Byzantine Fault Tolerant Consensus in Open Wireless Networks via an Abstract MAC Layer

Guanlin Jing, Yifei Zou, *Member, IEEE,* Zuyuan Zhang, Dongxiao Yu, *Senior Member, IEEE,* Falko Dressler, *Fellow, IEEE* and Xiuzhen Cheng, *Fellow, IEEE*

***Abstract*—The openness of wireless networks opens the door to Byzantine attacks on the physical channels, making the communications unreliable and resulting in more challenges in achieving consensus among mobile devices. To address this issue, this paper studies the Byzantine-fault-tolerant (BFT) consensus problem based on an unreliable Byzantine communication model. Different from the previous works requiring stable communications between the honest nodes, considering the unreliable communication makes our problem more realistic but also harder. Based on the unreliable communication model, we first implement a BFT abstract MAC (absMAC) layer with a distributed and randomized multi-channel communication algorithm. In the implemented absMAC layer, its acknowledgement and progress operations can be completed within $O(\frac{kn}{k-f}\log n)$ and $O(\frac{k}{k-f}\log n)$ rounds, respectively. n, f, and k are the numbers of nodes, Byzantine nodes, and channels, respectively. With the implemented absMAC layer, an efficient and elegant BFT consensus algorithm is designed, which can solve the binary consensus problem within $O(\frac{kn}{k-f}\log n)$ rounds in expectation. Even though a series of works have discussed how to achieve consensus with a specific absMAC layer provided, to the best of our knowledge, this paper is the first one that implements a BFT absMAC layer.**

***Index Terms*—Fault tolerant consensus; Open wireless networks; Byzantine-resilience; Abstract MAC layer**

I. Introduction

The openness of wireless technology ensures a convenient network connection for the massive Internet-of-Things (IoT) devices and enables a flexible exchange of data and services among different objects and various applications. Due to its openness feature, the open wireless network can be regarded as a foundation of the Internet of Everything and has attracted lots of attention from the wireless communities. A recent concept based on the open wireless network is the Open Radio Access Network (O-RAN) [1], which aims to drive innovation and competition in wireless networks by promoting greater diversity in network devices and services through open and standardized wireless network interfaces. Even though the openness of wireless networks allows for more flexibility and innovation in the design of network architecture and applications. This also presents a hurdle in reaching consensus amidst Byzantine attacks, which potentially undermine the network's dependability and security [2].

Specifically, the Byzantine fault tolerant (BFT) consensus is very important for the IoT devices in the wireless networks [3], [4], because most of them are organized in a decentralized pattern and face malicious attacks from the Byzantine agents. Only with a BFT consensus, can the honest devices achieve a series of reliable and safety agreements for their cooperation in the network services and application. Due to its significance, there have been some BFT protocols presented in the past decades, such as [5]–[7] with reliable communications, [8]–[11] on the message loss model, [12], [13] on the communication failure model, [14] on the communication collision model, [15]–[18] on the heard-of model, and [2], [19] on the dynamic omission failures model. Whereas, most of them assume a non-Byzantine physical channel and focus on Byzantine attacks from the protocol layer and information layer, which are higher than the physical layer.[1] Compared with the previous works that rely on stable communications between honest devices, the BFT consensus problem in an open wireless network is harder and more complex due to the following reasons. First of all, with a loss permission scheme in the open wireless network, the physical channel is exposed to the Byzantine nodes. By breaking the physical channel (e.g. with a jamming signal), the Byzantine nodes disrupt the communications between honest devices and mislead the consensus process. Therefore, to solve the BFT consensus problem in an open wireless network, the Byzantine attacks from physical, protocol, and information layers should be carefully considered, which makes the BFT problem harder than that in the previous works with reliable communication environments. Furthermore, such a cross-layer consideration increases the complexity of solving the BFT consensus problem. In detail, the algorithm design of BFT consensus has to start from the unreliable physical layer, cross the protocol layer, and finally reach a consensus on the information layer. When analyzing the reliability and security of the whole algorithm, the unstable communications in the physical layer, Byzantine violation in the protocol layer, and fake messages in the information layer increase the complexity of reaching a theoretical result with rigorous proof, especially in some worst cases.

Guanlin Jing, Yifei Zou (Corresponding author), Dongxiao Yu and Xiuzhen Cheng are with Institute of Intelligent Computing, the School of Computer Science and Technology, Shandong University, Qingdao, 266200, P.R. China. E-mail: 202120681@mail.sdu.edu.cn, {yfzou, dxyu, xzcheng}@sdu.edu.cn.

Zuyuan Zhang is with the School of Engineering and Applied Science, George Washington University, D.C.,20052, U.S. E-mail: zuyuan.zhang@email.gwu.edu .

F. Dressler is with the School of Electrical Engineering and Computer Science, TU Berlin, Berlin, 10587, Germany. E-mail: dressler@ccs-labs.org .

Manuscript received XX XX, XXXX; revised XX XX, XXXX.

[1]In the protocol layer, the Byzantine devices can violate the consensus protocols they are specifically designed for. In the information layer, Byzantine agents can declare fake information in their messages.

Some examples of Byzantine attacks in multiple layers are presented in the following.

- **Jamming Attack in Physical Layer.** In [20], [21], by sending a jamming signal with sufficient large transmission power in the physical channel, an adversarial device can prevent legitimate communications from being decoded. An energy budget is assumed on the adversarial device so that it can only jam the channel in a fraction of rounds for each time window.
- **Double-Spending Sybil Attack in Protocol Layer.** A double-spending sybil attack is proposed in the Bitcoin decentralized network [22]. In [22], the attacker first broadcasts a transaction to the network. Before the first transaction is confirmed, the attacker broadcasts another transaction using the same digital currency to a different node. If the attacker can control a significant portion of the network (e.g., through a 51% attack), they can create a longer chain that includes the second transaction, making it a valid transaction while invalidating the first. By violating the protocol, the attacker misleads the honest nodes in the Bitcoin decentralized network
- **False Data Attack in Information Layer.** In an information system, Byzantine nodes can transmit false data to the fusion center, which can lead to incorrect information aggregation, thereby degrading the overall detection performance and potentially causing the system to make erroneous decisions [23].

To design an efficient and elegant consensus algorithm against Byzantine attacks from multiple layers, a BFT abstract MAC layer based on the multi-channel technique is implemented in this paper. Specifically, the concept of abstract MAC (absMAC) layer was first proposed by Kuhn et al. [24], which expresses key guarantees of real MAC layers with respect to the local broadcast operation. In an abstract MAC layer, two message delivery operations are provided with the latency bounds: the *acknowledgment* operation and the *progress* operation. In the acknowledgment (ack. for short), a node has its message received by all its neighbors[2]. In the progress (prog. for short), a node receives one message from its neighbors. The f_{ack} and f_{prog} are the time bounds for the ack. and prog. operations, respectively. With the concept of abstract MAC layer, the BFT consensus problem in open wireless networks can be divided into two independent and manageable components: (1) to implement the BFT absMAC layer over a physical network, and (2) to solve the consensus problem based on ack. and prog. operations provided by the absMAC layer, as illustrated in Fig 1.

Implementing the BFT absMAC layer in the open wireless networks is not an easy task since the Byzantine nodes can arbitrarily disrupt the communications on the physical channels and arbitrarily violate the protocol specifically designed for the honest nodes. Fortunately, the multi-channel technique can help against those Byzantine behaviors. By dividing the shared frequency spectrum into sufficient non-overlapping channels and assuming that each of the Byzantine nodes and honest

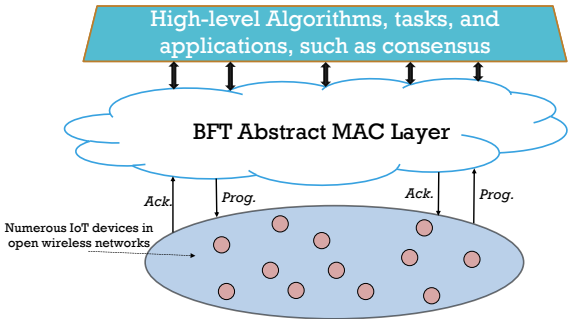


Fig. 1: Abstract MAC layer can simply the algorithm design for the consensus problem over open wireless networks.

nodes can only access a single channel in each transmission[3], a distributed and randomized multi-channel communication algorithm is designed to implement a BFT absMAC layer. Different from the existing channel hopping technique in which the transmitters and receivers synchronously jump to a clean channel, in this paper, our absMAC layer algorithm focuses on how to find and use the residual clean channels hidden by the byzantine nodes, to guarantee reliable communications from a distributed view.

With the BFT absMAC layer implemented, all the honest nodes have reliable and delay-bounded communications with each other. However, the Byzantine nodes can still mislead the consensus by delivering inconsistent or fake messages. To address this problem, the honest nodes have to know sufficient information from others through the absMAC layer, against the inconsistent or fake messages from the Byzantine nodes. Based on the above idea, our BFT consensus algorithm branches for two cases with $f \in (0, \frac{n}{8})$ and $f \in [\frac{n}{8}, \frac{n}{3})$, in which n and f are the numbers of nodes and Byzantine nodes, respectively. In the first case, each node broadcasts its own opinion and receives the opinions from other nodes through the implemented absMAC layer, which is enough to achieve a consensus. Whereas, in the second case, the honest nodes have to broadcast what they have received in the last execution of absMAC layer algorithm. Otherwise, too many inconsistent opinions from Byzantine nodes can fail the consensus. As for the case with $f \geq \frac{n}{3}$, it has been proved impossible for the BFT consensus problem in [25] even with reliable communications, and will not be discussed.

In general, this paper studies the Byzantine fault tolerant consensus problem in open wireless networks from the physical layer. Compared with most of the previous works relying on the non-Byzantine communication environment, this paper additionally considers the Byzantine behaviors on multiple physical channels, which makes the communications between honest nodes no longer reliable. To avoid making the algorithm design complex, we first implement a BFT absMAC layer that can complete the ack. and prog. operations with a bounded

[2]We say two nodes are neighbors if they are within the transmission range of each other.

[3]We have this assumption because most of the light-weight IoT devices are equipped with a single radio that can only access to a single channel.

time delay. Then, an efficient BFT consensus algorithm is designed based on the implemented absMAC layer. Generally, the novelty of our work is comprised of two components. In part one, different from the previous works assuming/relying on reliable communications, this paper directly implements a Byzantine-resilient abstract MAC layer to provide reliable communications with the presence of jamming attacks across multiple channels. In part two, a consensus algorithm is designed based on our absMAC layer, which can tolerate up to $n/3$ Byzantine faults. Compared with most of the Byzantine consensus algorithms relying on atomic reliable communication[4], our algorithm does not require the atomic property and has a smaller time complexity on achieving consensus. Compared with a recently proposed Byzantine consensus algorithm [27] in absMAC layer that can tolerate at most $n/5$ Byzantine faults, our Byzantine consensus algorithm has stronger Byzantine-resilience.[5]

The detailed contributions of this paper are listed in the following.

- This study investigates the BFT consensus problem in open wireless networks, in which the communications of honest nodes are no longer reliable due to the openness of wireless channels. In previous works, most of them rely on a reliable communication environment, and a few of them are considered on a single physical channel suffered from the Byzantine attacks with energy budget constraints. Compared with the previous works, this paper considers the BFT consensus problem under a more comprehensive and realistic Byzantine communication model, in which Byzantine nodes can cooperatively fail multiple physical channels with the energy budget removed.
- Based on the Byzantine communication model, a BFT absMAC layer is implemented with time bound $O(\frac{kn}{k-f}\log n)$ and $O(\frac{k}{k-f}\log n)$ for its ack. and prog. operations, respectively. n, k, and f are the numbers of nodes, Byzantine nodes, and channels separately. In other words, with the absMAC layer implemented, a node has its message received by all neighbors within $O(\frac{kn}{k-f}\log n)$ rounds, and a node receives one message from its neighbors within $O(\frac{k}{k-f}\log n)$ rounds. To the best of our knowledge, this is the first work implementing a BFT absMAC layer in Byzantine wireless networks.
- Based on the implemented absMAC layer, a BFT consensus algorithm is designed to help honest nodes achieve a consensus with its requirements on agreement, validity, and termination satisfied. When $f \in (0, \frac{n}{3})$, it takes $O(\frac{kn}{k-f}\log n)$ rounds in expectation for honest nodes to reach a consensus. Compared with a non-Byzantine fault-tolerant result $O(n^3 \log n)$ with absMAC layer in [28] and the classical fault-tolerant Paxos in [29], our BFT consensus algorithm is faster. Compared with a Byzantine fault-tolerant consensus algorithm in absMAC layer [27] that can tolerate up to $n/5$ Byzantine faults, our BFT consensus algorithm has a stronger Byzantine resilience.

Both theoretical analysis and numerical simulations are given to show the performance of our BFT consensus algorithm.

Roadmap. This paper is structured as follows. Sec. II gives the necessary related work. Sec. III gives the Byzantine network model in open wireless networks and defines the consensus problem. The implementation of the absMAC layer and the algorithm design for the consensus problem are given in Sec. IV, with the analysis part followed in Sec. V. Sec. VI presents the simulation results for our consensus algorithm. The final conclusion and future work is given in Sec. VII.

II. Related Works

BFT consensus has always been an important research topic in the past years. One of the early works is [25] proposed by Lamport et al. in 1982. Subsequent researches on Byzantine fault-tolerant consensus in wireless networks include [5]–[7], in which communication-efficient protocols using overlay networks, digital signatures, and failure detectors, or message and route redundancy schemes are proposed under a reliable communication environment. Except for those works on reliable communications, [8], [10], [13], [14] consider the BFT consensus problem in some unreliable environments. In [8], a message loss model assumes that the communications between honest nodes are reliable. While the communications from Byzantine nodes are unreliable because of malicious message loss, reordering, insertion, or duplication. Based on the message loss model, the message tagging and stable storage logging techniques are used in [8] to guarantee the correctness, consistency, and termination of consensus. The similar unreliable communication models also include the communication failure model [13], [19], communication collision model [14], and the heard-of model [17]. The study presented in [17] focused on examining the issues of node failures and message losses resulting from collisions. It addressed how to achieve consensus in a single-hop network environment. Subsequently, the concept of communication failure evolved into the dynamic omission failure model as discussed in [2], [19], which takes into account the possibility of Byzantine behaviors impacting communication among faithful devices. Within the framework of the dynamic omission failures model, [2], [19] delved into a binary consensus problem involving a subset of nodes in the network, aiming for at least the subset of nodes to achieve a majority consensus on a binary decision.

Recently, there are also some works considering a more realistic scenario, in which the communications between honest nodes are no longer reliable because of the malicious behaviors from Byzantine nodes, such as the dynamic omission failure model [2], [19] and the jamming model in [21]. As an extended version of the communication failure model, the dynamic omission failure model additionally assumes that the Byzantine behaviors may affect the communications between honest devices. Whereas, the works in [2], [19] no further discuss how legitimate communications can be affected. In [20], [21], an energy-budget jamming attack from the Byzantine nodes is considered, which can fail the legitimate communications

[4]Atomic reliable communications not only require that all the nodes have reliable communications with each other, but also require that the total order of messages received by each node should be the same [26].

[5]A detailed description for the novelty of our work can be found in the Appendix.

TABLE I: Related Works on Communication Failure Models

Reference	Communication Model	Detailed descriptions
[5]–[7]	Reliable communication model	Communications between nodes are reliable
[8]–[11]	Message loss model	(1) Physical channel is reliable. (2) Communications between honest nodes are stable. (3) Messages from Byzantine nodes can be dropped, reordered, inserted or duplicated.
[12], [13]	Communication failure model	(1) Physical channel is reliable. (2) Communications between honest nodes are stable. (3) Messages from Byzantine nodes can be omitted, added and corrupted.
[14]	Communication collision model	(1) Physical channel is reliable. (2) Communications between honest nodes may fail due to benign collisions and interference. (3) Byzantine nodes can send fake messages.
[15]–[18]	Heard-of model	(1) Physical channel is reliable. (2) Communications between honest nodes may fail due to benign failures. (3) Byzantine communications are not discussed.
[2], [19]	Dynamic omission failure model	(1) Physical channel is reliable. (2) Communications between honest nodes may fail with some probability because of Byzantine attacks. (3) Byzantine nodes can send fake messages.
[21]	Byzantine jamming model	(1) Single physical channel is no longer stable due to jamming attacks. (2) All communications fail if the channel is jammed in current time slots. During an interval, at most constant fraction of time slots are jammed due to an energy budget constraint. (3) Byzantine nodes can send fake messages.
Our work	Byzantine communication model	(1) Multiple physical channels are no longer stable due to jamming attacks. (2) Byzantine nodes can cooperatively jam multiple physical channels with stable and sufficient energy supply. All communications in the jammed channels fail. (3) Byzantine nodes can send fake messages.

TABLE II: Related Works on Abstract MAC Layer

Reference	Detailed descriptions
[20], [21], [30], [31]	Implement an abstract MAC layer under dynamic and jamming environments without the presence of Byzantine.
[27], [28], [32]	Solving the Byzantine-resilient consensus, fault-tolerant consensus, and consensus problems with assumed absMAC layer.
Our work	(1) Implement an abstract MAC layer under Byzantine communication failure model. (2) Prove that the absMAC layer indeed provides reliable communications. (3) Solve the BFT consensus problem with the help of implemented abstract MAC layer

between honest nodes. However, considering the jamming attack together with the packet omissions, corruptions, and additions results in a complex algorithm design, as the authors in [20], [21] declare that considering multiple Byzantine attacks indicate a sharp increase in the difficulties of algorithm design. A detailed comparison on the communication models is listed in Table I.

As for the concept of the abstract MAC layer, it was first proposed by Kuhn et al. in [24], to reduce the difficulties of algorithm design in complex networks. In the following years, the research on the abstract MAC layer branched in two directions: how to implement an abstract MAC layer based on complex networks and how to solve the corresponding problems with a provided abstract MAC layer. Calvin Newport et al. in [32] show that with an abstract MAC layer, the famous PAXOS consensus algorithm can be completed within $\Omega(DF_{ack})$ time steps, in which D is the diameter of the wireless network, and F_{ack} is the time complexity of the acknowledgement in the provided abstract MAC layer. On the basis of their previous work, Newport et al. in [28] demonstrated that their protocol can achieve non-Byzantine fault-tolerant consensus among n nodes within $O(n^3 \log n)$ time steps. Recently, an Byzantine approximate consensus algorithm is proposed in [27], in which an eventual message delivery service is required. Additionally, the works in [20], [21], [30], [31] consider how to implement an abstract MAC layer under dynamic and jamming environments. To the best of our knowledge, few of the previous works consider the usage and implementation of abstract MAC layer in the Byzantine-fault tolerant area. A detailed comparison on the absMAC layer technique is listed in Table II.

In general, a large fraction of the previous works are considered with reliable communications between honest nodes. A few of them assume that the legitimate communications between honest nodes no longer keep stable due to the malicious attacks from Byzantine nodes in a single channel. As the sacrifice, their algorithms become much complex. In this paper, we also adopt the harsh assumption that the legitimate communications can be destroyed by the Byzantine nodes with physical jamming attacks, but consider a more general multi-channel scenario. Our novelty relies on that we are the first one considering the BFT consensus problem under a multi-channel Byzantine communication model. Implementing the absMAC layer in a Byzantine environment and use the implemented absMAC layer to make our BFT consensus algorithm efficient and elegant are our technical novelty and contribution.

III. Network Model and Problem Definition

In this paper, we consider a single-hop wireless network in which f Byzantine nodes and $(n-f)$ honest nodes are arbitrarily deployed within a two-dimensional Euclidean space. V is the set of all nodes. V_l and V_b are the set of honest nodes and Byzantine nodes, respectively. All nodes have the same global clock and wake-up initially. By transmitting or listening to the wireless channels, the nodes exchange their messages with each other. In message transmission, each honest node has a unique ID that cannot be forged by others. The physical contention and interference are considered in wireless networks, which results in the unreliable communications between nodes. Based on the unreliable communication model, we consider the implementation of a BFT absMAC layer and then study the BFT consensus problem based on a three-layer Byzantine failure model. The unreliable communication model, Byzan-

tine failure model, definitions for the BFT absMAC layer, and consensus problems are given in the following.

Unreliable Communication Model. The communications between nodes are synchronized and roundly based, i.e., the time in our wireless network is divided into synchronized rounds, each of which is a time unit for the nodes to transmit or receive a message. In each round, the nodes exchange their messages through a shared medium divided into k sub-channels with IDs from 1 to k. Similar with [33], we assume that the k channels are not overlapped and are sufficiently discrete on the spectrum. Each of the nodes is equipped with a single and half-duplex radio. Thus, in each round, the nodes can choose one channel to transmit or listen to, but cannot do both. Only the simultaneous signals in the same wireless channel interfere with each other and the interference across multiple channels is not considered.

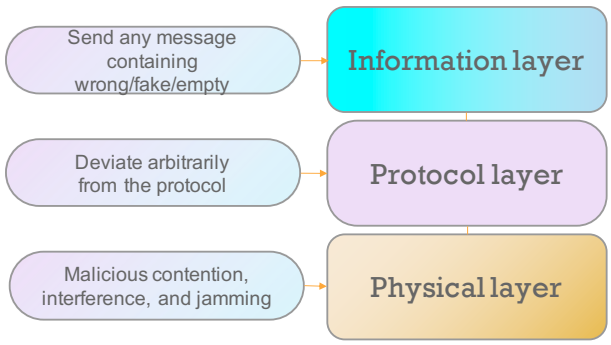


Fig. 2: Three-layer Byzantine Failure Model

In each round, the nodes that choose to transmit/receive are termed transmitters/receivers for short. For a signal from the transmitter u to the receiver v in the same channel, it is denoted by the vector $\vec{S}_{u,v}$ that not only includes the strength but also the phase knowledge of the signal. Whether the signal $\vec{S}_{u,v}$ can be decoded by the receiver v is formulated by the following SINR (Signal to Interference plus Noise Ratio) equations.

$$|\vec{S}_{u,v}| = P_u * d(u,v)^{-\alpha}, \quad |\vec{S}_{W,v}| = |\sum_{u \in W} \vec{S}_{u,v}|, \\ SINR(u,v) = |\vec{S}_{u,v}|/(|\vec{S}_{W\setminus\{u\},v}| + N). \tag{1}$$

In the above SINR equations, $|\vec{S}_{u,v}|$ is the strength of the signal from u to v, which gets weak with the distance $d(u,v)$ between u and v. P_u is the transmission power of node u. The path-loss exponent α is a constant determined by the wireless medium and within 2 to 6 in usual. When the signals from multiple transmitters accumulate at a receiver, the process can be regarded as the sum of vectors. Let W be the set of transmitters that are within the same channel with v, $\vec{S}_{W,v}$ is the mixed signal sensed by v. For the transmission from u to v, it succeeds if $SINR(u,v)$ is larger than β, in which $|\vec{S}_{u,v}|$ is the strength of the signal from u and received by v, $|\vec{S}_{W\setminus\{u\},v}|$ is the interference from other transmitters that in the same channel, N is the ambient noise determined by the environment, and β is the threshold determined by the hardware of v, greater than 1. The transmission power can be determined by the transmitter itself and has the lower and upper bounds P_{min} and P_{max}, respectively. To guarantee a single hop wireless network, we have $\frac{P_{min}}{d^{\alpha}(u,v)N} \geq \beta$ for any pair of nodes u and v.

Three-layer Byzantine Failure Model. Similar to [21], a three-layer Byzantine failure model is adopted to depict the malicious behaviors of totally f Byzantine nodes on the physical layer, protocol layer, and information layer. On the physical layer, each Byzantine node can arbitrarily choose a channel to transmit, listen to, or jam with a jamming signal in each round[6]. Evidence for this can be found in the occurrence of jamming attacks or in the installation of infected firmware when attackers have physical access to these devices [34]. On the protocol layer, the Byzantine nodes can deviate arbitrarily from the protocol to disrupt the task, such as malicious competition in a leader election task and message omission in a message dissemination task. On the information layer, the Byzantine nodes can send any messages containing wrong/fake information to mislead the other honest nodes on their next or final decisions. We assume that the f Byzantine nodes have reliable and real-time communications with each other. Thus, they can cooperatively launch their malicious attacks on the three layers after a full discussion with each other. As shown in Fig 2.

Overall, our model not only encompasses physical layer jamming capabilities but also addresses malicious behaviors at the protocol and information layers, thereby offering a holistic view of potential Byzantine attacks in real-world scenarios. At the physical layer, Byzantine nodes possess the capability to selectively jam communication channels, a technique that indeed might necessitate firmware modifications in practical implementations. In reality, someone can use an add-on to jam channels, which illustrates a practical concern at the physical layer. This aspect underscores the technical feasibility and potential threat of such attacks in real environments where adversaries might gain control over devices with sufficient privileges to alter their firmware for jamming purposes. At the protocol layer, Byzantine nodes may deviate from established communication protocols to disrupt operations, such as by maliciously competing during leader election tasks or omitting messages. At the information layer, these nodes further exploit their capabilities to disseminate false or misleading information, aiming to corrupt the decision-making processes of honest nodes. This layered approach reflects a nuanced understanding of Byzantine behaviors, extending beyond mere jamming to include sophisticated strategies that adversaries might employ.

Reliable and real-time communications among Byzantine nodes facilitate coordinated attacks across these layers, significantly elevating the threat level and highlighting the practical challenges in defending against such multifaceted attacks. This model draws parallels with real-world attack scenarios, such as coordinated network attacks involving both jamming and false information to undermine system integrity.

Problem Definition for the BFT AbsMAC layer. Based on the unreliable communication model and three-layer Byzantine

[6]Due to the hardware limitation (i.e., single radio), each Byzantine node can only get access to a single channel in each round.

failure model, we implement the following BFT absMAC layer in a single-hop wireless network, to provide reliable message exchanges for high-level algorithms and applications. Specifically, there are two operations in our absMAC layer: *acknowledgement* (ack. for short) and *progress* (prog. for short). In an ack. operation, each honest node has its message received by all the other honest nodes; and in a prog. operation, each honest node at least has one message received from the other honest nodes. The f_{ack} and f_{prog} are used to denote the timing bounds to complete the ack. and prog. operations, respectively. A series of works [28], [32] have shown that with an absMAC layer provided, the consensus problem can be solved elegantly. However, few of them discuss how to implement a BFT absMAC layer.

To implement an absMAC layer, the policy of the nodes in an interval includes their actions (transmit or listen in which channel) in each round of the interval. For each node v, tuple $\langle a_{v,t}, b_{v,t}\rangle$ is used to denote its action in the round t. If v transmits, Boolean variable $a_{v,t} = 1$; otherwise, $a_{v,t} = 0$. The variable $b_{v,t}$ is used to record the ID of the channel chosen by v in the round t. Combining with the SINR model and our Byzantine failure model, our problem can be formulated as minimizing the length of the interval with the constraints on ack. and prog. operations, given in the following.

$$
\begin{aligned}
&\textbf{Minimize} \quad |I| \\
&s.t.\ a_{v,t} = 1,\ a_{u,t} = 0,\ b_{v,t} = b_{u,t},\ SINR(v,u) \geq \beta \\
&\qquad \text{for } \exists\ t \in I,\ \forall\ v \in V_l,\ \forall\ u \in V_l \setminus \{v\} \\
&\qquad a_{v,t} = 0,\ a_{u,t} = 1,\ b_{v,t} = b_{u,t},\ SINR(u,v) \geq \beta \\
&\qquad \text{for } \exists\ t \in I,\ \forall\ v \in V_l,\ \forall\ u \in V_l \setminus \{v\} \\
&with\ a_{v,t} \text{ and } a_{u,t} \text{ are binary} \in \{0,1\}, \\
&\qquad b_{v,t} \text{ and } b_{u,t} \text{ are integer} \in \{1,2,...,k\}, \\
&\qquad SINR(v,u) = \frac{|a_{v,t} * \vec{S}_{v,u}|}{(|\sum_{w \in W \setminus \{v\}} a(w,t) * \vec{S}_{w,u}| + N)} \\
&\qquad \text{for } \forall\ t \in I,\ \forall\ v \in V,\ \forall\ u \in V \setminus \{v\}
\end{aligned}
\tag{2}
$$

In the above equation, the first and the second constraints require that the ack. and prog. operations for all honest nodes should be completed within the interval I. Let $\langle a_{V_l,t}, b_{V_l,t}\rangle = \{\cup_{v \in V_l} \langle a_{v,t}, b_{v,t}\rangle\}$ be the action set of all the honest nodes in the round t, and $\langle a_{V_l,I}, b_{V_l,I}\rangle = \{\cup_{t \in I} \langle a_{V_l,t}, b_{V_l,t}\rangle\}$ be the action set of all the honest nodes in the interval I, which is also termed as the policy of the honest nodes in interval I. Similarly, we have $\langle a_{V_b,I}, b_{V_b,I}\rangle$ to denote the policy of the Byzantine nodes which can be arbitrarily determined by the Byzantine nodes. In this paper, our objective is to design a distributed algorithm, by running which an efficient policy $a_{V_l,I}$ can be generated for the absMAC layer, despite the malicious policy from Byzantine nodes. Such an optimization problem can be formulated as a link scheduling problem that has been proven to be NP-complete in [35].

Problem Definition for the BFT consensus problem. We study the classical binary consensus problem among n nodes that contains f Byzantine nodes. Initially, all honest nodes have their binary opinions from $\{0,1\}$ for one task or event. By exchanging opinions with each other, each honest node can choose to insist on or change its opinion. Finally, all the honest nodes are required to achieve a consensus on their opinions with the following properties satisfied [36]. *Agreement*: all honest nodes should hold the same opinion; *Validity*: the final opinion held by the honest nodes should come from an honest node; *Termination*: this consensus should be achieved within a finite time

TABLE III: Table of key notations.

Notations	Definitions
n	Total number of nodes in the network
V	Set of all nodes
f	Number of Byzantine nodes
V_l	Set of honest nodes
V_b	Set of Byzantine nodes
P_u	Transmission power of node u
$d(u,v)$	Distance between node u and node v
α	Path-loss exponent
W	Set of transmitters in the same channel with v
$SINR(u,v)$	Signal to Interference plus Noise Ratio from u to v
β	Threshold determined by the hardware of v
N	Ambient noise
P_{min}	Minimum transmission power
P_{max}	Maximum transmission power
I	Interval duration for ack. and prog. operations
$Op_{v,t}$	Opinion of node v at round t

For each node v, variable $Op_{v,t}$ is used to denote its opinion at the round t, with $Op_{v,0}$ as its initial value. For the honest nodes, their initial opinions come from the set $\{0,1\}$. Whereas, the opinions of Byzantine nodes at any round can be arbitrary and determined by themselves, which may mislead the honest nodes on achieving a consensus.

Knowledge of Nodes and Necessary Assumptions. In our system model, the honest nodes are endowed with essential knowledge of the wireless network's key characteristics. This includes an understanding of the total number of nodes in the network n, the number of available communication channels k, and the maximum number of Byzantine nodes f. The nodes are identified by simplified IDs ranging from 1 to n. These aspects are crucial for nodes to participate effectively in the distributed protocol.

Total number of nodes n should be known by all the nodes. Each node is cognizant of the total count of nodes, which is vital for them to gauge the network's scale and adjust their behavior accordingly. This is in line with the principles outlined in Nancy Lynch's [37], where network size awareness is emphasized for efficient protocol functioning.

Number of communication channels k should be known by all the nodes. Nodes are aware of the total number of communication channels. This knowledge is crucial, especially in scenarios where channel interference is a possibility, such as in the presence of Byzantine nodes. This aspect is highlighted in the work by Gafni and Bertsekas [38], underscoring the importance of channel awareness in maintaining robust communication.

Maximum number of Byzantine nodes $f < n/3$. It is assumed that f, the number of Byzantine nodes, is less than one-third of the total nodes n. This assumption is grounded in the classic theory of Byzantine Fault Tolerance [25]. It's established that maintaining system safety and consistency is unattainable if the Byzantine nodes reach or exceed one-third

of the total nodes.

The number of channels $k > f$. The requirement that the number of channels should exceed the number of Byzantine nodes $k > f$ is critical. This is to prevent the Byzantine nodes from completely jamming the communication channels, which could render the consensus problem unfeasible. This assumption is supported by Dolev and Strong's [39], which emphasizes the necessity of redundant communication paths in the presence of Byzantine faults.

These assumptions form the bedrock of our proposed Byzantine Fault-Tolerant system. They are derived from established theories and research in distributed computing and network communications, ensuring that the system remains effective and secure despite the presence of Byzantine nodes.

IV. Algorithm Description

In this part, we show how to implement a BFT absMAC layer and use the implemented absMAC layer to achieve a consensus despite the malicious behaviors from f Byzantine nodes. Firstly, we discuss the challenges in our algorithm design and their corresponding solutions. Secondly, the algorithm to implement a BFT absMAC layer is presented. Finally, we show how to achieve the consensus efficiently and elegantly with the help of the implemented absMAC layer.

A. Challenges and Solutions

The first challenge is that the unstable communication between the honest nodes increases the difficulty on design a BFT algorithm. As has been mentioned in the related work section, most of the previous works require reliable communications between honest nodes and address the Byzantine behaviors from the protocol layer and information layer. Whereas, the Byzantine behaviors on the physical layer (e.g. the jamming attack and malicious contention) result in the unreliable communications between honest nodes, which may impact or even fail the BFT methods in the previous works. The second challenge is that the Byzantine behaviors on the protocol layer and information layer will mislead the absMAC layer. In general, most of the previous works rely on feedbacks from neighbors to design an efficient absMAC layer. For example, in [20], once a node has its messages received by all the neighbors, it will receive a feedback from its neighbors and terminate. However, in a Byzantine environment, the malicious feedback from Byzantine nodes is nearly impossible to detect and misleads the honest nodes.

To address the challenges mentioned above, in this paper, we first adopt a randomized communication scheme based on the multi-channel technique, to help the honest nodes find clean channels and obtain some reliable communications. We say a channel is clean if it does not contain any Byzantine behaviors. Otherwise, it is polluted by the Byzantine nodes, in which the communications are unreliable. Then, based on the reliable communications, an absMAC layer is implemented without relying on the feedback of neighbors. Refusing the feedback from neighbors fundamentally avoids the negative impacts of Byzantine behaviors on the protocol and information layers. Thus, it strengthens the Byzantine fault tolerance of our algorithm in open wireless networks. As a tradeoff, it takes longer time for our absMAC layer to complete the ack. and prog. operations, compared with the previous non-Byzantine works using feedback schemes.

B. Implementation of BFT absMAC layer

As has been defined, an absMAC layer consists of two operations: the ack. in which all honest nodes have their messages received by the other honest nodes and the prog. in which all honest nodes receive a message from other honest nodes. To complete these two operations despite the Byzantine attacks on the physical, protocol, and information layers, a randomized communication scheme based on the multi-channel technique is designed. Specifically, at the beginning of each round, each honest node v randomly and uniformly chooses a channel j from the k non-overlapped channels to transmit with probability $1/2$ or listen to with the other $1/2$ probability. Note that multiple transmitters transmitting with similar power in the same channel are likely to result in collision and heavy interference. The following *power selection rule* from [40] is used to help nodes randomly separate their transmission powers and reduce the contention/interference between communications. In detail, for each node v, $d = i + \lceil 2\log_2 n \rceil$ with probability $1/2^i$. Let D be an integer randomly selected from $[2^d \log_2^2 n, 2^{d+1} \log_2^2 n)$, we have $P_v = P_{min} * D^{\gamma D}$, where γ is a positive constant and P_v is the transmission power of node v. Specifically, γ is set as a constant larger than $\max(1, s\alpha + 1 + \log\beta)$. Constants α and β are the SINR parameters. We normalize the shortest distance between any pair of nodes in the network as 1 and use R to denote the longest distance between any pair of nodes. $s = \log_n R$ is a constant since we assume that R can be bounded by a polynomial of n. The same setting can be found in [40] and [41]. With such a power selection rule, we can prove that the node with the loudest transmission power has its message received by all the receivers if they stay in a clear channel. In [40], such a power selection scheme is used for leader election in a single-hop wireless network. In this paper, we extend it to a multi-channel scenario. When a node v listens to the channel j, it saves the messages from other transmitters if the signals from other transmitters can be decoded. By repeating such a randomized communication scheme for $\frac{ckn}{k-f}\log n$ times, we show that the ack. and prog. operations can be completed and the BFT absMAC layer gets implemented, in which c is a sufficiently large constant. The pseudocode to implement our absMAC layer is given in the Algorithm 1. Theorem 1 is given to show the performance of our BFT absMAC layer and proved in the analysis section.

Theorem 1: Within $\frac{ckn}{k-f}\log n$ rounds, our BFT absMAC layer can be implemented w.h.p. [7], with $f_{ack} = O(\frac{ckn}{k-f}\log n)$ for ack. and $f_{prog} = \frac{ck}{k-f}\log n$ for prog.

C. BFT Consensus Algorithm with absMAC layer

With the BFT absMAC layer implemented, we design an efficient and elegant BFT consensus algorithm to solve the

[7] with high probability for short, at least with probability of $1 - \frac{1}{n^{c_0}}$ for some constant $c_0 > 1$

Algorithm 1: BFT absMAC layer for node v

$\mathcal{M}_v :=$ Message of v; $\mathcal{S}_v := \phi$;
$P_v :=$ Transmission power of v ;
for $\frac{ckn}{k-f} \log n$ *rounds* **do**
  $j \leftarrow Random(\{1, 2, ..., k\})$;
  $x \leftarrow \{0, 1\}$ uniformly and randomly;
  if $x = 1$ **then**
    obtain its transmission power P_v according to the power selection rule;
    transmit $\mathcal{M}_v$ with power P_v in the channel j;
  else
    listen the channel j;
    if *received a message* $\mathcal{M}_u$ *from node* u **then**
      $\mathcal{S}_v = \mathcal{S}_v \cup \{\mathcal{M}_u\}$;

Power selection rule
$d \leftarrow i + \lceil 2 \log_2 n \rceil$ with probability $1/2^i$, $i \in N+$;
$D \leftarrow$ an integer randomly selected from interval $[2^d \log_2^2 n, 2^{d+1} \log_2^2 n)$;
$P_v = P_{min} * D^{\gamma D}$;

Algorithm 2: BFT consensus for node v

1 Op_v: opinion of v; ID_v: ID of v; $\mathcal{S}_v = \phi$;
2 Ds_v: decision of v; $\mathcal{M}_v$: message of v; $\hat{\mathcal{S}}_v = \phi$;
3 $X_v[n][n]$: two-dimension matrix with size of $n \times n$;
4 **if** $0 < f < \frac{n}{8}$ **then**
5 Branch-I ( );
6 **else if** $\frac{n}{8} \leq f < \frac{n}{3}$ **then**
7 Branch-II ( );
8 Output Ds_v;

Branch-I ( )
9 **while** $|\hat{\mathcal{S}}_v| < \frac{7}{8}n$ **do**
10 $\mathcal{M}_v \leftarrow \langle ID_v, Op_v \rangle$; $\mathcal{S}_v \leftarrow \{Op_v\}$;
11 transmit message $\mathcal{M}_v$ through the absMAC layer;
12 **for** *each received* $\mathcal{M}_u$ *from the absMAC layer* **do**
13 $\mathcal{S}_v \leftarrow \mathcal{S}_v \cup \{Op_u\}$;
14 $\hat{\mathcal{S}}_v \leftarrow$ major opinions in set $\mathcal{S}_v$;
15 $Ds_v \leftarrow$ value of opinions in the set $\hat{\mathcal{S}}$;
16 $y \leftarrow \{0, 1\}$ uniformly and randomly;
17 **if** $|\hat{\mathcal{S}}_v| < \frac{5+y}{8}n$ **then**
18 $Ds_v \leftarrow 0$;
19 $Op_v \leftarrow Ds_v$;

Branch-II ( )
20 $\mathcal{M}_v \leftarrow \langle ID_v, Op_v \rangle$; $\mathcal{S}_v \leftarrow \{\langle ID_v, Op_v \rangle\}$;
21 transmit message $\mathcal{M}_v$ through the absMAC layer;
22 **for** *each received* $\mathcal{M}_u$ *from the absMAC layer* **do**
23 $\mathcal{S}_v \leftarrow \mathcal{S}_v \cup \{\langle ID_u, Op_u \rangle\}$;
24 $\mathcal{M}_v \leftarrow \langle ID_v, \mathcal{S}_v \rangle$;
25 transmit its message $\mathcal{M}_v$ through the absMAC layer;
26 **for** *each received* $\mathcal{M}_u$ *from the absMAC layer* **do**
27 **for** *each tuple* $\mathcal{S}_w$ *in the set* $\mathcal{S}_u$ **do**
28 $X_v[ID_u][ID_w] \leftarrow Op_w$;
29 **for** *each tuple* Op_u *in the set* $\mathcal{S}_v$ **do**
30 **if** *at least* $n - f$ *nodes (denoted by* $\{w_1, w_2, ..., w_{n-f}\}$*) satisfy that* $X_v[ID_{w_1}][ID_u] = X_v[ID_{w_2}][ID_u] = ... = X_v[ID_{w_{n-f}}][ID_u]$ **then**
31 $Op_u = X_v[ID_{w_1}][ID_v]$;
32 $\hat{\mathcal{S}}_v \leftarrow \{Op_u\}$;
33 $Ds_v \leftarrow$ value of the major opinions in the set $\hat{\mathcal{S}}_v$;

binary consensus problem. Specifically, our algorithm has two branches to handle the cases $0 < f < \frac{n}{8}$ and $\frac{n}{8} \leq f < \frac{n}{3}$, respectively. As proved in our analysis section, in the first branch with $f < \frac{n}{8}$, directly exchanging opinions with each other for multiple times is enough for the honest nodes to achieve a binary consensus. Whereas, when there are more than $\frac{n}{8}$ Byzantine nodes, those honest nodes may be misled when the Byzantine nodes transmit inconsistent opinions [42]. To avoid this, a more complex communication scheme is designed in our second branch to help the honest nodes find legitimate opinions. The detailed descriptions are given in the following.

Branch-I. In the first branch with $0 < f < \frac{n}{8}$, the message to be transmitted by an honest node only contains its ID and opinion. Thus, it is a constant-size message and communication efficient. Through the BFT absMAC layer, each honest node v successfully broadcast its opinion to the other nodes and knows the opinions of other nodes. Set $\mathcal{S}_v$ is used by node v to store the opinions from other nodes and its own opinion. Set $\hat{\mathcal{S}}_v$ is used to record the major opinions in the set $\mathcal{S}_v$ and Ds_v is used to record the value of the opinions in the set $\hat{\mathcal{S}}_v$.[8] With the help of $\mathcal{S}_v$ and $\hat{\mathcal{S}}_v$, the consensus problem can be directly solved if it is in a non-Byzantine case. For example, all the honest nodes choose the value of major opinions as the final agreement. Whereas, the honest nodes may receive inconsistent opinions from the Byzantine nodes, which can make the set $\mathcal{S}_v$ various for different honest nodes v. Thus, we further have the following operations: variable y has the value of 0 with probability $1/2$ and 1 with the other probability $1/2$. If the number of opinions in the set $\hat{\mathcal{S}}_v$ is smaller than $\frac{5+y}{n}$, Ds_v is set to 0. All the honest nodes set Ds_v as their opinions. The above steps (lines 10-19 in Algorithm 2) are repeated until $|\hat{\mathcal{S}}_v| \geq \frac{7}{8}n$. In other words, only when the number of major opinions is at least $\frac{7}{8}n$, the loop in Branch-I terminates and the decision Ds_v of each node v is output as the final agreement.

Branch-II. In the second branch with $n/8 \leq n < n/3$, the BFT absMAC layer algorithm is executed twice to detect the inconsistent opinions from the Byzantine nodes. Specifically, in the first execution of the absMAC layer algorithm (line 21 in the Algorithm 2), the message to be transmitted by each honest node contains its ID and opinion. Through the absMAC layer,

[8]When the major opinions are chosen in the algorithm 2, the opinions with the value of "0" will be chosen if the opinions with the value of "0" has the same number with the opinions with the value of "1".

each honest node v receives the opinions from the other nodes and stores those opinions in the set $\mathcal{S}_v$. Then, in the second execution of the absMAC layer (line 25 in the Algorithm 2), each honest node v broadcasts its set $\mathcal{S}_v$ to all nodes and receives the set $\mathcal{S}_u$ from other nodes u through the absMAC layer. A matrix $X_v[n][n]$ is used by node v to handle the opinions from its set $\mathcal{S}_v$ and the sets $\mathcal{S}_u$ received from other nodes u. Let's take the nodes u, v, and w as an example. For node v, if it receives $\mathcal{S}_w$ from the node w, which contains the opinion Op_u, we say v hears the opinion of u from node w, and has $X_v[ID_w][ID_u] = Op_u$. When both w and u are honest nodes, the honest node v knows the true opinion of u. When w or u is the Byzantine node, v may hear nothing about u or receive a wrong opinion Op_u from w. When v hears nothing about w, it has $X_v[ID_w][ID_u] = null$. When v hears a wrong opinion Op_u from w, the wrong opinion will also be recorded by $X_v[ID_u][ID_w]$ since the wrong opinion cannot be directly detected. With the help of matrix $X_v[n][n]$, node v knows the opinion of u from other nodes. We say an opinion Op_u is legitimate in node v if v hears the same Op_u from at least $n-f$ nodes $\{w_1, w_2, ..., w_{n-f}\}$ (line 30 in the Algorithm 2). Set $\hat{\mathcal{S}}_v$ is used to store all the legitimate opinions Op_u received by v. Finally, node v chooses the value of the major opinions in the set $\hat{\mathcal{S}}_v$ as its final decision. The performance of our BFT consensus algorithm is presented in the Theorem 2 and proved in the next section.

Theorem 2: With the help of implemented BFT absMAC layer, all the honest nodes achieve a consensus within $O(\frac{ckn}{k-f}\log n)$ rounds in expectation with its properties agreement, validity, and termination satisfied.

V. Theoretical Analysis

In this part, we prove the correctness and efficiency of our BFT absMAC layer algorithm and BFT consensus algorithm.

A. Analysis for BFT AbsMAC Layer Algorithm

As mentioned in our model section, the absMAC layer contains the ack. and prog. operations. In an ack. operation, all the honest nodes have their messages received by the other honest nodes. In a prog. operation, all the honest nodes receive at least one message from the honest nodes. In the following, the Lemma 1 and Lemma 2 are given to prove the ack. and prog. operations, respectively.

Lemma 1: Within $O(\frac{ckn}{k-f}\log n)$ rounds, all the honest nodes have their messages received by the other honest nodes w.h.p..

Proof: Since the Byzantine attacks on the physical layer can directly fail the communications in a channel, we firstly show that there are always $(k-f)$ clean channels that can be used by the honest nodes to transmit or listen to in the Claim 1. Then, we focus on the randomized process and communication efficiency of honest nodes in the clean channel in Claim 2 and 3, which proves this lemma.

Claim 1: In each round, there are at least $(k-f)$ clean channels despite the f Byzantine nodes.

Proof: The Claim 1 holds because there are at most f Byzantine nodes, each of which at most chooses one channel to pollute in each round. ∎

Claim 2: In each round, the node with the loudest transmission power has its message received by all the other nodes in the same clean channel with high probability.

Proof: Let's consider an arbitrary round t and a clean channel j. W is the set of transmitters in channel j. u is the node with the maximum transmission power in the set W. v is a receiver in the channel j. According to our SINR equation,

$$SINR(u,v) \geq \frac{P_u/d^\alpha(u,v)}{\sum_{w\in W\setminus\{u\}} P_w/d^\alpha(w,v)+N} \tag{3}$$

With the Claim 4 in the Appendix, we have $\frac{P_w}{d^\alpha(w,v)} < \frac{P_u}{d^\alpha(u,v)\beta n}$ for arbitrary node $w \in W\setminus\{u\}$ w.h.p.. Additionally, we have $N \leq \frac{P_{min}}{d^\alpha(u,v)\beta}$ in model section and $|W\setminus\{u\}| \leq n-1$. Thus, we have

$$\begin{aligned} SINR(u,v) &\geq \frac{P_u/d^\alpha(u,v)}{\sum_{w\in W\setminus\{u\}} P_w/d^\alpha(w,v)+N} \\ &\geq \frac{P_u/d^\alpha(u,v)}{\sum_{w\in W\setminus\{u\}} P_u/\left(d^\alpha(u,v)\beta n\right)+N} \\ &\geq \frac{P_u/d^\alpha(u,v)}{\frac{n-1}{n}*\frac{P_u}{d^\alpha(u,v)\beta}+\frac{P_{min}}{d^\alpha(u,v)\beta}} \\ &\geq \frac{P_u/d^\alpha(u,v)}{\frac{n-1}{n}*\frac{P_u}{d^\alpha(u,v)\beta}+\frac{P_u}{d^\alpha(u,v)\beta}} = \beta. \end{aligned} \tag{4}$$

According to our SINR communication model, $SINR(u,v) \geq \beta$ means that v receives the message from u, which proves the result in claim 2. ∎

Claim 3: For arbitrary two honest nodes u and v, v receives the message from u w.h.p. within $O(\frac{ckn}{k-f}\log n)$ rounds.

Proof: According to the results in Claim 2, a sufficient condition for v receiving the message from u is that (a) both v and u choose the same clean channel j; (b) u becomes the node with the maximum transmission power and v listens; For the event (a), it occurs at least with a probability $\frac{k-f}{k}*\frac{1}{k}$. Because there are at least $(k-f)$ clean channels hidden behind the k channels and both u and v randomly choose a channel in each round. For the event (b), it occurs at least with a probability $\frac{1}{2n_j}*\frac{1}{2}$, in which $n_j \in \Theta(n/k)$ is the number of nodes choosing the channel j. Because each node transmits or listens with probability $1/2$ in its channel and has the equal probability to become the node with the maximum transmission power. Multiplying these probabilities together, we obtain a result that for arbitrary two honest nodes u and v, v receives the message from u with a probability of $\Theta(\frac{k-f}{nk})$. By applying a Chernoff bound, we obtain that within $\frac{ckn}{k-f}\log n$ rounds, v receives the message from u with high probability of $1-\frac{1}{n^{\Theta(c)}}$. The introduction and application of Chernoff bound are given in the Appendix. ∎

By setting constant c sufficiently large and taking a union bound for any pair of honest nodes from the set V_l, the Lemma 1 gets proved. ∎

Lemma 2: Within $O(\frac{ck}{k-f}\log n)$ rounds, all the honest nodes at least receive one message from the other honest nodes w.h.p..

Proof: With Claims 1 and 2, we have already shown that the nodes with the maximum transmission power will have their message received by all the receiving nodes in the same

clean channel. Thus, a sufficient condition for an honest node v to receive one message from the other honest nodes is to choose a clean channel and become the listener. Consider that (1) there are at least $(k-f)$ clean channels hidden behind the k channels and (2) v randomly choose a clean channel to transmit or listen with a probability $1/2$. The probability for an honest node v to receive one message from the other honest nodes in a round is at least $\frac{k-f}{2k}$. By applying a Chernoff bound, we obtain that within $\frac{ck}{k-f}\log n$ rounds, v receives at least one message from an honest node. By setting constant c sufficiently large and taking a union bound on all the honest nodes, the Lemma 2 gets proved. ■

B. Analysis for BFT Consensus Algorithm

Since our algorithm branches for the cases $0 < f < \frac{n}{8}$ and $\frac{n}{8} \leq f < \frac{n}{3}$, we prove the agreement, validity, termination of algorithms in branch-I and branch-II, respectively.

Proof for Branch-I. We start the analysis in Branch-I from a worst-case $f = \lfloor \frac{n}{8} \rfloor - 1$. Let $\mathcal{V}_0$ and $\mathcal{V}_1$ be the set of honest nodes with the opinion "0" and the opinion "1", respectively. In branch-I, through the absMAC layer, all the honest nodes receive opinions from the other honest nodes and the Byzantine nodes. For each honest node v, $\mathcal{S}_v$ has been defined as the set of opinions it received from the honest nodes and Byzantine nodes. The major opinions in set $\mathcal{S}_v$ are recorded by the set $\hat{\mathcal{S}}_v$. num_b is the number of opinions in set $\hat{\mathcal{S}}_v$ contributed by the Byzantine nodes. According to the line 17-18 in Algorithm 2, the honest nodes in branch-I set their opinions as 0 if $\hat{\mathcal{S}}_v < \frac{5+y}{8}n$. Otherwise, v chooses the opinion from the set $\hat{\mathcal{S}}_v$ as its opinion. Our following analysis branches into the following five cases.

- Case 1: $|\mathcal{V}_0| \in [\frac{n}{2}, n-f]$. In this case, the major opinion in set $\mathcal{S}_v$ must be opinion 0. According to the line 15-19, no matter whether the event $|\hat{\mathcal{S}}_v| < \frac{5+y}{8}n$ occurs or not, all the honest nodes choose 0 as their opinions.
- Case 2: $|\mathcal{V}_0| \in [\frac{n-f}{2}, \frac{n}{2})$. In this case, even though $|\mathcal{V}_0| \geq |\mathcal{V}_1|$, the major opinions in set $\mathcal{S}_v$ can be 0 or 1 because the f Byzantine nodes can arbitrarily vote for opinions from 0 and 1. What we can make sure is that $|\hat{\mathcal{S}}_v| < \frac{5n}{8}$ because $|\hat{\mathcal{S}}_v| \leq \max\{|\mathcal{V}_0|, |\mathcal{V}_1|\} + num_b \leq |\mathcal{V}_0| + f < \frac{5n}{8}$. num_b is the number of opinions in set $\hat{\mathcal{S}}_v$ contributed by the Byzantine nodes. Since the Byzantine nodes can arbitrarily vote for opinions 0 or 1, num_b can be an arbitrary value from 0 to f determined by the Byzantine nodes. Thus, all the honest nodes choose the value 0 as their opinion according to the lines 18 and 19 in the Algorithm 2.
- Case 3: $|\mathcal{V}_0| \in [\frac{n}{2} - f, \frac{n-f}{2})$. In this case, even though $|\mathcal{V}_0| < |\mathcal{V}_1|$, the major opinions in set $\mathcal{S}_v$ can be 0 or 1 because the f Byzantine nodes can arbitrarily vote for opinions from 0 and 1. What we can make sure is that $|\hat{\mathcal{S}}_v| < \frac{5n}{8}$ because $|\hat{\mathcal{S}}_v| \leq \max\{|\mathcal{V}_0|, |\mathcal{V}_1|\} + num_b \leq |\mathcal{V}_1| + f = n - |\mathcal{V}_0| \leq \frac{5n}{8}$. Thus, all the honest nodes choose the value 0 as their opinion according to the line 18 and 19 in the Algorithm 2.
- Case 4: $|\mathcal{V}_0| \in [\frac{n}{8} - f, \frac{n}{2} - f)$. In this case, $|\hat{\mathcal{S}}_v| = |\mathcal{V}_1| + num_b \in [\frac{n}{2}, \frac{7n}{8}]$. For the case $|\hat{\mathcal{S}}_v| \in [\frac{n}{2}, \frac{5n}{8})$, all the honest nodes have $|\hat{\mathcal{S}}_v| < \frac{5+y}{8}n$ no matter $y = 0$ or $y = 1$ and choose 0 as their opinions. For the case $|\hat{\mathcal{S}}_v| \in [\frac{3n}{4}, \frac{7n}{8})$, all the honest nodes have $|\hat{\mathcal{S}}_v| \geq \frac{5+y}{8}n$ and choose the major opinion 1 as their opinion. As for the middle case $|\hat{\mathcal{S}}_v| \in [\frac{5n}{8}, \frac{3n}{4})$, $|\hat{\mathcal{S}}_v| < \frac{5+y}{8}n$ when $y = 1$ and $|\hat{\mathcal{S}}_v| \geq \frac{5+y}{8}n$ when $y = 0$. In the branch-I, y is a variable randomly and uniformly chosen from $\{0, 1\}$. In other words, each honest node v chooses 0 as its opinion with probability $1/2$ and 1 as its opinion with the remaining probability $1/2$. By doing this, $|\mathcal{V}_0|$ falls into the scope of $[\frac{3n}{8} - f, n - f]$ with at least $1/2$ in the next round of consensus process, because the expectation of $|\mathcal{V}_0|$ is $\frac{n-f}{2}$ in the next round. When $|\mathcal{V}_0|$ falls into the scope of $[\frac{3n}{8} - f, n - f]$ in the next round, all the honest nodes achieve a consensus on the opinion 0. This is because when $|\mathcal{V}_0| \in [\frac{n}{2} - f, n - f]$ in the next round, all the honest nodes choose the value 0 as their opinion, according to the analyses in cases 1, 2, and 3, respectively. In the remaining case when $|\mathcal{V}_0| \in [\frac{3n}{8}, \frac{n}{2} - f)$, we have $|\mathcal{V}_1| = n - f - |\mathcal{V}_0| \in (\frac{n}{2}, \frac{5n}{8} - f]$. Considering that $|\hat{\mathcal{S}}_v| = |\mathcal{V}_1| + num_b$ and $num_b \in [0, f]$, we have $|\hat{\mathcal{S}}_v| \in [\frac{n}{2}, \frac{5n}{8}]$ and all the honest nodes choose 0 as their opinion, according to the previous analysis in Case 4.
- Case 5: $|\mathcal{V}_0| \in [0, \frac{n}{8} - f)$. In this case, the major opinion in set $\mathcal{S}_v$ must be opinion 1 and $|\hat{\mathcal{S}}_v| \geq \frac{5+y}{8}n$ no matter $y = 0$ or $y = 1$. Thus, all the honest nodes choose 1.

From the above cases, we can see that (1) if $|\mathcal{V}_0|$ falls into the cases 1-3 and 5, all the honest nodes reach an agreement on the valid opinion in one round consensus, (2) if $|\mathcal{V}_0|$ falls into the case 4, all the honest nodes reach an agreement on the valid opinion at least with probability of $1/2$ in each round of consensus. Additionally, when all the honest nodes hold the same opinion, the loop in the branch-I terminates because $|\hat{\mathcal{S}}_v| \geq \frac{7n}{8}$. Thus, the agreement and validity are satisfied within constant rounds of consensus in expectation.

Proof for Branch-II. The branch-II is designed for the case $f \in [\frac{n}{8}, \frac{n}{3})$. Through the absMAC layer, all the honest nodes v know the opinions of other nodes and store them in the set $\mathcal{S}_v$. Then, with the help of matrix $X_v[n][n]$, the legitimate opinions are selected from the set $\mathcal{S}_v$ and stored in the set $\hat{\mathcal{S}}_v$. Finally, all the honest nodes select the value of the major opinions in the set $\hat{\mathcal{S}}_v$ as the final agreement. Obviously, our algorithm terminates after executing the absMAC layer algorithm twice, by which the property of termination is proved. In the next, we prove the agreement and validity of the consensus.

Lemma 3: For any two honest nodes u and v, $\hat{\mathcal{S}}_u = \hat{\mathcal{S}}_v$ w.h.p..

Proof: We first consider the case that w is an honest node. Through our absMAC layer, node u receives the opinion Op_w from all the other honest nodes. In this case, the condition in line 30 of Algorithm 2 gets satisfied and $\hat{\mathcal{S}}_u \leftarrow \{Op_w\}$. With similar proof, we have $\hat{\mathcal{S}}_v \leftarrow \{Op_w\}$. Thus, all the opinions from honest nodes can be found in the sets $\hat{\mathcal{S}}_u$ and $\hat{\mathcal{S}}_v$.

Secondly, we consider the case that w is a Byzantine node, but Op_w is always a consistent value in the absMAC layer. With the similar proof, u finds the same Op_w from its own set $\hat{\mathcal{S}}_u$ and the sets $\hat{\mathcal{S}}_x$ from all the other $(f-1)$ honest nodes

x. Thus, we have Op_w appeared in both $\hat{\mathcal{S}}_u$ and $\hat{\mathcal{S}}_v$.

Thirdly, we consider the case that w is a Byzantine node that transmits inconsistent opinions in the absMAC layer. For example, w transmits $Op_w = 1$ in the round t_1 and transmits $Op_w = 0$ in the round t_2. According to the Claim 2, the opinion of w will be received by $\Theta(n/k)$ nodes if w broadcasts successfully. Thus, w's voting for opinion 0 in the round t_1 and voting for opinion 1 in the round t_2 will be witnessed by at least $\Theta(n/k)$ nodes separately. In the second execution of our absMAC layer (line 25 of Algorithm 2), all the nodes (including the honest nodes) exchange what they have received from others. Then, the inconsistent behavior of the Byzantine node will be caught. Considering that the Byzantine nodes prefer to mislead the consensus process without being be caught by the honest nodes. The Byzantine nodes will not choose to transmit the inconsistent opinion under the witness of $\Theta(n/k)$ nodes. In other words, execution of the absMAC layer twice to exchange $\mathcal{S}_v$ prevents the inconsistent opinions from Byzantine nodes.

Combining the results in the above three cases proves the Lemma 3. ∎

Since all the honest nodes v have the same set $\hat{\mathcal{S}}_v$ and choose the value of the major opinions in the set $\hat{\mathcal{S}}_v$ as the final opinion. The property of agreement has been satisfied. Besides, the number of Byzantine nodes is smaller than $n/3$. By choosing the major opinions from the set $\hat{\mathcal{S}}_v$, the validity of the final agreement can be satisfied.

C. Single Byzantine Jamming across Multiple Channels

In our model section, we assume that each Byzantine node can arbitrarily choose a channel to jam in each communication round. Based on this assumption, we design our absMAC algorithm, consensus algorithm and present the theoretical proofs. In this part, we further discuss a harder but more realistic case in which a Byzantine node can disrupt communications across multiple channels. A new parameter J is defined to quantify a Byzantine node's maximum jamming capability, which allows us to rigorously assess the worst-case scenario. In other words, f Byzantine node could interfere with normal communications across $J * f$ channels. In this case, there are remaining $k - J * f$ clean channels that can be used by honest nodes to deliver messages. Note that in our previous analysis, f Byzantine nodes at most jam f channels cooperatively, leaving $k - f$ clean channels. And the time complexity for ack. operation, prog. operation in absMAC layer and BFT-consensus problem are $O(\frac{kn}{k-f}\log n)$ rounds, $O(\frac{k}{k-f}\log n)$ rounds, and $O(\frac{kn}{k-f}\log n)$ rounds, respectively. In this setting when a Byzantine node can at most jam J channels, there are only $k - J * f$ clean channels left, and the efficiency for communication is reduced for $\frac{k-J*f}{k-f}$ times. With similar proofs, we can show that the time complexity for ack. operation, prog. operation in absMAC layer and BFT-consensus problem are $O(\frac{kn}{k-J*f}\log n)$ rounds, $O(\frac{k}{k-J*f}\log n)$ rounds, and $O(\frac{kn}{k-J*f}\log n)$ rounds, respectively. The constraints would be $f < n/3$ and $k - J * f > 0$, i.e., $f < \min\{n/3, k/J\}$.

VI. Performance in Simulation

In this section, we investigate the performance of our BFT consensus algorithm with various network parameters. Specifically, the number of rounds used by our algorithm to achieve a consensus is observed when the numbers of honest nodes, Byzantine nodes, and channels vary.

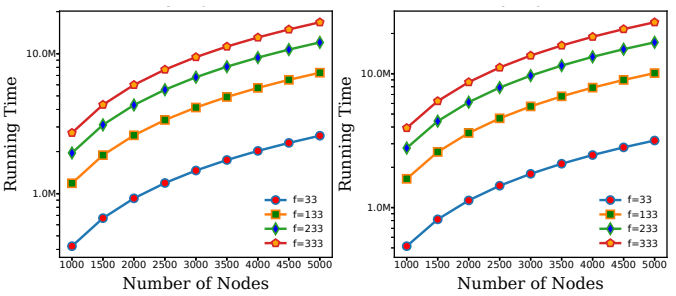
Parameter Settings. In our simulation, n nodes are randomly and uniformly distributed within a circular area with a radius of 100m. The transmission range R of each node is $200m$, ensuring a single-hop network environment. Initially, all the n nodes randomly and uniformly choose their opinions from $\{0, 1\}$. As for the malicious behaviors from the Byzantine nodes, ζ-fraction of Byzantine nodes cooperatively jam $\lfloor \zeta * f \rfloor$ channels, and the rest of Byzantine nodes choose to release the fake messages. Specifically, when a Byzantine node decides to jam a channel, a sufficient large jamming signal will be launched. When a Byzantine node w plan to deliver an opinion Op_v to another node u, the fake opinion $\bar{Op}_v$ will be used to replace the true opinion Op_v. By doing this, the Byzantine nodes mislead the consensus process. $m = k - \lfloor \zeta * f \rfloor$ is defined as the number of clean channels. For the SINR parameters, we have $\alpha = 3$ and $\beta = 2$.

TABLE IV: The parameters in simulation.

Para.	Definition	Value
n	Number of devices	$[1000, 5000]$
f	Number of Byzantine devices	$(0, 333]$
m	Number of clean channels in progress	$\{25, 50\}$
ζ	Parameter to depict Byzantine behaviors	$[0.1, 0.9]$
α	Path loss exponent in SINR model	3
β	Threshold in SINR model	2
R	Transmission range	200m

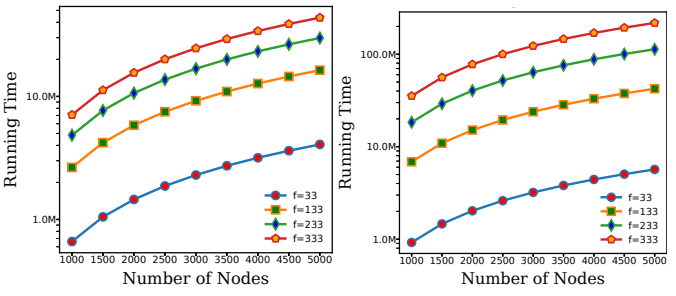
Numerical Results. The performance of our algorithm is presented in Fig. 3, in which the X-axes and Y-axes represent the number of nodes and the time used to achieve the consensus, respectively. Round is used to describe the unit of the running time. By observing the curves in the Fig. 3 with $n \in [1000, 5000]$, $f \in (0, 333]$ $m \in \{25, 50\}$, $\zeta \in [0.1, 0.9]$, the following results can be obtained.

- By fixing on the same m, f, and ζ, the running time of our consensus algorithm increases when n gets larger. For example, in Fig 3 (a) with $f = 233$, when n gets larger from 1000 to 5000, the running time increases from 1.96×10^6 to 1.21×10^7. This tendency verifies the theoretical time complexity $O(\frac{nk\log n}{k-f})$ of our algorithm.
- By fixing on the same n, m, and ζ, it takes a longer time to achieve the consensus when there are more Byzantine nodes. For example, by comparing the four curves in Fig 3 (a), the numbers of rounds with $f = 133$, $f = 233$, and $f = 333$ is about 1.82, 3.64, and 5.46 times larger than that with $f = 33$ when $n = 3000$. This is because more Byzantine nodes result in a longer time to achieve reliable communications and consensus.
- With the same n, f, and m, the running time of our algorithm increases when ζ gets larger. By comparing the four curves with $f = 133$ in Fig 3 (a)-(d), we can see that when $n = 1000$, the numbers of rounds are 1.19×10^6 with $\zeta = 0.1$, 1.64×10^6 with $\zeta = 0.4$, 2.65×10^6 with $\zeta = 0.7$, and 6.88×10^6 with $\zeta = 0.9$.



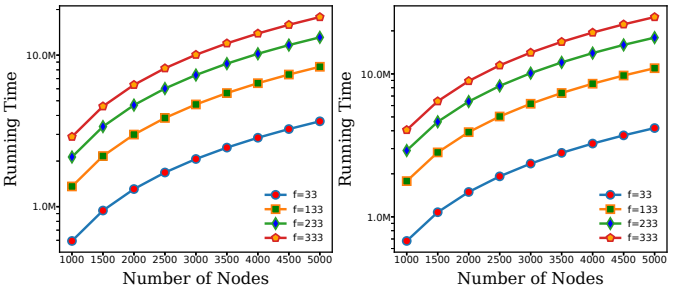
(a) $m = 25$ and $\zeta = 0.1$

(b) $m = 25$ and $\zeta = 0.4$



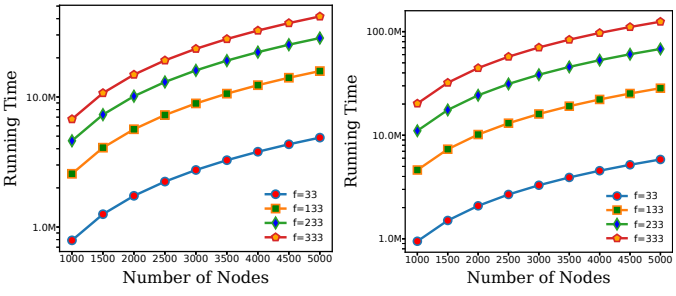
(c) $m = 25$ and $\zeta = 0.7$

(d) $m = 25$ and $\zeta = 0.9$



(e) $m = 50$ and $\zeta = 0.1$

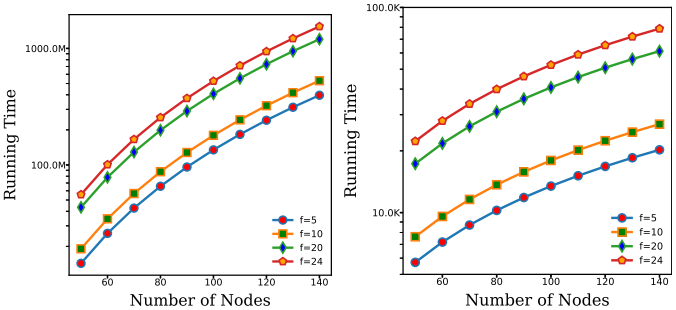
(f) $m = 50$ and $\zeta = 0.4$



(g) $m = 50$ and $\zeta = 0.7$

(h) $m = 50$ and $\zeta = 0.9$

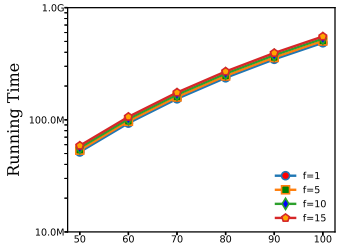
Fig. 3: The running time of our algorithm when the numbers of nodes, Byzantine nodes and clean channels vary.



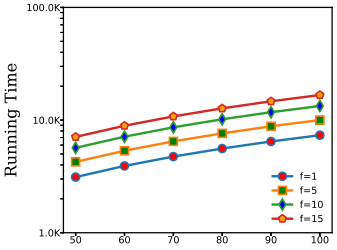
(a) Paxos

(b) Our work

Fig. 4: The running time of Paxos compared with our algorithm when the numbers of nodes, Byzantine nodes vary, $m = 50$ and $\zeta = 0.9$.



(a) PBFT



(b) Our work

Fig. 5: The running time of PBFT compared with our algorithm when the numbers of nodes, and Byzantine nodes vary, $m = 50$ and $\zeta = 0.1$.

From this observation, we can see that the jamming attack from Byzantine nodes has a heavier impact than the fake messages with respect to the running time.

- With the same n, f, and ζ, the running time of our algorithm decreases when there are more clean channels. For example, by comparing the curves with $f = 233$ in Fig 3 (d) and (h), we can see the running time is 1.84×10^6 rounds when $m = 25$ and $n = 1000$. Whereas, when m doubles, the number of rounds is 1.10×10^7, which is 0.4 times smaller than that in Fig 3 (d). This is because more clean channels improve the efficiency of communications in our absMAC layer.
- By fixing on the same n, m, and ζ, we compare the performance of our proposed BFT algorithm with the non-BFT Paxos algorithm [29] in the face of channel jamming. The numerical results are reported in Fig 4. According to the curves in Fig 4, we observe that Paxos exhibits a considerably high time cost reaching consensus, especially as the number of nodes scales. This is because Paxos requires atomic multicast and thus must contend with the implications of jamming at the communication layer. Given Paxos's requirement for a total order, this adds a significant time complexity to its operation in our context, effectively making a global consensus in $O(n^3 \log n)$. In contrast, our proposed algorithm demonstrates better scalability and resilience to jamming attacks, with the numbers of rounds required to reach consensus being about $1.22 * 10^4$, $1.67 * 10^4$, $1.70 * 10^4$, and $1.73 * 10^4$ times larger than that when $n = 140$ and $f = 5$, 10, 20, and 24, respectively.
- In Figure 5, we compare the performance of our proposed BFT consensus algorithm with the PBFT algorithm [9] in the face of channel jamming with $n \in [50, 100]$, $f \in [1, 15]$, $m = 50$ and $\zeta = 0.1$. According to the curves in Figure 5, we can observe that the PBFT requires an exceedingly high temporal cost to reach consensus, even with a small value of ζ and a small number of nodes. Compared with the results of PBFT, our proposed Byzantine consensus algorithm demonstrates better scalability and resilience to jamming attacks with the numbers of rounds required to reach consensus being about 8.3×10^3, 1.2×10^4, 1.6×10^4, 2.1×10^4, 2.7×10^4, and 3.3×10^4 times smaller when $f = 15$, and $n = 50$, 60, 70, 80, 90, 100, respectively. The comparative results further demonstrate the efficiency and resilience of our

proposed BFT consensus algorithm.

Overall, the numerical results verify the correctness and efficiency of our BFT consensus algorithm. When the number of nodes n, the number of Byzantine nodes f, and the jamming ratio ζ get larger, it takes longer time for our algorithm to achieve a consensus. Meanwhile, more clean channels $(k-f)$ are helpful to improve the communication efficiency and result in a smaller running time.

VII. Conclusion

This paper explores the Byzantine fault tolerant consensus problem in open wireless networks. Different from most of the previous works that require reliable communications between honest nodes, this study considers a more challenging Byzantine model in which the communications can be unstable due to the Byzantine attacks on physical channels. To make our BFT consensus algorithm elegant and efficient, we first implement a BFT abstract MAC layer with its acknowledgement and progress operations completed within $O(\frac{kn}{k-f}\log n)$ and $O(\frac{k}{k-f}\log n)$ rounds, respectively. Then, a BFT consensus algorithm is designed based on the implemented absMAC layer. We show that repeating the absMAC layer algorithm for constant times is enough for the honest nodes to achieve a BFT consensus. Both theoretical proofs and numerical results are presented to show the correctness and efficiency of our work. Implementing a similar BFT absMAC layer to solve the other complex problems in distributed computing and networking will be our work in the future.

Acknowledgement

This work was supported in part by the National Key R&D Program of China (No. 2023YFB2703600), National Natural Science Foundation of China (NSFC) under Grant 62102232, 62122042, and Shandong Science Fund for Excellent Young Scholars (No.2023HWYQ-007).

References

[1] A. Garcia-Saavedra and X. Costa-Pérez, "O-ran: Disrupting the virtualized ran ecosystem," *IEEE Communications Standards Magazine*, vol. 5, no. 4, pp. 96–103, Dec. 2021.

[2] H. Moniz, N. F. Neves, and M. Correia, "Byzantine Fault-Tolerant Consensus in Wireless Ad Hoc Networks," *IEEE Transactions on Mobile Computing*, vol. 12, no. 12, pp. 2441–2454, Dec. 2013.

[3] R. Guo, Z. Guo, Z. Lin, and W. Jiang, "A hierarchical byzantine fault tolerance consensus protocol for the internet of things," *High-Confidence Computing*, 2023.

[4] Y. Zou, L. Yang, G. Jing, R. Zhang, Z. Xie, H. Li, and D. Yu, "A survey of fault tolerant consensus in wireless networks," *High-Confidence Computing*, p. 100202, 2024.

[5] B. Awerbuch, R. Curtmola, D. Holmer, C. Nita-Rotaru, and H. Rubens, "ODSBR: An on-demand secure Byzantine resilient routing protocol for wireless ad hoc networks," *ACM Trans. Inf. Syst. Secur.*, vol. 10, no. 4, pp. 6:1–6:35, Jan. 2008.

[6] V. Drabkin, R. Friedman, and M. Segal, "Efficient Byzantine broadcast in wireless ad-hoc networks," in *DSN*, Jun. 2005, pp. 160–169.

[7] M. Yu, S. Kulkarni, and P. Lau, "A new secure routing protocol to defend Byzantine attacks for ad hoc networks," in *MICC*, vol. 2, Nov. 2005, p. 6 pp.

[8] R. Boichat, P. Dutta, S. Frølund, and R. Guerraoui, "Deconstructing paxos," *SIGACT News*, vol. 34, no. 1, pp. 47–67, Mar. 2003.

[9] M. Castro and B. Liskov, "Practical Byzantine fault tolerance," in *OSDI*, Feb. 1999, pp. 173–186.

[10] L. Lamport, "The part-time parliament," *ACM Trans. Comput. Syst.*, vol. 16, no. 2, pp. 133–169, May 1998.

[11] J.-P. Martin and L. Alvisi, "Fast Byzantine Consensus," *IEEE Transactions on Dependable and Secure Computing*, vol. 3, no. 3, pp. 202–215, Jul. 2006.

[12] N. Santoro and P. Widmayer, "Time is not a healer," in *STACS*, B. Monien and R. Cori, Eds., 1989, pp. 304–313.

[13] Santoro, Nicola and Widmayer, Peter, "Agreement in synchronous networks with ubiquitous faults," *Theoretical Computer Science*, vol. 384, no. 2, pp. 232–249, Oct. 2007.

[14] G. Chockler, M. Demirbas, S. Gilbert, C. Newport, and T. Nolte, "Consensus and collision detectors in wireless Ad Hoc networks," in *PODC*, Jul. 2005, pp. 197–206.

[15] M. Biely, J. Widder, B. Charron-Bost, A. Gaillard, M. Hutle, and A. Schiper, "Tolerating corrupted communication," in *PODC*, Aug. 2007, pp. 244–253.

[16] F. Borran, R. Prakash, and A. Schiper, "Extending Paxos/LastVoting with an Adequate Communication Layer for Wireless Ad Hoc Networks," in *SRDS*, Oct. 2008, pp. 227–236.

[17] B. Charron-Bost and A. Schiper, "The Heard-Of model: Computing in distributed systems with benign faults," *Distrib. Comput.*, vol. 22, no. 1, pp. 49–71, Apr. 2009.

[18] U. Schmid, B. Weiss, and I. Keidar, "Impossibility Results and Lower Bounds for Consensus under Link Failures," *SIAM J. Comput.*, vol. 38, no. 5, pp. 1912–1951, Jan. 2009.

[19] H. Moniz, N. F. Neves, M. Correia, and P. Veríssimo, "Randomization Can Be a Healer: Consensus with Dynamic Omission Failures," in *Distributed Computing*, I. Keidar, Ed., 2009, pp. 63–77.

[20] Y. Zou, D. Yu, J. Yu, Y. Zhang, F. Dressler, and X. Cheng, "Distributed Byzantine-Resilient Multiple-Message Dissemination in Wireless Networks," *IEEE/ACM Transactions on Networking*, vol. 29, no. 4, pp. 1662–1675, Aug. 2021.

[21] G. Jing, Y. Zou, D. Yu, C. Luo, and X. Cheng, "Efficient Fault-Tolerant Consensus for Collaborative Services in Edge Computing," *IEEE Transactions on Computers*, vol. 72, no. 8, pp. 2139–2150, Aug. 2023.

[22] S. Zhang and J. Lee, "Double-spending with a sybil attack in the bitcoin decentralized network," *IEEE Transactions on Industrial Informatics*, vol. 15, pp. 5715–5722, 2019.

[23] B. Kailkhura, Y. Han, S. Brahma, and P. Varshney, "Distributed bayesian detection in the presence of byzantine data," *IEEE Transactions on Signal Processing*, vol. 63, pp. 5250–5263, 2013.

[24] F. Kuhn, N. Lynch, and C. Newport, "The Abstract MAC Layer," in *Distributed Computing*, I. Keidar, Ed., 2009, pp. 48–62.

[25] L. Lamport, R. Shostak, and M. Pease, "The Byzantine Generals Problem," *ACM Trans. Program. Lang. Syst.*, vol. 4, no. 3, pp. 382–401, Jul. 1982.

[26] V. Hadzilacos and S. Toueg, "A modular approach to fault-tolerant broadcasts and related problems," Cornell University, Tech. Rep., 1994.

[27] L. Tseng and C. Sardina, "Byzantine Consensus in Abstract MAC Layer," in *27th International Conference on Principles of Distributed Systems (OPODIS 2023)*, ser. Leibniz International Proceedings in Informatics (LIPIcs), vol. 286, 2024.

[28] C. Newport and P. Robinson, "Fault-Tolerant Consensus with an Abstract MAC Layer," in *DISC*, U. Schmid and J. Widder, Eds., vol. 121, 2018, pp. 38:1–38:20.

[29] F. Borran, R. Prakash, and A. Schiper, "Extending paxos/lastvoting with an adequate communication layer for wireless ad hoc networks," in *2008 Symposium on Reliable Distributed Systems*, 2008.

[30] Y. Zou, M. Xu, J. Yu, F. Zhao, and X. Cheng, "Fault-Tolerant Consensus with NOMA in Mobile Networks," *IEEE Wireless Communications*, vol. 29, no. 3, pp. 80–86, Jun. 2022.

[31] D. Yu, Y. Zou, Y. Zhang, H. Sheng, W. Lv, and X. Cheng, "An Exact Implementation of the Abstract MAC Layer via Carrier Sensing in Dynamic Networks," *IEEE/ACM Transactions on Networking*, vol. 29, no. 3, pp. 994–1007, Jun. 2021.

[32] C. Newport, "Consensus with an abstract MAC layer," in *PODC*, Jul. 2014, pp. 66–75.

[33] M. M. Halldórsson, Y. Wang, and D. Yu, "Leveraging multiple channels in ad hoc networks," *Distrib. Comput.*, vol. 32, no. 2, pp. 159–172, Apr. 2019.

[34] Y. Guan and X. Ge, "Distributed attack detection and secure estimation of networked cyber-physical systems against false data injection attacks and jamming attacks," *IEEE Transactions on Signal and Information Processing over Networks*, vol. 4, pp. 48–59, 2018.

[35] O. Goussevskaia, Y. A. Oswald, and R. Wattenhofer, "Complexity in geometric SINR," in *MobiHoc*, 2007, pp. 100–109.

[36] G. Coulouris, J. Dollimore, and T. Kindberg, *Distributed Systems - Concepts and Designs (3. Ed.)*, Jan. 2002, p. 452.

[37] N. A. Lynch, *Distributed Algorithms*, 1996, p. 122.

[38] E. Gafni and D. Bertsekas, "Distributed algorithms for generating loop-free routes in networks with frequently changing topology," *IEEE Transactions on Communications*, vol. 29, no. 1, pp. 11–18, 1981.

[39] D. Dolev and H. R. Strong, "Polynomial algorithms for multiple processor agreement," in *Symposium on the Theory of Computing*, 1982.

[40] M. M. Halldórsson, S. Holzer, E. A. Markatou, and N. A. Lynch, "Leader election in SINR model with arbitrary power control," *Theor. Comput. Sci.*, vol. 811, pp. 21–28, 2020.

[41] D. Yu, L. Ning, Y. Zou, J. Yu, X. Cheng, and F. C. M. Lau, "Distributed spanner construction with physical interference: Constant stretch and linear sparseness," *IEEE/ACM Transactions on Networking*, vol. 25, no. 4, pp. 2138–2151, 2017.

[42] M. O. Rabin, "Randomized byzantine generals," in *SFCS*, 1983, pp. 403–409.

[43] A. Miller, Y. Xia, K. Croman, E. Shi, and D. Song, "The honey badger of bft protocols," in *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*, 2016, pp. 31–42.

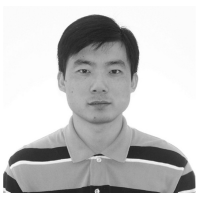
[44] T. Lorünser, B. Rainer, and F. Wohner, "Towards a performance model for byzantine fault tolerant services." in *CLOSER*, 2022, pp. 178–189.

[45] P.-L. Aublin, S. B. Mokhtar, and V. Quéma, "Rbft: Redundant byzantine fault tolerance," in *2013 IEEE 33rd international conference on distributed computing systems*. IEEE, 2013, pp. 297–306.

Guanlin Jing is currently pursuing a Ph.D. degree in computer science from Shandong University. He received the BS degree in electrical engineering from University of Minnesota, Twin Cities in 2019, and MS degree in computer science from George Washington University in 2021. He is currently in Prof. Xiuzhen Cheng's group, focusing on distributed computing and wireless networks.

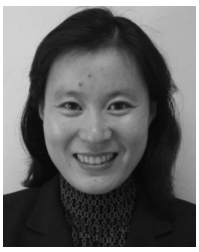
Yifei Zou received the B.E. degree in 2016 from Computer School, Wuhan University, and the PhD degree in 2020 from the Department of Computer Science, The University of Hong Kong. He is currently an Assistant Professor with the school of computer science and technology, Shandong University, Qingdao. His research interests include wireless networks, ad hoc networks and distributed computing.

Zuyuan Zhang Zuyuan Zhang received the B.S. degree from the Shandong University, China in 2023. He is currently a first year Ph.D. student and a Research Assistant in Electrical and Computer Engineering deparment at the George Washington University. His research interests include Reinforcement Learning.



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

Falko Dressler received his M.Sc. and Ph.D. degrees from the Dept. of Computer Science, University of Erlangen in 1998 and 2003, respectively. He is a full professor and Chair for Data Communications and Networking at the School of Electrical Engineering and Computer Science, TU Berlin. Dr. Dressler has been associate editor-in-chief for IEEE Trans. on Mobile Computing and Elsevier Computer Communications as well as an editor for journals such as IEEE/ACM Trans. on Networking, IEEE Trans. on Network Science and Engineering, Elsevier Ad Hoc Networks, and Elsevier Nano Communication Networks. He has been chairing conferences such as IEEE INFOCOM, ACM MobiSys, ACM MobiHoc, IEEE VNC, IEEE GLOBECOM. He authored the textbooks Self-Organization in Sensor and Actor Networks published by Wiley & Sons and Vehicular Networking published by Cambridge University Press. He has been an IEEE Distinguished Lecturer as well as an ACM Distinguished Speaker. Dr. Dressler is an IEEE Fellow as well as an ACM Distinguished Member. He is a member of the German National Academy of Science and Engineering (acatech). He has been serving on the IEEE COMSOC Conference Council and the ACM SIGMOBILE Executive Committee. His research objectives include adaptive wireless networking (radio, visible light, molecular communications) and embedded system design (from microcontroller to Linux kernel) with applications in ad hoc and sensor networks, the Internet of Things, and cooperative autonomous driving systems.



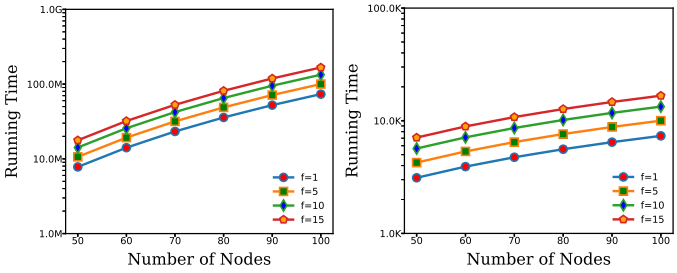
Xiuzhen Cheng received her MS and PhD degrees in computer science from University of Minnesota, Twin Cities, in 2000 and 2002, respectively. She was a faculty member at the Department of Computer Science, The George Washington University, from 2002-2020. Currently she is a professor of computer science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

APPENDIX

A. *Novelty of Our Work on Byzantine Consensus Algorithm*

The reliable communication provided by our absMAC layer only guarantees that the messages from the transmitter can be received by the receiver within a bounded delay (i.e. f_{ack} rounds). Our Byzantine consensus algorithm is based on reliable communications. Compared with reliable communications, atomic reliable communications not only require that all the nodes have reliable communications with each other, but also require that the total order of messages received by each node should be the same [26]. To the best of our knowledge, a large fraction of the existing Byzantine consensus algorithms are based on atomic reliable communications. For example, the Honey Badger in [43] and PBFT (Practical Byzantine Fault Tolerance) in [9]. Directly grabbing an existing Byzantine consensus algorithm and running it on our absMAC layer results in high time complexity. In general, we discuss two approaches to implement an existing Byzantine consensus algorithm that relies on atomic reliable communications on our absMAC layer with only reliable communications provided.

- **Approach I.** Directly run an existing Byzantine consensus algorithm on our absMAC layer regardless of its atomic communication requirement. However, the results in [26] have highlighted the importance of atomic communications for existing algorithms to reach consensus. The work in [44] showed that PBFT may fail when the atomic communications are destroyed by packet loss. In our simulation, we have presented the performance of our proposed algorithm and PBFT algorithm [9] when they are directly executed on our absMAC layer with only reliable communications provided, and reported the results in Fig. 6. According to the curves in Fig. 6 with $n \in [50, 100]$, $f \in [1, 15]$ and $\zeta = 0.1$, we can see that it takes PBFT an extremely high time cost to reach consensus, especially as the number of nodes increases. Whereas, our proposed Byzantine consensus algorithm has a better performance on the running time. Both of the results from [26], [44] and the numerical results from our simulation show that directly executing the existing Byzantine consensus algorithms that rely on atomic reliable communication on our absMAC layer results in high time cost.
- **Approach II.** Achieving atomic reliable communications on our absMAC layer with some additional schemes, and then implementing the existing Byzantine consensus algorithms on the achieved atomic reliable communication is a feasible approach. To guarantee atomic reliable communications, an intuitive solution is executing our absMAC layer for n times. Specifically, we assume that there are totally n nodes in a distributed system, denoted by the set $\{v_1, v_2, ..., v_n\}$. Then, in the i-th execution of our absMAC layer, only node v_i transmits its message $\mathcal{M}_i$ while the other nodes listen, with $i = 1, 2, ..., n$. By doing this, all the nodes can receive messages from others with the same order $\{\mathcal{M}_1, \mathcal{M}_2, ..., \mathcal{M}_n\}$, i.e., the atomic reliable communications can be guaranteed. Since an execution of our absMAC layer requires f_{ack} rounds. It takes $n * f_{ack}$ rounds to guarantee an atomic reliable communication for all nodes. According to [9], [43], [45], the Byzantine consensus algorithms PBFT, RBFT, and Honey Badger require constant times atomic reliable communications to achieve a consensus. Thus, the number of rounds required by PBFT, RBFT, and Honey Badger would be $\Theta(n * f_{ack})$ with $f_{ack} = O(\frac{kn}{k-f} \log n)$ rounds in our absMAC. Compared with PBFT, RBFT, and Honey Badger, our Byzantine consensus algorithm only needs constant times of reliable communications. Thus, its time complexity is $\Theta(f_{ack})$, which is $\Theta(n)$ times faster than that of PBFT, RBFT, and Honey Badger.
  There may be some potential solutions that can provide atomic reliable communications by executing our absMAC layer but is more simple or brief than the intuitive solution proposed in approach II. Even though we assume the existence of a brief solution that only needs constant times execution of our absMAC layer to provide atomic reliable communications, the time complexity of PBFT, RBFT, and Honey Badger on our absMAC layer would be $\Theta(f_{ack})$, which is the same with our Byzantine consensus algorithm. However, such a brief solution is not easy to find since the wireless channels are unreliable.



(a) PBFT in absMAC layer (b) Our Byzantine consensus

Fig. 6: The running time of PBFT and our algorithm with reliable communications in absMAC layer

In [27], the authors assume an absMAC layer to provide reliable (but not atomic) communications and design a Byzantine-resilient consensus algorithm based on the absMAC layer, which can tolerate up to $n/5$ Byzantine faults. Compared with the work in [27], the novelty of part one in our paper is clear. Because the work in [27] just assumes an absMAC layer to provide reliable communications despite Byzantine attacks while our work directly implements a Byzantine-resilient absMAC layer with provable time delay. Part two of our work is also a Byzantine-resilient consensus algorithm in the absMAC layer but with a different idea on algorithm design and a stronger threshold $n/3$ for Byzantine tolerance. The threshold $n/3$ has already been proved as the upper bound in Byzantine-resilient consensus problems with reliable communications [25].

Claim 4: For the node u with the maximum transmission power, and two arbitrary nodes $w, v \in W \setminus \{u\}$, we have $\frac{P_w}{d^\alpha(w,v)} < \frac{P_u}{d^\alpha(u,v)\beta n}$ with high probability in an arbitrary round t,

Proof: Recall that in the power selection scheme, each

node has $d = i + \lceil 2\log_2 n \rceil$ with probability $1/2^i$, randomly and uniformly selects an integer D from the interval $[2^d \log_2^2 n, 2^{d+1} \log_2^2 n)$, and finally adopts $P_{min} * D^{\gamma D}$ as its transmission power. Obviously, $d_u \geq d_w$. Otherwise, $P_u < P_w$, which directly violates the fact that $P_u \geq P_w$. Specifically, γ is set as a constant larger than $\max(1, s\alpha + 1 + \log \beta)$. Constants α and β are the SINR parameters. We normalize the shortest distance between any pair of nodes in the network as 1 and use R to denote the longest distance between any pair of nodes. $s = \log_n R$ is a constant since we assume that R can be bounded by a polynomial of n. The same setting can be found in [40] and [41]. In the next, we consider the cases of $d_u = d_w$ and $d_u > d_w$, respectively.

Case 1: $d_u = d_w$. In this case, $d_u = d_w = \Omega(\log n)$. This is because when n nodes select d values in our scheme, there are $(1/2)^{i+1}$ fraction of nodes having $d = i + \lceil 2\log_2 n \rceil$ in expectation. Thus, by applying the Chernoff bound, we can get $d_u = \Theta(\log n)$ with high probability since node u holds the largest d. Then, both of D_u and D_w are randomly and uniformly chosen from the interval $[2^{d_u} \log_2^2 n, 2^{d_u+1} \log_2^2 n)$. Note that $2^{d_u+1} \log_2^2 n - 2^{d_u} \log_2^2 n = \Omega(n^2 \log^2 n)$. It is easy to prove that when two integers are chosen from an interval with a length of $\Omega(n^2 \log^2 n)$, the two integers differ at least with a probability $1 - \frac{1}{n^2 \log^2 n}$. In other words, $D_n - D_{n-1} \geq 1$ with high probability. We normalize the minimum distance between any pair of nodes as unit 1 and assume that the maximum distance between any pair of nodes is $poly(n)$ (i.e.polynomial of n). Then, we have

$$\begin{aligned} \frac{P_u/d^\alpha(u,v)}{P_w/d^\alpha(w,v)} &= \frac{P_{min} \cdot D_n^{\gamma D_n} \cdot d^\alpha(w,v)}{P_{min} \cdot D_{n-1}^{\gamma D_{n-1}} \cdot d^\alpha(u,v)} \\ &\geq \frac{D_n^{\gamma D_n}}{D_{n-1}^{\gamma D_{n-1}} poly(n)} \\ &\geq D_n^\gamma \end{aligned} \tag{5}$$

Case 2: $d_u > d_w$, in which $D_w \leq 2^{d_w+1} \log_2^2 n < 2^{d_u} \log_2^2 n \leq D_u$. Thus,

$$\begin{aligned} \frac{P_u/d^\alpha(u,v)}{P_w/d^\alpha(w,v)} &= \frac{P_{min} \cdot D_u^{\gamma D_u} \cdot d^\alpha(w,v)}{P_{min} \cdot D_w^{\gamma D_w} \cdot d^\alpha(u,v)} \\ &\geq \frac{D_u^{\gamma D_u}}{D_w^{\gamma D_w} \cdot poly(n)} \\ &\geq D_n^\gamma \end{aligned} \tag{6}$$

Combining the two cases, we get $\frac{P_u/d^\alpha(u,v)}{P_w/d^\alpha(w,v)} = \Omega(n^{2\gamma} \log^{2\gamma} n) > \beta n$ if constant γ is sufficient large. ■

Explanation of Chernoff bound. Let $X_1, X_2, \cdots, X_n$ be independent or negatively associated non-negative random variables with $X_i \leq 1$. Moreover, let $X = X_1 + X_2 + \cdots + X_n$, and $\mu = E[X]$. For $\delta > 0$, it holds that

$$Pr[X \geq (1+\delta)\mu] \leq \left(\frac{e^\delta}{(1+\delta)^{1+\delta}} \right)^\mu.$$

For every $\delta \in (0, 1)$, it holds that

$$Pr[X \leq (1-\delta)\mu] \leq \left(\frac{e^{-\delta}}{(1-\delta)^{1-\delta}} \right)^{\mu/\gamma} \leq e^{-\delta^2 \mu/2}.$$

Application of Chernoff Bound. From the proof of Claim 3, we obtain a result that for arbitrary two honest nodes u and v, v receives the message from u with a probability of $\Theta(\frac{k-f}{nk})$. Let $X_i = 1$ be the event that v receives the message from u at round i. Otherwise, $X_i = 0$. For an interval from round 1 to round t, we have $X = \sum_{i=1}^t X_i$ as a random variable and each X_i be an independent Bernoulli random variable. Assuming that $t = \frac{ckn}{k-f} \log n$, the probability of v successfully receiving a message from u in each round is $p = c_1 * \frac{k-f}{nk}$ for some constant c_1, we have $\mu = \mathbb{E}[X] = \frac{ckn}{k-f} \log n * c_1 * \frac{k-f}{nk} = c * c_1 \log n$. To apply the Chernoff bound, we are looking for the probability $\Pr(X < 1)$, i.e., not receiving any messages in all rounds. Using the Chernoff bound with $\delta = 1/2$, we have

$$\Pr(X < 1) \leq \Pr(X < \mu/2) \leq e^{-\frac{\mu}{8}} = e^{-\frac{c * c_1 \log n}{8}} = n^{-c * c_1/8},$$

which proves the last sentence of Claim 3.