 16b.$$

Moreover, the number of dominators in C_1, which simultaneously transmit with v_a is at most 4. For the transmission from v_a to v_b, the interference caused by these minors is at most

$$\mathcal{I}_{C_1} = 4P_i \cdot \left[(c-1)\frac{r_i}{\sqrt{2}}\right]^{-\alpha}.$$

Hence, the interference I_{v_b} at v_b for the transmission from v_a to v_b in our multi-hop blockchain network is bounded by

$$\mathcal{I}_{C_1} + \sum_{b=2}^{\infty} 16b \cdot P_i\left((b-1)(c-1)\frac{r_i}{\sqrt{2}}\right)^{-\alpha} \leqslant \left(32 \cdot \frac{\alpha-1}{\alpha-2} + 4\right) \cdot P_i \cdot \left((c-1)\frac{r_i}{\sqrt{2}}\right)^{-\alpha} = \left(32 \cdot \frac{\alpha-1}{\alpha-2} + 4\right) \cdot 2N\beta \cdot (c-1)^{-\alpha} \cdot 2^{\alpha/2}.$$

Setting $c = 2\left\lceil\left[(32 \cdot \frac{\alpha-1}{\alpha-2} + 4) \cdot 2\beta \cdot 2^{\alpha/2}\right]^{\frac{1}{\alpha}}\right\rceil + 2$, when v_a transmits, v_b can receive the message based on the SINR Eq. (2):

$$\text{SINR}(v_a, v_b) \geqslant \frac{P_i \cdot d(v_a, v_b)^{-\alpha}}{I_{v_b} + N} \geqslant \frac{P_i \cdot r_i^{-\alpha}}{I_{v_b} + N} \geqslant \beta.$$

Therefore, all miners v_b, v_c, and v_d in the same window with miner v_a can receive the message from v_a in round s and become inactive. Thus, v_a becomes the dominator of the miners in window window(v_a). For any miner in set V_0, the above analysis holds. For the windows containing less than four miners, we can still prove that the first transmitting miners will become the dominators. Thus, Lemma 1 is proven.

Lemma 2. Given set V_1 as the input, by executing Algorithm 1, the output set V_2 is a dominating set for set V_1 with respect to distance 2^i with $i = 2$

Proof. The proof of Lemma 2 is similar to that of Lemma 1. First, based on the findings, each miner v with the coordination (v_x, v_y) in set V_1 belongs to the cell cell(v) with a length of $r_1/\sqrt{2}$, $r_1 = 2^1$, and window

window(v) with a length of $2r_1/\sqrt{2}$. Accordingly, we can drive the following conclusions:

- There is at most one miner in each cell.
- There are four cells in each window and at most four miners in each window.

Thus, with a similar proof, we can show that when electing dominators from set V_1, the first transmitting miner v in a window window(v) will become the dominator, and other miners in the window window(v) will become the dominatees. The length of the window is $2r_1/\sqrt{2}$. Thus, Lemma 2 is proven.

Theorem 1. After $O(\log \Gamma)$ rounds, the root of our spanner can be elected.

Proof. In Lemma 5, we can see that in the first round which consists of $c \cdot c$ slots, V_1 is elected by Algorithm 1 as the dominator set for miners in V_0. In the second round, the dominator set V_2 is elected as the dominator set for miners in set V_1, according to Lemma 2. With a similar proof, we can easily prove that when Algorithm 1 is executed in the $O(\log \Gamma)$ rounds, set $V_{\log \Gamma}$ will be elected as the dominating set with respect to distance Γ and $V_{\log \Gamma}$ only contains one miner v, which will become the root of our spanner.

With Theorem 1, we had proven that the root will be elected in $O(\log \Gamma)$ rounds by repeatedly executing our dominating set election algorithm. The elected root will schedule the message transmissions in the block proposal, verification, and CUPs. To guarantee the consistency of the blockchain system, we then analyze the message dissemination/aggregation from/to the root via the following lemmas and theorem.

Lemma 3. In BPP, when the root transmits its block with power $P_{\max}$, all miners in the blockchain system can receive the block.

Proof. Because the root is the only transmitting root in the current slot, it is not hard to prove that each miner can receive the block according to the SINR Eq. (2).

In our BVP, each dominatee v transmits with power $\mathrm{Count}_1 \cdot U \cdot P_i/(N_2(v) - N_1(v))$ in Slot 3 and with power $\mathrm{Count}_2 \cdot U \cdot P_i/(N_2(v) - N_1(v))$ in Slot 4. $\mathrm{Count}_1(v)$ and $\mathrm{Count}_2(v)$ are the number of disagreeing miners and the total number of miners in the subtree of miner v, respectively, which should be aggregated by the dominator of v. In the following, how $\mathrm{Count}_1(v)$ is aggregated by the dominator of v in one round is shown.

Without loss of generality, we assume that at Slot t, there are one dominator v_a and three dominatees, i.e., v_b, v_c, and v_d.

Lemma 4. Let $X = (N_2(v_a)-N_1(v_a))/U$ be the parameter of dominator v_a in Slot 3 and dominatee v_x transmit with power $\mathrm{Count}_1(v_x) \cdot U \cdot P_i/(N_2(v_x)-N_1(v_x))$ with $x = b, c, d$. Then, the condition $\lfloor X \rfloor = \mathrm{Count}_1(v_a) + \mathrm{Count}_1(v_b) + \mathrm{Count}_1(v_c)$ holds.

Proof. As introduced before, N is the ambient noise in the environment. We have $N_1(v_a) = N_2(v_b) = N_2(v_c) = N_2(v_d) = N$. Thus, we have

$$N_2(u) = \sum_{v \in \{v_b, v_c, v_d\}} P_C \cdot d(u,v)^{-\alpha} + N + I =$$

$$\sum_{v \in \{v_b, v_c, v_d\}} \frac{\mathrm{Count}_1 \cdot \boldsymbol{U} \cdot P_i}{N_2(v)-N_1(v)} \cdot d(u,v)^{-\alpha} + N + I =$$

$$\sum_{v \in \{v_b, v_c, v_d\}} \frac{\mathrm{Count}_1 \cdot \boldsymbol{U} \cdot P_i}{\frac{P_i}{d(u,v)^{\alpha}} + N - N_1(v)} \cdot d(u,v)^{-\alpha} +$$

$$N + I =$$

$$(\mathrm{Count}_1(v_a) + \mathrm{Count}_1(v_b) + \mathrm{Count}_1(v_c)) \cdot \boldsymbol{U} +$$

$$N + I,$$

where I is the interference experienced by the dominator v_a and caused by the transmitting miners outside window(v_a), which has been proven to be a constant smaller than $\left(32 \cdot \dfrac{\alpha-1}{\alpha-2} + 4\right) \cdot 2N\beta \cdot (c-1)^{-\alpha} \cdot 2^{\alpha/2}$ in the proof of Lemma 5. Based on the value of $N_1(u)$, $N_2(u)$, $N_1(v)$, and $N_2(v)$, $X = \mathrm{Count}_1(v_a) + \mathrm{Count}_1(v_b) + \mathrm{Count}_1(v_c) + I/\boldsymbol{U}$. By setting $\boldsymbol{U}$ to as a constant larger than I, we proved Lemma 4 because $\mathrm{Count}_1(v_a)$, $\mathrm{Count}_1(v_b)$, and $\mathrm{Count}_1(v_c)$ are all positive integers.

With the above analysis, we show how the dominator collects the number of disagreeing miners from its dominatee in Slot 3. In Slot 4, the total number of miners from its dominatee will also be collected.

Lemma 5. By executing Algorithms 2 and 3 for $\log \Gamma$ rounds, the root knows the number of disagreeing miners and the total number of miners in the blockchain system.

Proof. Clearly, by executing Algorithms 2 and 3 for one round, the number of disagreeing miners and the total number of miners are aggregated from sets V_i to V_{i+1}. Then, i varies from 0 to $\log \Gamma$. Thus after executing Algorithms 2 and 3 for $\log \Gamma$ rounds, the root knows the number of disagreeing miners and the total number of miners in the blockchain system.

For the root miner v, let Count_1 and Count_2 be the number of disagreeing miners and the total number of miners in the blockchain system. If $\mathrm{Count}_1/\mathrm{Count}_2 > \zeta$, then the root will broadcast a message with transmission power $P_{\max}$ to inform all miners to discard the newly proposed block. In this situation, since the root is the only miner transmitting with power $P_{\max}$, all miners can receive the message from the root according to the SINR Eq. (2). If $\text{Count}_1/\text{Count}_2 \leqslant \zeta$, then the root transmits nothing and all miners will update the newly proposed block in their local chains.

Theorem 2. The time complexity of our consensus algorithm is $O(\log \Gamma)$. For each newly proposed block, when there are ζ fraction of nodes disagreeing, all miners reach a consensus to discard it. Otherwise, all miners will reach a consensus to add the newly proposed block to their local chains.

Proof. The time complexity for SCP, BPP, BVP, and CUP are $\Omega(\log \Gamma)$, $\Omega(1)$, $\Omega(\log \Gamma)$, and $\Omega(1)$, respectively. Then, the agreement, termination, and validity properties will be proven by Lemmas 6, 7, and 8, respectively.

Considering the lower bound $\Omega(\log n)$ for a successful transmission even without interference[40], the time complexity of our algorithm is asymptotically optimal, because $\Gamma \in \Theta(n)$ for any network with a constant density.

Lemma 6. The blockchain consensus protocol satisfies the validity requirement.

We prove the validity property via the following claim.

Claim 3. Each miner will help check whether the transactions in the new block are the same as what had happened in the history of the blockchain system and in making the decision of whether the block should be updated on the blockchain. If more than ζ fractions of miners disagree with updating the block, then the newly proposed block will be discarded. Otherwise, the new block will be updated on the blockchain. ζ is a constant determined by the administrator of a blockchain system.

Proof. Due to the spanner construction, there is only one root elected in the blockchain network in SCP in our algorithms. Then, the root will propose a new block and broadcast the block to all the other miners in BPP. Once a miner receives the message, it will verify the validity of the block based on the transactions or relevant information in this block. For this block, the number of disagreeing miners and total number of miners would be aggregated by the dominators layer by layer, and finally aggregated to the root. If the fraction of disagreeing miners is less than ζ, then the proposed block will be recorded to the local blockchain as the next building block for each miner in the blockchain system.

Lemma 7. After at most $\Omega(\log \Gamma)$ rounds, all miners will terminate the consensus process with high probability (w.h.p.).

Proof. The time complexity of our consensus algorithm directly proves the termination property.

Lemma 8. The blockchain consensus protocol satisfies the agreement property.

The following claims are used to prove Lemma 8. Claim 4 indicates that whether the proposed block is valid or not (determined by the fraction of disagreeing miners and the threshold ζ) the latest block of each miner in its local blockchain would be the same. Next, Claim 5 illustrates that the blocks in the local blockchain of each miner should be in the same order.

Claim 4. Regardless of whether new block B_u proposed by the elected leader u is valid or not, no two miners of the blockchain network have a different decision on either accepting or discarding B_u.

Proof. Claim 3 shows the following facts: (1) If the proposed block B_u is valid, then B_u will be accepted by all miners. (2) On the contrary, if the proposed block B_u is invalid, then B_u will be discarded by all miners. As a consequence, the latest block of each miner in its local blockchain would be the same.

Claim 5. For any two different miners u and v in a blockchain network, if blocks B_u^i and B_v^i are the i-th blocks in the local blockchain of u and v, then B_u^i and B_v^i are the same.

Proof. We prove this claim through a contradiction. Suppose that B_u^i and B_v^i are different. Clearly, B_u^i and B_v^i would be proposed by various valid leaders. However, there exists only one valid leader in a specific round in the blockchain network, which results in contradiction.

Combining Claims 4 and 5 above, we prove that the consensus protocol satisfies the agreement requirement. Moreover, our algorithm is energy saving. If we directly adopt the maximum transmission power for message transmission, such as the blockchain consensus protocol in Ref. [4], then we can easily obtain the consistency in the blockchain system. However, the power consumption is unmanageably large. In the following, we present two discussions to show the energy-saving property of our algorithm and how to extend our work to a Rayleigh model.

Discussion 1. Reference [4] has a similar framework to our work. In Phase 1, Ref. [4] used a leader election algorithm to elect a leader to propose a block, whereas our algorithm uses a spanner construction algorithm to elect a root to propose a block.

Let TP_1 and TP_2 be the total transmission power cost by our algorithm and by Ref. [4] in Phase 1, respectively.

$$\mathrm{TP}_1 = \sum_{i=0}^{\log\Gamma} P_i \cdot \frac{n}{4^i} = n \cdot \sum_{i=0}^{\log\Gamma} \frac{2N\beta r_i^{\alpha}}{4^i} =$$

$$2nN\beta \sum_{i=0}^{\log\Gamma} (2^i)^{\alpha-2} = 2nN\beta \cdot \frac{(2^{\alpha-2})^{1+\log\Gamma}-1}{2^{\alpha-2}-1},$$

and

$$\mathrm{TP}_{\max} \geqslant n \cdot P_{\log\Gamma} = 2nN\beta \cdot (2^{\log\Gamma})^{\alpha} = 2nN\beta \cdot \Gamma^{\alpha}.$$

The value of TP_1 is much less than TP_2. As mentioned before, the value of α is usually between 2 and 6, i.e., we set $\alpha = 3$ and $\Gamma = 2^9$. Then, we have $\mathrm{TP}_1/\mathrm{TP}_2 = 2^{-17}$. By comparing with the state-of-the-art work, we showed the energy-saving efficiency of our algorithm.

Discussion 2. In our work, we use the SINR model to depict the linear accumulation and received signal. While considering the blockage effect and shadow fading in the channel model, the multi-path reflections and fading interference/signal would lead to the dominators/root no longer exactly computing the number of disagreeing miners and the total number of miners. The Rayleigh fading model is a suitable path-loss model of wireless channels. The difference between the SINR model and the Rayleigh-fading mode lies in the following: Compared with the received signal strength $\mathrm{Signal}(v)$ under the SINR model, in the Rayleigh fading model, the received signal strength of a miner v will be a random variable that is exponentially distributed with a mean value $\mathrm{Signal}(v)$. Fortunately, our proposed multi-hop scheme can estimate the number of disagreeing miners and the total number of miners with a constant ratio at least with a constant probability, i.e., at least with a constant probability, the estimated numbers are only constant times larger/smaller than the real counts. With a similar analysis, the rough estimation will only reduce the accuracy and efficiency of our algorithm by a constant ratio.

6 Experimental Results

We examined the empirical performance of our multi-hop wireless blockchain consensus protocol in this section. Specifically, we focus on the time used for achieving consensus, spanner construction, and block verification in the multi-hop wireless blockchain network when the network size, network area, and SINR parameters vary.

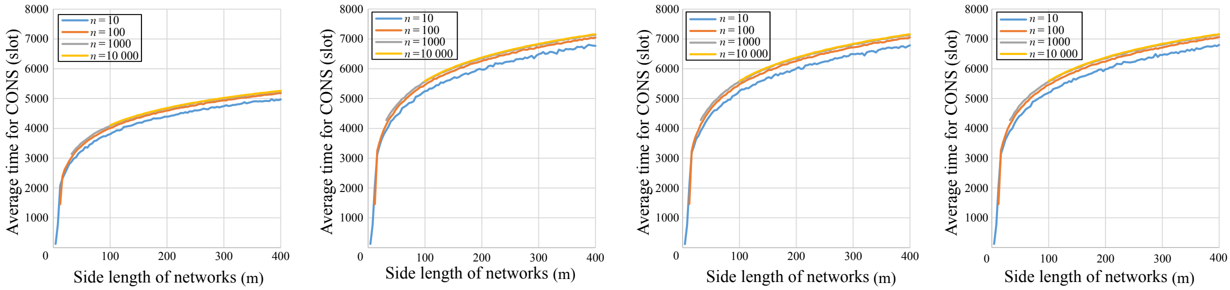
Parameter setting. In the simulation, we randomly and uniformly implemented n nodes into a square area of $D \times D$, where the size of D belongs to $[100\,\mathrm{m}, 400\,\mathrm{m}]$, and the minimum distance between any pair of nodes is normalized to 1 m. All nodes are initially in an active state ($\mathcal{A}$) for the spanner construction. The ambient noise at each node is normalized as 1.0 dB. For each layer i, with $i \in \{0, 1, \ldots, \log\Gamma\}$, we set the transmission range $r_i = 2^i$ and the transmission power to $P_i = 2N\beta \cdot r_i^{\alpha}$. The value of SINR parameters α and β and some other parameters used in our simulation are given in Table 2. All the experiments are conducted on the same platform with an Intel Xeon 2.4 GHz Linux workstation and 64 GB main memory, implemented in C++ and compiled by g++ compiler. W.l.o.g., for each reported result, we performed the simulation over 20 runs.

Protocol performance. The simulation results for achieving a consensus are given in Fig. 2, spanner construction is presented in Fig. 3, and block verification is shown in Fig. 4. Figure 2 illustrates the performance of our proposed consensus protocol in multi-hop wireless blockchain networks under different SINR parameters α and β, in which the x-axis and y-axis represent the side length of networks, and the average time for achieving a consensus (i.e., "CONS"), respectively. From Figs. 3a–3d, the curves depict the consensus time of our algorithm as the side length of networks and the number of nodes changes. As shown in Figs. 3a–3d, the consensus times is logarithmic with the side length of networks for the arbitrary number of nodes, which corroborates our analysis that the time complexity of our protocol is $O(\log\Gamma)$. Figures 3a–3d also illustrate that the delay bounds slightly change when α and β change. Thus, our algorithm is insensitive to the SINR parameters.

As mentioned before, our consensus protocol comprises four phases, namely, SCP, BPP, BVP, and CUP. Among them, spanner construction and block verification are two predominant phases of the execution time of our algorithm. Hence, we investigated the performances of our algorithm, including the average time for spanner construction and block verification

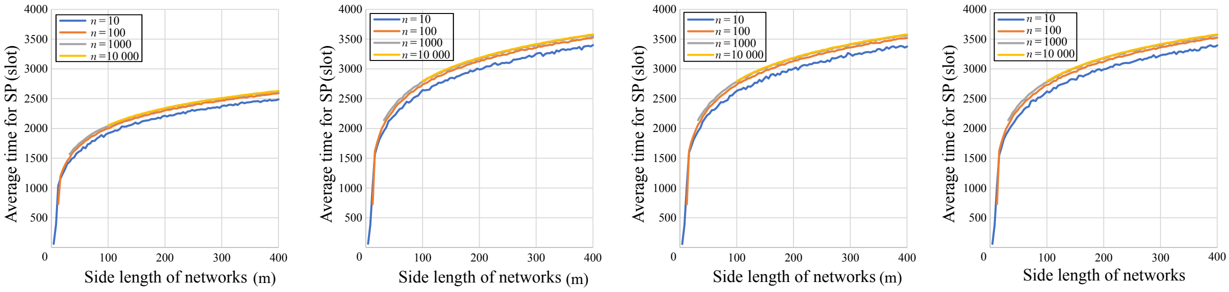
Table 2 Parameters in the simulation.

Parameter	Value
N (dB)	1.0
D (m)	$[4, 400]$
α	$\in \{3.0, 4.0\}$
β	$\in \{1.5, 2.0\}$
n	$\in \{10, 100, 1000, 10\,000\}$
r_i	2^i
P_i	$2N\beta \cdot r_i^{\alpha}$



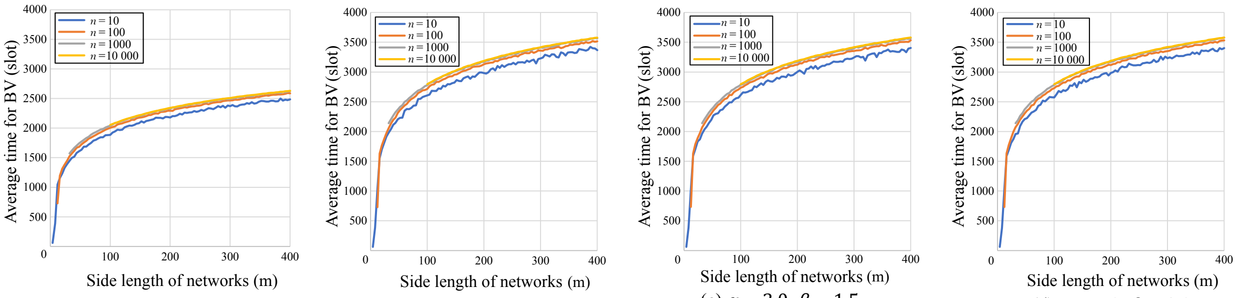
(a) $\alpha = 4.0$, $\beta = 1.5$ (b) $\alpha = 3.0$, $\beta = 2.0$ (c) $\alpha = 3.0$, $\beta = 1.5$ (d) $\alpha = 4.0$, $\beta = 2.0$

Fig. 2 Performance of our algorithm for achieving a consensus (CONS) under different SINR parameters α and β.



(a) $\alpha = 4.0$, $\beta = 1.5$ (b) $\alpha = 3.0$, $\beta = 2.0$ (c) $\alpha = 3.0$, $\beta = 1.5$ (m) (d) $\alpha = 4.0$, $\beta = 2.0$

Fig. 3 Performance of our algorithm on the spanner (SP) construction under different SINR parameters α and β.

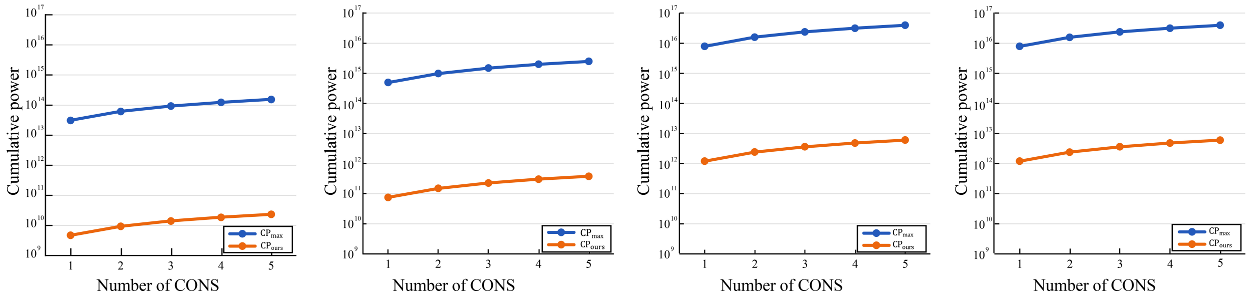


(a) $\alpha = 4.0$, $\beta = 1.5$ (b) $\alpha = 3.0$, $\beta = 2.0$ (c) $\alpha = 3.0$, $\beta = 1.5$ (d) $\alpha = 4.0$, $\beta = 2.0$

Fig. 4 Performance of our algorithm on block verification (BV) under different SINR parameters α and β.

under different values of α and β. The simulation results are shown in Figs. 3 and 4. One can draw the conclusion that both of their delay bounds are logarithmic in the side length of networks and are also insensitive in terms of the SINR parameters.

We also evaluated the power consumption of our algorithm by comparing it with the energy consumption in Ref. [4] with different side lengths of network D, which varies from 100 m to 400 m. As shown in Fig. 5, LCP$_{\max}$ represents the energy consumption of the consensus protocol in Ref. [4], in which all miners transmit with the maximum transmission power $P_{\max}$. LCP$_{\text{ours}}$ represents the transmission consumption of our algorithm. The x-axis and y-axis represent the number of consensuses and the cumulative energy consumption, respectively. Compared with the energy consumption in



(a) $D = 100$ (b) $D = 200$ (c) $D = 300$ (d) $D = 400$

Fig. 5 Power consumption of our algorithm under different side lengths of network D. The cumulative power is normalized by N in this work.

Ref. [4], our algorithm is rather energy saving.

7 Conclusion

This study is the first to consider the consensus problem for a blockchain system in multi-hop wireless networks under the SINR model. The proposed distributed protocol, which aims to solve the consensus problem not only has an asymptotically optimal performance on time complexity, but is also energy-saving. The theoretical analysis and empirical simulation show the non-trivial performances of our algorithm on the running time and energy consumption. It is believed that our multi-hop consensus protocol can make blockchain technology suitable for applications and services in IoT networks. Considering our consensus problem in a byzantine scenario will be our next step.

Acknowledgment

This work was partially supported by the National Key Research and Development Program of China (No. 2020YFB1005900) and the National Natural Science Foundation of China (NSFC) (Nos. 6212200494, 61971269, and 6210070740).

References

[1] A. Zanella, N. Bui, A. Castellani, L. Vangelista, and M. Zorzi, Internet of things for smart cities, *IEEE Internet Things J.*, vol. 1, no. 1, pp. 22–32, 2014.

[2] S. Huckle, R. Bhattacharya, M. White, and N. Beloff, Internet of things, blockchain and shared economy applications, *Procedia Comput. Sci.*, vol. 98, pp. 461–466, 2016.

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

[4] Q. Xu, Y. F. Zou, D. X. Yu, M. H. Xu, S. K. Shen, and F. Li, Consensus in wireless blockchain system, in *Proc. 15th Int. Conf. Wireless Algorithms, Systems, and Applications*, Qingdao, China, 2020, pp. 568–579.

[5] Z. P. Cai and Q. Chen, Latency-and-coverage aware data aggregation scheduling for multihop battery-free wireless networks, *IEEE Trans. Wirel. Commun.*, vol. 20, no. 3, pp. 1770–1784, 2021.

[6] J. W. T. Chan, F. Y. L. Chin, D. S. Ye, and Y. Zhang, Online frequency allocation in cellular networks, in *Proc. 19th Annu. ACM Symp. Parallel Algorithms and Architectures*, San Diego, CA, USA, 2007, pp. 241–249.

[7] W. T. Chan, Y. Zhang, S. P. Y. Fung, D. S. Ye, and H. Zhu, Efficient algorithms for finding a longest common increasing subsequence, in *Proc. 16th Int. Symp. Algorithms and Computation*, Sanya, China, 2005, pp. 665–674.

[8] D. X. Yu, Q. S. Hua, Y. X. Wang, and F. C. M. Lau, An $O(\log\ n)$ distributed approximation algorithm for local broadcasting in unstructured wireless networks, presented at the 2012 IEEE 8th Int. Conf. Distributed Computing in Sensor Systems, Hangzhou, China, 2012, pp. 132–139.

[9] Q. Chen, Z. P. Cai, L. L. Cheng, and H. Gao, Structure-free general data aggregation scheduling for multihop battery-free wireless networks, *IEEE Trans. Mob. Comput.*, doi: 10.1109/TMC.2021.3053557.

[10] J. Li, A. M. V. V. Sai, X. Z. Cheng, W. Cheng, Z. Tian, and Y. S. Li, Sampling-based approximate skyline query in sensor equipped IoT networks, *Tsinghua Science and Technology*, vol. 26, no. 2, pp. 219–229, 2021.

[11] J. Li, M. Siddula, X. Z. Cheng, W. Cheng, Z. Tian, and Y. S. Li, Approximate data aggregation in sensor equipped IoT networks, *Tsinghua Science and Technology*, vol. 25, no. 1, pp. 44–55, 2020.

[12] D. X. Yu, Y. F. Zou, Y. X. Wang, J. G. Yu, X. Z. Cheng, and F. C. M. Lau, Implementing the abstract MAC layer via inductive coloring under the Rayleigh-fading model, *IEEE Trans. Wirel. Commun.*, vol. 20, no. 9, pp. 6167–6178, 2021.

[13] D. X. Yu, Y. F. Zou, M. H. Xu, Y. C. Xu, Y. Zhang, B. Gong, and X. S. Xing, Competitive age of information in dynamic IoT networks, *IEEE Internet Things J.*, vol. 8, no. 20, pp. 15160–15169, 2021.

[14] Y. Zhang, J. C. Chen, F. Y. L. Chin, X. Han, H. F. Ting, and Y. H. Tsin, Improved online algorithms for 1-space bounded 2-dimensional bin packing, in *Proc. 21st Int. Symp. Algorithms and Computation*, Jeju, Korea, 2010, pp. 242–253.

[15] S. King and S. Nadal, PPcoin: Peer-to-peer crypto-currency with proof-of-stake, https://decred.org/ research/ king2012.pdf, 2012.

[16] Bitcoinwiki, Proof of stake, https://en.bitcoin.it/wiki/ Proof_of_Stake, 2014.

[17] S. Dziembowski, S. Faust, V. Kolmogorov, and K. Pietrzak, Proofs of space, in *Proc. 35th Annu. Cryptology Conf.*, Santa Barbara, CA, USA, 2015, pp. 585–605.

[18] W. T. Chan, Y. Zhang, S. P. Y. Fung, D. S. Ye, and H. Zhu, Efficient algorithms for finding a longest common increasing subsequence, *J. Comb. Optim.*, vol. 13, no. 3, pp. 277–288, 2007.

[19] D. X. Yu, L. Ning, Y. F. Zou, J. G. Yu, X. Z. Cheng, and F. C. M. Lau, Distributed spanner construction with physical interference: constant stretch and linear sparseness, *IEEE/ACM Trans. Netw.*, vol. 25, no. 4, pp. 2138–2151, 2017.

[20] Y. Xiao, N. Zhang, W. J. Luo, and Y. T. Hou, Modeling the impact of network connectivity on consensus security of proof-of-work blockchain, presented at the IEEE INFOCOM 2020—IEEE Conf. Computer Communications, Toronto, Canada, 2020, pp. 1648–1657.

[21] I. Bentov, C. Lee, A. Mizrahi, and M. Rosenfeld, Proof of activity: Extending Bitcoin's proof of work via proof of stake [extended abstract] y, *ACM SIGMETRICS Performance Evaluation Review*, vol. 42, no. 3, pp. 34–37, 2014.

[22] S. Nakamoto, Bitcoin: A peer-to-peer electronic cash system, https://bitcoin.org/bitcoin.pdf, 2008.

[23] K. J. O'Dwyer and D. Malone, Bitcoin mining and its energy footprint, presented at the 25th IET Irish Signals & Systems Conf. 2014 and 2014 China-Ireland Int. Conf. Information and Communications Technologies (ISSC 2014/CIICT 2014), Limerick, Ireland, 2014, pp. 280–285.

[24] J. Bonneau, A. Miller, J. Clark, A. Narayanan, J. A. Kroll, and E. W. Felten, SoK: Research perspectives and challenges for Bitcoin and Cryptocurrencies, presented at the 2015 IEEE Symp. Security and Privacy, San Jose, CA, USA, 2015, pp. 104–121.

[25] A. Kiayias, A. Russell, B. David, and R. Oliynykov, Ouroboros: A provably secure proof-of-stake blockchain protocol, in *Proc. 37th Annu. Int. Cryptology Conf.*, Santa Barbara, CA, USA, 2017, pp. 357–388.

[26] P. Gaži, A. Kiayias, and A. Russell, Stake-bleeding attacks on proof-of-stake blockchains, presented at the 2018 Crypto Valley Conf. Blockchain Technology (CVCBT), Zug, Switzerland, 2018, pp. 85–92.

[27] M. F. Yin, D. Malkhi, M. K. Reiter, G. G. Gueta, and I. Abraham, HotStuff: BFT consensus with linearity and responsiveness, in *Proc. 2019 ACM Symp. Principles of Distributed Computing*, Toronto, Canada, 2019, pp. 347–356.

[28] M. Castro and B. Liskov, Practical byzantine fault tolerance, in *Proc. 3rd Symp. Operating Systems Design and Implementation*, New Orleans, LA, USA, 1999, pp. 173–186.

[29] W. T. Chan, F. Y. L. Chin, D. S. Ye, G. C. Zhang, and Y. Zhang, On-line scheduling of parallel jobs on two machines, *J. Discrete Algorithms*, vol. 6, no. 1, pp. 3–10, 2008.

[30] F. Y. L. Chin, B. Fu, J. L. Guo, S. G. Han, J. L. Hu, M. H. Jiang. G. H. Lin, H. F. Ting, L. P. Zhang, Y. Zhang, et al., Competitive algorithms for unbounded one-way trading, *Theor. Comput. Sci.*, vol. 607, pp. 35–48, 2015.

[31] X. Y. Fan, M. Z. Dai, C. X. Liu, F. Wu, X. D. Yan, Y. Feng, Y. Q. Feng, and B. Q. Su, Effect of image noise on the classification of skin lesions using deep convolutional neural networks, *Tsinghua Science and Technology*, vol. 25, no. 3, pp. 425–434, 2020.

[32] Z. Y. Hu, D. S. Li, and D. K. Guo, Balance resource allocation for spark jobs based on prediction of the optimal resource, *Tsinghua Science and Technology*, vol. 25, no. 4, pp. 487–497, 2020.

[33] D. X. Yu, Y. F. Zou, J. G. Yu, X. Z. Cheng, Q. S. Hua, H. Jin, and F. C. M. Lau, Stable local broadcast in multihop wireless networks under SINR, *IEEE/ACM Trans. Netw.*, vol. 26, no. 3, pp. 1278–1291, 2018.

[34] D. X. Yu, Y. F. Zou, J. G. Yu, Y. Zhang, F. Li, X. Z. Cheng, F. Dressler, and F. C. M. Lau, Implementing the abstract MAC layer in dynamic networks, *IEEE Trans. Mob. Comput.*, vol. 20, no. 5, pp. 1832–1845, 2021.

[35] D. X. Yu, Y. F. Zou, Y. Zhang, F. Li, J. G. Yu, Y. Wu, X. Z. Cheng, and F. C. M. Lau, Distributed dominating set and connected dominating set construction under the dynamic SINR model, presented at the 2019 IEEE Int. Parallel and Distributed Processing Symp. (IPDPS), Rio de Janeiro, Brazil, 2019, pp. 835–844.

[36] D. X. Yu, Y. F. Zou, Y. Zhang, H. Sheng, W. F. Lv, and X. Z. Cheng, An exact implementation of the abstract MAC layer via carrier sensing in dynamic networks, *IEEE/ACM Trans. Netw.*, vol. 29, no. 3, pp. 994–1007, 2021.

[37] Y. F. Zou, M. H. Xu, H. Sheng, X. S. Xing, Y. C. Xu, and Y. Zhang, Crowd density computation and diffusion via internet of things, *IEEE Internet Things J.*, vol. 7, no. 9, pp. 8111–8121, 2020.

[38] Y. F. Zou, D. X. Yu, L. B. Wu, J. G. Yu, Y. Wu, Q. S. Hua, and F. C. M. Lau, Fast distributed backbone construction despite strong adversarial jamming, presented at the IEEE INFOCOM 2019—IEEE Conf. Computer Communications, Paris, France, 2019, pp. 1027–1035.

[39] C. Dwork, N. Lynch, and L. Stockmeyer, Consensus in the presence of partial synchrony (preliminary version), in *Proc. 3rd Annu. ACM Symp. Principles of Distributed Computing*, Vancouver, Canada, 1984, pp. 103–118.

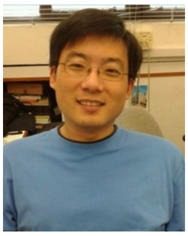
[40] J. Schneider and R. Wattenhofer, What is the use of collision detection (in wireless networks), in *Proc. 24th Int. Symp. Distributed Computing*, Cambridge, MA, USA, 2010, pp. 133–147.

Li Yang received the MS degree from Fuzhou University, Fuzhou, China, in 2020. He is currently pursuing the PhD degree at the School of Computer Science and Technology, Shandong University, Qingdao, China. His research interests include wireless networks and distributed computing.

Yifei Zou received the BEng degree from Wuhan University, Wuhan, China, in 2016, and the PhD degree from University of Hong Kong, Hongkong, China, in 2020. He is currently an assistant professor at the School of Computer Science and Technology, Shandong University, Qingdao, China. His research interests include wireless networks, ad hoc networks, and distributed computing.

Minghui Xu received the BS degree in physics from Beijing Normal University, Beijing, China, in 2018, and the PhD degree in computer science from George Washington University, Washington, DC, USA, in 2021. He is currently an assistant professor at the School of Computer Science and Technology, Shandong University, Qingdao, China. His research focuses on blockchain, distributed computing, and quantum computing.

Yicheng Xu received the PhD degree from Beijing University of Technology, Beijing, China, in 2018. Currently he is an associate professor at Shenzhen Institute of Advanced Technology, Chinese Academy of Sciences, Shenzhen, China. His research interests include algorithm design and discrete mathematics.



Dongxiao Yu received the BS degree from Shandong University, Qingdao, China, in 2006, and the PhD degree from University of Hong Kong, Hongkong, China, in 2014. He became an associate professor at the School of Computer Science and Technology, Huazhong University of Science and Technology, Wuhan, China, in 2016. He is currently a professor at the School of Computer Science and Technology, Shandong University, Qingdao, China. His research interests include wireless networks, distributed computing, and graph algorithms.

Xiuzhen Cheng received the MS and PhD degrees in computer science from University of Minnesota, Twin Cities, USA, in 2000 and 2002, respectively. She was a faculty member at the Department of Computer Science, George Washington University, during 2002–2020. Currently she is a professor at the School of Computer Science and Technology, Shandong University, Qingdao, China. She is a fellow of IEEE. Her research focuses on blockchain computing, security and privacy, and Internet of Things.