Distributed Byzantine-Resilient Multiple-Message Dissemination in Wireless Networks

Yifei Zou, *Student Member, IEEE,* Dongxiao Yu, *Member, IEEE,* Jiguo Yu, *Senior Member, IEEE,* Yong Zhang *Senior Member, IEEE*, Falko Dressler, *Fellow, IEEE,* Xiuzhen Cheng, *Fellow, IEEE*

Abstract—The byzantine model is widely used to depict a variety of node faults in networks. Previous studies on byzantine-resilient protocols in wireless networks assume reliable communications and do not consider the jamming behavior of byzantine nodes. Such jamming, however, is a very critical and realistic behavior to be considered in modern wireless networks. In this paper, for the first time, we integrate the jamming behavior of byzantine nodes into the network setting. We show that, in this much more comprehensive and harsh model, efficient distributed communication protocols can be still devised with elaborate protocol design. In particular, we developed an algorithm that can accomplish the basic multiple-message dissemination task close to the optimal solution in terms of running time. Empirical results validate the byzantine-resilience and efficiency of our algorithm.

Index Terms—Multiple-message dissemination, Byzantine-resilient, SINR model.

1 Introduction

With the rapidly increasing deployment of sensor networks and the Internet-of-Things, many distributed protocols are being designed to implement core wireless networking functionalities. At the same time, various kinds of faults have to be dealt with in wireless networks, such as transmission failures, tampering on transmitted messages, and malicious attacks from an adversary. Hence, the design of fault-tolerant, resilient protocols has been attracting more attention due to very high demands on reliable communication.

Since the quality of a fault model directly determines the performance of the designed protocols in reality, an accurate and comprehensive model to depict as many types of faults as possible is required. A common approach is assuming Byzantine models. Specifically, it is assumed that there are a group of byzantine nodes in the network, which can deviate arbitrarily from the protocol they are specified to execute to cause a wide variety of faults in the network, e.g., [5], [13], [15], [16].

- *Y. Zou (Corresponding author) and F.C.M. Lau are with Department of Computer Science, The University of Hong Kong, Hong Kong, P.R. China. E-mail:{yfzou,fcmlau}@cs.hku.hk*
- *D. Yu and X. Cheng are with Institute of Intelligent Computing, School of Computer Science and Technology, Shandong University, P.R. China. E-mail:{dxyu, fli, xzcheng}@sdu.edu.cn*
- *Y. Zhang is with Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences, Shenzhen, P.R. China. E-mail: zhangyong@siat.ac.cn*
- *J. Yu is with School of Computer Science and Technology, Qilu University of Technology (Shandong Academy of Science), Shandong Computer Center (National Supercomputer Center in Jinan), Jinan, School of Information Science and Engineering, Qufu Normal University, Rizhao, P.R. China. E-mail: jiguoyu@sina.com*
- *F. Dressler is with the School of Electrical Engineering and Computer Science, TU Berlin, Berlin, 10587 Berlin, Germany. E-mail: dressler@ccs-labs.org.*

Manuscript received MM DD, YYYY; revised MM DD, YYYY.

There have been many previous works (e.g., [1], [2], [5]) focusing on byzantine behavior tolerance in wireless networks. However, to the best of our knowledge, the previous wireless byzantine models assume reliable communications, and deal with faults above the MAC layer. However, due to the feature of wireless network that communications are commonly using a shared wireless channel, the byzantine nodes not only can generate faults on upper layers, but also can disrupt the transmissions between normal nodes by jamming the channel, e.g., through injecting noise on the channel [12], [18], [19], [24]. Obviously, ignoring the jamming behavior of byzantine nodes makes the byzantine models designed before fail to depict the faults in real wireless networks comprehensively, and consequently the protocols devised under those models may perform poorly in reality.

In this work, we, for the first time, integrate the jamming behavior into the wireless byzantine model. In particular, we assume that the nodes not only can deviate arbitrarily from the protocol, e.g., generate fake messages or stop help disseminate messages as a intermediate node, but also can jam the channel unpredictably. Note that most of previous jamming models have the strong assumption that the jammer has a very limited energy budget, so that it can only jam the channel in a small fraction of slots during a fixed interval [18]. However, this assumption may not be realistic since the jammer may have a very strong, reliable and sustained power supply. Thus, it can jam the channel arbitrarily and only leave some unpredictable slots unjammed. Furthermore, we adopt the hardest *reactive* setting, where the byzantine nodes know the protocol, all history information, and current states of nodes in the network. They can use these information to perform the so called "global jamming" attack [19], i.e., jamming the network at any round at will, by injecting strong enough noise on the channel to disrupt

all transmissions in the network. We believe that protocols devised under such a strong byzantine model can work well in reality.

We study a basic communication problem, multiple-message dissemination, which is to deliver the messages stored at k source nodes to all nodes in the network, in spite of the existence of byzantine nodes. The hard non-spontaneous setting is considered, where the nodes will not participate the dissemination process until receiving a message. In this setting, the nodes cannot construct a structure or implement a coordination scheme in prior, such as a backbone network, to facilitate the message dissemination, and hence it is harder to handle comparing the spontaneous case where nodes start the algorithm execution simultaneously.

We present a first distributed multiple-message dissemination algorithm, which is resilient to the strong byzantine setting as defined in this work. The algorithm is efficient, i.e., it can disseminate all messages within $\mathscr{T}(O((D+F)(\log n+F)+kF))$ rounds with high probability,[1] where $\mathscr{T}(A)$ denotes the number of rounds in the interval from the beginning of the algorithm execution that contains A clean rounds,[2] D, F, n, k are the diameter of the network, the upper bound on the number of byzantine nodes, the number of nodes, and the number of messages in network, respectively. In our definition, each round contains constant slots, each of which is the minimum time unit needed for one message transmission. The algorithm is close to the optimal solution, considering the lower bound for dissemination time of $\mathscr{T}(\Omega(D+\log n+k))$.

Roadmap. The remainder of the paper is organized as follows. Sec. 2 presents the related work. Sec. 3 introduces our byzantine model and the problem definition. The multiple-message dissemination algorithm is given in Sec. 4, followed by the analysis in Sec. 5. Sec. 6 shows the simulation results, and the paper is concluded in Sec 7.

2 Related work

Byzantine-resilient communications have been extensively studied in wireless networks, e.g, the cryptography (e.g., digital signatures) based protocols, which ensure authentication and integrity of transmitted messages, and the cryptography-free protocols [1], [3], [5], [10], [13], [15], [16], [17]. As for message dissemination, byzantine-resilient protocols for disseminating a single message have been studied in [5], [13], [15], [16]. However, all these results relied on reliable transmissions on wireless channels and did not consider the byzantine behavior in terms of transmission disruption via jamming. Some state of the art jamming works can be found in [6], [7], [12], [18], [19], [24]. However, none of them is considered under a byzantine-resilient framework. Hence, the performances of the above protocols in reality were not guaranteed.

As a fundamental building block for communications in upper layers, the multiple-message dissemination problem has already been extensively studied by the distributed computing community since the 1990s. Most works considered the radio network model which simplifies the interference into a binary and local phenomenon. Under this setting, the best known randomized distributed algorithms were presented in [4], [11], which can accomplish multiple-message dissemination in $\min\{O(k\log n\log\Delta+(D+n/\log n)\log n\log\Delta), O((k\Delta\log n+D)\log\Delta)\}$ rounds with high probability, where D and k are the diameter of the network and the number of messages, respectively. Recently, more attentions were turned to the realistic SINR model. In particular, randomized algorithms for the spontaneous case with running time of $O(D+k+\log^2 n)$ were presented in [21], [22], while for the non-spontaneous case, an algorithm with time complexity of $O((D+k)\log n+\log N^2)$ was given in [23]. All these results were derived without considering byzantine faults.

3 Models and Problem Definitions

A multi-hop wireless network with n agents arbitrarily placed possibly in a worst-case fashion is considered in our work. In our model, the network is regarded as a two dimensional Euclidean space, and agents are regarded as the nodes in the space. Specifically, we normalize the minimum distance between any pair of nodes to 1, and let $d(u,v)$ be the Euclidean distance between the pair of nodes u and v. All nodes can get access to the channel in network in each time slot, sending a message or receiving the signal in channel.

Communication Model. Initially, there is no prior or apparent structure in the network, which means that nodes know nothing about other nodes and can only try to communicate with neighbors by transmitting messages and listening. In our work, we divide the time in network into synchronized rounds, each of which contains constant synchronized slots. A slot is the minimum time unit needed for one message transmission depending on the message size, e.g., $50\mu s$ in IEEE 802.11.

A realistic and widely-adopted SINR (Signal-to-Interference-plus-Noise Ratio) model is used to depict the message reception between nodes in each slot. In the SINR model, it is assumed that the strength of a signal gets weaker with the distance between the transmitter and the receiver with a environment-determined path-loss exponent, and the strength of signals from different transmitting nodes are cumulative at each listening node. Specifically, a message sent by a node u can be correctly received by a node v if and only if the following defined SINR equation $SINR(u,v)\geq\beta$ holds.

$$SINR(u,v)=\frac{P_u * d(u,v)^{-\alpha}}{N+\sum_{w\in W}P_w * d(w,v)^{-\alpha}} \qquad (1)$$

where P_u is the transmission power of node u, $\alpha\in(2,6]$ is the path-loss exponent; N is the ambient noise in environment, W is the set of nodes simultaneously transmitting with u, $\sum_{w\in W}P_w * d(w,v)^{-\alpha}$ denotes the interference experienced by v when u transmits, and β is the hardware-defined threshold, and is larger than 1 in usual.

An uniform power setting assumes that nodes transmit with a same power, which is 'friendly' for algorithm design and analysis. However, it is hard for non-spontaneous

1. A high probability means with a probability $1-n^{-c}$ for some constant $c>0$.

2. A round is clean when no byzantine behavior occurs in the network during the round.

wake-up nodes to get an agreement on the transmission power in network without any prior structure. Thus, we assume that the transmission power between nodes can be various, to get close to the reality. Let $[P_{min}, P_{max}]$ be the interval where the transmission power of all nodes mathematically locates in. The only assumption is that P_{max}/P_{min} should be a constant. We say nodes u, v are neighbors if they can receive messages from each other according to the SINR equation.

By normalizing the minimum distance between two nodes to 1, we assume that the nodes in network are connected with respect to distance R.[3] Then, the SINR model gives an inherently upper bound $R_B = (\frac{P_{min}}{N\beta})^{1/\alpha}$ for R. Obviously, if $R > R_B$, it is impossible for the transmission between nodes u, v that can succeed, with $d(u, v) = R$ and $P_u = P_v = P_{min}$. However, the upper bound R_B is also very 'strict'. By setting $R = R_B$, for nodes u, v with previous assumption, it can be seen that the transmission between u, v only succeed when there are no other nodes simultaneously transmitting, which is nearly impossible in reality. Here we take a standard assumption which sets distance R to be slightly smaller than the upper bound R_B. Specifically, let $R = R_B/(1 + \epsilon)$, where ϵ is a positive constant.

Problem definition. Here we consider a multiple-message dissemination problem under the byzantine model. Given a network containing n nodes, which includes k source nodes, f byzantine nodes, and $(n - k - f)$ normal nodes. Initially, each source node s holds a source message $\mathcal{M}(s)$. The normal nodes wake up in a non-spontaneous mode, and know nothing about the source nodes and source messages after waking up. In each time slot, all source nodes and normal nodes can only exchange messages with neighbours by transmitting and listening. However, byzantine nodes can deviate arbitrarily from the protocol they are specified to execute and work in collusion, to cause any faults in message transmissions. In our problem, it is required to give a distributed protocol for source nodes and normal nodes, by executing which all source messages can be disseminated to each node (not including the byzantine nodes) in the network.

Byzantine Nodes. In our model, the byzantine nodes are considered with an adversarial and reactive assumption, i.e., each byzantine node knows the protocol, all history information and current states of nodes in network, and can deviate arbitrarily from the protocol to disrupt the transmissions.

From the view of MAC layer, byzantine nodes can not only arbitrarily transmit, but also jam the network. For any node v, if the interference experienced by v is too large to receive a transmission, node v is regarded as jammed. It is not assumed any constraint on the energy budget for the byzantine nodes. Thus, byzantine nodes can jam all nodes in network at any round at will, by adopting a sufficient large transmission power to make all nodes be under large enough interference, such that the transmissions fails, as shown in the SINR formula. This jamming mode is known as a "roundly-based global jamming" [18], [19]. In this global jamming setting, some clean rounds are necessary for message transmissions.

Denote a path from node s to w as $\mathcal{P}(s \to w) = \{v_0 \to v_1 \cdots \to v_{m-1} \to v_m\}$, in which $v_0 = s$, $v_m = w$, and $d(v_i, v_{i+1}) \leq R$, $i \in \{0, 1, ..., k-1\}$. $\mathcal{P}(s \to w)$ is a *safe path* if none of the nodes on the path is a byzantine node.

Necessary assumptions in our work are listed here.

- For any pair of source node s and normal node w, there is at least a safe path between them;
- Each node has a unique ID, and byzantine nodes cannot forge the IDs of other nodes;
- The source nodes are not too close with each other;

Obviously, the first assumption is inherently indispensable. The second assumption is also necessary as proved in [8]. As for the third assumption, if all source nodes are very close with each other, the message exchange between them will be easy. Then, the multiple messages in k different nodes can be regarded as k messages in one node, and the multiple-message dissemination problem degenerates as finding a single-message dissemination protocol and apply it for k times. To avoid this trivial case, we assume that for any pair of source nodes u and v, $d(u, v) \geq R$.

Knowledge and Capability of Nodes. Here, non-spontaneous wakeup mode is adopted for nodes, which is realistic and energy saving. Specifically, only the source nodes and byzantine nodes are active initially. The normal nodes wake up when receiving any message from neighbours. Each node has a unique identifier and is equipped with a full-duplex transceiver, i.e, in each time slot, a node can transmit, listen, or do both. The knowledge for each node is the same, including a logarithmic estimate on n, the SINR parameters α, β, P_{max}/P_{min}, and an upper bound F on the number of byzantine nodes. The nodes also know its location information when waking up, which is easy to access by the widely used GPS service. The physical carrier sensing is not required. Also, the nodes know nothing about their neighbours, the number of source nodes, and the accurate number of byzantine nodes.

4 Algorithm Design and Description

A distributed and randomized byzantine-resilient multiple-message dissemination algorithm in wireless network is given in this section. In general, our algorithm finishes the multiple-message dissemination task by two steps. The first step is to elect leaders for each local area in network. The elected leaders must be connected with each other and the transmission ranges of all leaders should cover all the nodes in network, which are called as the connectivity and coverage properties of leaders; The second step is message dissemination by the elected leaders, which guarantees that each source message is disseminated to all source nodes and normal nodes in network after a sufficient long time under the negative impact of byzantine nodes. The first and second steps are also called as leader election period and message dissemination period in sequel.

4.1 The challenges in algorithm design

Even through our algorithm only contains two steps and the high-level idea sounds brief, the algorithm design process

3. In usual, the ratio of the transmission range and the minimum distance between two nodes cannot be exponentially large, which means that R can be set to be bounded by $poly(n)$, and $\log R \in O(\log n)$.

for leader election step and message dissemination step is not easy, let along the challenges taken by arbitrary behaviors from byzantine nodes and wireless network, which makes our work non-trivial.

A kind of the challenges deserving to note are caused by the behaviors from byzantine nodes, including: jamming the network, malicious competition, and adversarial transmission. Since byzantine nodes can easily jam the network, our protocol should be inherently jamming-resilient when it is designed. Besides, the arbitrary jamming in non-spontaneous wake up mode makes the global clock in network no longer help to coordinate the operations of nodes. In particular, since byzantine nodes can arbitrary jam the network, nodes can't acquire information by itself on how many rounds were jammed in the past and how many steps the algorithm has been executed. Thus, a coordinate scheme is needed to replace the global clock. Second, since there are multiple-times leader election processes in each local network in our algorithm, the byzantine nodes can repeat to compete to the leaders by malicious competition, which will break the leader election processes. A scheme should be added in algorithm to detect and prevent the malicious competition from byzantine nodes. Third, byzantine nodes can mislead other nodes in network by adversarial transmissions, i.e. in any round, a byzantine node can drop the received source message and release a fake source transmission. Since it is also impossible to detect and prevent the adversarial transmission from byzantine nodes, the designed message dissemination step should also be fault-tolerant w.r.t. this point.

Another kind of challenges are the inherently problems in leader election and message dissemination step. First, the leaders should be carefully elected. Too many leaders elected will cause a flooding of transmission in the messages dissemination period, i.e., too many leaders transmitting the source messages causes too much interference, thus no transmission successes. Also, if the elected leaders are not enough to guarantee the connectivity and coverage properties of leaders, the message dissemination to all nodes will fail; Second, for each leader in message dissemination period, it has several messages to transmit. Thus the leaders should carefully select a transmission scheme to make sure the correctness and efficiency of message dissemination.

The final kind of challenges are the global and accumulative interference in MAC layer, which is seldom considered by the previous byzantine-resilient works.

The challenges mentioned above are the technical gaps we need to handle in this work. To be honest, the solutions to single challenge may can be found in previous works. For example, some previous works [9], [24] provide a leader election scheme with an asymptotically optimal performance without/with the jamming fault considered. However, our work is the first one considering the byzantine model in MAC layer, and integrating all the challenges together in one work. Considering multiple challenges in one solution sharply increases the level of difficulties in our algorithm design. let alone some traditional solutions for single challenge conflict with each other. The detail examples are not listed here because of page limitation. So it may not be surprising of designing solutions for each single challenge, but designing a protocol solving all the challenges together makes our work non-trival.

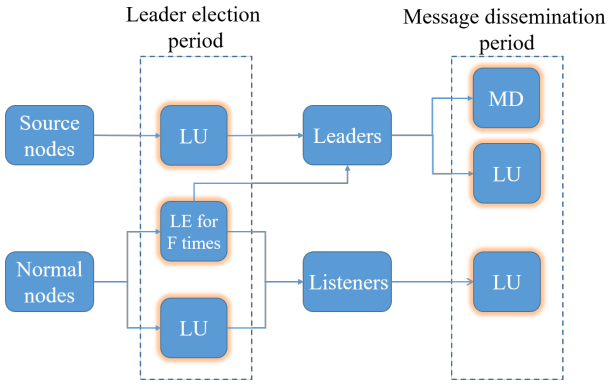


Fig. 1: The process of our algorithm. Listen and Update, Leader Election, and Message Dissemination are written as LU, LE and MD for short.

Solutions for challenges. Here, we briefly introduce how the challenges mentioned above are solved. The challenges taken by byzantine nodes include jamming, uncoordinate of algorithm, the malicious competition, and adversarial transmission. By each node accumulating the message received from neighbours, nodes can coordinate locally with its neighbours. By adopting the leader election for F times in each local area , our protocol survives from the malicious competition and adversarial transmission from byzantine nodes. Besides, the F times leader election in each local area guarantees the connectivity and coverage properties of leaders for messages dissemination, and also avoids the flooding of transmission. Thus, the inherently problems in leader election are also solved. Also the correctness and efficiency of the message dissemination are achieved by letting each leader select the fewest transmitted source message to transmit in each round. For the global accumulative interference in wireless network, a traditional griding and coloring scheme similar with [24] is adopted. Besides, the randomized transmission is helpful to avoid fraction of interference in transmission.

4.2 Detail description for algorithm

During the process of our algorithm, all nodes keep listening to collect the source messages. Meanwhile, all source nodes and part of the normal nodes are elected as the leaders in election period, to transmit the source messages in message dissemination period, as illustrated in Figure 1. Algorithm 1, 2, and 3 give the pseudo-code for source nodes and normal nodes respectively. Obviously, the source nodes are always the leader to disseminate source messages. For normal nodes, we use three states to schedule their executions.

- State $\mathbb{I}$. All normal nodes are initially inactive in state $\mathbb{I}$. A node in state $\mathbb{I}$ only do the listening in each round. When receiving a source message from neighbours, it becomes active and changes to state $\mathbb{A}$;
- State $\mathbb{A}$. A node in state $\mathbb{A}$ means that it has already waken up and been active. In the first round when a node is in state $\mathbb{A}$, it immediately executes the F times leader election. If elected as a leader, it changes to state $\mathbb{B}$. Otherwise, the node stays in state $\mathbb{A}$, and

Algorithm 1: *Functions called by a node v in color j :*

Function: *Listen and update ()*

Listen in current round;
if *receive any* $\mathcal{M}_u = \{In(s), In(u)\}$ *from node* u *and* $d(u, g_v) \leq (1 + \frac{2}{3}\epsilon)R$ **then**
  $t_v + +$;
  if $In(s)$ *in message* $\mathcal{M}_u$ *is new* **then**
    Add element $\{In(s), 0\}$ in set S_v;
  if $state_v = \mathbb{B}$ *and* v *is in same cell with* u **then**
    Replace the element $\{In(s), m\}$ in S_v with $\{In(s), m + 1\}$;
  if $state_v = \mathbb{I}$ **then**
    $state_v = \mathbb{A}$;
    if v *is in same cell with source node* s **then**
      $t_v = F * k_1 * \log n + 1$;

Function: *Message dissemination ()*

Select element $\{In(s), m\}$ with the smallest m from S_v;
Transmit $\mathcal{M}_v = \{In(s), In(v)\}$ in slot j;
if t_v *increased in current round* **then**
  replace the selected element in set S_v with $\{In(s), m + 1\}$;

Algorithm 2: MD for source nodes

Initially for Source node s:
ID: n_s; Location: (x_s, y_s); Source message: $\mathcal{M}(s)$;
$state_s = \mathbb{B}$; $In(s) = (n_s, x_s, y_s, t_u, \mathcal{M}(s))$; $t_v = 0$;
Source message set: $S_s = \{In(s), 0\}$;

In each round, for each source node s in color j:
Execute *Message dissemination ()*;
Execute *Listen and update ()*;

Algorithm 3: MD for normal nodes

Initially for normal node v:
Node ID: n_v; Location: (x_v, y_v); Set: $S_v = L_v = null$;
$state_v = \mathbb{I}$; $In(v) = (n_v, x_v, y_v, t_v, null)$;
$t_v = k_v = 0$;

For normal nodes v in state $\mathbb{I}$:
Execute *Listen and update ()* in each round;

For normal nodes v in color j, and state $\mathbb{A}$:
$flag_v = 1$;
for i *from* 1 *to* F **do**
  while $t_v \leq i * k_1 * \log n$ **do**
    Execute *Listen and update ()*;
    if $flag_v = 1$ **then**
      if $t_v = i * k_1 * \log n$ **then**
        $state_v = \mathbb{B}$; $k_v = i$; $L_v = L_v \cup \{v\}$;
        Transmit $\mathcal{M}''_v = In(v)$ in slot $c * c + j$;
      else
        Transmit message $\mathcal{M}'_v = In(v)$ with constant probability p in slot $c * c + j$;
        if *only listen, receive message* $\mathcal{M}'_u$ *in slot* $c * c + j$ *and* u *is not in set* L_v **then**
          $flag_v = 0$;
    else if *receive* $\mathcal{M}''_u$ *in slot* $c * c + j$ **then**
      $flag_v = 1$; $t_v = t_u$; $L_v = L_v \cup \{u\}$;
while $t_v > F * k_1 * \log n$ **do**
  Execute *Listen and update ()* in each round;

For normal nodes v in color j, and state $\mathbb{B}$:
while $t_v \geq 0$ **do**
  Execute *Listen and update () in each round*;
  if $t_v \ mod \ |L_v| = k_v$ **then**
    Execute *Message dissemination ()* in current round;

keeps listening to collect the source messages in the following rounds;

- State $\mathbb{B}$. A node in state $\mathbb{B}$ means that it has been elected as a leader during the F times leader election. In the first round when a node is in state $\mathbb{B}$, it immediately starts to disseminate the source messages in the following rounds.

As is mentioned above, the global coordination is ruined by the jamming behaviors from the byzantine nodes, we try to design a local coordinated protocol. Specifically, we grid and coloring the network by many cells, and make sure that nodes in each cell have a coordinated execution.

Preliminary on cells and nodes. Initially, we grid the network by square cells with size of $\frac{\epsilon R}{3\sqrt{2}} \times \frac{\epsilon R}{3\sqrt{2}}$. Point $(0, 0)$ is assumed to be the grid origin. Each cell includes its left side without the top endpoint, and its bottom side without the right endpoint, and does not include its right and top sides. For a cell with its bottom left corner locates at $(\frac{\epsilon R}{3\sqrt{2}} * i, \frac{\epsilon R}{3\sqrt{2}} * j)$, it has the coordinate of (i, j), and is denoted as $g(i, j)$, for $(i, j) \in \mathbb{Z}^2$. For a node v with coordinate (x, y) on the plane, it is in cell $g(i, j)$ when $i * \frac{\epsilon R}{3\sqrt{2}} \leq x < (i+1) * \frac{\epsilon R}{3\sqrt{2}}$ and $j * \frac{\epsilon R}{3\sqrt{2}} \leq y < (j + 1) * \frac{\epsilon R}{3\sqrt{2}}$. After the griding process, the cells and nodes are colored as follows: the cell $g(i, j)$ and nodes in cell $g(i, j)$ get the color $c*(i \mod c)+(j \mod c)$, c is a constant determined by constants α, ϵ and P_{max}/P_{min}.

Algorithm execution in each cell. Let's take the nodes in a non-empty cell g as an example to further describe our algorithm. Generally, each node in cell g executes function *listen and update* in each round to collect the source messages. If cell g contains any source node s in cell g, s becomes the only leader in g. Otherwise, normal nodes in g execute *leader election* for F times to try to elect F leaders. When a node becomes the leader, it will do the *message dissemination* in the following rounds, either it is a source node, or a normal node. The detail descriptions for *listen and update*, *leader election*, and *message dissemination* are given in the following.

Listen and update. All nodes in network keep listening in each round. For a node u and a cell g_v, $d(u, g_v)$ denotes the distance from node u to cell g_v. $d(u, g_v) = 0$ if u is in cell g_v. Otherwise, $d(u, g_v)$ is the maximum distance from u to any point in cell g_v. The node v in cell g_v only accepts the source message from u when $d(u, g_v) \leq (1 + \frac{2}{3}\epsilon)$ to keep the coordination in cell g_v. When receiving a source

message, nodes in state $\mathbb{I}$ wake up; the nodes in state $\mathbb{A}$ and $\mathbb{B}$ add the source message into their source message set if firstly receiving such a source message. It deserves to note that each node v counts the number of rounds when it successfully receives any source message from neighbours by parameter t_v, to estimate the clean rounds it experienced to coordinate with its neighbours, i.e. t_v can be regarded as a lower bound on the number of clean rounds v experienced.

leader election (LE). The normal nodes in state $\mathbb{A}$ take the leader election for F times, each of which consists of $\mathscr{T}(k_1 * \log n)$ rounds. During a leader election process, each node v in $\mathbb{A}$ transmits message $\mathcal{M}'_v = \{In(v)\}$ with constant probability p in slot $c * c + j$, j is the color of the node v. If v receives message $\mathcal{M}'_u = \{In(v)\}$ from node u in slot $c*c+j$, and u has never been a leader, v gives up the leader competition in current leader election. Each node v has a set L_v to record the elected leaders in its own cell. The message $\mathcal{M}'$ from an leader can not let v give up the current leader election to prevent the malicious competition from byzantine nodes. As we will proved later, within $\mathscr{T}(k_1 * \log n)$ rounds, there is only one normal node v still compete for leader in cell g w.h.p, and it becomes the leader. It deserves to note that nodes can't directly detect how many clean rounds it experienced. Fortunately, a node v can estimate the number of clean rounds it experienced by parameter t_v, as is mentioned above. Thus, when $t_v = i * k_1 * \log n$, the node v who still competes for leader becomes the i-th leader in the cell, and give an announcement $\mathcal{M}''_v$. When receiving $\mathcal{M}''_v$, the other nodes u in state $\mathbb{A}$ and from a same cell get a new coordination by setting $t_u = t_v$, and start to compete for the next leader until the end of F-th leader election. The F times leader election requires $\mathscr{T}(F * k_1 * \log n)$ rounds in total. When F times leader election completes in cell g, $\min\{F, |g|\}$ leaders are elected as leaders w.h.p., $|g|$ is the number of nodes in cell g.

Message dissemination (MD). When a node is elected as a leader, it starts to disseminate the messages in its source message set. The elected leaders in a cell transmit in each clean round in turn. Specifically, let's take a leader v in cell g_v as an example. v has the leader set L_v recording all the elected leaders in cell g_v. The leader v has a parameter k_v, which means that v is the k_v-th elected leader in g_v. Thus, v transmits a source message in each $(i * |L_v| + k_v)$-th clean round, $i = 0, 1, 2, ...$ Each time when v is to transmit a source message, it selects the source message, which is fewest transmitted by leaders in g_v, from its source message set S_v to transmit. In set S_v, each element has the format of $\{In(s), m\}$. Parameter m is used to record the number of times a source message $\mathcal{M}(s)$ transmitted by leaders in cell g_v in clean rounds. Each time, v selects the element $\{In(s), m\}$ with the smallest m to disseminate. With the help of message disseminations by leaders, all nodes in network can receive all the source messages finally even through under the negative inpacts of byzantine nodes.

By the above description, we can give a conclusion to our algorithm, which will be proved later in analysis.

Theorem 1. *Within $\mathscr{T}(O((D+F)(\log n+F)+kF))$ rounds, each node receives all the k source messages in network w.h.p.*

5 Algorithm analysis

We design and describe our algorithm by leader election period and message dissemination period. The analysis also corresponds to the two periods in the following:

Theorem 2. *Within $\mathscr{T}(D*k_1*\log n+2F*k_1*\log n)$ rounds, each non-empty cell g finishes the F times leader election w.h.p. If g contains a source node, the source node will become the only leader in g; Otherwise, $\min\{F, |g|\}$ leaders will be elected in cell g.*

Assume that at round T_1, all nodes in network finishes the leader election period.

Theorem 3. *Within $\mathscr{T}((D + 2k + 2F - 4) * F)$ rounds after round T_1, each non-byzantine node receives all the k source messages w.h.p.*

When combining the theorems for LE period and MD period, Theorem 1 can be directly proved. In the next, we prove the theorems for two periods one by one.

Proof for leader election period. Obviously, Theorem 2 holds when cell g contains a source node. So, we next consider the case that g only contains normal and byzantine nodes. Two basic lemmas are presented first to make the proof brief for reading. The proofs of which are given in the next subsection.

Lemma 1. *For a cell g with nodes in state $\mathbb{A}$, when all nodes in state $\mathbb{A}$ execute the leader election for $\mathscr{T}(k_1 * \log n)$ rounds, a node in state $\mathbb{A}$ is elected as the leader w.h.p.*

Lemma 2. *When a leader v transmits a source message in a clean round, all nodes within distance $(1+\frac{2}{3}\epsilon)R$ from v can receive the source message.*

We say a cell is active if it contains any nodes in state $\mathbb{A}$ or $\mathbb{B}$. Let t_g be the first round when cell g becomes active.

Lemma 3. *$\mathscr{T}(F * k_1 * \log n)$ rounds after t_g, cell g has $\min\{F, |g|\}$ leaders elected w.h.p.*

Proof. Obviously, all nodes in cell g become active at round t_g. Because in our algorithm, each node w only receives the message from node u when $d(u, g_w) \leq (1+\frac{2}{3}\epsilon)R$. So when a node w in cell g receives message from u, all the other nodes in g also receive the message from u according to Lemma 2. After t_g, all nodes in g start to do the F times leader election, each of which requires for $\mathscr{T}(k_1 * \log n)$ rounds according to Lemma 1. Hence, Lemma 3 is proved. □

With the lemmas above, we know that nodes in cell g finish the leader election period within $t_g + \mathscr{T}(F*k_1*\log n)$ rounds w.h.p. However, t_g may differ for different cell g. In the next lemma, an upper bound on $|t_g|$ is given.

Lemma 4. *For any cell g, it becomes active within $\mathscr{T}(D * k_1 * \log n + F * k_1 * \log n)$ rounds w.h.p.*

Proof. Here, we analysis how the inactive nodes in cell g receive a source message. Let node w be a normal node in cell g, $\mathcal{P}(v_0 \to v_m) = \{v_0 \to v_1 \cdots \to v_{m-1} \to v_m\}$ is a safe path from any source node s to normal node w, in which $v_0 = s$, $v_m = w$, and $\forall i \in [0, m-1]$, $d(v_i, v_{i+1}) \leq R$. W.l.o.g, we assume that each node v_i is in cell g_i, f_i is the number of byzantine nodes in g_i, for $i \in [0, m]$. Since v_0 is

initially active and always transmits the source message, at the end of $\mathscr{T}(1)$, v_1 become active according to Lemma 2, i.e., $t_{g_1} = \mathscr{T}(1)$. By executing leader election for f_i+1 times, at least a normal node is elected as the leader to disseminate the source message, i.e. at round $t_{g_1}+\mathscr{T}((f_i+1)*k_1*\log n)$, a non-byzantine leader v_1' in g_1 is elected out to transmit the source message according to Lemma 1. Also $d(v_1', g_2) \leq (1+\frac{2}{3}\epsilon)R$ because $d(v_1, v_2) \leq R$. Thus, all nodes in g_2 receives the source message from v_1' and become active according to Lemma 2, and $t_{g_2} \leq t_{g_1} + \mathscr{T}((f_i+1)*k_1*\log n)$. By continuing this analysis, we get

$$\forall i \in [0,m] : t_{g_i} \leq \begin{cases} \mathscr{T}(1) & i = 1 \\ t_{g_{i-1}} + \mathscr{T}((f_i+1)*k_1*\log n) & i > 1 \end{cases}$$

Thus, $t_{g_w} \leq \mathscr{T}(1) + \sum_{i=2}^{m} \mathscr{T}((f_i+1)*k_1*\log n)$. For any safe path $\mathcal{P}(v_0 \to v_m)$ from a source node s to the node w, we have $m \leq D$, and $\sum_{i=2}^{m} f_i < F$. Thus, $t_{g_w} \leq \mathscr{T}(D*k_1*\log n + F*k_1*\log n)$, and the lemma is proved. □

Directly combining Lemma 3 and 4, we prove Theorem 2.

Proof for message dissemination period. During MD period, leaders start to disseminate the source messages in their source message set, and the other nodes keep listening to collect the source messages from leaders. If a source message is from a source node, we say it is a true message. Otherwise, it is a fake message from byzantine nodes. It is very likely for a byzantine node to declare itself as a source node to release a fake message. Also, a byzantine leader can replace the true messages in its source message set by a fake message, and transmit the fake message. In our work, we assume that it is impossible for normal nodes to prevent and detect the fake messages. When receiving a message, the leaders only disseminate it no matter it is true or fake.

We first prove the connectivity between any node w and source node s through the elected non-byzantine leaders.

Lemma 5. *For any nodes w and source node s, they are connected by the elected non-byzantine leaders.*

Proof. According to the previous assumption in model, there is a safe path $\mathcal{P}(v_0 \to v_m) = \{v_0 \to v_1 \cdots \to v_{m-1} \to v_m\}$ between source node s and node w, in which $v_0 = s$, $v_m = w$, and $d(v_i, v_i+1) \leq R$ for $i \in [0, m-1]$. Assume that each node v_i is in cell g_i. Since F is an upper bound on the number of byzantine nodes, there must be at least one non-byzantine node v_i' elected as the leader in each cell g_i. Thus, for each safe path $\mathcal{P}(v_0 \to v_m) = \{v_0 \to v_1 \cdots \to v_{m-1} \to v_m\}$, there is a corresponding safe path $\mathcal{P}'(v_0 \to v_m) = \{v_0' \to v_1' \cdots \to v_{m-1}' \to v_m'\}$, in which $v_0' = s, v_m' = w, v_i'$ is the non-byzantine leader in cell g_i for $i \in [0, m-1]$. We also get $d(v_i', v_{i+1}') \leq d(v_i', g_{i+1}) \leq (1+\frac{2}{3}R)$ for $i \in [0, m-1]$. Since leaders in cell g_i disseminate the source messages in turn, at most within $\mathscr{T}(F)$ rounds, a non-byzantine leader v_i' transmits a source message. And the source message is received by all nodes in cell g_{i+1} according to Lemma 2. So, the safe path $\mathcal{P}'$ guarantees the connectivity for any pair of nodes s and w. □

The following Lemma 6 is given in [14], which presents the pipelining effect of the multiple-message broadcast process. Let F_{prog} denote the maximum number of rounds needed for a successful transmission. For a graph G, define $d_G(u,v)$ as the number of edges in the shortest path from u to v in G.

Lemma 6. *Assume that at round t_0, a node u receives a new message $\mathcal{M}$. Let v be a node at distance $d = d_G(u,v)$ from v. For integers $l \geq 1$, we define $t_{d,l} = t_0 + (d+2l-2)*F_{prog}$. Then for all integers $l \geq 1$, at least one of the following two statements is true:*

(i) v received the message $\mathcal{M}$ by the time $t_{d,l}$;

(ii) there exists a set M, $|M| = \min\{l,k\}$, such that every message in M has been received by v by the timeslot $t_{d,l}$;

Lemma 7. *For any node w, it receives all the k source messages within $T_1 + \mathscr{T}((D+2k+2F-4)*F)$ rounds.*

Proof. For the fake messages released by the byzantine nodes, they are also disseminated as the source messages by the leaders. Thus, there are at most $(k+F-1)$ source messages disseminated in network. To simplify the analysis, we assume that all leaders synchronously start the broadcast process from the timeslot T_1. Clearly, this assumption does not make the analysis of the completion time of the multiple-message dissemination worse, since some leaders have started the message dissemination period before T_1. In the analysis of Lemma 5, we get $F_{prog} = F$. Thus, combining with Lemma 6, within $T_1 + \mathscr{T}((D+2k+2F-4)*F)$ rounds, each node w receives all the $(k+F-1)$ messages disseminated in network, including the k true messages from source nodes. □

Thus, we finish the proof for Theorem 3.

5.1 Technical proofs

Proof for Lemma 1. Here, we consider a leader election process in cell g, which is in color j, and does not contain any source node. The analysis starts from the moment t when all nodes in state $\mathbb{A}$ start the leader election. Let set $\hat{A}$ be the nodes in cell g and in leader competition. $|\hat{A}| > 1$, otherwise, the proof for Lemma 1 already finishes. Then, we divide the nodes v in set $\hat{A}$ into classes $\{V_i : i = 0, 1, \ldots, \log\frac{\epsilon}{3}R\}$ according to the distance between v and its nearest neighbour u in set $\hat{A}$. Specifically, v is in the set V_i for $0 \leq i \leq \log\frac{\epsilon}{3}R - 1$ if $d(u,v) \in [2^i, 2^{i+1})$, and is in the set $V_{\log\frac{\epsilon}{3}R}$ otherwise. The above division is only used for analysis purpose and the nodes know nothing about it. The number of nodes in V_i reduces in each clean round when some nodes give up the current leader competition because of receiving competition messages from other nodes in $\hat{A}$. When nodes in all V_i for $i \in \{0, 1, \ldots, \log\frac{\epsilon}{3}R - 1\}$ are reduced to empty, only one node is left in set $V_{\log\frac{\epsilon}{3}R}$, and becomes the leader. The following analysis contains two steps: firstly it is proved that each set V_i reduces at least with a constant ratio with a constant probability in each clean round; Second we prove that with a sufficient large parameter k_1, at the end of $\mathscr{T}(k_1 \log n)$, each set v_i reduces to empty for $i = \{0, 1, \ldots, \log\frac{\epsilon}{3}R - 1\}$.

For $i \in \{0, 1, \ldots, \log\frac{\epsilon}{3}R\}$, we use $V_{<i}(t)$ to denote the sets of nodes in classes V_js for $j < i$ at the beginning of a round t. Let $n_i(t) = |V_i(t)|$ and $n_{<i}(t) = |V_{<i}(t)|$. We observe that when transmission probability p is a sufficient

small constant, constant fraction of nodes in set V_i experience with a limited interference from nodes in $V_{\geq i}(t)$; Also, the interference from nodes in $V_{<i}(t)$ is bounded with the assumption that $n_{<i}(r) \leq \frac{1-(2^{1-\alpha/2})}{2} n_i(r)$. Thus, in each clean round, with constant probability such nodes in V_i receive messages from other nodes in $\hat{A}$ and give up the current leader election. In this way, we prove the following Lemma 8.

Lemma 8. *At each clean round t_1, V_i reduces with a constant fraction γ with probability $1 - e^{\Omega(n_i(r))}$ for $i \in \{0, 1, \ldots, \log \frac{\epsilon}{3} R - 1\}$, when $n_{<i}(t_1) \leq \frac{1-(2^{1-\alpha/2})}{2} * n_i(t_1)$.*

Proof. Let $A(u, d)$ be the set of active nodes within distance d from u. The *exponential annulus* $E_t^i(u) = A(u, 2^{t+1}2^i) \backslash A(u, 2^t 2^i)$. An active node u in cell g is defined to be a *Sparse Node* if for every $t \in \{0, 1, ..., \log \frac{\epsilon}{3} R - 1\}$, $E_t^i(u) \cap V \leq 48 * 2^{t(\alpha/2+1)}$. $S_i \subseteq V_i$ is the largest subset of *Sparse Node*s in V_i and for any pair of nodes u, v in S_i, the distance $d(u, v) \geq (s+2)2^i$, where $s = (\frac{P_{max}}{P_{min}} \cdot \frac{3 \cdot 2^{2\alpha+7}\beta}{2^\alpha - \epsilon^\alpha/(1+\epsilon)^\alpha} \cdot \frac{1}{1-2^{1-\alpha/2}})^{\frac{1}{\alpha/2-1}}$.

Claim 1. *At any round r, for $i \in \{0, 1, ..., \log \frac{\epsilon}{3} R - 1\}$, if $n_{<i}(r) \leq \frac{1-(2^{1-\alpha/2})}{2} * n_i(r)$, then a constant fraction of the nodes in V_i are sparse.*

Proof. For a node $u \in V_i$, if for every $t \in \{0, 1, ..., \log \frac{\epsilon}{3} R - 1\}$, it holds that $|E_t^i(u) \cap V_{\geq i}| \leq 24 * 2^{t(\alpha/2+1)}$ and $|E_t^i(u) \cap V_{<i}| \leq 24 * 2^{t(\alpha/2+1)}$, we say u is an *excellent node*. Clearly, an excellent node must be a sparse node. We next show that a fraction of the nodes in V_i are excellent nodes, by which a lower bound on the fraction of the sparse nodes can be obtained.

We first show the condition of $|E_t^i(u) \cap V_{\geq i}|$ in an excellent node. Because the nodes in $V_{\geq i}$ have distance at least 2^i with each other, the disks centered at nodes in $V_{\geq i}$ with radius 2^{i-1} are disjoint. Considering any given annulus $E_t^i(u)$, using an area argument shown in the following (2), it can be shown that for each node $u \in V_i$, $|E_t^i(u) \cap V_{\geq i}| \leq 24 * 2^{2t}$, which is smaller than $24 * 2^{t(\alpha/2+1)}$.

$$\begin{aligned} &\frac{\pi(2^{t+1}2^i + 2^{i-1})^2 - \pi(2^t 2^i - 2^{i-1})^2}{\pi 2^{2(i-1)}} \\ &= 3 * 2^{t+2} * (2^t + 1) \leq 3 * 2^{2t+3} < 24 * 2^{t(\alpha/2+1)} \end{aligned} \tag{2}$$

Then, we consider $|E_t^i(u) \cap V_{<i}|$ for node $u \in V_i$. Fix i and t. Let Γ_t^i be the sum of the nodes in $E_t^i(u) \cap V_{<i}$ for all the nodes in V_i. Then, we have

$$\begin{aligned} \Gamma_t^i &= \sum_{u \in V_i} |E_t^i(u) \cap V_{<i}| = \sum_{u \in V_{<i}} |E_t^i(u) \cap V_i| \\ &\leq n_{<i}(r) * 24 * 2^{2t} \leq \frac{1-(2^{1-\alpha/2})}{2} * n_i(r) * 24 * 2^{2t} \end{aligned} \tag{3}$$

From (3) and the definition of excellent nodes, it is easy to see that there are at most $\frac{1-(2^{1-\alpha/2})}{2} 2^{t(1-\alpha/2)}$ fraction of the nodes in V_i that are not excellent ones in annulus $E_t^i(u)$ for each node $u \in V_i$, as otherwise the above inequality would be violated. Then, we sum up the number of non-excellent nodes at each annulus as follows, which is an upper bound on the number of non-excellent nodes in V_i.

$$\begin{aligned} &\sum_{t=0}^{\log \frac{\epsilon}{3} R - 1} n_i(r) * \frac{1-(2^{1-\alpha/2})}{2} * 2^{t(1-\alpha/2)} \\ =& n_i(r) * \frac{1-(2^{1-\alpha/2})}{2} * \sum_{t=0}^{\log \frac{\epsilon}{3} R - 1} (2^{1-\alpha/2})^t \\ \leq& n_i(r) * \frac{1-(2^{1-\alpha/2})}{2} * \frac{1}{1-(2^{1-\alpha/2})} = \frac{1}{2} n_i(r). \end{aligned}$$

Thus, with the assumptions in Claim 1, at least half of the nodes in V_i are sparse nodes. □

Claim 2. *For any V_i, $i \in \{0, 1, ..., \log \frac{\epsilon}{3} R - 1\}$, at least $\frac{1}{(2s+5)^2}$ fraction of the sparse nodes are in the set S_i.*

Proof. Because S_i is the largest subset of the sparse nodes that have distance $(s+2)2^i$ pairwise, the disks with radii $(s+2)2^i$ centered at nodes in S_i can cover all the sparse nodes in V_i. To get $|S_i|/|V_i|$, it suffices to upper-bound the number of sparse nodes covered by a node in S_i. This can be done using an area argument.

Now consider a node $v \in S_i$ and the sparse nodes in V_i within distance 2^i. Let D_v and D'_v be the disks centered at v that have radius $(s+2)2^i$ and $(s+\frac{5}{2})2^i$, respectively. Notice that each pair of the sparse nodes in D_v have distance at least 2^i. This means that the disks centered at these nodes with radii 2^{i-1} are disjoint, and all these disks are covered by D'_v. Then one can see that the number of sparse nodes in D_v is at most

$$\frac{\pi * ((s+2)2^i + 2^{i-1})^2}{\pi * (2^{i-1})^2} = (2s+5)^2$$

The Claim then follows. □

Claim 3. *At each un-jammed round r of $\mathfrak{P}_1$, for $i \in \{0, 1, ..., \log \frac{\epsilon}{3} R - 1\}$, a constant fraction of the nodes in S_i become inactive with probability of $1 - e^{-\Omega(|S_i|)}$.*

Proof. We firstly bound the probability under which $u \in S_i$ receives a message from its nearest neighbor in the same cell. Let $\mathcal{E}$ be the event that u listens and its nearest neighbor (in the same cell) transmits. Obviously, $Pr(\mathcal{E}) = p(1-p)$. Then, under the assumption that $\mathcal{E}$ occurs, we calculate the number of nodes receiving messages from their nearest neighbors.

According to the SINR model, the interference at each node $u \in S_i$ matters in message reception ended at the node u. Let T_i be the set of nearest neighbors of all nodes S_i. The interference can be divided into two components, namely the interference from the nodes in $S_i \cup T_i$ and that from the other nodes.

We first bound the interference from the nodes in $S_i \cup T_i$. Consider a node $u \in S_i$. Notice that each node in S_i has distance at least $(s+2)2^i$ from u and has distance with its nearest neighbor in the range $[2^i, 2^{i+1})$. Thus the nodes in $(S_i \cup T_i) \setminus \{u, v\}$ have distance at least $s * 2^i$ from u. Then the interference I_1 at u from the nodes in $(S_i \cup T_i) \setminus \{u, v\}$ can be bounded as follows:

$$I_1 = \sum_{t=\log s}^{\infty} |E_t^i(u)| \frac{P_{max}}{(2^i 2^t)^\alpha} \leq \frac{48 P_{max}}{2^{i\alpha}} \cdot \frac{1}{s^{\alpha/2-1}} \cdot \frac{1}{1-2^{1-\alpha/2}}. \tag{4}$$

In the next, we bound the interference from the nodes not in $S_i \cup T_i$. Generally speaking, we show that with a

moderate probability, a constant fraction of the nodes in S_i experience the interference that is caused by the nodes not in $S_i \cup T_i$ and is not large. Combining the previous results, we can finally prove the claim.

Let $\hat{I}(v)$ be the interference at the nodes in S_i that is caused by a node $v \notin S_i \cup T_i$. For a node $v \notin S_i \cup T_i$, $\hat{I}(v)$ can also be recorded as the sum of the interference on the nodes in $E_t^i(v) \cap S_i$ over all annulus. Using an area argument as before, it can be obtained that $|E_t^i(v) \cap S_i| \leq 24 * 2^{2t}$. Then

$$\begin{aligned}\hat{I}(v) &\leq \sum_{t=0}^{\infty} |E_t^i(v) \cap S_i| \frac{P_{max}}{(2^i 2^t)^\alpha} = \frac{P_{max}}{2^{i\alpha}} \sum_{t=0}^{\infty} \frac{|E_t^i(v) \cap S_i|}{2^{t\alpha}} \\ &\leq \frac{P_{max}}{2^{i\alpha}} \sum_{t=0}^{\infty} \frac{24 * 2^{2t}}{2^{t\alpha}} = \frac{24 P_{max}}{2^{i\alpha}} \sum_{t=0}^{\infty} \frac{1}{2^{t(\alpha-2)}} \\ &< \frac{24 * P_{max}}{2^{i\alpha}} (\frac{1}{1 - 2^{2-\alpha}}) \end{aligned} \tag{5}$$

Let $C_{max} = \frac{48}{1-2^{1-\alpha/2}}$ and we have $\hat{I}(v) < c_{max} P_{max}/2^{i\alpha}$. We next prove the conclusion that for any constant c_1, by setting $p = c_1/(4c_{max})$, with probability $1 - e^{-\frac{c_1^2}{24 c_{max}^2}|S_i|}$, at least half of the nodes in S_i experience the interference that is caused by the nodes not in $S_i \cup T_i$ and is not larger than $c_1 P_{max}/2^{i\alpha}$.

We prove the conclusion in two cases.

Case I. $c_1 \geq c_{max}$.

Consider a node $u \in S_i$. Let I_2 denote the interference experienced by u that are caused by the nodes not in $S_i \cup T_i$. Then

$$\begin{aligned} I_2 &\leq \sum_{t=0}^{\infty} |E_t^i(u)| \frac{P_{max}}{(2^t 2^i)^\alpha} = \frac{P_{max}}{2^{i\alpha}} \sum_{t=0}^{\infty} \frac{|E_t^i(u)|}{2^{t\alpha}} \\ &\leq \frac{P_{max}}{2^{i\alpha}} \sum_{t=0}^{\infty} \frac{48 * 2^{t(\alpha/2+1)}}{2^{t\alpha}} = \frac{48 P_{max}}{2^{i\alpha}} \sum_{t=0}^{\infty} \frac{1}{2^{t(\alpha/2-1)}} \\ &< \frac{48 P_{max}}{2^{i\alpha}} (\frac{1}{1 - 2^{1-\alpha/2}}) \leq c_{max} P_{max}/2^{i\alpha} \end{aligned}$$

Case II. $c_1 < c_{max}$.

We define a random variable x_v

$$x_v = \begin{cases} \hat{I}(v) 2^{i\alpha}/(c_{max} P_v) & \text{when node } v \text{ transmits} \\ 0 & \text{when node } v \text{ listens} \end{cases}$$

Then we have

$$\begin{aligned} \mathbb{E}\Big[\sum_{v \notin S_i \cup T_i} x_v\Big] &= \sum_{v \notin S_i \cup T_i} p * \hat{I}(v) 2^{i\alpha}/(c_{max} P_v) \\ &= p \sum_{v \notin S_i \cup T_i} \hat{I}(v) 2^{i\alpha}/(c_{max} P_v) \end{aligned}$$

For the case when $|S_i| c_1 P_{max}/2^{i\alpha+1} > \sum_{v \notin S_i \cup T_i} \hat{I}(v)$, this claim can be directly proved. For the other case when $|S_i| c_1 P_{max}/2^{i\alpha+1} \leq \sum_{v \notin S_i \cup T_i} \hat{I}(v) \leq |S_i| c_{max} P_{max}/2^{i\alpha}$, we can get $(c_1^2/8c_{max}^2)|S_i| \leq \mathbb{E}\Big[\sum_{v \notin S_i \cup T_i} x_v\Big] \leq c_1 P_{max} |S_i|/(4 P_{min} c_{max})$. Let $\mu = \mathbb{E}[\sum_{v \notin S_i \cup T_i} x_v]$. Notice that $x_v \in [0, 1)$. Then using the standard Chernoff bound for the set of independent random variables $\{x_v : v \notin S_i \cup T_i\}$, it follows that

$$\begin{aligned} &Pr\Big(\sum_{v \notin S_i \cup T_i} x_v \geq 2 * (c_1 P_{max} |S_i|/(4 P_{min} c_{max}))\Big) \\ \leq &Pr\Big(\sum_{v \notin S_i \cup T_i} x_v \geq 2\mu\Big) \leq e^{-\mu/3} \leq e^{-\frac{c_1^2}{24 c_{max}^2}|S_i|} \end{aligned}$$

Thus, we prove with probability at least $1 - e^{-\frac{c_1^2}{24 c_{max}^2}|S_i|}$,

$$\begin{aligned} \sum_{v \notin S_i \cup T_i} \hat{I}(v) &= \sum_{v \notin S_i \cup T_i} x_v * c_{max} P_v/2^{i\alpha} \\ &\leq (2 c_1 P_{max} |S_i|/(4 P_{min} c_{max})) * c_{max} P_v/2^{i\alpha} \\ &= c_1 |S_i| \frac{P_{max}^2}{P_{min}}/2^{i\alpha+1} \end{aligned}$$

Therefore it is impossible for more than half of the nodes in S_i to experience interference from the nodes not in $S_i \cup T_i$ larger than $c_1 \frac{P_{max}^2}{P_{min}}/2^{i\alpha}$.

Combining all the above results together and setting $c_1 = \frac{P_{min}^2}{P_{max}^2} * \frac{2^\alpha - \epsilon^\alpha/(1+\epsilon)^\alpha}{2^{2\alpha+1}\beta}$, we can prove that with probability at least $1 - e^{-\frac{c_1^2}{24 c_{max}^2}|S_i|}$, at least half of the nodes $u \in S_i$ have the interference not larger than $2 c_1 \frac{P_{max}^2}{P_{min}}/2^{i\alpha}$. Then, according to the SINR condition, u can receive a message from its nearest neighbor v as follows:

$$SINR(v, u) > \frac{P_{min}/2^{\alpha(i+1)}}{2 c_1 \frac{P_{max}^2}{P_{min}}/2^{i\alpha} + N} \geq \beta.$$

Note that the above analysis is based on the assumption that u listens and its nearest neighbor v transmits, which occurs with probability $p(1-p)$. Hence, under the condition that at least half of the nodes in $|S_i|$ can receive messages from their nearest neighbors, $p(1-p) * |S_i|/2$ nodes become inactive in expectation. Using the Chernoff bound, the Claim is then proved.

□

With the above claims, one can see that for $i \in \{0, 1, 2, ... \log \frac{\epsilon}{3} R - 1\}$, (1) when $n_{<i}(r) \leq \frac{1-(2^{1-\alpha/2})}{2} * n_i(r)$, $\frac{|S_i|}{|V_i|} \geq \frac{1}{2(2s+5)^2}$; (2) at each un-jammed round of $\mathfrak{P}_1$, with probability at least $1 - e^{\Omega(|V_i|)}$, more than $\frac{p(1-p)}{4}|S_i|$ nodes become inactive. Thus Lemma 8 is proved. □

Even with Lemma 8 illustrating the reduction tendency of set V_i in each clean round, the tendency between different rounds is still uncertain, not only because of the jamming, but also because nodes in $V_{<i}$ may join into the set V_i when their nearest neighbours give up the leader competition.

A series of vectors $\{m_i(t) : t \geq 0 \; and \; 0 \leq i \leq \log \frac{\epsilon}{3} R - 1\}$ is define as an upper bound on the reduction process between rounds for each set V_i.

$$\forall t \geq 0 : m_i(t) = \begin{cases} n/\gamma_1 & t \leq T_i \\ \lfloor m_i(t-1) * \gamma_2 \rfloor & t > T_i \end{cases}$$

Here, $\gamma_1 = 1 - \gamma$ and $\gamma_2 = \gamma_1 + \rho/(1-\rho)$, where ρ is a sufficiently small constant; and $T_i = i * h$ and $h = \lceil \log_{\gamma_2} \rho \rceil$.

Considering that we have the assumption $\log R \in O(\log n)$ from reality, by setting $\hat{T} = O(\log n)$, we have $m_i(\hat{T}) = 0$, $\forall 0 \leq i \leq \log \frac{\epsilon}{3} R - 1$. Define random events

$\mathcal{E}(j)$s for $j \geq 0$: $\mathcal{E}(j)$ occurs when $n_i(t) \leq m_i(j)$ for all $i \in \{0, 1, \ldots, \log \frac{\epsilon}{3}R - 1\}$ at some round t. Then, $\mathcal{E}(0)$ always occurs and when $\mathcal{E}(\hat{T})$ occurs, $n_i = 0$ for $i \in \{0, 1, \ldots, \log \frac{\epsilon}{3}R - 1\}$. We next analyse when $\mathcal{E}(\hat{T})$ occurs.

Lemma 9. *When $\mathcal{E}(j)$ occurs, and $n_i(t) \leq \frac{\gamma_1}{\gamma_2} m_i(j+1)$ at some round t, then $n_i(t+1) \leq m_i(j+1)$.*

Proof. If the network is jammed at round t, no transmission succeed and $n_i(t+1) = n_i(t) \leq \frac{\gamma_1}{\gamma_2} m_i(j+1) \leq m_i(j+1)$. If t is not a jamming round, the lemma is prove by considering two cases. Case 1: when $m_i(j) = n/\gamma_1$, $n_i(t+1) \leq n < m_i(j+1)$; case 2: when $m_i(j) < n/\gamma_1$, we get

$$n_i(t+1) \leq n_i(t) + \sum_{s=0}^{i-1} n_s(t) \leq \frac{\gamma_1}{\gamma_2} m_i(j+1) + \sum_{s=0}^{i-1} m_s(j)$$
$$\leq m_i(j)\gamma_2 - m_i(j)\frac{\rho}{1-\rho} + m_i(j)\frac{\rho}{1-\rho} = m_i(j+1)$$

So the lemma gets proved by considering all the cases. □

Lemma 10. *When $\mathcal{E}(j)$ occurs at a clean round t, $n_i(t+1) < m_i(j+1)$ occurs with probability at least $1 - e^{-\Omega(n_i(t))}$, where $i \in \{0, 1, \ldots, \log \frac{\epsilon}{3}R - 1\}$.*

Proof. Obviously, $m_i(j) = n/\gamma_1$, or $n_i(t) < \frac{\gamma_1}{\gamma_2} m_i(j+1)$ are the sufficient conditions to prove the lemma. In the next, we consider the remaining case that $m_i(j) < n/\gamma_1$ and $n_i(t) \geq \frac{\gamma_1}{\gamma_2} m_i(j+1)$.

Because $\mathcal{E}(j)$ occurs and $m_i(j) < n/\gamma_1$, mathematically we get $n_{<i}(t) \leq \frac{\rho n_i(t)}{\gamma_1(1-\rho)} \leq \frac{1-(2^{1-\alpha/2})}{2} * n_i(t)$, where the last inequality holds by setting the constant ρ to be small enough to make sure $\frac{\rho}{\gamma_1(1-\rho)} \leq \frac{1-(2^{1-\alpha/2})}{2}$. Then, in the un-jammed round t, by Lemma 8, with probability $1 - e^{-\Omega(n_i(t))}$, we have

$$n_i(t+1) \leq \gamma_1 n_i(t) + \sum_{s=0}^{i-1} n_s(t) \leq \gamma_1 m_i(j) + \sum_{s=0}^{i-1} m_s(j)$$
$$= \frac{\gamma_1}{\gamma_2} m_i(j+1) + \sum_{s=0}^{i-1} m_s(j) \leq m_i(j+1)$$

□

Since we already get the reduction speed for V_i when $\mathcal{E}(j)$ occurs, the next step is to see when $\mathcal{E}(j+1)$ occurs. Let a_1 be the constant behind the Ω notation in the probability guarantee in Lemma 10, and $\hat{c} = \max\{\frac{2\gamma_1/\gamma_2}{a_1(1-\gamma_2)}, 2\gamma_1/\gamma_2\}$.

Lemma 11. *$\mathscr{T}(\hat{c})$ rounds after $\mathcal{E}(j)$ occurs, $\mathcal{E}(j+1)$ occurs with probability at least $1/2$.*

Proof. For $i \in \{0, 1, \ldots, \log \frac{\epsilon}{3}R - 1\}$, when $m_i(j) = 0$ or $n_i = 0$, it is easy to get that $n_i \leq m_i(j) \leq m_i(j+1)$. For the case $m_i(j) \geq n_i > 0$, the probability that at least one n_i is larger than $m_i(j+1)$ after $\mathscr{T}(\hat{c})$ rounds is bounded by

$$\sum_{i=0}^{\log \frac{\epsilon}{3}R-1} e^{-2\gamma_1 n_i/(\gamma_2(1-\gamma_2))} \leq \sum_{i=0}^{\log \frac{\epsilon}{3}R-1} \gamma_2(1-\gamma_2)/(2\gamma_1 n_i)$$
$$\leq \sum_{i=0}^{\log \frac{\epsilon}{3}R-1} (1-\gamma_2)/(2m_i(j+1)) \leq \frac{1-\gamma_2}{2} \sum_{i=0}^{+\infty} \gamma_2^i \leq \frac{1}{2}$$

Hence, with probability at least $1/2$, $\mathcal{E}(j+1)$ occurs $\mathscr{T}(c_1)$ rounds after $\mathcal{E}(j)$ occurs, and the lemma is proved. □

By taking a Chernoff bound, $\mathcal{E}(\hat{T})$ occurs within $\mathscr{T}(\hat{c} * \log n)$ rounds with high probability. By setting the constant k_1 to be sufficiently larger than $\hat{c}$, we prove that finally a node v will be left in leader competition. If v is a normal node, it will become the leader according to the executed algorithm. If v is a byzantine node, it could choose to keep silent and not become the leader. However, as is presented in the following proof for Lemma 2, the latest leader-competition message from v will be recorded by all its neighbors. So, if v choose not to become the leader, all its neighbors can find that v is a byzantine node. To avoid being "caught" by all its neighbors, v can only choose to become the leader. Thus, the Lemma 1 gets proved.

Proof for Lemma 2. We focus on the transmission from any leader v to u with $d(v,u) \leq (1 + \frac{2}{3}\epsilon)R$ in a clean round. Assume that nodes u, v are in color j_1 and j_2 respectively. Since the current round is clean, the interference at node u from other simultaneously transmitting leaders determines whether u can receive the message from the leader v.

Centered at node u, we firstly set a series of circles $\{C_b : b \geq 2\}$, each of which has the radius of $(b-1)(c-1)*(\frac{\sqrt{2}\epsilon}{6}R)$. For any of $b \geq 2$, let annulus A_b be the space between circles C_b and C_{b+1}, and L_b be the set of leaders in color j_2 and located in A_b. Considering that in our MD period each cell at most has one leader transmitting the source message in each round, any two transmitting leaders in slot $c*c+j_2$ are separated by a distance at least $(c-1)*(\frac{\sqrt{2}\epsilon}{6}R)$ because of the $c*c$ coloring scheme. Hence, the circles centered at the transmitting leaders in A_b with radius of $(c-1)*(\frac{\sqrt{2}\epsilon}{12}R)$ are disjoint. Also extending the two sides of annulus A_b by $(c-1)*(\frac{\sqrt{2}\epsilon}{12}R)$, we can get these circles are in the annulus with distance from u between $(b-\frac{3}{2})(c-1)*(\frac{\sqrt{2}\epsilon}{6}R)$ and $(b+\frac{1}{2})*(c-1)*(\frac{\sqrt{2}\epsilon}{6}R)$. Then the number of transmitting leaders in slot $c*c+j_2$ at each set A_b is upper bounded:

$$\frac{\pi(\frac{\sqrt{2}\epsilon}{6}R)^2 * ((b+\frac{1}{2})^2(c-1)^2 - (b-\frac{3}{2})^2(c-1)^2)}{\pi((c-1)*(\frac{\sqrt{2}\epsilon}{12}R))^2} \leq 16 * b$$

Furthermore, the number of simultaneous transmitting leaders in C_1, is at most 5, including v. Thus, the interference T_u at node u caused by the leaders who simultaneously transmit with v is at most:

$$I_u \leq \frac{4P_{max}}{((\frac{\sqrt{2}\epsilon(c-1)}{6})R)^\alpha} + \sum_{b=2}^{\infty} \frac{16b * P_{max}}{((b-1)(c-1) * \frac{\sqrt{2}\epsilon}{6}R))^\alpha}$$
$$\leq (4 + 32 * \frac{\alpha-1}{\alpha-2}) * P_{max} * (\frac{\sqrt{2}\epsilon(c-1)}{6})^{-\alpha} * R^{-\alpha}$$

By setting $c = \lceil ((\frac{P_{max}}{P_{min}} * \frac{32\frac{\alpha-1}{\alpha-2}+4}{(1+\epsilon/2)^{-\alpha}-(1+\epsilon)^{-\alpha}})^{\frac{1}{\alpha}}) * \frac{3\sqrt{2}}{2\epsilon} + 1 \rceil$, transmission from v to u successes because of the SINR ratio:

$$SINR(v,u) \quad \geq P_{min} * d(v,u)^{-\alpha} / (N + I_u) \geq \beta.$$

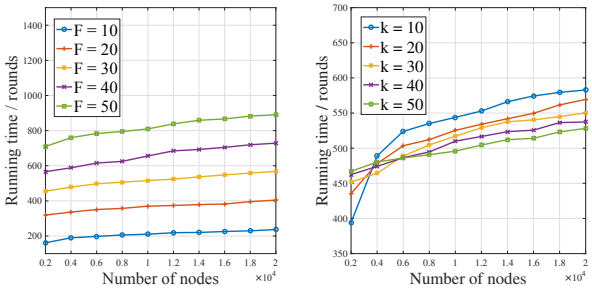
5.2 Lower bound proof

Theorem 4. *$\Omega(D + \log n + k)$ clean rounds is a lower bound for k-message dissemination under our byzantine model.*

Proof. Since byzantine nodes arbitrarily jam the network, any transmission in an un-clean round becomes unreliable.

TABLE 1: Parameters in simulation

$n \in \{0.2, 0.4, 0.6, 0.8, 1.0, 1.2, 1.4, 1.6, 1.8, 2.0\} * 10^4$			
$F \in \{1, 2, 3, 4, 5\} * 10$		$N = P_{min}/((1+\epsilon)^\alpha R^\alpha \beta)$	
$k \in \{1, 2, 3, 4, 5\} * 10$		$\zeta \in \{0.4, 0.8\}$	
$R = 30m$	$\alpha = 3$	$\beta = 1.5$	$P_{min} = R^\alpha \beta$
$p = 0.2$	$c = 10$	$\epsilon = 1.0$	$P_{max} = 4R^\alpha \beta$



(a). when $k = 30$ (b). when $F = 30$

Fig. 2: Running time of MMD when $\zeta = 0.4$

Considering a lower bound of $\Omega(\log n)$ reliable rounds for a successful transmission [20], we get the $\mathscr{T}(\Omega(D+\log n+k))$ rounds as the lower bound for disseminating k messages in a network with n nodes and diameter D. □

6 Simulation Result

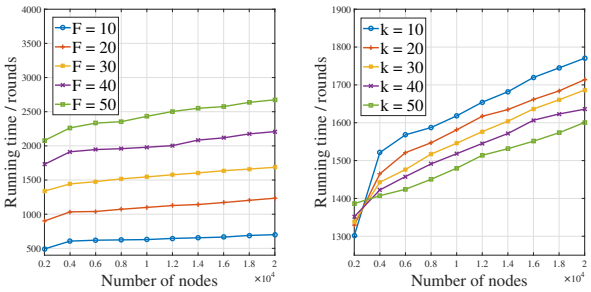
In this section, we investigate the empirical performances of our multiple-message dissemination algorithm under the byzantine model. Specifically, we consider the running time of our algorithm under the byzantine behaviors when the number of normal nodes, byzantine nodes, and source nodes vary.

Byzantine behaviors. Here, we assume that byzantine nodes jam each round in network with probability $\zeta \in (0, 1)$. Larger the ζ, more frequently the byzantine jamming occurs in network. Also, when receiving source messages, byzantine leaders take the worst behaviors to deal with the source messages. Specifically, in leader election period when leaders broadcast the source message to wake up the inactive normal nodes around, the byzantine leaders just keep silent. In message dissemination period, byzantine leaders disseminate the fake messages, to overload the transmissions in network.

Parameters. Basically, n nodes including k source nodes and $(F-1)$ byzantine nodes are randomly and uniformly distributed into a network with size of $300m \times 300m$. Each node randomly selects a transmission power between P_{min} and P_{max}, and has a constant transmission probability $p = 0.2$. Table 1 presents the parameters used in our simulation. Over 20 executions of the simulation have been carried out for each reported result. All experiments are conducted on a Linux machine with Intel Xeon CPU E5-2670@2.60GHz and 64 GB main memory, implemented in C++.

6.1 Algorithm Performance

We count the number of rounds used by our algorithm to disseminate all source messages to every node in network under various parameter settings. The extensive results on



(a). when $k = 30$ (b). when $F = 30$

Fig. 3: Running time of MMD when $\zeta = 0.8$

running time are presented in figure 2 and figure 3, in which the x-axes and y-axes represent the number of nodes in network and the running time T_M for multiple-message dissemination (MMD). The results in Figure 2 and Figure 3 are in the setting of $\zeta = 0.4$ and $\zeta = 0.8$, respectively. From (a) and (b) in Figure 2, we can see that T_M increases when n gets larger in any cases of F and k. Also, by comparing T_M when the value of n is fixed and F varies in Figure 2(a), it can be observed that with F getting larger, T_M linear increases, which is better result than that in our theoretical analysis. Figure 2(b) illustrates the relationship between k and T_M. According to our observation in simulation, when k is larger, it takes less time for nodes to wake up, but more time to collect all k source messages. Thus, in Figure 2(b), When the network is not so spare, the T_M is smaller when k is larger because more source nodes better facilitates the non-spontaneous wake-up process for inactive nodes .

Figure 3 depicts a similar tendency between T_M and parameters n, k, and F. Also, considering that the clean rounds used by our algorithm in expectation is $(1-\zeta) * T_M$, and the T_M in Figure 3 is about three times as large as that in Figure 2 when F, k and n are the same, we get the conclusion that a more frequent byzantine jamming will not increase the demand of our algorithm on the number of clean rounds, which indicates the byzantine resilience of our algorithm.

6.2 Summary

In conclusion, our algorithm is verified to be byzantine-resilient by the above simulation results. Also, the empirical running time of MMD in simulation indicates that the performance of our algorithm in reality should be better than the theoretical result w.r.t parameters k and F, since the theoretical result in analysis is only an upper bound in worst cases.

7 Conclusion

In this work, we proposed the first byzantine model which considers the jamming behavior of byzantine nodes in wireless networks, and is much more comprehensive and harsh than the byzantine models in previous works that rely on reliable communications. Under the new proposed byzantine model, a distributed and randomized byzantine-resilient algorithm was presented that can complete the basic multiple-message dissemination task within $\mathscr{T}(O((D+$

$F)(\log n + F) + kF))$ rounds w.h.p. Extensive simulations reveal the efficiency of our algorithm in reality.

The new proposed byzantine model and the protocol designed in this work shed some light for distributed fault-tolerant protocol design that may implement in real wireless networks. It will be interesting to investigate some other fundamental problems, such as consensus and link scheduling, under the proposed byzantine model.

REFERENCES

[1] E. Alchieri, A. Bessani, J. Fraga, F. Greve. Byzantine Consensus with Unknown Participants. In *OPODIS*, 2008.

[2] E. Alchieri, A. Bessani, F. Greve, J. Fraga. Knowledge Connectivity Requirements for Solving Byzantine Consensus with Unknown Participants. In *IEEE Trans. Dependable Sec. Comput.*, 15(2): 246-259, 2018.

[3] J. Augustine, G. Pandurangan, P. Robinson. Fast byzantine agreement in dynamic networks. In *PODC*, 2013.

[4] R. Bar-Yehuda, A. Israeli, A. Itai. Multiple communication in multihop radio networks. In *SIAM Journal on Computing*, 22: 875-887, 1993.

[5] S. Bonomi, G. Farina, S. Tixeuil. Reliable Broadcast in Dynamic Networks with Locally Bounded Byzantine Failures. In *SSS*, 2018.

[6] S. Bräuer, A. Zubow, S. Zehl, M. Roshandel, S. Mashhadi-Sohi. On practical selective jamming of Bluetooth Low Energy advertising. In *CSCN*, 2016.

[7] T.X. Brown, J.E. James, and A. Sethi. Jamming and Sensing of Encrypted Wireless Ad Hoc Networks, in *MobiHoc*, 2006.

[8] J. Douceur. The sybil attack. In *IPTPS*, 2002.

[9] J.T. Fineman, S. Gilbert, F. Kuhn, C.C. Newport. Contention Resolution on a Fading Channel. In *PODC*, 2016.

[10] R. Guerraoui, F. Huc, A. Kermarrec. Highly dynamic distributed computing with byzantine failures. In *PODC*, 2013.

[11] M. Khabbazian, F. Kuhn, D.R. Kowalski, N.A. Lynch. Decomposing broadcast algorithms using abstract MAC layers. In *DialM-PODC*, 2010.

[12] F. Klingler, F. Dressler. Poster Abstract: Jamming WLAN Data Frames and Acknowledgments using Commodity Hardware, in *INFOCOM Workshops*, 2019.

[13] C. Koo. Broadcast in radio networks tolerating byzantine adversarial behavior. In *PODC*, 2004.

[14] F. Kuhn, N.A. Lynch, C.C. Newport. The abstract MAC layer. In *Distributed Computing*, 24(3-4): 187-206, 2011.

[15] A. Maurer, S. Tixeuil. Byzantine broadcast with fixed disjoint paths. In *J. Parallel Distrib. Comput.*, 74(11): 3153-3160, 2014.

[16] A. Maurer, S. Tixeuil. Containing Byzantine Failures with Control Zones. In *IEEE Trans. Parallel Distrib. Syst.*, 26(2): 362-370, 2015.

[17] A. Maurer, S. Tixeuil, X. Défago. Communicating Reliably in Multihop Dynamic Networks Despite Byzantine Failures. In *SRDS*, 2015.

[18] A. Ogierman, A.W. Richa, C. Scheideler, S. Schmid, J. Zhang. Competitive MAC under adversarial SINR. In *INFOCOM* 2014.

[19] A. Ogierman, A.W. Richa, C. Scheideler, S. Schmid, J. Zhang: Sade: competitive MAC under adversarial SINR. In *Distributed Computing*, 31(3): 241-254, 2018.

[20] D. Yu, Q. Hua, Y. Wang, F.C.M. Lau. An O(log n) Distributed Approximation Algorithm for Local Broadcasting in Unstructured Wireless Networks. In *DCOSS* 2012.

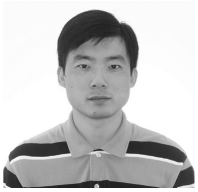
[21] D. Yu, Q. Hua, Y. Wang, H. Tan, F.C.M. Lau. Distributed multiple-message broadcast in wireless ad hoc networks under the SINR model. In *SIROCCO*, 2012.

[22] D. Yu, Q. Hua, Y. Wang, H. Tan, F.C.M. Lau. Distributed multiple-message broadcast in wireless ad hoc networks under the SINR model. In *Theor. Comput. Sci.*, 610: 182-191, 2016.

[23] D. Yu, Q. Hua, Y. Wang, J. Yu, F.C.M. Lau. Efficient distributed multiple-message broadcasting in unstructured wireless networks. In *INFOCOM*, 2013.

[24] Y. Zou, D. Yu, L. Wu, J. Yu, Y. Wu, Q. Hua, F.C.M. Lau. Fast Distributed Backbone Construction Despite Strong Adversarial Jamming. In *INFOCOM*, 2019.

Yifei Zou received the B.E. degree in 2016 from Computer School, Wuhan University. He is currently a PhD student in Department of Computer Science, The University of Hong Kong. His research interests include wireless networks, ad hoc networks and distributed computing.

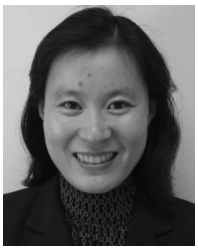


Dongxiao Yu received the BSc degree in 2006 from the School of Mathematics, Shandong University and the PhD degree in 2014 from the Department of Computer Science, The University of Hong Kong. He became an associate professor in the School of Computer Science and Technology, Huazhong University of Science 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.

Jiguo Yu received his Ph.D. degree in School of mathematics from Shandong University in 2004. He became a full professor in the School of Computer Science, Qufu Normal University, Shandong, China in 2007. Currently he is a full professor in Qilu University of Technology (Shandong Academy of Sciences), Shandong Computer Science Center (National Supercomputer Center in Jinan), and a professor in School of Information Science and Engineering, Qufu Normal University. His main research interests include privacy-aware computing, wireless networking, distributed algorithms, peer-to-peer computing, and graph theory. Particularly, he is interested in designing and analyzing algorithms for many computationally hard problems in networks. He is a senior member of IEEE, a member of ACM and a senior member of the CCF (China Computer Federation).

Yong Zhang received his PhD degrees in Computer Science from Fudan University, China, in 2007. Currently he is a professor in Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences. Prior to joining SIAT, he has worked as Post-Doctoral Fellow in TU-Berlin and senior research associate in the University of Hong Kong. His research interests include design and analysis of algorithms, graph algorithms, online algorithms, distributed computing, etc.

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 M.S. and Ph.D. degrees in computer science from the University of Minnesota – Twin Cities in 2000 and 2002, respectively. She is a professor in the School of Computer Science and Technology, Shandong University. Her current research interests include cyber physical systems, wireless and mobile computing, sensor networking, wireless and mobile security, and algorithm design and analysis. She has served on the editorial boards of several technical journals and the technical program committees of various professional conferences/workshops. She also has chaired several international conferences. She worked as a program director for the US National Science Foundation (NSF) from April to October in 2006 (full time), and from April 2008 to May 2010 (part time). She received the NSF CAREER Award in 2004. She is Fellow of IEEE and a member of ACM.