# Implementation and Experimental Evaluation of a Frog-Chorus-Based Autonomous Decentralized Collision Avoidance Method for LoRa Networks


Dai Kojima
*Department of Electrical Engineering*
*Tokyo University of Science*
Tokyo, Japan
4325531@ed.tus.ac.jp

Hiroyuki Yasuda
*International Research Center for Neurointelligence*
*The University of Tokyo*
Tokyo, Japan
yasuda@g.ecc.u-tokyo.ac.jp

Song-Ju Kim
*SOBIN Institute LLC*
Hyogo, Japan
kim@sobin.org

Maki Arai
*College of Engineering*
*Shibaura Institute of Technology*
Tokyo, Japan
m-arai@shibaura-it.ac.jp

Jin Nakazato
*Department of Electrical Engineering*
*Tokyo University of Science*
Tokyo, Japan
jin-nakazato@rs.tus.ac.jp

Mikio Hasegawa
*Department of Electrical Engineering*
*Tokyo University of Science*
Tokyo, Japan
hasegawa@ee.kagu.tus.ac.jp



***Abstract*—Low-power wide-area (LPWA) technologies for the Internet of Things (IoT) enable long-range and long-term data acquisition with minimal energy consumption. However, as the number of devices transmitting within the same coverage area increases, packet collisions become more frequent, leading to degraded communication stability and efficiency. Conventional collision avoidance schemes are not well suited to high-density and dynamically changing IoT environments, and often fail to meet the low-power requirements of battery-operated nodes. To address this issue, we previously proposed an autonomous and decentralized collision avoidance scheme based on the Kuramoto model with smallest-phase-difference intermittent inhibitory coupling, and demonstrated its ability to reduce the packet error rate (PER) on LoRa devices. Nevertheless, the Kuramoto model exhibits only a small phase response for small phase differences, resulting in slow separation of transmission timings and prolonged collision occurrences. In this study, we propose an autonomous decentralized collision avoidance method based on the frog chorus model, whose phase response is strong for small phase differences and weak for large phase differences. The contributions of this paper are threefold: (i) we design and formulate a frog-chorus-based inhibitory coupling rule that accelerates the separation of transmission timings in dense networks, (ii) we conduct a fair simulation-based comparison using normalized coupling strengths and show substantially faster convergence than the Kuramoto-based approach, and (iii) we implement the proposed algorithm on 50 LoRa devices and validate its effectiveness through PER measurements, demonstrating its applicability to large-scale IoT networks.**



***Index Terms*—LoRa, TDMA, nonlinear oscillator, collision avoidance, synchronization, autonomous decentralized protocol, Frog Chorus Model.**


## I. Introduction

With the widespread deployment of the Internet of Things (IoT), the number of simultaneously operating sensor nodes and edge devices has been increasing [1], [2]. In particular, in large-scale networks with dynamically changing topologies, such as condition monitoring of factory equipment and wide-area sensing in urban infrastructure, a large number of devices need to share the same channel for communication [3]–[6]. In such high-density environments, packet collisions are unavoidable, and it has been pointed out that they lead to a significant degradation in communication quality [7].

Conventionally, representative collision avoidance schemes, such as Carrier Sense Multiple Access/Collision Avoidance (CSMA/CA) and Time Division Multiple Access (TDMA), have been widely used. However, the performance of CSMA/CA degrades in high-density environments because channel utilization decreases, which leads to longer waiting times for transmission and a higher probability of packet collisions even when carrier sensing is performed. In contrast, although TDMA guarantees collision-free communication by design, it relies on static or centrally controlled scheduling and therefore cannot respond effectively to dynamic IoT environments where the number of active devices frequently changes, making it unsuitable for autonomous and decentralized operation [8]. Furthermore, many IoT devices are assumed to be battery powered, and both low power consumption and long-term operation are required. Consequently, there are an increasing number of cases in which existing schemes alone cannot ensure sufficient performance, such as maintaining low packet error rates, ensuring timely data transmission, and sustaining stable communication efficiency under varying network densities.

To address this issue, methods that exploit autonomous and decentralized synchronization phenomena to distribute transmission timings have attracted considerable attention [9]–[11]. In particular, a method that applies intermittent inhibitory coupling with the smallest phase difference based on the Kuramoto model to communication control has been proposed,

and implementation experiments using LoRa devices have reported a reduction in packet error rate (PER) [12]. Because each node adjusts its transmission phase based only on local information, this method offers high adaptability to dynamic networks and low power consumption, which are major advantages of this method.

However, in the Kuramoto model, the magnitude of the phase update becomes very small when the phase difference is small. Consequently, it takes a long time for the phases of devices that are initially close to each other to separate sufficiently, and in high-density environments, the problem of delayed collision avoidance remains. Therefore, to suppress collisions more rapidly, a coupling design is required in which the repulsive interaction becomes stronger as the phase difference decreases and weaker as the phase difference increases.

A model that satisfies this requirement is the frog chorus model, which was originally designed for wireless sensor networks (WSNs) [13]. The frog chorus model is inspired by the calling behavior of frogs and features a nonlinear phase response characteristic. By generating strong repulsion when the phases are close, the transmission timings among the nodes can be quickly separated.

In this study, we implement the frog chorus model on LoRa devices and experimentally verify the effectiveness of a collision avoidance scheme that overcomes the limitations of conventional Kuramoto-model-based approaches. In addition, through simulations, we compare the speed of convergence to desynchronization between the proposed method and the Kuramoto method. Furthermore, we construct a large-scale testbed consisting of up to 50 LoRa devices and evaluate the collision suppression performance of the proposed scheme.

The remainder of this paper is organized as follows. Section II introduces the theoretical background of the frog chorus model. Section III describes our model improvements that adapt the frog chorus model for IoT environments. Section IV presents simulation results comparing the conventional Kuramoto-based method and the proposed frog chorus-based method. Section V presents the performance evaluation in a real experimental environment and discusses the results. Finally, Section VI concludes this paper and outlines future research directions.

## II. A Collision Avoidance Method for IoT Devices Using the Frog Chorus Model

This section describes an autonomous decentralized method that applies the anti-phase synchronization phenomenon of nonlinear oscillators, which has been proposed in WSNs to suppress wireless communication collisions among IoT devices. This method is based on the frog chorus model and incorporates several modifications to adapt it to IoT environments. In the following, we explain each stage of the model modifications and their mathematical background.

### A. Description of the Frog Chorus Model

The frog chorus model is a nonlinear phase oscillator model inspired by the calling behavior of frogs. It exhibits a characteristic phase response, in which strong repulsion appears when the phase difference of the oscillators is small. In this model, the temporal evolution of the phase $\theta_i$ of the $i$th oscillator among $N$ oscillators coupled to the other oscillators is given by the following equation [13],

$$\frac{d\theta_i}{dt} = \omega_i + \frac{K_l}{N} \sum_{\substack{j=1 \\ j \neq i}}^{N} \frac{-1}{\pi} \left( \left( (\theta_j - \theta_i) \bmod 2\pi \right) - \pi \right) \quad (1)$$

where $\omega_i$ denotes the natural angular frequency of oscillator $i$, and $K_l$ represents the coupling strength between the oscillators, and $x \bmod 2\pi$ is defined as the unique value in $[0, 2\pi)$ such that $x = 2\pi q + (x \bmod 2\pi)$ for some integer $q$. Figure 1 illustrates the phase difference and the corresponding rate of phase change induced by the coupling function in Eq. (1). For collision avoidance, we exploit the property of anti-phase synchronization under inhibitory coupling with $K_l < 0$, in which the phases are evenly distributed.

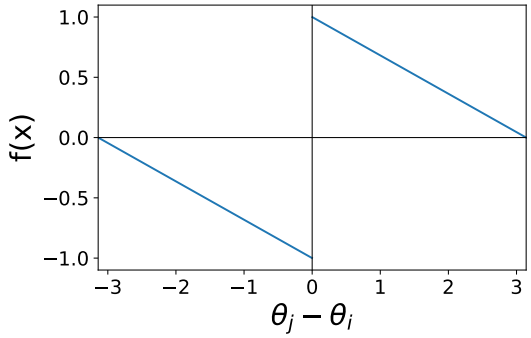



Fig. 1: Coupling Function for the Phase Difference in (1)

### B. Description of the assumed IoT environment

In this study, we assume a periodic-reporting IoT system in which each device periodically transmits sensing data to a gateway. Under normal operation, each data packet is intended to be received and processed solely by the gateway. However, in the proposed scheme, we additionally assume that each transmitting device can overhear packets sent by other devices and measure their reception timings.

Equation (1) represents the evolution of the internal phase of each device, and the transmission timing is determined at the moment when this phase reaches a predefined value. In this framework, the reception timings of packets from other devices are treated as coupling events in (1). Upon detecting these receptions, each device updates its phase according to the oscillator dynamics, thereby determining its next transmission timing.

In this way, (1) links the transmission and reception timings within the network, enabling each device to autonomously adjust its transmission instant based solely on locally observed packet receptions.

## III. Application of the Frog Chorus Model to IoT Devices

### A. Model Modification for Intermittent Coupling

Because many IoT devices are battery powered, they cannot perform continuous transmission and reception and instead operate by periodically switching between sleep and active states. In light of this operating pattern, it is not appropriate to assume continuous coupling in the model, and it is necessary to modify the model so that intermittent coupling is allowed. Therefore, we modify the model to the following intermittent form, in which coupling is restricted to a specific interval $[-r\pi, r\pi]$ in the phase space.

$$\frac{d\theta_i}{dt} = \begin{cases} \omega_i + \frac{K_l}{N} \sum_{j=1}^{N} \frac{-1}{\pi} \left( \left( (\theta_j - \theta_i) \bmod 2\pi \right) - \pi \right) \\ \omega_i \end{cases} \tag{2}$$

where $r$ is the ratio of the listening window and takes values in the range $0 < r < 1$, and the condition applies for $-r\pi \leq \theta_i < r\pi$ and otherwise.

Because coupling with other nodes is disabled during the sleep period, intermittent operation can be achieved.

### B. Extension to the smallest phase difference Coupling Model

Because listening is impossible during the sleep period in intermittent operation, we change the model to the smallest phase difference coupling model in which the coupling target is limited not to all nodes but to a single node whose phase difference from node $i$ is the smallest. In this model, the phase is updated as follows,

$$\frac{d\theta_i}{dt} = \begin{cases} \omega_i - \frac{K_l}{\pi} \left( \left( (\theta_j - \theta_i) \bmod 2\pi \right) - \pi \right) \\ \omega_i \end{cases} \tag{3}$$

$$j = \arg\min_{j \neq i} |\theta_i - \theta_j|, \quad \theta_i - \theta_j \in [-r\pi, r\pi] \tag{4}$$

where $j$ is the index of the other node whose phase difference from node $i$ is the smallest, as defined in (4). This modification also contributes to more efficient communication.

### C. Implementation based on discrete-time models

To run the model described above on a microcontroller or system-on-chip, it must be converted into a discrete-time form. By applying Euler discretization with a step size $\Delta t$, the discrete-time model is defined as follows,

$$\theta_i(t_{n+1}) = \begin{cases} \theta_i(t_n) + \Delta t(\omega_i - \frac{K_l}{\pi} \left( \left( (\theta_j - \theta_i) \bmod 2\pi \right) - \pi \right)) \\ \theta_i(t_n) + \Delta t \omega_i \end{cases} \tag{5}$$

Here, $t_n$ denotes the $n$th discrete time instant. Node $i$ is designed to transmit a packet at the time when $\theta_i = 0$, and by updating its phase in each period based on the received information, it realizes autonomous collision avoidance behavior.

In the next subsection, we describe the algorithm for implementing the proposed collision avoidance method and present simulation results comparing the Kuramoto and frog chorus models based on this algorithm.

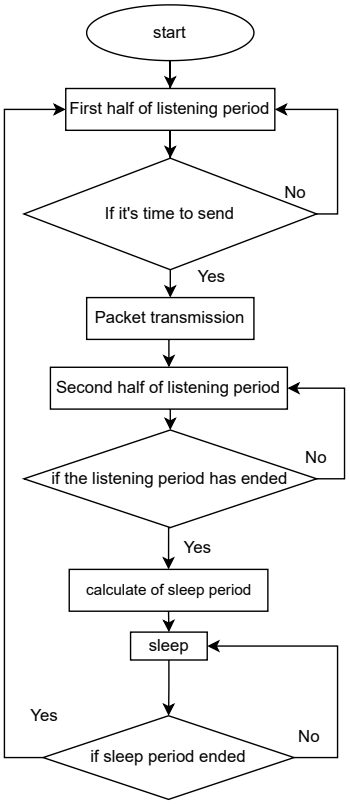



Fig. 2: Proposed flowchart.

### D. Implementation of an algorithm for phase controlled operation

The most important aspect of applying the proposed method to LoRa devices is that each device autonomously adjusts its transmission timing so that a sufficiently dispersed set of communication timings is formed. In this study, we realize phase updating based on anti-phase synchronization inside each device by controlling the sleep time of the devices according to (5). The overall processing flow of the implemented algorithm is shown in Fig. 2. The operation in each phase is outlined below.

- **First half of listening period**: After startup, the device listen to frames from other devices during the first half of the listening window ($-r\pi \leq \theta_i \leq 0$) and records the transmission time $t_j$ of the smallest phase difference node.
- **Packet transmission**: The device transmits a packet at the timer instant corresponding to $\theta_i = 0$, which represents the center of the listening window. The transmission time at this moment is denoted by $t_i$.
- **Second half of listening period**: After transmission, the device listen to frames from other devices during the second half of the listening window ($0 \leq \theta_i \leq r\pi$) and records the transmission time $t_j$ of the smallest phase difference node.
- **Calculation of sleep period**: Using the transmission time $t_j$ of the smallest phase difference node obtained from the

received frame, the next sleep period is calculated as

$$\alpha = K_l \times T \times \frac{1}{\pi}\left(\frac{t_i - t_j}{T} \times 2\pi - \pi\right) \tag{6}$$

where $K_l$ is the coupling strength and $T$ is the duration of one operation cycle. In standard LoRa communication, we set $\alpha = 0$ and do not adjust the timing.

- **Sleep**: Based on the computed $\alpha$, the device determines the sleep period and restarts at $t = T + \alpha$, thereby autonomously updating the transmission timing for the next cycle. After waking up from sleep, $t$ is reset to 0.

Through this sequence of processing, each device adjusts its own phase according to the surrounding transmission activity, and ultimately an evenly dispersed set of transmission timings is formed among nodes, thereby realizing autonomous decentralized collision avoidance.

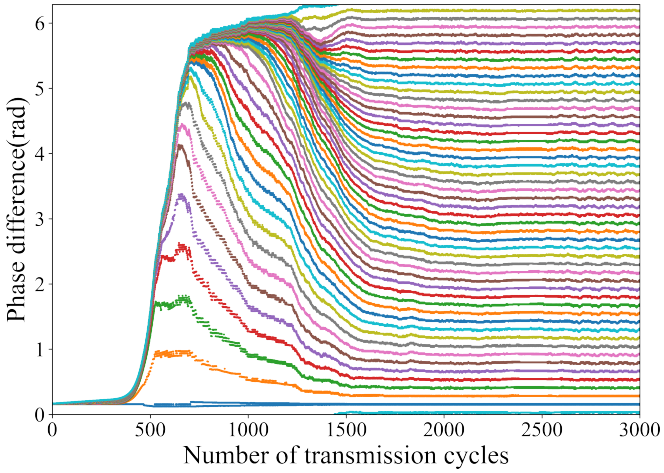
## IV. Comparison of the Kuramoto Model and Frog Chorus Model by Simulation

### A. Parameter settings for simulation

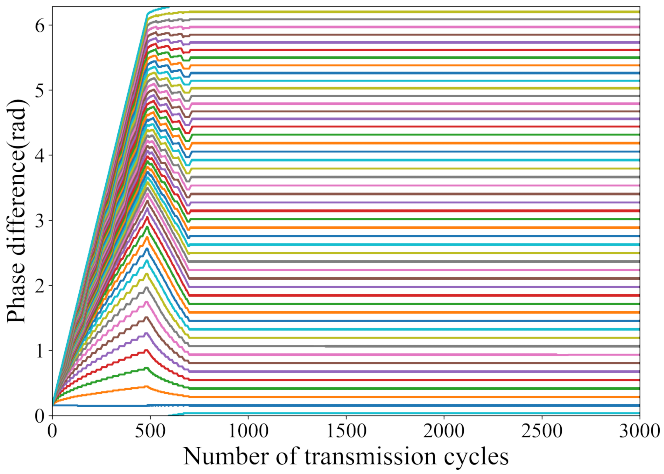
Based on the algorithm described in Section III-D, we compare the behavior of the Kuramoto model and the frog chorus model by simulation. Here, the coupling functions of the Kuramoto and frog chorus models are given by $K_s \sin(\theta_j - \theta_i)$ and $K_l/\pi(\theta_j - \theta_i - \pi)$, where $K_s$ and $K_l$ denote the coupling strengths, respectively. Because the convergence speed depends on the repulsive strength of the coupling function, the coupling coefficients were determined such that the maximum value of the coupling function within the coupling range becomes identical for both methods. This normalization is appropriate because, under the smallest-phase-difference inhibitory coupling, the separation process is driven by the node pair experiencing the strongest repulsion, which governs the worst-case time required to resolve collisions. We evaluate the case of $N = 50$ nodes and set the coupling strengths to $K_s = -0.0080$ and $K_l = -0.0010$, respectively. The simulation conditions are as follows: transmission interval of 30 s, listening ratio $r = 0.25$, simulation length of 3,000 cycles, and an initial offset of 1 ms between the first transmission times of the devices.

### B. Results of the speed comparison in simulation

The simulation results are shown in Figs. 3(a) and 3(b). In these figures, the vertical axis represents the phase difference with a base device, and the horizontal axis represents the number of transmissions of each device. In addition, the initial phase differences are clustered in a small range and increase over time, eventually approaching almost equal spacing among the devices. However, we can observe that the phase differences in the frog chorus model increase more rapidly than in the Kuramoto model. This is because, in the frog chorus model, the amount of phase update is large when the phase difference is close to 0 or $2\pi$ and becomes smaller as the phase difference increases, so that the states with phase differences of 0 or $2\pi$ do not become stable.




(a) Kuramoto model.




(b) Frog chorus model.

Fig. 3: Simulation results for the temporal evolution of phase differences among devices.

Next, we evaluate the speed of phase dispersion by representing the phase of each device as a unit vector and computing the magnitude of the vector sum. The more evenly the phases are distributed, the smaller this magnitude becomes. Figure 4 shows the temporal evolution of the vector sum magnitude for each model. From the results, we see that the magnitude decreases more rapidly in the frog chorus model, indicating that the phases become evenly dispersed more quickly and that collision avoidance is achieved earlier compared with the Kuramoto model.

## V. Comparison of Packet Error Rates Using Devices

In this section, we present the experimental results obtained using LoRa devices to demonstrate that the proposed collision avoidance method is effective in reducing the packet error rate (PER).

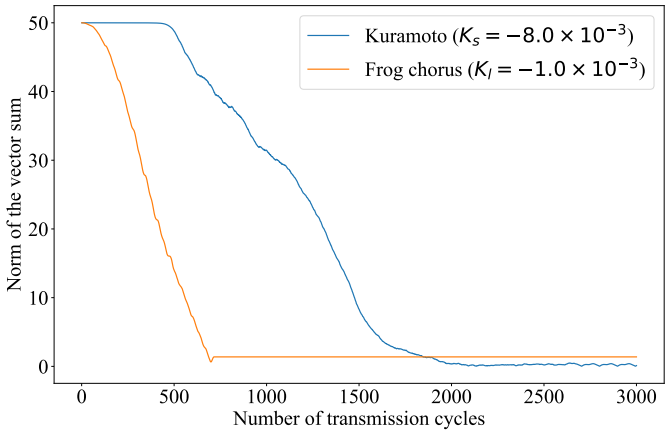
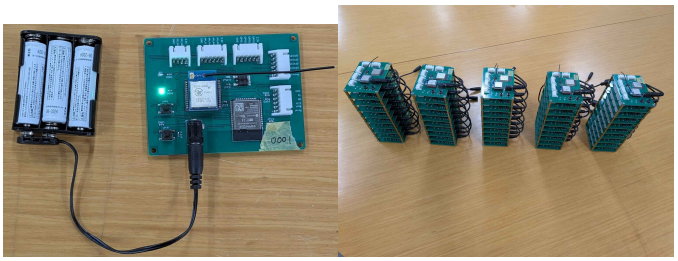



Fig. 4: Temporal evolution of the magnitude of the vector sum for each model.



(a) Single device. (b) Fifty devices.

Fig. 5: Implemented devices.

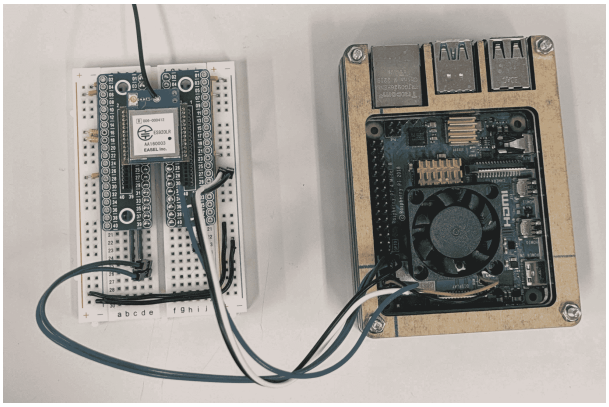
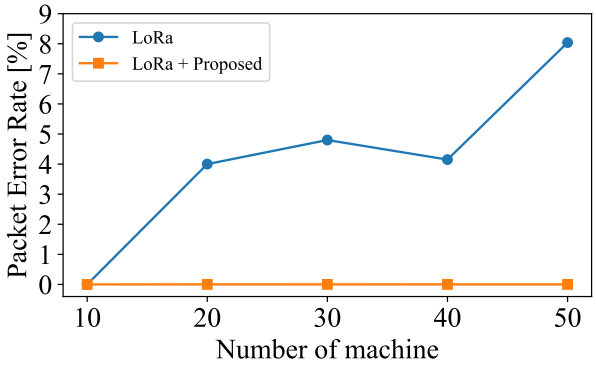


Fig. 6: Receiver.

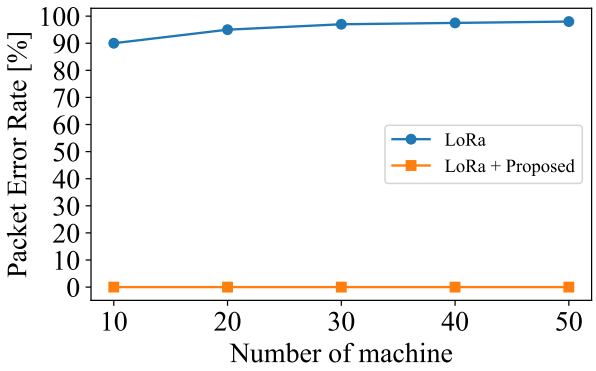
### A. Hardware components and implementation

To measure PER with real devices, we designed and constructed the following two types of hardware configurations.

*a) Terminal devices:* Each transmitting node is composed of the ES920LR LoRa communication module [14] and an ESP32-WROOM-32E microcontroller for control [15]. To emulate a large-scale network environment, we prototyped 50 devices and used them in our experiments. Photographs of a single device and all 50 devices are shown in Figs. 5(a) and 5(b), respectively.




(a) Random start.




(b) Synchronous start.

Fig. 7: Experimental relationship between the number of devices and PER.

*b) Receiver:* To collect and analyze the data transmitted from the devices, we prototyped a dedicated receiver composed of the ES920LR LoRa communication module [14] and a Raspberry Pi 4 single-board computer [16]. This receiver records information such as the transmission timing and device ID of each device, and the PER is calculated based on these records to evaluate the collision avoidance performance of the proposed method. A photograph of the prototype receiver is presented in Fig. 6.

### B. Measurement results on the devices

Each device performs a broadcast transmission, and on the receiver side, the source device ID and reception time are recorded. In this experiment, the following parameters are set identically for all devices: transmission interval 30 s, number of measurement trials 50, center frequency 920.4 MHz, bandwidth 500 kHz, spreading factor (SF) 7, payload length 1 byte, coupling strength $K_l = -0.0010$ and listening ratio $r = 0.25$. The success rate is defined as the ratio of the number of packets received by the receiver to the total

number of transmission attempts, and PER is obtained as the complement of this success rate.

First, we configure the system so that the startup time of each device is randomly distributed within one transmission interval, and perform measurements under this condition. The results are shown in Fig. 7(a). Fig. 7(a). indicate that when the proposed method is applied, PER is consistently low. Even in the case of 50 devices, PER remains at 0 %, indicating higher performance than standard LoRa communication.

Next, we configure the condition so that the startup times of all devices are close to each other, and perform measurements again. The results are shown in Fig. 7(b). Fig. 7(b) shows that, while standard LoRa exhibits a high PER regardless of the number of devices, the proposed method maintains a low PER, similar to the case with random startup times.

From these observations, it can be inferred that in standard LoRa, sufficient diversity in transmission timing among devices is not formed, and simultaneous startup causes the transmission timings to overlap frequently, resulting in frequent collisions that manifest as an increase in PER. In contrast, the anti-phase synchronization property of the proposed method enables each device to autonomously adjust its transmission timing, thereby forming appropriate temporal dispersion and suppressing collisions. These results confirm that, even in large-scale deployments, the proposed method based on the frog chorus model can maintain a stable communication quality and functions as an effective approach for reducing PER.

## VI. Conclusion

In this study, we applied an autonomous decentralized collision avoidance method based on the frog chorus model to LoRa communication and verified its effectiveness in reducing the PER in large-scale LoRa networks. By combining standard LoRa modules with microcontrollers, we constructed 50 communication terminals and conducted evaluation experiments in a real environment. The results confirmed that the proposed method can suppress packet collisions more effectively than conventional LoRa communications. These findings indicate that the proposed method is a promising approach for achieving high communication stability, even in large-scale LoRa networks. In particular, it was clearly shown that the proposed method converges faster than schemes based on the conventional Kuramoto model, thereby enabling more efficient communication control.

In future work, we plan to optimize various parameters, including the coupling strength, to achieve even faster convergence. In addition, we will conduct a more detailed comparative evaluation of the convergence speed and verify the adaptability of the proposed method to increases in the number of devices and changes in the network topology in real environments. Furthermore, to confirm the reliability and practicality of the proposed method, we will evaluate its performance under more complex and realistic operating conditions, with the goal of contributing to the overall performance improvement of IoT systems. In particular, we will test the robustness of the proposed method against hidden-node effects and topology variations, as well as external interference sources. Moreover, we will evaluate its performance under mild mobility, or alternatively emulate mobility-induced link dynamics using controlled attenuation and asymmetric reception conditions.

## Acknowledgment

This work was supported in part by the JSPS KAKENHI Grant Number JP22H05197 and JP23K13331, JST Moonshot R&D Grant Number JPMJMS2021, and the International Research Center for Neurointelligence (WPI-IRCN) at The University of Tokyo Institutes for Advanced Study (UTIAS).

## References

[1] D. Nguyen *et al.*, "6g internet of things: A comprehensive survey," *IEEE Internet of Things Journal*, vol. 9, no. 1, pp. 359–383, 2022.

[2] C. Alwis *et al.*, "Survey on 6g frontiers: Trends, application, requirements, technologies and future research," *IEEE Open Journal of the Communications Society*, vol. 2, pp. 836–886, 2021.

[3] M. L. Liya and M. Aswathy, "Lora technology for internet of things (iot): A brief survey," in *International Conference on I-SMAC*, 2020, pp. 8–13.

[4] U. Raza, P. Kulkarni, and M. Sooriyabandara, "Low power wide area networks: An overview," *IEEE Communications Surveys & Tutorials*, vol. 19, no. 2, pp. 855–873, 2017.

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

[6] M. Jouhari, N. Saeed, M.-S. Alouini, and E. M. Amhoud, "A survey on scalable lorawan for massive iot: Recent advances, potentials, and challenges," *IEEE Communications Surveys & Tutorials*, vol. 25, no. 3, pp. 1841–1876, 2023.

[7] A. Malla, M. El-Kadi, S. Olariu, and P. Todorova, "A fair resource allocation protocol for multimedia wireless networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 14, no. 1, pp. 63–71, 2003.

[8] T. S. Rappaport, *Wireless Communications: Principles and Practice*. Prentice Hall, 1996.

[9] J. Degesys, I. Rose, A. Patel, and R. Nagpal, "Desync: Self-organizing desynchronization and tdma on wireless sensor networks," in *Proceedings of the International Conference on Information Processing in Sensor Networks (IPSN)*, 2007, pp. 11–20.

[10] R. Pagliari, Y. W. P. Hong, and A. Scaglione, "Pulse-coupled oscillators' primitives for collision-free multiple access with application to body area networks," in *International Symposium on Applied Sciences in Biomedical and Communication Technologies*, 2008, pp. 1–5.

[11] T. Osada, H. Yasuda, A. Li, S.-J. Kim, and M. Hasegawa, "Design and implementation of decentralized tdma for low power iot devices based on desynchronization of nonlinear oscillators," in *International Conference on Artificial Intelligence in Information and Communication*, 2023, pp. 602–606.

[12] D. Kojima, H. Yasuda, T. Osada, A. Li, M. Arai, and M. Hasegawa, "Application of nonlinear oscillator desynchronization phenomena to collision avoidance in lora and its experimental evaluation," in *International Conference on Artificial Intelligence in Information and Communication*, 2025, pp. 243–246.

[13] I. Aihara, D. Kominami, Y. Hirano, and M. Murata, "Mathematical modelling and application of frog choruses as an autonomous distributed communication system," *Royal Society Open Science*, vol. 6, p. 181117, 2019.

[14] Easel, "Es920lr wireless module product information," Online, 2025, accessed: 2025-12-27. [Online]. Available: https://easel5.com/service/products-information/products/wireless-module/es920lr/

[15] E. Systems, "Esp32-wroom-32e and esp32-wroom-32ue datasheet," Online, 2025, accessed: 2025-12-27. [Online]. Available: https://www.espressif.com/sites/default/files/documentation/esp32-wroom-32e_esp32-wroom-32ue_datasheet_en.pdf

[16] R. Pi, "Raspberry pi 4 model b," Online, 2025, accessed: 2025-12-27. [Online]. Available: https://www.raspberrypi.com/products/raspberry-pi-4-model-b/