






AirCon: Over-the-Air Consensus for Wireless Blockchain Networks

Xin Xie , Cunqing Hua , *Member, IEEE*, Jianan Hong , Pengwenlong Gu , *Member, IEEE*,
and Wenchao Xu , *Member, IEEE*

Abstract—Blockchain has been deemed as a promising solution for providing security and privacy protection in the next-generation wireless networks. Large-scale concurrent access for massive wireless devices to accomplish the consensus procedure may consume prohibitive communication and computing resources, and thus may limit the application of blockchain in wireless conditions. As most existing consensus protocols are designed for wired networks, directly apply them for wireless users equipment (UEs) may exhaust their scarce spectrum and computing resources. In this paper, we propose AirCon, a byzantine fault-tolerant (BFT) consensus protocol for wireless UEs via the over-the-air computation. The novelty of AirCon is to take advantage of the intrinsic characteristic of the wireless channel and automatically achieve the consensus in the physical layer while receiving from the UEs, which greatly reduces the communication and computational cost that would be caused by traditional consensus protocols. We implement the AirCon protocol integrated into an LTE system and provide solutions to the critical issues for over-the-air consensus implementation. Experimental results are provided to show the feasibility of the proposed protocol, and simulation results to show the performance of the AirCon protocol under different wireless conditions.

Index Terms—Consensus protocol, over-the-air computation, wireless blockchain network, lattice coding.

I. INTRODUCTION

THE next-generation wireless networks are expected to provide ubiquitous access to heterogeneous devices with ultra-high throughput, reliability, and extremely low latency [1]. This will bring great challenges to the security management in current 4G/5G mobile systems. Blockchain, a generic distributed ledger technology (DLT), has received extensive attention due to the significant advantages from decentralization, immutability, and security in recent years. The blockchain technology enables registering and updating transactions in a decentralized fashion

via consensus among participants, which has become the foundation of new security architecture for future wireless networks.

The consensus protocol plays an important role in the blockchain system. The efficiency of the consensus protocol determines blockchain system security bounds (fault tolerances) and performance such as transaction throughput, delay, and node scalability [2]. Among many consensus protocols, the proof-based algorithm (PoX) is often used in a public chain, such as proof-of-work (PoW) [3], proof-of-stake (PoS) [4], whereby participants can join/leave the network without authentication. These PoX-based consensus protocols achieve good scalability with the cost of high resource consumption. For instance, the PoW protocol employed in Bitcoin consumes a huge amount of power resources to compute a meaningless hash value. And the confirmation delay and throughput of these consensus protocols also limit the application scenario. The voting-based consensus protocols, such as PBFT [5], RAFT [6], relies heavily on inter-participant communications to achieve consensus. Therefore, the communication resource is critical for the voting-based protocols, which can only be applicable for small/middle scale networks.

Most of the existing consensus protocols are primarily designed over wired communication networks, some new challenges may arise when they are deployed in wireless networks. First, wireless devices usually have limited resources (energy, storage, computation, etc.), so the resource-consuming PoX protocols are not suitable. Second, the wireless communication channel condition varies dynamically, not to mention that users often suffer from limited bandwidth resources, which significantly affects the consensus performance, especially for the voting-based protocols.

As a solution, the wireless channel features can be utilized to solve this dilemma. For the voting-based protocols, a block is generated based on the decision from the majority participants [7]. For instance, in the PBFT protocol, each participant determines the consensus results by counting the number of messages with consistent hash from other participants. For traditional “communicate-then-consensus” solutions, the consensus process can be conducted only after all messages from other participants are successful received and decoded. Instead of collecting messages from all participants, the consensus process only require the result of a function of these messages, e.g., the number of messages with consistent hash from other participants, rather than the details bits of each message.

Manuscript received 14 October 2022; revised 9 June 2023; accepted 3 July 2023. Date of publication 10 July 2023; date of current version 4 April 2024. This work was supported in part by the National Key Research and Development Program of China under Grants 2022YFB2702302 and 2020YFB1807504, and in part by the Natural Science Foundation of China under Grants 62171278, 62101327, and 62202290. Recommended for acceptance by J. Wu. (*Corresponding author: Cunqing Hua.*)

Xin Xie, Cunqing Hua, and Jianan Hong are with the School of Cyber Science and Engineering, Shanghai Jiao Tong University, Shanghai 200240, China (e-mail: xiexin_312@sjtu.edu.cn; cqhua@sjtu.edu.cn; hongjin@sjtu.edu.cn).

Pengwenlong Gu is with the Department INFRES, LTCI, Telecom Paris, Institut Polytechnique de Paris, 91120 Palaiseau, France (e-mail: pengwenlong.gu@sjtu.edu.cn).

Wenchao Xu is with the Department of Computing, Hong Kong Polytechnic University, Hong Kong, SAR, China (e-mail: wenchao.xu@polyu.edu.hk).

Digital Object Identifier 10.1109/TMC.2023.3292898

Therefore, a “communicate-and-consensus” scheme can be adopted to reduce the communication and computational complexity. Specifically, the communicating and consensus tasks can be accomplished simultaneously by leveraging the over-the-air computation (AirComp) technology [8] in wireless networks, which is a novel technology that exploits the superposition property of wireless multiple access channel (MAC) to aggregate signals from multiple participants over the same wireless channel. In this way, the destructive interferences from different participants are turned into a constructive one by appropriately matching the structure of the wireless channel and the function of the signals can be computed from the aggregated signal.

In this paper, we propose “AirCon”, an over-the-air consensus protocol for wireless networks, which significantly reduces the communication and computational overhead in a wireless network by using the AirComp technique. The novelty of this protocol is two-folded. First, the hash symbols of all participating users are encoded using lattice codes and transmitted to the base station (BS) simultaneously over the same wireless spectrum via the AirComp technique, which greatly reduces the wireless resource usage. Second, by leveraging the structural property of the lattice codes, each user can verify the consistency of its hash value with respect to the aggregated hash symbols without decoding the signal, whereby the consensus can be achieved with extreme low computational complexity. To the best of our knowledge, this paper is the first attempt to apply AirComp and lattice coding techniques to the consensus protocol in wireless blockchain systems.

To show the feasibility of the AirCon protocol, we implement AirCon on an LTE system, where two key issues are solved: 1) Synchronization problem, which is the key to ensure that all participants transmit data in the same frequency simultaneously such that the signals can be accurately aligned at the BS. We will discuss the details of LTE synchronization mechanisms and show that these intrinsic mechanisms are sufficient for satisfying the synchronization requirement for the implementation of AirComp in an LTE system. 2) Uplink channel estimation and feedback problem. The participants need to learn the uplink channel state information (CSI) so that the channel fading can be pre-compensated. To this end, we propose a flexible pilot symbol assignment scheme, which achieves a better tradeoff between estimation accuracy and estimation latency.

The main contributions of the paper are summarized as follows:

- We propose a novel AirComp-based consensus protocol (AirCon), which enables all participants to transmit their hash symbols to the BS simultaneously using the same wireless channel, and thus significantly reduces the communication cost of consensus messages.
- We propose a hash consistency verification scheme in the physical layer, wherein each participant determines whether its hash value is consistent with the aggregated hash symbols or not without decoding the signal, which significantly reduces the complexity of consensus computation.

- We implement AirCon based on srsLTE¹ [9], an open-source LTE soft defined radio (SDR) platform. Extensive experiments are conducted to validate the practical performance of AirCon in a real-world LTE testbed. We also provide simulation results to evaluate the performance of the proposed schemes under more general network conditions.

The rest of this paper is organized as follows. In Section II, we discuss the research progress in blockchain-enabled wireless networks and the AirComp technique. In Section III, we introduce the system model and propose a modified PBFT-based over-the-air consensus (AirCon) procedure for wireless blockchain systems. The detailed design of AirCon protocol is presented in Section IV and the details for implementation of AirCon in the LTE system are discussed in Section V. The analysis of key metrics for the AirCon protocol is presented in Section VI. Experimental and simulation results are provided in Section VII. Finally, we summarize this work in Section VIII.

Notations: Table I summarizes the symbols and notations adopted in this paper.

II. RELATED WORKS

In this section, we give a brief introduction to the research progress on blockchain-enabled wireless networks and AirComp technique.

A. Blockchain-Enabled Wireless Networks

Blockchain has been studied extensively in literature as a security and privacy protection scheme for various application scenarios in wireless networks. In [10], the authors examined the application of blockchain in the Internet of things (IoT). They concluded that the blockchain-IoT combination is powerful and can lead to significant transformations across several industries. Blockchain technology also can be applied in the internet of vehicles [11] to provide cybersecurity protection for vehicular communications, including the dynamic control of source reliability, and the integrity and validity of the information exchanged. The incorporation of blockchain into the next-generation radio access network (RAN) also attracted great interest. A unified framework of the blockchain radio access network (B-RAN) was proposed in [12] as a trustworthy and secure paradigm for upcoming 6 G networking. Some critical elements of B-RAN, such as the deployment of smart contract [13], trustworthy access [14], [15], mathematical modeling [16] were also explored.

Due to the resource-limited feature of wireless devices, the blockchain system needs to be improved for the wireless environment. The authors in [12] proposed proof-of-devices (PoD) as a low-cost consensus protocol, which is based on the fact that B-RAN is comprised of a tremendous number of devices and attackers cannot control 51% devices of the whole network. In an edge computing scenario, the authors in [17] considered

¹The latest version of srsLTE has been renamed as srsRAN and to support 5G standards.

constructing a collaborative mining network (CMN) to execute mining tasks for mobile blockchain. Miners can offload their mining tasks to non-mining devices within a CMN when the resources are insufficient. The authors in [18] proposed a lightweight blockchain system to reduce the computational cost and speed up the block generation rate in the industrial IoT (IIoT).

On the other hand, some research works were concerned with the impact of a wireless channel on blockchain performance. In [19], the authors analyzed the trade-off between communication reliability and computing power in blockchain security and presented a lower bound to the computing power that is needed to conduct an attack with given communication reliability. Based on the widely used CSMA/CA mechanism, the impact of communication transmission delay on the confirmation delay, transaction per second, and transaction loss probability is analyzed in [20]. The authors in [21] analyzed the impact of node geographical distribution in the spatial domain and designed an algorithm to determine the optimal full-function node deployment. In [2], the authors evaluated the impact of scarce frequency spectrum resources on blockchain performance.

B. Over-the-Air Computation

In this paper, we design a novel consensus protocol based on the AirComp technology, which exploits the intrinsic superposition property of the wireless channel to achieve efficient transmissions of multiple users.

AirComp was proposed by B. Nazer and M. Gastpar in their cornerstone work [8], where nested lattice coding is used as a joint source-channel coding strategy to reliably reconstruct a function of sources over a multiple-access channel. Compared with linear source coding, nested lattice coding can approach the performance of a standard random coding [22]. The lattice-coded AirComp was extended to the relay-assisted network by proposing a compute-and-forward relaying scheme in [23], where the relay decodes a linear combination of all messages instead of ignoring interferences as noise. Furthermore, the authors in [24] investigated the computation rate of coded AirComp over a multi-cluster network with constant channel gains. Instead of assuming ideal MACs, the authors in [25] studied the coded AirComp over fading MACs, where an opportunistic in-network computation framework is proposed to ensure that the computation rate is not limited by deep-fading devices. Furthermore, the authors in [26] extended coded AirComp from narrow-band to wide-band fading channels and derived the computation rate. The transceiver design is taken into account to maximize the achievable computation rate of hierarchical AirComp [27] and to balance the trade-off between the computation rate and the transmission delay for CSI acquisition [28].

The AirComp technique can be applied in a wide range of application scenarios. In [29], [30], AirComp was proposed for fast data fusion, whereby the fusion center (FC) attempts to compute a specific function from the data of all sensor nodes (e.g., average reading). AirComp also has been extended for model aggregation in the distributed edge learning system [31], [32], [33], which is shown to reduce the communication

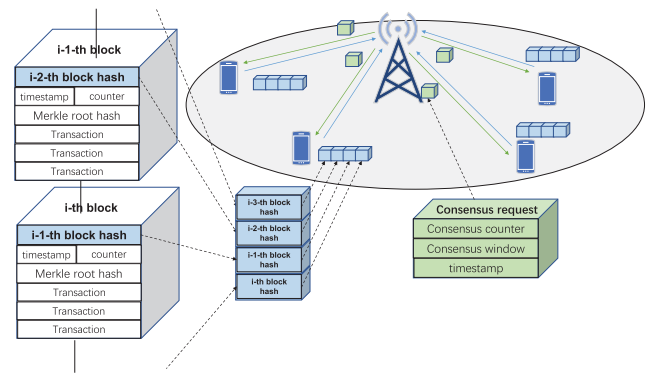


Fig. 1. Blockchain-enabled wireless network model.

latency without significant loss of the learning accuracy [34]. In a multi-agent system, AirComp can be used for distributed consensus control [35], [36] to reach an agreement over a set of variables of common interest, such as velocity, acceleration, and trajectory in vehicular platooning. Our work extends the application of AirComp to the consensus protocol in wireless blockchain systems.

Despite the extensive theoretical work in literature, the research on the implementation issues for AirComp is relatively scarce. In [29] and [30], the authors proposed to modulate the value to be computed through AirComp as the mean value or power of a random sequence, so the accurate synchronization and CSI is not necessary for implementing the AirComp technique. In [37], the authors proposed to achieve system synchronization via the “AirShare” scheme [38], and the uplink CSI was obtained by using channel reciprocity between uplink and downlink transmissions. In [39], the uplink CSI was obtained by feedback from the fusion center. These implementation schemes were designed for analog function computation (AFC) in a wireless network. In this paper, we implement the AirComp technique based on the universal LTE system, which demonstrate that AirComp can be realized in the current digital communication systems.

III. SYSTEM MODEL

We consider a blockchain-enabled wireless network as illustrated in Fig. 1, where a set of K users are served by a BS. These users maintain the same blockchain. The transaction blocks are generated periodically based on a consensus protocol, and stored in a distributed manner in all users.

In a blockchain system, the block appending procedure can be separated into three stages: Request, Consensus and Confirm. The function of each stages are described roughly as follows:

Request: The client sends a request to the primary user to trigger a transaction. After collecting enough transactions, the primary user generates a candidate block and broadcasts it to other replica users and requests to start the consensus procedure.

Consensus: After the candidate block is broadcast to all replica users in the consensus network, the consensus users verify the validity of the candidate block and trigger a protocol to achieve the consensus across the consensus network.

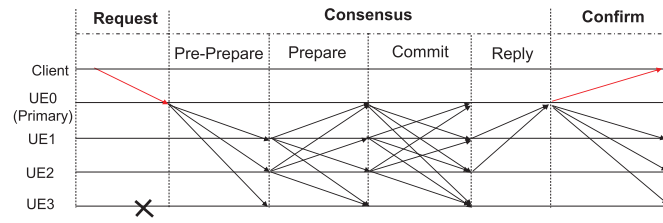


Fig. 2. Diagram of the traditional PBFT consensus protocol. Hash verification for the candidate block is achieved using the orthogonal channels.

Confirm: When the consensus has been achieved, the primary user broadcasts a confirm message to all replica users and the client, then the candidate block is appended in each blockchain which is distributively stored at each replica user.

We consider employing the PBFT protocol in the consensus stage. As mentioned in Section I, the PBFT is one of the classic voting-based consensus protocols, which relies heavily on inter-node communications to achieve consensus. The block appending procedure based on the traditional PBFT protocol is illustrated in Fig. 2. The PBFT consensus protocol consists of four phases as follows:

- **Pre-prepare:** After receiving the candidate block and request from the primary user, all replica users verify the transactions of the candidate block, then multicast the hash of the candidate block to other replica users and enter the Prepare phase.
- **Prepare and Commit:** Each consensus user cross-verifies the hash from other users. If the number of hash (m) that is matched with the local hash is enough (specifically, $m \geq 2f$ where f is the tolerable number of faulty node [5]), the user enters the next phase.
- **Reply:** If the consensus users have entered the Reply phase, they can send the Reply message to the primary user and the primary user determines whether the consensus is achieved or not.

From the communication perspective, the bottleneck of the consensus protocol shown in Fig. 2 lies in the *Prepare* and the *Commit* phases. The hash value generated by different users should be transmitted to each other via orthogonal wireless channels (such as TDMA, FDMA, or OFDMA in 4G/5G networks) for hash cross-validation among users. Therefore, the wireless resource usage in these two phases increases with the number of consensus users, which cannot guarantee reliable and low latency consensus for larger network size.

In this paper, we revise the PBFT protocol for wireless networks based on the AirComp technology, which is illustrated in Fig. 3. Specifically, we mainly focus on the procedures revision of the *Prepare* and *Commit* phases in the traditional PBFT protocol to address the communication bottleneck problem. The details of the proposed consensus protocol will be discussed in Section IV and the rough description are presented as follows:

- **Pre-prepare:** After receiving the candidate block and request from the primary user, all replica users verify the transactions of the candidate block, then send the hash symbols of the candidate block to the BS in the same uplink

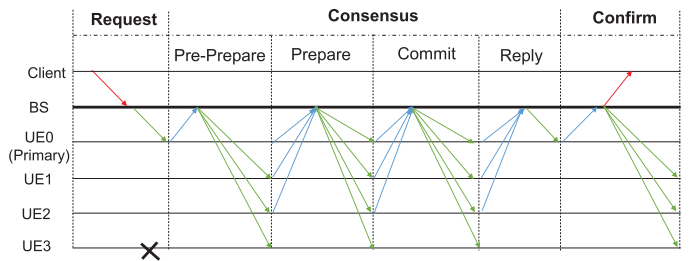


Fig. 3. Diagram of the proposed AirCon protocol. Hash verification for the candidate block is achieved through the BS using the multiple access and broadcast channels.

channel and enter the *Prepare* phase. Note that user-specific information (such as user ID) should not be included in the generation of the hash value.

- **Prepare and Commit:** The BS obtains the superposed hash symbols and broadcasts them to all users. All consensus users adopt a threshold-based two-round hash verification method to determine the consensus result.
- **Reply:** Only those users satisfying the threshold return the Reply message to the BS via the AirComp technique. Then the superposed Reply message is forwarded to the primary user. The primary user also adopts the same threshold-based method to determine whether the consensus is achieved or not.

Note that in the proposed consensus protocol, the only function of the BS is to receive and forward the superposed hash symbols by acting as a data fusion center. In other words, the consensus result is only determined by these distributed users via the proposed hash verification method. The users in the networks can be classified as normal users and abnormal users. These normal users will behave honestly and generate the same output if they get the same input. On the other hand, these abnormal users can be further classified as *crash users* and *Byzantine users*. During the consensus process, crash users behave abnormally due to some sudden factors, such as communication channel outage and user system crash. Therefore, the output of crash users is random and independent of each other. On the contrary, the output of Byzantine users can be elaborately designed by malicious attackers to tamper with consensus results. Therefore, it is more destructive to the blockchain network. In our protocol design in Section IV-C, both crash users and Byzantine users are considered.

IV. AIRCON PROTOCOL DESIGN

In this section, we provide the detailed design of the over-the-air consensus (AirCon) protocol to address the transmission bottleneck problem in the *Prepare* and the *Commit* phases based on the AirComp and the lattice coding techniques. The idea is to modulate the hash bits of different users using the lattice codes, which can be transmitted simultaneously to the BS over the same wireless channel. Due to the structural property of the lattice codes, the BS can adopt the AirComp technique to aggregate the linear combination of all codes (which is a codeword itself),

and forwards it back to all users, which can be used by each user to make a consensus decision without decoding the signal.

In the following, we first introduce the preliminary background on the AirComp and lattice coding techniques. We then present the hash consistency verification scheme based on these two techniques, with which a two-round consensus procedure for the AirCon protocol is designed.

A. Over-the-Air Computation

We consider an OFDM-based wireless system, which consists of a single BS and a set of K users. All users transmit signals to the BS through the same uplink channel. The received signal at the BS is expressed as:

$$\mathbf{y} = \sum_{k=0}^{K-1} \mathbf{H}_k \mathbf{B}_k \mathbf{x}_k, \quad (1)$$

where $\mathbf{x}_k \in \mathbb{C}^N$ is the symbol vector transmitted by user k , $\mathbf{B}_k = \text{Diag}\{b_{k0}, b_{k1}, \dots, b_{kN-1}\} \in \mathbb{C}^{N \times N}$ is the pre-processing matrix for user k . $\mathbf{H}_k = \text{Diag}\{h_{k0}, h_{k1}, \dots, h_{kN-1}\} \in \mathbb{C}^{N \times N}$ is the channel matrix from user k to the BS. N is the total number of sub-carriers. K is the total number of users.

Under the condition that each user k has perfect knowledge of \mathbf{H}_k , and its transmit power is not bounded, then the channel fading can be pre-compensated by a simple channel-inversion operation as $\mathbf{B}_k = \mathbf{H}_k^{-1}$. In this way, the signals transmitted by all users will be aggregated at the BS as follows:

$$\mathbf{y} = \sum_{k=0}^{K-1} \mathbf{x}_k, \quad (2)$$

which is equivalent to the computation of the summation of all \mathbf{x}_k s over the wireless channel.

Motivated by this property, the AirComp technique can be generalized to support a wide range of mathematical operations, which are based on the property of nomographic function in the following form:

$$f(s_0, s_1, \dots, s_{K-1}) = \psi \left(\sum_{k=0}^{K-1} \phi(s_k) \right), \quad (3)$$

where $\phi(\cdot)$ and $\psi(\cdot)$ denote pre- and post-processing functions, respectively.

Some well-known nomographic functions include:

- 1) *Arithmetic Mean*: $f(s_0, s_1, \dots, s_{K-1}) = \frac{1}{K} \sum_{k=0}^{K-1} s_k$, with $\phi(s_k) = s_k$ and $\psi(y) = y/N$;
- 2) *Euclidean Norm*: $f(s_0, s_1, \dots, s_{K-1}) = \sqrt{\sum_{k=0}^{K-1} s_k^2}$, with $\phi(s_k) = s_k^2$ and $\psi(y) = \sqrt{y}$;
- 3) *Number of Active Node*: $f(s_0, s_1, \dots, s_{K-1})$ is the number of active node, with $\phi(s_k) = 1$ (active) or 0 (inactive) and $\psi(y) = y$.

B. Nested Lattice Code

A lattice is an infinite discrete set of points in the euclidean space that are regularly arranged and are closed under addition [40]. As an important channel coding technique, the structural properties of nested lattice coding are well suited for multiple access channel in wireless networks, which allows multiple transmitters to effectively share the same radio resources and can protect against channel noise. Specifically, a d -dimensional lattice Λ in the euclidean space \mathbb{R}^d can be generated as follows:

$$\Lambda = \{\mathbf{G}\mathbf{u} : \mathbf{u} \in \mathbb{Z}^d\}, \quad (4)$$

where $\mathbf{G} = [\mathbf{g}_1, \mathbf{g}_2, \dots, \mathbf{g}_d]$ is a full-rank generator matrix.

A lattice Λ_C is *nested* in some lattice Λ_F if $\Lambda_C \subseteq \Lambda_F$, i.e., Λ_C is a sublattice of Λ_F . In this case, Λ_F is denoted as the fine lattice, which defines the codewords, while Λ_C is denoted as the coarse lattice, which is used for shaping. Specifically, the *nested lattice code* \mathcal{L} is the set of all points of a fine lattice Λ_F that is within the fundamental Voronoi region \mathcal{V}_C of a coarse lattice Λ_C :

$$\mathcal{L} = \Lambda_F \cap \mathcal{V}_C = \{\mathbf{x} : \mathbf{x} = \lambda \bmod \Lambda_C, \lambda \in \Lambda_F\}, \quad (5)$$

where the *fundamental Voronoi region*, \mathcal{V}_C , of the lattice Λ_C , is the set of all points in \mathbb{R}^d that are closest to the zero vector:

$$\mathcal{V}_C = \{\mathbf{z} : \|\mathbf{z}\| \leq \|\mathbf{z} - \lambda\|, \forall \lambda \in \Lambda_C, \mathbf{z} \in \mathbb{R}^d\}. \quad (6)$$

For each user k , a d -dimension nested lattice codeword $\mathbf{x}_k \in \mathcal{L}$ can be generated based on its hash value $s_k \in \mathbb{F}_p^l$ by the encoding function $\phi(\cdot)$ as follows:

$$\begin{aligned} \phi(s_k) : \mathbb{F}_p^l &\rightarrow \mathbb{R}^d \\ s_k &\rightarrow \mathbf{x}_k \end{aligned} \quad (7)$$

where $\mathbb{F}_p = \{0, 1, \dots, p-1\}$ forms a finite field under integer arithmetic modulo p . Then for a nested lattice codebook \mathcal{L} , the following property is held for all $\mathbf{x}_k \in \mathcal{L}$:

$$\left[\sum_{k=0}^{K-1} \mathbf{x}_k \bmod \Lambda_C \right] \in \mathcal{L}, \quad (8)$$

that is, the sum of lattice codewords modulo the shaping lattice is a codeword itself. Due to this linearity preserving characteristic, there exists a post-processing function $\psi(\cdot)$ that satisfies [23]:

$$\psi \left(\sum_{k=0}^{K-1} \phi(s_k) \bmod \Lambda_C \right) = \bigoplus_{k=0}^{K-1} s_k, \quad (9)$$

which is a nomographic function, that is, $f(s_0, s_1, \dots, s_{K-1}) = \bigoplus_{k=0}^{K-1} s_k$.

Based on this property, AirComp can be used to transmit and compute the superposition of hash values from all users. Specifically, each user k maps its hash value s_k to a lattice codeword \mathbf{x}_k using the mapping operation in (7), then all users can transmit their lattice codes simultaneously to the BS using the AirComp technique. At the BS, the received signal is $\mathbf{y} = \sum_{k=0}^{K-1} \mathbf{x}_k + \mathbf{w}$, which is a d -dimension linear combination of the lattice codes of all users with the N -dimensional noise vector $\mathbf{w} = [w_0, w_1, \dots, w_{N-1}]^T$.

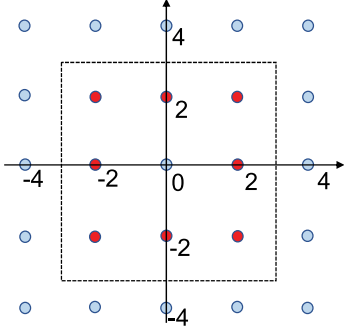


Fig. 4. Nested \mathbb{Z}^2 cubic lattice. The points are Λ_F points and the Voronoi region of the coarse lattice Λ_C is drawn in dashed lines. Only the eight codes (red points) in the outer tier of the Voronoi region are used.

In theory, there are many choices for the lattice codes with different codeword dimensions. However, in practice, the transmitted symbol can only represent information in a 2-dimensional space (in-phase and quadrature dimension). Therefore, for the high dimensional hash value, such as the 128-bits hash generated by the MD5 algorithm [41], it needs 2^{64} codewords at each dimension to represent the hash, which is impossible in a real communication system.

In this work, we consider an \mathbb{Z}^2 cubic lattice Λ_F as illustrated in Fig. 4, where the points are Λ_F points and the fundamental Voronoi region of the coarse lattice Λ_C is drawn in dashed lines. Only the eight codes in the outer tier of the Voronoi region are used (shown in red color), so the code rate is $B = \log_2 8 = 3$ bits/symbol. Therefore, for an L -bits hash value, it can be partitioned into $N = \lceil L/3 \rceil$ symbols.

C. Hash Consistency Verification

Based on the \mathbb{Z}^2 cubic lattice in Fig. 4, each user k can construct a hash symbol vector $\mathbf{x}_k = [x_{k0}, x_{k1}, \dots, x_{kN-1}]^T$, where x_{kn} is mapped to one of the eight codes. Then each element of \mathbf{x}_k is transmitted in an OFDM sub-carrier. The received symbol at the BS in sub-carrier n is:

$$y_n = \sum_{k=0}^{K-1} x_{kn} + w_n. \quad (10)$$

The received symbol at sub-carrier n can be quantized to the nearest point $t_n \in \mathbb{Z}^2$ in the Λ_F as follows:

$$t_n = \arg \min_{\lambda \in \Lambda_F} \|y_n - \lambda\|, \quad n \in [0, N-1], \quad (11)$$

which are then broadcasted to all users through the downlink channel. A vector $\mathbf{t} = [t_0, t_1, \dots, t_{N-1}]^T \in \mathbb{Z}^{2N}$ is constructed by each user, which is the complete linear aggregation of the lattice codes of all users.

Based on the received vector \mathbf{t} , we design a consensus algorithm by exploiting the geometric characteristics of lattice codes. Let C_m denote a set of m users that transmit the consistent hash symbol vector, and C_{K-m} denote the rest $K-m$ users that transmit different hash symbol vectors. Then \mathbf{t} can be re-written

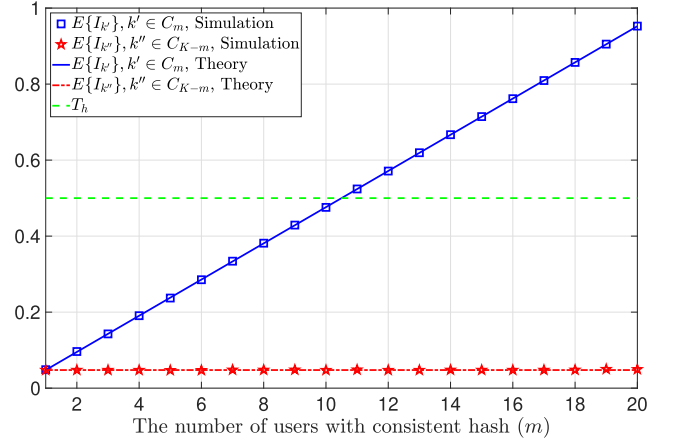


Fig. 5. $\mathbb{E}\{I_{k'}\}$ and $\mathbb{E}\{I_{k''}\}$ versus m ($K = 21$).

as follows:

$$\mathbf{t} = \sum_{k \in C_m} \mathbf{x}_k + \sum_{k \in C_{K-m}} \mathbf{x}_k + \mathbf{e}, \quad (12)$$

where \mathbf{e} is the random error pattern after symbols quantization in (11) with $\mathbb{E}\{\mathbf{e}\} = \mathbf{0}$.

We assume that the hash symbol vector elements are uniformly mapped from the lattice codewords as shown in Fig. 4. For each user $k \in C_m$, we assume the transmitted hash symbol vector is $\bar{\mathbf{x}}$, i.e., $\mathbf{x}_k = \bar{\mathbf{x}}, \forall k \in C_m$. For each user $k \in C_{K-m}$, we first consider the case that the hash symbol vector is independent of each other, i.e., these users are crash users. Then for $k \neq j$, we have

$$\mathbb{E}\{\mathbf{x}_k^T \mathbf{x}_j\} = \begin{cases} N\sigma_s^2, & \text{If } \forall k, j \in C_m \\ 0, & \text{Otherwise} \end{cases} \quad (13)$$

where σ_s^2 is the variance of codewords. Furthermore, we define I_k as the hash consistency factor (HCF) of user k , which is the normalized inner product of its hash symbol vector \mathbf{x}_k and vector \mathbf{t} , that is:

$$I_k = \frac{\mathbf{t}^T \mathbf{x}_k}{K|\mathbf{x}_k|^2}, \quad (14)$$

From (13), we can obtain the expectation of I_k as follows:

$$\mathbb{E}\{I_k\} = \begin{cases} \frac{m}{K}, & \forall k \in C_m \\ \frac{1}{K}, & \forall k \in C_{K-m} \end{cases} \quad (15)$$

In Fig. 5, we plot $\mathbb{E}\{I_k\}$ with 21 users ($K = 21$) for m varying from 1 to 20. It can be seen that for $m \geq 2$, $\mathbb{E}\{I_{k'}\} > \mathbb{E}\{I_{k''}\}, \forall k' \in C_m, k'' \in C_{K-m}$. Thus I_k can be used as a reliable metric for each user to determine whether its hash symbol vector is consistent with the majority of hash vectors in \mathbf{t} or not, that is, it belongs to one of the m users sending the same hash or not. Therefore, this metric can be used to check if the consensus is reached or not. Specifically, in the proposed AirCon protocol, the consensus can be reached if $m > \lfloor K/2 \rfloor$ users generate the same hash vector. According to (15), we can set a threshold as $T_h = 0.5$. After receiving the superposed hash vector \mathbf{t} , each user k calculates I_k as defined in (14). If $I_k \geq T_h$, the user

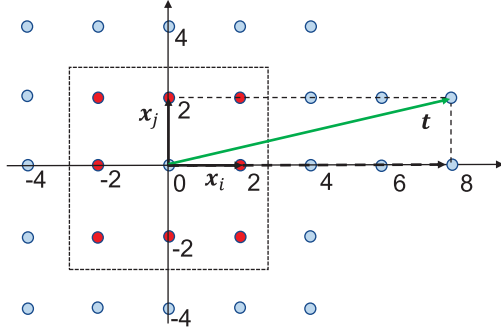


Fig. 6. An example of hash symbols aggregation. $\mathbf{x}_i = [2, 0]$, $\mathbf{x}_j = [0, 2]$, $\mathbf{t} = 4\mathbf{x}_i + \mathbf{x}_j = [8, 2]$. Then $I_i = 0.8$, $I_j = 0.2$, which suggests that the codeword \mathbf{x}_i is the majority in \mathbf{t} . The consensus can be achieved due to $I_i > 0.5$.

assumes that the consensus can be achieved and sends the result to the primary user.

For a better understanding, an example of hash symbols aggregation is shown in Fig. 6, where a linear combination $\mathbf{t} = 4\mathbf{x}_i + \mathbf{x}_j$ is formed by $\mathbf{x}_i = [2, 0]$ and $\mathbf{x}_j = [0, 2]$, which gives $\mathbf{t} = [8, 2]$. Then we can compute the HCF defined in (14) as $I_i = 0.8$ and $I_j = 0.2$, which suggests that the codeword \mathbf{x}_i is the majority in \mathbf{t} and the consensus can be achieved due to $I_i > 0.5$.

However, if some users are controlled by an attacker, the consensus performance will be degraded based on this ideal threshold. For instance, consider the case where m users are honest and the rest of $K - m$ users are Byzantine users controlled by a malicious attacker. The hash symbol vector transmitted by honest users is $\bar{\mathbf{x}}$, while the hash symbol vector transmitted by malicious users is $\hat{\mathbf{x}} = -\bar{\mathbf{x}}$, then the aggregated vector at the BS is $\mathbf{t} = m\bar{\mathbf{x}} + (K - m)\hat{\mathbf{x}} = (2m - K)\bar{\mathbf{x}}$. In this case, $I_k = (2m - K)/K$ for a honest user k , so the consensus cannot be achieved with a threshold of $T_h = 0.5$ unless $m > \lceil 3K/4 \rceil$. Therefore, this one-round consensus procedure is vulnerable to this kind of conspiracy attack.

We also give an example of hash symbols conspiracy attack in Fig. 7, where three honest users transmit hash symbol $\mathbf{x}_i = [2, -2]$ and two malicious users transmit hash symbol $\mathbf{x}_j = [-2, 2]$. Then the aggregated symbol $\mathbf{t} = 3\mathbf{x}_i + 2\mathbf{x}_j = [2, -2]$ and the HCF of honest users and malicious users are $I_i = 0.2$ and $I_j = -0.2$, respectively. As a result, the consensus cannot be achieved due to $I_i < 0.5$ and $I_j < 0.5$, although the number of honest users is more than half of the total number of users.

As a solution, we propose a two-round consensus procedure to improve the security level of the proposed consensus procedure. Specifically, in the first round, malicious users can be filtered by setting a proper threshold and only honest users enter the second round to decide the consensus results. In the following, we will analyze the optimal threshold for the first round and the corresponding fault tolerance of the proposed consensus protocol.

Before getting into the details, we first make some assumptions about the capabilities of an attacker:

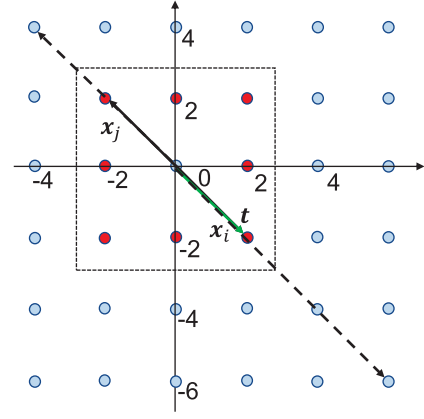


Fig. 7. An example for hash symbols conspiracy attack. $\mathbf{x}_i = [2, -2]$, $\mathbf{x}_j = [-2, 2]$, $\mathbf{t} = 3\mathbf{x}_i + 2\mathbf{x}_j = [2, -2]$. Then $I_i = 0.2$, $I_j = -0.2$. The consensus cannot be achieved due to $I_i < 0.5$ and $I_j < 0.5$.

- 1) AS_1 : An attacker knows the hash symbols transmitted by each honest user. This is feasible because the attacker can calculate the legal hash by acting as an honest user;
- 2) AS_2 : An attacker cannot manipulate the parameters of the physical layer, such as the transmitting power.
- 3) AS_3 : An attacker cannot manipulate the consensus protocol, such as the consensus threshold.

We assume that an attack is successful if the consensus cannot be achieved even if the number of honest users is more than half of the total number of users (i.e., $m > \lfloor K/2 \rfloor$).

Based on the aforementioned attacking model, we assume all honest users transmit $\bar{\mathbf{x}}$ and all malicious users transmit $\hat{\mathbf{x}}$. Then \mathbf{t} can be re-written as follows:

$$\mathbf{t} = m\bar{\mathbf{x}} + (K - m)\hat{\mathbf{x}} + \mathbf{e}, \quad (16)$$

and the expectation of I_k can be re-written by

$$\mathbb{E}\{I_k\} = \begin{cases} 1 - \alpha + \alpha\rho, & \forall k \in C_m \\ \alpha + (1 - \alpha)\rho, & \forall k \in C_{K-m} \end{cases} \quad (17)$$

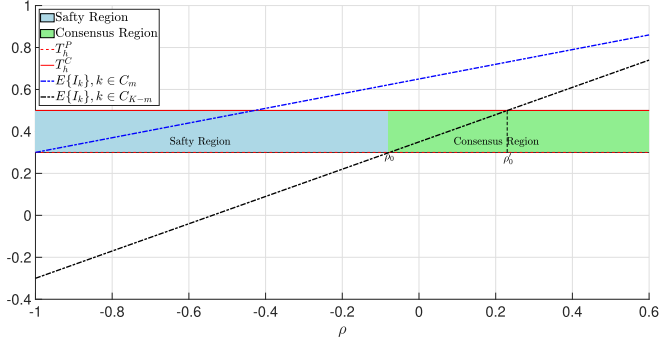
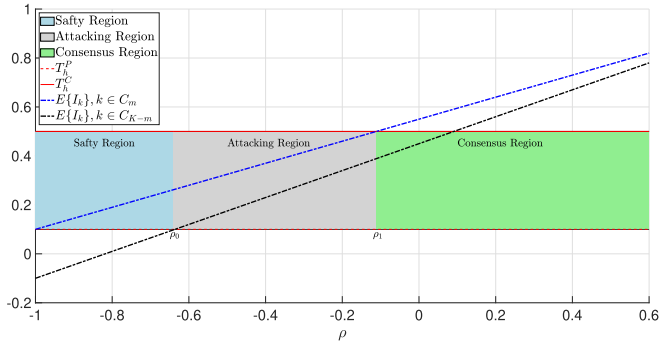
where $\alpha = \frac{K-m}{K}$ is the percentage of malicious users and $\rho = \mathbb{E}\{\frac{\bar{\mathbf{x}}^T \bar{\mathbf{x}}}{\|\bar{\mathbf{x}}\|^2}\} = \mathbb{E}\{\frac{\hat{\mathbf{x}}^T \hat{\mathbf{x}}}{\|\hat{\mathbf{x}}\|^2}\}$ denote the correlation coefficient between $\hat{\mathbf{x}}$ and $\bar{\mathbf{x}}$. The attacker can change I_k of the honest users by manipulating ρ between $[-1, 1]$ with properly setting of $\hat{\mathbf{x}}$.

In the first round of consensus (Prepare phase), a threshold should be set such that all honest users can get into the second round while the malicious users should be filtered as many as possible. To this end, the threshold should be set to

$$T_h^P = \min_{\rho \in [-1, 1]} (1 - \alpha + \alpha\rho) = 1 - 2\alpha. \quad (18)$$

From the attacker's perspective, it must find a ρ satisfying $\alpha + (1 - \alpha)\rho > T_h^P = 1 - 2\alpha$ such that malicious users also can get into the second round. Otherwise, all malicious users will be filtered in the first round. Therefore, the range of ρ can be given by

$$\rho > \frac{1 - 3\alpha}{1 - \alpha}. \quad (19)$$

Fig. 8. Consensus performance under fault tolerance, $\alpha = 0.35$.Fig. 9. Consensus performance over fault tolerance, $\alpha = 0.45$.

In the second round of consensus (Commit phase), if the number of honest users is more than half of the total number of users, the HCF I_k of honest users should be more than $T_h^C = 0.5$ such that the consensus can be achieved even if all malicious users satisfy the threshold in the first round of consensus by setting ρ as in (19), that is

$$I_k = 1 - \alpha + \alpha\rho = 1 - \alpha + \alpha \frac{1 - 3\alpha}{1 - \alpha} > \frac{1}{2}, \forall k \in C_m. \quad (20)$$

Then we can obtain the range of α as follows:

$$\alpha < \frac{\sqrt{17} - 1}{8} \approx 0.39 := \alpha^* \quad (21)$$

Substituting (21) into (18), we can obtain the optimal value of T_h^P as follows:

$$T_h^{P*} := 0.22. \quad (22)$$

The range of α in (21) specifies the fault tolerance of the two-round consensus protocol. If the percentage of malicious users does not exceed 0.39, then the consensus can be achieved in the second round. Otherwise, the consensus cannot be achieved even if $m > \lfloor K/2 \rfloor$.

We show two scenarios in Figs. 8 and 9. In Fig. 8, we plot the HCF for all users with $\alpha = 0.35$, which is under the threshold α^* . The range of ρ can be divided into two parts by the critical point ρ_0 that satisfies $\alpha + (1 - \alpha)\rho_0 = T_h^P = 1 - 2\alpha$. When $\rho < \rho_0$, the region is denoted as *Safety Region*, all malicious users will be filtered by T_h^P and cannot enter the second round. When $\rho > \rho_0$, the region is denoted as *Consensus Region*, malicious

users can enter the second round, but they cannot affect the consensus results because the HCF of all honest users always exceeds $T_h^C = 0.5$ such that all honest users can enter the Reply phase and send the Reply message to the primary user. Note that in the second round, if $\rho_0 < \rho < \rho'_0$, the HCF of malicious users is less than T_h^C such that they cannot send the Reply message. If $\rho > \rho'_0$, malicious users also send a Reply message. However, regardless of whether malicious users can send a Reply message or not, it will not affect the consensus result because all honest users always send the Reply message. Therefore, malicious users cannot affect consensus results whatever ρ is set when $\alpha = 0.35$.

In Fig. 9, we plot the HCF of all users with $\alpha = 0.45$, which is over the threshold α^* . In this case, we can divide the range of ρ into three parts by two critical point ρ_0 and ρ_1 , where ρ_0 satisfies $\alpha + (1 - \alpha)\rho_0 = T_h^P = 1 - 2\alpha$ and ρ_1 satisfies $1 - \alpha + \alpha\rho_1 = T_h^C = 0.5$. Similarly, when $\rho < \rho_0$, the consensus can be achieved because malicious users cannot enter the second round. When $\rho > \rho_1$, the consensus also can be achieved because the HCF of honest users always exceeds T_h^C . However, when $\rho_0 < \rho < \rho_1$, we denote it as *Attacking Region*, malicious users can enter the second round, and the HCF of all honest users does not exceed T_h^C , so the consensus cannot be achieved and the attack is successful in this region.

D. Two-Round Consensus Procedure

Based on the hash consistency verification scheme discussed in the previous subsection, we propose a two-round consensus procedure as follows.

In the first round (Prepare phase), all users send their hash symbols to the BS via the AirComp technique. The BS feeds back the aggregated hash symbol vector \mathbf{t} to all users. Each user k calculates the HCF I_k according to (14) and compares it with the threshold $T_h^P = 0.22$. If $I_k > T_h^P$, then the user changes to the prepared state and enters the second round of the consensus process.

In the second round (Commit phase), only users who are in the prepared state send the hash symbols to the BS via the AirComp technique. The BS feeds back the new aggregated hash symbol vector \mathbf{t}' to all users. Each prepared user k calculates the new HCF I'_k once again using \mathbf{t}' and compares with the threshold $T_h^C = 0.5$. Only those users with $I'_k > T_h^C$ enter the Reply phase and return the Reply message to the BS via the AirComp technique. The aggregated Reply message is forwarded to the primary user by the BS.

The final consensus result is determined by the primary user based on the aggregated Reply message from the BS. The primary user also can calculate the HCF in (14) between its Reply message and the aggregated Reply message. If the HCF exceeds $T_h^R = 0.5$, which suggests that more than half of the total users return the consistent Reply message, then the consensus is achieved.

For the example of conspiracy attack illustrated in Fig. 7, based on the threshold $T_h^P = 0.22$, malicious users will be filtered after the first round consensus because $I_j = -0.2 < T_h^P$. Therefore, only honest users can enter the second round consensus and the new aggregated hash symbols is $\mathbf{t}' = 3\mathbf{x}_i = [6, -6]$,

whose HCF is $I'_i = 0.6 > T_h^C$. Therefore, the consensus can be achieved after two round of consensus.

V. AIRCON IMPLEMENTATION BASED ON LTE SYSTEM

In this section, we consider the implementation of AirCon protocol based on the open-source srsLTE platform, which provides the standard LTE protocols. We present solutions to some of the critical issues for practical AirComp implementation. Specifically, two problems are considered for AirComp implementation: 1) Synchronization problem; 2) Uplink channel estimation and feedback problem.

A. Synchronization

The first challenging problem for AirComp implementation is to achieve strict timing/frequency synchronization across different users so that accurate signal superposition can be obtained at the BS.

Fortunately, the LTE system has a complete set of synchronization mechanisms that can meet the synchronization requirement of AirComp. Specifically, in the downlink of the LTE system, the BS broadcasts synchronization channels (including PSS and SSS) periodically. If a user wants to connect to the BS, the first step is to search for these two channels to get timing and frequency synchronization. As long as the user is connected to the BS, it will keep tracking the timing/frequency synchronization via PSS/SSS. The user also compensates for the frequency offset of uplink channels according to the estimated value from the downlink channels.

In the uplink, a “timing advance” mechanism is adopted by all users for timing synchronization with the BS. Specifically, when the user connects to the BS, the BS measures the propagation delay and calculates a timing offset, named “Timing Advance (TA)”, which is fed back to the corresponding user. Whenever the user transmits data to the BS, it should transmit at the time with an advance of TA so that the signals from different users arrive at the BS without timing offset.

In addition, the physical layer of the LTE system is based on the OFDM technique, which adopts a cyclic prefix (CP) to cope with the multipath interference problem. Due to the cyclic property, the CP can also be used for channel synchronization. That is, as long as the timing offset is within a CP, the impact of the timing offset can be treated as a channel phase shift and compensated as a part of channel fading.

In summary, the existing synchronization mechanisms in the LTE system are sufficient for the implementation of AirComp, so there is no need to design a dedicated AirComp synchronization scheme in an LTE system, which is part of the reason we choose the LTE system as the implementation platform of the AirCon protocol.

B. Channel Estimation and Feedback

The purpose of uplink channel estimation is to cope with the channel fading problem via pre-compensation on the user side. In practice, there are two different ways to obtain uplink CSI. One way is that the BS estimates the uplink channel and sends

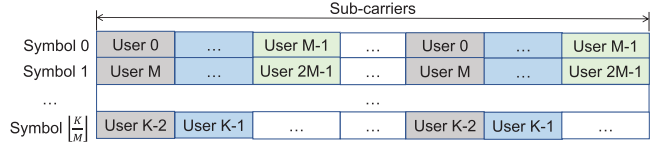


Fig. 10. Assignment of pilot symbols.

feedback to the corresponding user, the other way is that a user infers the uplink CSI from the downlink signals by leveraging the channel reciprocity of the TDD channel. In this work, we choose the first solution since it can be used for both FDD and TDD systems.

To get uplink CSI at the BS, a user needs to transmit pilot symbols in the uplink channel. Ideally, the pilot symbols should be transmitted at each sub-carrier so that the channel can be estimated accurately for all sub-carriers. However, this will consume too many channel resources and take a longer time for channel estimation for a large number of participating users. To this end, we exploit the property that the sub-carriers within coherence bandwidth may have a similar fading coefficient, so the pilot symbols are only needed for every M sub-carriers² (M is named as *Sharing Factor*). As a result, at most M users can share an OFDM symbol, therefore a total of $\lceil K/M \rceil$ OFDM symbols are needed for pilot symbol transmissions. Fig. 10 illustrates the assignment of the pilot symbols for different users.

Upon receiving the pilot symbols, the BS extracts the received symbols and estimates the CSI in the order of user index one by one. The simplest estimation method is the least square (LS) method. Taking user 0 as an example, the BS first constructs the received symbol vector $\mathbf{y} = [y_0, y_M, \dots, y_{\lfloor N/M \rfloor M}]$ from symbol 0, the corresponding transmitted pilot symbol for user 0 is $\mathbf{x} = [x_0, x_1, \dots, x_{\lfloor N/M \rfloor}]$, then the LS estimation for uplink channel knowledge of user 0 can be expressed as

$$\mathbf{h}_{\text{LS}} = \begin{bmatrix} \frac{y_0}{x_0}, \frac{y_M}{x_1}, \dots, \frac{y_{\lfloor N/M \rfloor M}}{x_{\lfloor N/M \rfloor}} \end{bmatrix}^T. \quad (23)$$

However, the LS method does not consider the noise in the pilot symbols. The LMMSE method [42] can be employed to further reduce the noise impact by utilizing the cross-correlation between sub-carriers, which is given by [42]:

$$\mathbf{h}_{\text{LMMSE}} = \mathbf{R}_{\text{hh}} \left(\mathbf{R}_{\text{hh}} + \frac{\beta}{\text{SNR}} \mathbf{I} \right)^{-1} \mathbf{h}_{\text{LS}}, \quad (24)$$

where $\beta = \mathbb{E}\{|x_n|^2\} \mathbb{E}\{|1/x_n|^2\}$ is a constant depending on the pilot symbol, $\mathbf{R}_{\text{hh}} = \mathbb{E}\{\mathbf{h}\mathbf{h}^\dagger\}$ is the channel autocorrelation matrix,³ \mathbf{I} is the identity matrix. The superscript $(\cdot)^\dagger$ denotes Hermitian transpose.

Based on the channel estimation h_{kn} for each subcarrier, the BS can set the coefficient of \mathbf{B}_k as $b_{kn} = h_{kn}^\dagger / |h_{kn}|^2$, which can compensate the channel fading and achieve the ideal signal

² M is a parameter depending on the coherence bandwidth. Note that this scheme is also adopted by the uplink sounding reference signal (SRS) in the LTE system.

³ The matrix \mathbf{R}_{hh} can be obtained from either a typical channel model [42] or the channel LS estimation.

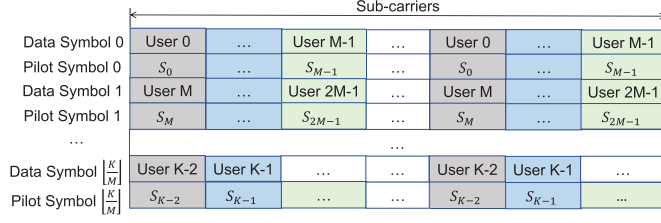


Fig. 11. Assignment of downlink feedback symbols.

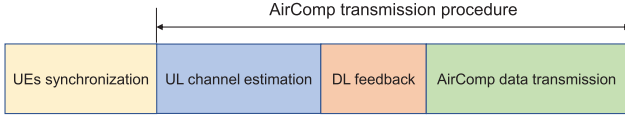


Fig. 12. AirComp signaling and data transmission procedures.

aggregation as shown in (1) and (2). This parameter should be fed back to the users as shown in Fig. 11. Similar to the uplink channel estimation, two OFDM symbols are shared by every M users, where the uplink CSI is transmitted in the *Data Symbol* and downlink pilot symbols are transmitted in the *Pilot Symbol* for downlink channel estimation. Therefore, totally $2\lceil K/M \rceil$ OFDM symbols are needed for downlink feedback transmissions. We still take user 0 as example: the coefficient vector $\mathbf{b}_0 = [b_{00}, b_{0M}, \dots, b_{0\lfloor N/M \rfloor M}]$ is transmitted at the corresponding sub-carrier in the *Data Symbol* 0. In the *Pilot Symbol* 0, the downlink pilot symbol s_0 is transmitted at the corresponding sub-carrier. After receiving these two symbols, user 0 first estimates the downlink channel using the received symbols from *Pilot Symbol* 0. Then the user can obtain the coefficient vector \mathbf{b}_0 from *Data Symbol* 0 by canceling the impact of downlink channel fading. Since each user only utilizes $1/M$ sub-carriers, the coefficients for the rest of sub-carriers can be estimated by interpolation [43], [44] at the users.

In the low-SNR scenario, in addition to the channel estimation algorithm, a retransmission scheme (i.e., the uplink pilot symbols and downlink feedback symbols are transmitted multiple times) can be adopted to further improve the performance. The impact of retransmission times on the consensus accuracy will be studied in Section VII.

C. AirCon Implementation

The transmission procedure of the AirComp protocol is summarized in Fig. 12, which consists of all procedures discussed in previous subsections. First, the synchronization between the users and the BS will be established. Then the BS estimates uplink CSI for all users based on the uplink pilot symbols. The coefficients of \mathbf{B}_k are computed based on the channel estimation results and fed back to all users. Finally, all users transmit data using the AirComp technique.

We implement the AirCon protocol based on the above AirComp procedures in the LTE system. The AirCon protocol is started when all users are in a stable state, such as the RRC IDLE state. In our implementation, we mainly focus on the feasibility of AirComp technology on hash verification of blockchain

consensus. Therefore, we simplify the AirCon protocol in our implementation. That is, only hash bits are generated in each user, rather than broadcasting an entire block from the primary user. In this case, we do not appoint a specific primary user. The consensus request is generated in the BS, which is a dedicated system information block (SIB) to notify all users that the BS is ready for consensus and users can transmit an uplink pilot symbol for AirComp uplink channel estimation. Since a user usually does not receive SIBs when it is in RRC IDLE state, the BS can set the *systemInfoModification* field in the paging message (all users must receive the paging message periodically) to ensure all users receive the SIB (i.e., the consensus request). Similarly, the consensus result is determined by the BS instead of the primary user in the Reply phase.

VI. KEY METRICS ANALYSIS AND COMPARISON BETWEEN THE AIRCON AND PBFT PROTOCOL

In this section, we analyze and compare the AirCon and PBFT protocol from the perspective of liveness, safety, resilience, and complexity.

A. Liveness

In the PBFT protocol, the view-change protocol provides liveness by allowing the system to make progress when the primary user fails [5]. In the blockchain consensus procedure, the primary user is in charge of candidate block generation, starting the consensus procedures in the Pre-prepare phase, and consensus result determination in the Reply phase. If the primary user behaves improperly, then other replica users can trigger the view-change protocol to re-select the primary user.

The view-change procedure can be triggered under the following situations: (i) The primary user broadcasts invalid transactions in the Pre-prepare phase, which can be detected by hash verification at each replica user. (ii) The primary user delivers an opposite consensus result in the Confirm stage. That is, if an honest user finds that the consensus result delivered by the primary user is opposite to the local result of the user, it can trigger the view-change procedure. (iii) The primary user does not broadcast the candidate block and consensus request after collecting enough transactions from the clients. A replica user can run a counter for transaction collections and a timer for consensus execution. The timer is started when the transactions counter exceeds a predefined threshold. When the timer expires in the current view, the replica user starts the view change to move the system to the next view.

Note that the proposed AirCon protocol mainly focuses on the revision of the Prepare and Commit phases in the traditional PBFT protocol to address the communication bottleneck problem, which hardly affects the view-change procedure of the PBFT protocol. Therefore, the liveness of the proposed AirCon protocol also can be guaranteed by the traditional view-change procedure in the PBFT protocol. Based on the valid information, such as view number, candidate block index, and received aggregated hash, similar view-change procedure as in the traditional PBFT protocol can be employed to re-select the primary user in the AirCon protocol when other users detect the improper

behaviors of the primary user. Due to space limitations, we do not provide the details of the view-change procedure in this paper.

As can be seen, the BS plays an important role in the AirCon protocol, if the BS is compromised by a cyber attack, the liveness of the consensus network will be destroyed. Therefore, we also need to design a BS-change procedure to solve the dilemma wherein the BS is not able to function properly (e.g., compromised by a cyber attack, or a sudden system crash).

In general, the target of cyber attack is to prevent the consensus from being reached in the case that honest users account for the majority. If the serving BS is controlled by the attacker, the normal consensus process can be corrupted by two kinds of behaviors: 1) The BS keeps silence when it needs to forward the aggregated signals. 2) The BS forwards the fake signals. The first behavior can be easily detected by the honest users via simply detecting the strength of the received signal. The second behavior also can be detected by the honest users via slightly complicated process. On the one hand, if the BS tampers with the broadcast messages from a single transmitter, it can be detected by honest users via checking the integrity of message that is protected by the digital signature of the transmitter. Specifically, these kinds of messages include the message from the clients in the Request stage, the messages from the primary user at the Pre-prepare phase and Confirm stage. On the other hand, if the BS tampers with the aggregated signals, the tampering behavior can be detected by checking the value of HCF I_k at each honest users, which is based on the fact that the HCF I_k of each honest user is always larger than the pre-defined threshold (i.e., T_h^P , T_h^C and T_h^R) in the corresponding phase, as long as the ratio of malicious users is under the fault tolerance, that is, $\alpha = 0.39$. If the HCF of honest users are under the corresponding threshold, they can determine that the BS is not able to function properly. Note that the message from the BS is always broadcast to all users, even if the message is only targeted for one specific user. Therefore, all honest users still can monitor the messages from the BS that are only for the primary user at the Request stage and Reply phase to determine whether the BS works well or not. Overall, all honest users can monitor the message from the BS at all stages and phases. The BS-change procedure can be triggered by honest users as long as they detect the unexpected behavior of the serving BS at any slot.

Considering the ultra-dense deployment of base stations in the next-generation networks, the consensus networks can maintain a candidate serving BS list. Once the BS-change procedure has been triggered, the consensus users can switch to the next candidate serving BS for the BS-change message transmission and the subsequent AirCon procedures. The choice of candidate serving BS from the list depends on comprehensive factors, such as the history behaviors of the BS, the current channel condition, etc. All in all, with effective BS-change procedures, the liveness of AirCon can be ensured even if the serving BS is compromised by a cyber attack and is unable to function properly.

B. Safety

The PBFT protocol ensures the safety property by guaranteeing that all non-faulty replicas agree on a total order for

the execution of requests despite failures [5]. Specifically, the safety is guaranteed from two aspects: (i) The assumption about the strength of message digests ensures that the probability that $m_0 \neq m_1$ and $Hash(m_0) = Hash(m_1)$ is negligible. (ii) After collecting enough matched Prepare messages from other replica users ($m \geq 2f$) at each non-faulty replica user, the protocol can guarantee that the non-faulty replica users will agree on a total order for the execution because non-faulty replica users would not send two conflicting Prepare messages in the same view. In the scenario of blockchain consensus, the safety property implies that honest users will obtain the same hash verification results for the same candidate block.

The safety of the AirCon protocol is guaranteed by the utilization of AirComp technology and downlink broadcast schemes. Specifically, in the uplink transmission, all users must transmit hash symbols in the same wireless channel resource, which naturally avoids the situation that a user transmits conflicting messages to different users. In the downlink, the BS broadcasts the aggregated hash symbols to all users. Therefore, each user will compute the HCF defined in (14) by the same t from the BS. Therefore, all honest users always obtain the same HCF, which guarantees that honest users will obtain the same hash verification results for the same candidate block.

C. Resilience

The resilience of the traditional PBFT protocol is $3f + 1$ [5], which implies that the maximum ratio of malicious users that can be tolerated by the system is 0.33. According to the obtained result in (21), the fault tolerance of the AirCon protocol is 0.39, which is slightly higher than the traditional PBFT protocol. However, considering that the view-change procedure of the AirCon is similar as the traditional PBFT protocol, the effective resilience of the AirCon protocol is also limited to $3f + 1$.

D. Complexity

We analyze the complexity of the consensus protocol from four aspects: communication overhead, computational complexity, memory consumption and energy consumption. As aforementioned, we mainly focus on the complexity in the *Prepare* and *Commit* phases.

1) *Communication Overhead*: The total number of messages in the Prepare and Commit phases of the traditional PBFT protocol is

$$\begin{aligned} \text{Num}_{\text{PBFT}}^{\text{message}} &= (K-1)(K-1) + K(K-1) \\ &= (2K-1)(K-1). \end{aligned} \quad (25)$$

Assuming each message consumes N wireless resource elements (REs), then the number of wireless REs consumed by the PBFT protocol is

$$\text{Num}_{\text{PBFT}}^{\text{REs}} = 2N \cdot \text{Num}_{\text{PBFT}}^{\text{message}} = 2N(2K-1)(K-1), \quad (26)$$

where the REs consumption in both the uplink and downlink directions is considered.

For the AirCon protocol, the total number of wireless REs consumed in the Prepare and Commit phases is

$$\text{Num}_{\text{AirCon}}^{\text{REs}} = 4N. \quad (27)$$

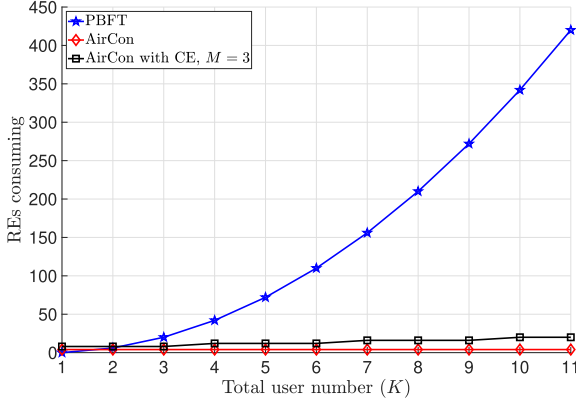


Fig. 13. Consensus protocol REs consuming, $N = 1$.

Therefore, the communication complexity of the traditional PBFT protocol is $\mathcal{O}(K^2)$, while the communication complexity of the AirCon protocol is $\mathcal{O}(1)$.

If the overhead of channel estimation (CE) and feedback are also considered, the extra overhead is $4\lceil K/M \rceil N$, which has a complexity of $\mathcal{O}(K)$. In this case, the total number of consumed REs is

$$\text{Num}_{\text{AirConCE}}^{\text{REs}} = 4N + 4\lceil K/M \rceil N. \quad (28)$$

Note that the complexity of channel estimation depends on the system. If channel reciprocity is utilized to estimate the uplink channel, the complexity of channel estimation can be reduced to $\mathcal{O}(1)$. The REs consumption in both consensus protocols under different users number is illustrated in Fig. 13.

2) *Computational Complexity*: As for the computational complexity, the hash verification procedures in the traditional PBFT protocol are carried out in the application layer. Therefore, demodulation, decoding in the physical layer are required. However, in our AirCon protocol, the consensus procedures are conducted in the physical layer, and the demodulation/decoding procedures are not required when using the proposed hash consistency verification method. Therefore, the computational complexity only comes from $\mathcal{O}(K)$ multiplication operations.

3) *Memory Consumption*: The memory consumption of the consensus protocol depends on the communication and computation schemes. Therefore, we analyze memory consumption from the perspective of communication and computation. (i) Communication memory consumption. For the convenience of analysis, we assume that each hash message consumes 1 memory unit. In the AirCon protocol, each user only receives the aggregated hash message of all users forwarded by the BS, so each user can allocate 1 memory unit to store the message. In the traditional PBFT consensus, each user determines the consensus result by decoding all messages from other users. Therefore, each user needs to store all messages. (ii) Computation memory consumption. It is not easy to quantitatively analyze the memory consumption of the traditional PBFT protocol because it depends on the particular demodulation and decoding schemes in the physical layer. However, from the perspective of qualitative analysis, according to practical experience, demodulation and

decoding procedures usually occupy a significant ratio of the total memory consumption in the physical layer procedures. On the contrary, the HCF calculation in (14) hardly consumes extra memory because it is a simple vector inner-product operation.

4) *Energy Consumption*: Similar to the memory consumption analysis, we analyze energy consumption from the perspective of communication and computation. (i) Communication energy consumption. The communication energy consumption depends on the communication times among consensus users. We assume each message transmission consumes 1 energy unit. Based on the communication overhead analysis in Section VI-D1, the energy consumption of the AirCon protocol is $\mathcal{O}(1)$ while the energy consumption of the PBFT protocol is $\mathcal{O}(K^2)$. (ii) We also cannot quantitatively analyze the energy consumption of the consensus protocol. From the perspective of qualitative analysis, the computational complexity also can reveal the level of energy consumption: the more complicated the computation is, the more energies are consumed.

In summary, the proposed AirCon protocol has a significant improvement from the four complexity metrics compared to the traditional PBFT protocol.

VII. PERFORMANCE EVALUATION

In this section, we show the implementation of the AirCon testbed and provide experimental results to demonstrate the performance under the real-world testbed. Due to the limitation of testbed conditions, we also present simulation results to evaluate the performance of the AirCon protocol under more general channel conditions.

A. AirCon Testbed Design

We implement the AirCon protocol based on srsLTE (version 20.04), an open-source LTE platform consisting of complete UE/eNodeB protocol stacks and a lightweight core network (CN) protocol stack. The software defines radio (SDR) board USRP B210 [45] is used to transmit and receive RF signals. Based on the srsLTE platform, the following modules are re-used for AirComp implementation: 1) user attaching procedures, which guarantees all users are synchronized to the eNodeB; 2) The OFDM waveform generation, which can generate waveform based on the LTE frame structure. We then develop all other modules related to the AirCon protocol, which include: 1) pilot symbols, hash bits generation, and lattice code modulation; 2) channel estimation module; 3) downlink feedback module; 4) hash consistency verification module and 5) AirCon protocol control module.

All devices of the testbed and setup of the test environment are shown in Fig. 14. Totally eight USRP boards are used in our experiment, where one board is used as BS and the other seven boards are used as users. All these USRP boards are driven by four PCs via USB3.0 interfaces.

For each experiment setup, the consensus procedures are repeated 1000 times. For a given number of m users with consistent hash symbols, we define consensus error ratio (CER(m)) as a

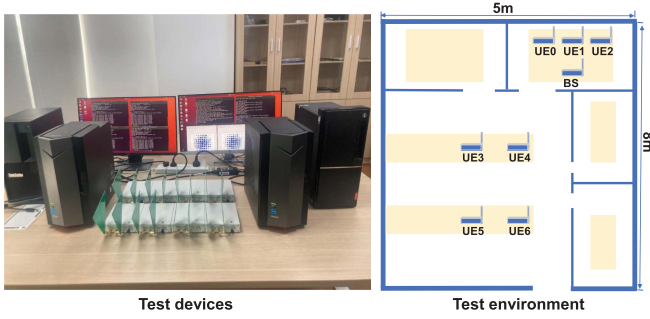


Fig. 14. Testbed devices and environment.

measure of consensus performance, which is given by:

$$\text{CER}(m) = \frac{\text{Number of consensus errors}}{\text{Total number of consensuses}}, \quad (29)$$

whereby a consensus error corresponds to one of the following two cases: 1) The consensus is achieved when $m \leq \lfloor \frac{K}{2} \rfloor$; 2) The consensus is not achieved when $m > \lfloor \frac{K}{2} \rfloor$.

B. Testbed Results

First, we demonstrate the symbol superposition results of the AirComp technique. As shown in Fig. 15(a), if all users transmit the same hash symbols, the received symbols at the BS are scaled with the number of users. On the contrary, as shown in Fig. 15(b), if all users transmit different hash symbols, the superposed symbols should be distributed on all lattice nodes randomly. From these two figures, it can be seen that the channel fading is effectively eliminated using the proposed AirComp implementation schemes (in particular the channel estimation, feedback, and pre-compensation schemes), and the desired channel superposition is achieved. Therefore, it is feasible to apply the AirComp technology in a digital communication system.

Second, we show the superposed hash symbols at the BS at different consensus phases. In Fig. 16, we show the case that all users transmit consistent hash symbols. It can be seen that all users can get into the Commit and Reply phases. In the second case, only four users transmit consistent hash symbols, while the other three users transmit random hash symbols. As shown in Fig. 17, these three users with random hash symbols are filtered in the Commit phase, and only four users with consistent hash symbols enter the Reply phase.

The performance of the AirCon protocol are shown in Figs. 18 and 19 with different users number (K). For $K = 5$, UE5 and UE6 in Fig. 14 are excluded in the experiments. It can be observed that when the number of users with consistent hash symbols (m) is close to the threshold for reaching a consensus ($m = 3$ for $K = 5$ and $m = 4$ for $K = 7$), it may lead to consensus error. However, even in the worst case, the consensus error is lower than 1%, which demonstrates that the proposed AirCon protocol is reliable in practice. By carefully examining the experimental results, we observe that the consensus error is mainly caused by inaccurate downlink feedback.

C. AirCon Simulation Setup

We further evaluate the performance of the proposed AirCon protocol under more general conditions using simulations. In the simulation, the frame structure is the same as the practical LTE system with a 1.92 MHz sampling rate. The channel gains for different users are generated independently.

In our simulation, three wireless channels are considered: 1) AWGN; 2) flat fading; 3) EPA (Extended Pedestrian A model), which is a multi-path channel model defined in the 3GPP TS 36.104 [46] for the typical pedestrian wireless environments. The path delay and the corresponding delay power of the EPA channel are shown in Table II. We introduce the average CER (ACER) as a metric to compare the performance difference among different simulation setups, which is given by

$$\text{ACER} = \frac{1}{K} \sum_{m=1}^K \text{CER}(m). \quad (30)$$

D. Simulation Results

In Fig. 20, we show the ACER of the AirCon protocol under different SNR conditions. First, it can be seen that the consensus performance is degraded among all types of channels when SNR decreases. Second, the performance gap between different types of channels is increasing in the low-SNR region, which suggests that the channel estimation accuracy loss is the dominant factor of the consensus performance loss in the low-SNR region. If the channel fading is well pre-compensated (e.g., an AWGN case), the average consensus error ratio of AirCon is about 1% even if the SNR is only 0 dB.

In the low-SNR region, the retransmission scheme can help improve consensus accuracy. In Fig. 21, we show the consensus performance improvement under different retransmission numbers. It can be observed that the consensus error ratio decreases gradually with the increase of the retransmission number. Note that the retransmission will consume more wireless resources. Therefore, the trade-off between retransmission number and consensus performance needs to be considered in practice.

We also study the influence of total user number (K) on consensus accuracy. Due to space limitation, we only show the AWGN channel case. It can be observed from Fig. 22 that the AirCon protocol performs better when K increases. The improvement of consensus performance comes from two aspects: 1) The consensus error only appears when m is around the threshold for reaching the consensus (namely, $m = \lceil \frac{K}{2} \rceil$ and $m = \lceil \frac{K}{2} \rceil + 1$ in our simulation). Therefore, the proportion of cases that will not incur consensus error increases when the total number of users increases. 2) Even in the cases that $m = \lceil \frac{K}{2} \rceil$ and $m = \lceil \frac{K}{2} \rceil + 1$, the consensus error ratio also decreases, which is illustrated in Fig. 23. That is because when more users participate in the AirComp process, the average effect of noise on each user in the superposed signal is reduced, so the consensus performance becomes better.

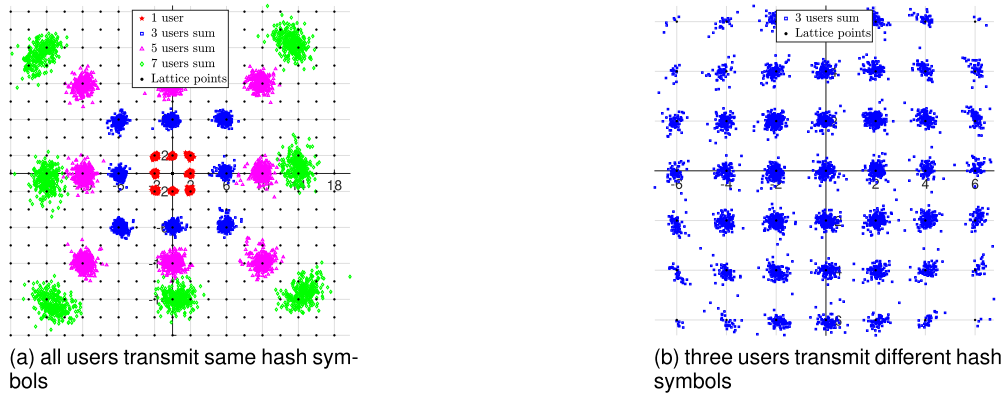


Fig. 15. Symbol superposition effects.

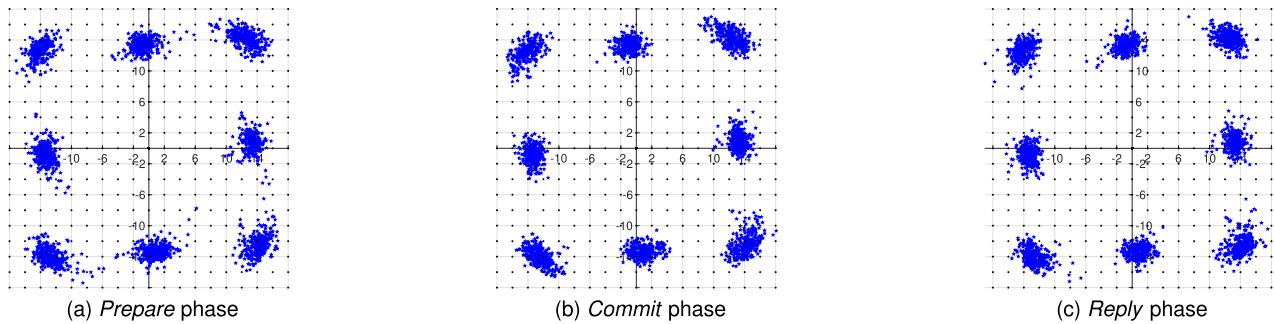


Fig. 16. Seven users transmit consistent hash symbols, all users determine the final consensus result.

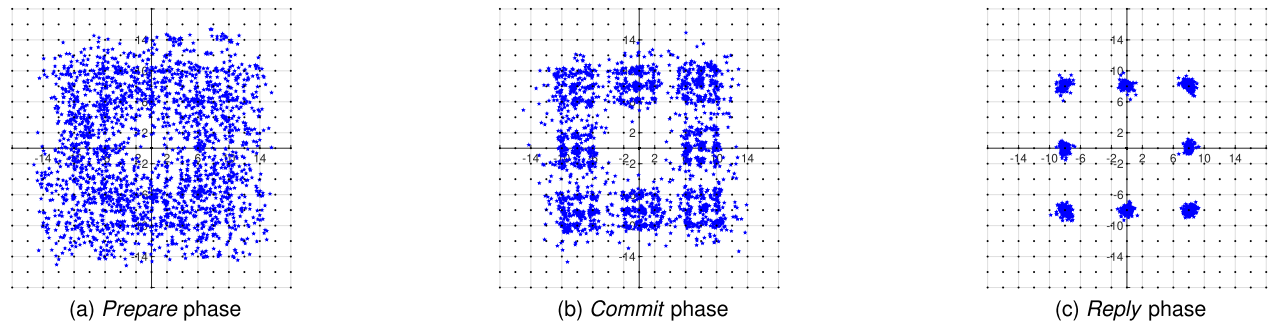


Fig. 17. Four users transmit consistent hash symbols, three users transmit random hash symbols, those users with random hash symbols are filtered.

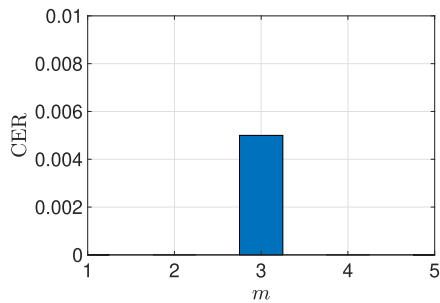


Fig. 18. Consensus error ratio for real testbed result with $K = 5$.

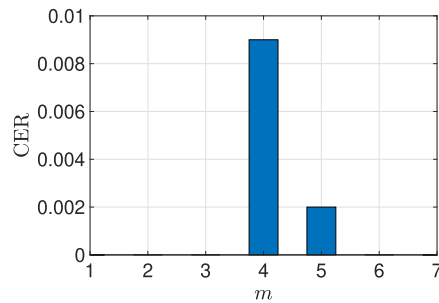


Fig. 19. Consensus error ratio for real testbed result with $K = 7$.

TABLE I
SYMBOL AND NOTATION DEFINITIONS

Symbols	Definitions
k	user index
n	sub-carrier index
m	the number of users that transmit the consistent hash
f	the maximum tolerable number of malicious users
α	the percentage of malicious users
ρ	the correlation between honest and malicious symbols
β	a constant for LMMSE channel estimation
w_n	the receiving noise at sub-carrier n at the BS
y_n	the received signal at sub-carrier n at the BS
t_n	the quantized (to fine lattice) signal at sub-carrier n
σ_s^2	variance of lattice codewords
x_{kn}	the transmitted codeword of user k at sub-carrier n
b_{kn}	the pre-processing factor of user k at sub-carrier n
h_{kn}	uplink channel fading of user k at sub-carrier n
K	total number of users
L	bits length of hash
N	total number of sub-carriers
B	lattice coding rate
Λ	lattice
\mathcal{L}	nested lattice code
\mathcal{V}_C	fundamental Voronoi region
Λ_C	coarse lattice code
Λ_F	fine lattice code
I_k	the hash consistency factor of user k
C_m	the set of users that transmit the consistent hash
C_{K-m}	the set of users that transmit different hash
T_h	threshold for hash verification
T_h^P	threshold for hash verification in the Prepare phase
T_h^C	threshold for hash verification in the Commit phase
T_h^R	threshold for hash verification in the Reply phase
\mathbb{R}^d	the $d \times 1$ real number set
\mathbb{Z}^d	the $d \times 1$ integer number set
\mathbb{C}^N	the $N \times 1$ complex number set
\mathbb{F}_p	the finite field under integer arithmetic modulo p
$\mathbb{C}^{N \times N}$	the $N \times N$ complex number set
\mathbf{e}	the BS receiving error after quantization
\mathbf{u}	integer vector
\mathbf{t}	the aggregated hash lattice in the Prepare phase
\mathbf{t}'	the aggregated hash lattice in the Commit phase
\mathbf{w}	the noise signal vector at the BS
\mathbf{y}	the received signal vector at the BS
s_k	hash value of user k
\mathbf{x}_k	the transmitted lattice vector of user k
$\bar{\mathbf{x}}$	the hash vector transmitted by honest users
$\hat{\mathbf{x}}$	the hash vector transmitted by malicious users
\mathbf{h}_{LS}	the LS estimation of uplink CSI
\mathbf{h}_{LMMSE}	the LMMSE estimation of uplink CSI
\mathbf{G}	the generator matrix of lattice
\mathbf{I}	the identity matrix
\mathbf{B}_k	the pre-processing matrix of user k
\mathbf{H}_k	the channel fading matrix of user k
\mathbf{R}_{hh}	the channel autocorrelation matrix
$(\cdot)^{-1}$	the inversion of the matrix
$(\cdot)^\dagger$	the Hermitian transpose of the matrix
$(\cdot)^T$	the transpose of the matrix
$\lceil \cdot \rceil$	to round up the argument inside
$\lfloor \cdot \rfloor$	to round down the argument inside
$\mathbb{E}\{\cdot\}$	the expectation of the argument inside
$\text{Diag}\{\cdot\}$	the diagonal matrix
$\text{mod}\Lambda$	modulo operation on lattice Λ

TABLE II
PARAMETERS OF EPA CHANNEL

Delay (ns)	0	30	70	90	110	190	410
Power (dB)	0	-1	-2	-3	-8	-17.2	-20.8

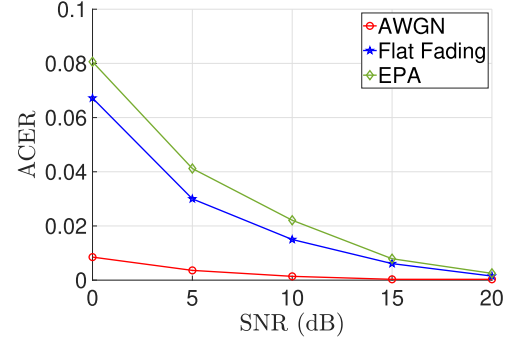


Fig. 20. Consensus error ratio under different SNR, $K = 11$.

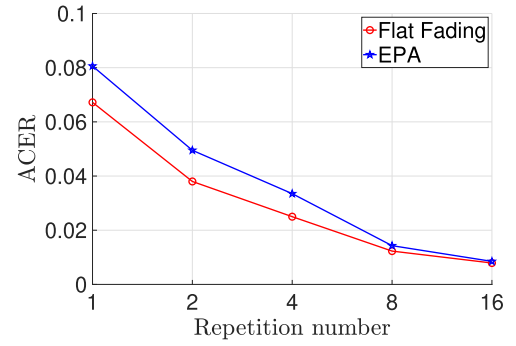


Fig. 21. Consensus error ratio under different repetition number, SNR = 0 dB.

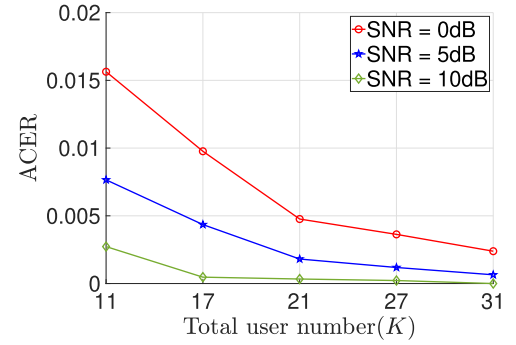


Fig. 22. Averaging consensus error ratio under different K , AWGN channel.

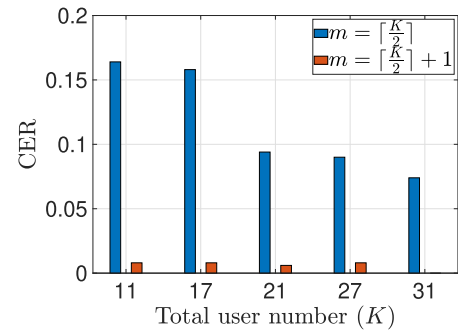


Fig. 23. Consensus error ratio around threshold under different K , SNR = 0 dB.

VIII. CONCLUSION

In this paper, we have proposed the AirCon to achieve low complexity consensus for blockchain-enabled wireless networks, which is novel in that the hash symbols of all users are transmitted to the BS simultaneously over the same wireless spectrum via the AirComp and lattice coding techniques, and the consensus can be done in the physical layer without decoding the hash symbols. We have shown that the AirCon protocol can significantly reduce the transmission and computational complexity during the consensus procedure. We have shown the design of AirCon based on the universal LTE system and provided solutions for the critical issues involved in the AirComp implementation, such as channel estimation and feedback. We also have implemented AirCon based on a srsLTE testbed, and real-world measurement results demonstrate that the proposed AirCon protocol is feasible in a practical LTE system. In addition, we have also conducted extensive simulation experiments and shown the accuracy and robustness of the proposed consensus protocol under different network conditions. The proposed AirCon is inspiring for designing efficient consensus schemes over wireless networks, where a more efficient channel status acquiring scheme can bring significant performance gain for AirCon.

REFERENCES

- [1] J. Wang, X. Ling, Y. Le, Y. Huang, and X. You, "Blockchain-enabled wireless communications: A new paradigm towards 6G," *Nat. Sci. Rev.*, vol. 8, no. 9, 2021, Art. no. nwab069.
- [2] L. Zhang, H. Xu, O. Onireti, M. A. Imran, and B. Cao, "How much communication resource is needed to run a wireless blockchain network?," *IEEE Netw.*, vol. 36, no. 1, pp. 128–135, Jan./Feb. 2022.
- [3] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," Oct. 2008. [Online]. Available: <http://bitcoin.org/bitcoin.pdf>
- [4] P. Vasin, "Blackcoin's proof-of-stake protocol v2," 2014. [Online]. Available: <http://blackcoin.org/blackcoin-pos-protocol-v2-whitepaper.pdf>
- [5] M. Castro and B. Liskov, "Practical Byzantine fault tolerance," in *Proc. 3rd Symp. Operating Syst. Des. Implementation*, 1999, pp. 173–186.
- [6] D. Ongaro and J. Ousterhout, "In search of an understandable consensus algorithm," in *Proc. USENIX Annu. Tech. Conf.*, 2014, pp. 305–320.
- [7] G. T. Nguyen and K. Kim, "A survey about consensus algorithms used in blockchain," *J. Inf. Process. Syst.*, vol. 14, no. 1, pp. 101–128, 2018.
- [8] B. Nazer and M. Gastpar, "Computation over multiple-access channels," *IEEE Trans. Inf. Theory*, vol. 53, no. 10, pp. 3498–3516, Oct. 2007.
- [9] Open source SDR LTE software suite. Accessed: Apr. 18, 2022. [Online]. Available: <http://github.com/srsRAN/>
- [10] K. Christidis and M. Devetsikiotis, "Blockchains and smart contracts for the Internet of Things," *IEEE Access*, vol. 4, pp. 2292–2303, 2016.
- [11] V. Ortega, F. Bouchmal, and J. F. Monserrat, "Trusted 5G vehicular networks: Blockchains and content-centric networking," *IEEE Veh. Technol. Mag.*, vol. 13, no. 2, pp. 121–127, Jun. 2018.
- [12] X. Ling, J. Wang, T. Bouchoucha, B. C. Levy, and Z. Ding, "Blockchain radio access network (B-RAN): Towards decentralized secure radio access paradigm," *IEEE Access*, vol. 7, pp. 9714–9723, 2019.
- [13] Y. Le, X. Ling, J. Wang, and Z. Ding, "Prototype design and test of blockchain radio access network," in *Proc. IEEE Int. Conf. Commun. Workshops*, 2019, pp. 1–6.
- [14] X. Ling, Y. Le, J. Wang, and Z. Ding, "Hash access: Trustworthy grant-free IoT access enabled by blockchain radio access networks," *IEEE Netw.*, vol. 34, no. 1, pp. 54–61, Jan./Feb. 2020.
- [15] B. Zhang, X. Ling, Y. Le, J. Wang, C. Cai, and Z. Tang, "Analysis and evaluation of hash access for blockchain radio access networks," in *Proc. Int. Conf. Wirel. Commun. Signal Process.*, 2020, pp. 62–67.
- [16] X. Ling, Y. Le, J. Wang, Z. Ding, and X. Gao, "Practical modeling and analysis of blockchain radio access network," *IEEE Trans. Commun.*, vol. 69, no. 2, pp. 1021–1037, Feb. 2021.
- [17] S. Guo, Y. Dai, S. Guo, X. Qiu, and F. Qi, "Blockchain meets edge computing: Stackelberg game and double auction based task offloading for mobile blockchain," *IEEE Trans. Veh. Technol.*, vol. 69, no. 5, pp. 5549–5561, May 2020.
- [18] Y. Liu, K. Wang, Y. Lin, and W. Xu, "LightChain: A lightweight blockchain system for Industrial Internet of Things," *IEEE Trans. Ind. Informat.*, vol. 15, no. 6, pp. 3571–3581, Jun. 2019.
- [19] H. Wei, W. Feng, Y. Chen, C.-X. Wang, and N. Ge, "Rethinking blockchains in the Internet of Things era from a wireless communication perspective," *IEEE Netw.*, vol. 34, no. 6, pp. 24–30, Nov./Dec. 2020.
- [20] B. Cao, M. Li, L. Zhang, Y. Li, and M. Peng, "How does CSMA/CA affect the performance and security in wireless blockchain networks," *IEEE Trans. Ind. Inform.*, vol. 16, no. 6, pp. 4270–4280, Jun. 2020.
- [21] Y. Sun, L. Zhang, G. Feng, B. Yang, B. Cao, and M. A. Imran, "Blockchain-enabled wireless Internet of Things: Performance analysis and optimal communication node deployment," *IEEE Internet of Things J.*, vol. 6, no. 3, pp. 5791–5802, Jun. 2019.
- [22] R. Zamir, "Lattices are everywhere," in *Proc. Inf. Theory Appl. Workshop*, 2009, pp. 392–421.
- [23] B. Nazer and M. Gastpar, "Compute-and-forward: Harnessing interference through structured codes," *IEEE Trans. Inf. Theory*, vol. 57, no. 10, pp. 6463–6486, Oct. 2011.
- [24] M. Goldenbaum, H. Boche, and S. Stańczak, "Nomographic functions: Efficient computation in clustered Gaussian sensor networks," *IEEE Trans. Wireless Commun.*, vol. 14, no. 4, pp. 2093–2105, Apr. 2015.
- [25] S.-W. Jeon and B. C. Jung, "Opportunistic function computation for wireless sensor networks," *IEEE Trans. Wireless Commun.*, vol. 15, no. 6, pp. 4045–4059, Jun. 2016.
- [26] F. Wu, L. Chen, N. Zhao, Y. Chen, F. R. Yu, and G. Wei, "Computation over wide-band multi-access channels: Achievable rates through sub-function allocation," *IEEE Trans. Wireless Commun.*, vol. 18, no. 7, pp. 3713–3725, Jul. 2019.
- [27] L. Chen, N. Zhao, Y. Chen, X. Qin, and F. R. Yu, "Computation over MAC: Achievable function rate maximization in wireless networks," *IEEE Trans. Commun.*, vol. 68, no. 9, pp. 5446–5459, Sep. 2020.
- [28] L. Chen, N. Zhao, Y. Chen, F. R. Yu, and G. Wei, "Toward optimal rate-delay tradeoff for computation over multiple access channel," *IEEE Trans. Commun.*, vol. 69, no. 7, pp. 4335–4346, Jul. 2021.
- [29] S. Sigg, P. Jakimovski, and M. Beigl, "Calculation of functions on the RF-channel for IoT," in *Proc. IEEE 3rd Int. Conf. Internet Things*, 2012, pp. 107–113.
- [30] A. Kortke, M. Goldenbaum, and S. Stańczak, "Analog computation over the wireless channel: A proof of concept," in *Proc. IEEE SENSORS Conf.*, 2014, pp. 1224–1227.
- [31] K. Yang, T. Jiang, Y. Shi, and Z. Ding, "Federated learning via over-the-air computation," *IEEE Trans. Wireless Commun.*, vol. 19, no. 3, pp. 2022–2035, Mar. 2020.
- [32] G. Zhu, Y. Du, D. Gündüz, and K. Huang, "One-bit over-the-air aggregation for communication-efficient federated edge learning: Design and convergence analysis," *IEEE Trans. Wireless Commun.*, vol. 20, no. 3, pp. 2120–2135, Mar. 2021.
- [33] M. Mohammadi Amiri and D. Gündüz, "Machine learning at the wireless edge: Distributed stochastic gradient descent over-the-air," *IEEE Trans. Signal Process.*, vol. 68, pp. 2155–2169, 2020.
- [34] G. Zhu, J. Xu, K. Huang, and S. Cui, "Over-the-air computing for wireless data aggregation in massive IoT," *IEEE Wireless Commun.*, vol. 28, no. 4, pp. 57–65, Aug. 2021.
- [35] F. Molinari, S. Stanczak, and J. Rausch, "Exploiting the superposition property of wireless communication for average consensus problems in multi-agent systems," in *Proc. Eur. Control Conf.*, 2018, pp. 1766–1772.
- [36] F. Molinari, N. Agrawal, S. Stańczak, and J. Rausch, "Max-consensus over fading wireless channels," *IEEE Control Netw. Syst.*, vol. 8, no. 2, pp. 791–802, Jun. 2021.
- [37] O. Abari, H. Rahul, and D. Katabi, "Over-the-air function computation in sensor networks," 2016. [Online]. Available: <http://arxiv.org/abs/1612.02307>
- [38] O. Abari, H. Rahul, D. Katabi, and M. Pant, "AirShare: Distributed coherent transmission made seamless," in *Proc. IEEE Conf. Comput. Commun.*, 2015, pp. 1742–1750.
- [39] U. Altun, S. T. Başaran, H. Alakoca, and G. K. Kurt, "A testbed based verification of joint communication and computation systems," in *Proc. 25th Telecommun. Forum*, 2017, pp. 1–4.
- [40] L. Natarajan, Y. Hong, and E. Viterbo, "Lattice codes achieve the capacity of common message Gaussian broadcast channels with coded side information," *IEEE Trans. Inf. Theory*, vol. 64, no. 3, pp. 1481–1496, Mar. 2018.

- [41] R. L. Rivest, "The MD5 message-digest algorithm," RFC 1321, Apr. 1, 1992, doi: [10.17487/RFC1321](https://doi.org/10.17487/RFC1321).
- [42] O. Edfors, M. Sandell, J.-J. van de Beek, S. Wilson, and P. Borjesson, "OFDM channel estimation by singular value decomposition," *IEEE Trans. Commun.*, vol. 46, no. 7, pp. 931–939, Jul. 1998.
- [43] Y. Zhao and A. Huang, "A novel channel estimation method for OFDM mobile communication systems based on pilot signals and transform-domain processing," in *Proc. IEEE 47th Veh. Technol. Conf. Technol. Motion*, 1997, pp. 2089–2093.
- [44] X. Dong, W.-S. Lu, and A. C. Soong, "Linear interpolation in pilot symbol assisted channel estimation for OFDM," *IEEE Trans. Wireless Commun.*, vol. 6, no. 5, pp. 1910–1920, May 2007.
- [45] Ettus research USRP B210. Accessed: Apr. 18, 2022. [Online]. Available: <http://www.ettus.com/all-products/UB210-KIT/>
- [46] 3GPP, "Evolved universal terrestrial radio access (E-UTRA); base station (BS) radio transmission and reception," 3rd Generation Partnership Project (3GPP), Technical Specification (TS) 36.104, Mar. 28, 2017, version 14.3.0. [Online]. Available: <https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=2412>

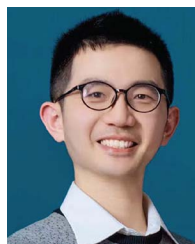


Xin Xie received the BEng degree in communication engineering from Tongji University, in 2014, and the MEng degree in communication and information system from the University of Chinese Academy of Sciences, Shanghai, China, in 2017. He is currently working toward the PhD degree with the School of Cyber Science and Engineering, Shanghai Jiao Tong University, Shanghai. His research interests include advanced wireless techniques, security, and privacy for the wireless systems.



Cunqing Hua (Member, IEEE) received the BEng degree in electronics engineering from the University of Science and Technology of China, in 2000, and the MPhil and PhD degrees in information engineering from the Chinese University of Hong Kong, in 2002 and 2006, respectively. He is currently a professor with the School of Cyber Science and Engineering, Shanghai Jiao Tong University, P. R. China. He joined Shanghai Jiao Tong University in July 2011. From September 2006 to December 2008, he was a postdoctoral research fellow with the University of

Houston, and was with the Department of Information Science and Electronic Engineering, Zhejiang University, China, as an associate professor from December 2008 to June 2011. His research interests include the general area of wireless networking and mobile communication, in particular focusing on network architecture design for 5G/B5G systems and integrated satellite terrestrial networks, advance wireless techniques (MIMO, Beamforming, network coding, etc.), Security, privacy and trust issues for wireless systems. He served on the technical program committees of the international conferences including INFOCOM, ICC, Globecom, WCNC etc. He is currently an editor of the *Wireless Networks Journal*.



Jianan Hong received the PhD degree from the Department of Electronic Engineering and Information Science (EEIS), USTC, in 2018. From 2018 to 2021, he was a research engineer at Huawei Shanghai Research Institute, Shanghai, China. He is currently an assistant researcher with the School of Cyber Science and Engineering, Shanghai Jiao Tong University, Shanghai. His research interests include secure cloud computing and privacy preserving authentication.



Pengwenlong Gu (Member, IEEE) received the BS degree from the Beijing University of Posts and Telecommunications (BUPT), Beijing, P. R. China, in 2010, and the master's and PhD degrees from TELECOM ParisTech, Paris, France, in 2013 and 2018, respectively. He is currently a post-doctoral researcher with Department INFRES, TELECOM Paris, IP Paris. His research interests include vehicular communication systems, anti-jamming in wireless networks, beamforming and its applications, optimization theory, and intrusion detection in wireless networks.



Wenchao Xu (Member, IEEE) received the BE and ME degrees from Zhejiang University, Hangzhou, China, in 2008 and 2011, respectively, and the PhD degree from the University of Waterloo, Canada, in 2018. He is a research assistant professor with the Hong Kong Polytechnic University. In 2011, he joined Alcatel Lucent Shanghai Bell Company Ltd., where he was a software engineer for telecom virtualization. He has also been an assistant professor with the School of Computing and Information Sciences, Caritas Institute of Higher Education, Hong Kong.

His research interests include wireless communication, Internet of Things, distributed computing, and AI enabled networking.