IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received April 30, 2019, accepted May 18, 2019, date of publication May 27, 2019, date of current version July 17, 2019.

*Digital Object Identifier 10.1109/ACCESS.2019.2919130*

# Fair and Distributed Resource Allocation in Wireless Networks Using Frogs' Calling Behavior Algorithm

**DARA RON AND JUNG-RYUN LEE, (Member, IEEE)**
School of Electrical and Electronics Engineering, Chung-Ang University, Seoul 06974, South Korea

Corresponding author: Jung-Ryun Lee (jrlee@cau.ac.kr)

This work was supported in part by the MSIT (Ministry of Science and ICT), South Korea, under the Information Technology Research Center (ITRC) support program supervised by the Institute for Information and Communications Technology Planning and Evaluation (IITP), under Grant IITP-2019-2018-0-01799, in part by the Human Resources Development of the Korea Institute of Energy Technology Evaluation and Planning (KETEP) grant funded by the Korean Government Ministry of Trade, Industry and Energy, under Grant 20174030201810, and in part by the Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Science, ICT and Future Planning under Grant NRF-2019R1F1A1058587.

**ABSTRACT** Bio-inspired algorithms provide some notable characteristics, such as stability, scalability, convergence, and adaptability, which explains the reason why many researchers have attempted to apply bio-inspired algorithms to various kinds of engineering problems. In this paper, we propose a fair resource allocation method in wireless networks, which is inspired by the frogs' calling behavior algorithm. Because the frogs' calling behavior algorithm shows strict de-synchronization of the calling phase and adaptivity in the dynamically changing environment, it is suitable for nodes to achieve fair resource allocation in time-division multiple access (TDMA)-based wireless networks. The analysis of the proposed algorithm verifies the convergence criteria. The simulation results show that the proposed algorithm achieves strict de-synchronization for their phase-coupled oscillators and balanced distribution over all nodes, and thus enables the fair and distributed resource allocation over all nodes in a TDMA-based wireless network even with dynamically changing network topology.

**INDEX TERMS** Bio-inspired, fair resource allocation, de-synchronization, frogs' calling behavior.

## I. INTRODUCTION

In wireless communication, a medium access control (MAC) protocol moderates access to the shared medium which allows the end-devices to communicate with each other in order to gain maximum of channel utilization with minimum of interference and collisions. A meticulous design of the medium access control (MAC) protocol is to cope with various challenges, such as full bandwidth utilization, fairness, scalability, QoS guarantee, flexibility, and adaptability. The importance of these challenges of MAC protocol design is being emphasized in the internet of things (IoT) system, which emerged for wide range communication to realize, such as smart city, smart grid, smart home, intelligent transportation system, and environmental monitoring applications.

MAC protocol is classified according to the channel-access methods and fall into two categories: contention-based protocol (CBP) and scheduling-based protocol (SBP). In CBP, the transmission collision occurs when two or more nodes persist to transmit data simultaneously using the same channel. Because the number of collisions mainly depends on variable traffic loads in a network, it is difficult to ensure the fairness and scalability and to guarantee the quality of service (QoS) of each user [1]. A typical example of contention-based protocol is carrier sense multiple access with collision avoidance (CSMA/CA), which reduces collisions by transmitting packets opportunistically. In CSMA/CA, the network nodes persist to listen to the idle channel using random back-off time, when the medium was sensed as busy. However, this random back-off scheme can just reduce the simultaneous transmission of users, and so cannot guarantee complete collision avoidance. On the other hand, in SBP, the network nodes transmit data using pre-allocated resource, so collision can be avoided once the resource is allocated to each user. SBP can be categorized by two types according to the scheduling policy:

The associate editor coordinating the review of this manuscript and approving it for publication was Petros Nicopolitidis.

centralized scheduling and distributed scheduling. In centralized scheduling, nodes are allocated the resource by using a central coordinator, such as a base station and a access point. Because a central coordinator has all the information necessary to allocate resources to users, centralized scheduling shows high efficiency in resource allocation. However the adoption of centralized scheduling is limited due to the high overhead and complexity, and so it is difficult to ensure the scalability in a network. On the other hand, in distributed scheduling, the nodes exchange the information necessary for resource allocation without the aid from the centralized controller. The key feature of this algorithm is that it is simple to implement and requires less overhead compared to centralized scheduling. Because it is suitable for the dynamic changing of the network environment, and so it is possible to ensure scalability, adaptability, and flexibility in a network.

Among various scheduling-based protocols, time-division multiple access (TDMA) has been widely adopted and used in various networks. In TDMA, an available bandwidth is allocated to a user for a specific fraction of time and so it shows high channel usage efficiency specifically when traffic load is high [2]. Meanwhile, TDMA protocols periodically pre-assign nodes to the fixed time-slot, so it is not easy work to automatically change the time-slot assignment according to the moving, entering, and exiting of nodes in a network [3]. In addition, the wasting time slot occurs during when a node has no data to be transmitted, which mainly deteriorates channel usage efficiency in TDMA networks.

Recently, biologically inspired (bio-inspired) algorithms have gained the attentions of many researches. Bio-inspired algorithms are modeled on the behavior of organisms on Earth such as flies flashing, cardiac pacemaker cells, bees' cooperative searching for food, flocking of birds, schooling of fish, routing of ants, and frogs' calling behavior [4]–[6]. Bio-inspired algorithms have evolved with the goal of achieving given purposes and ultimately obtaining optimal results by encapsulating simple, heuristic rules for operation in a distributed way. As we can observe from previous successful attempts at developing bio-inspired algorithms that have been reported in the literature, bio-inspired algorithms have the excellent characteristics, such as stability, convergence, scalability, and adaptability. Drawing on this expertise, a resource allocation method based on bio-inspired algorithm is expected to be suitable approach to achieve efficient resource allocation in highly complicated, scalable and large-scale networks.

In this study, we propose a frogs' calling behaviors algorithm for fair and distributed resource allocation in the context of the TDMA protocol. The proposed algorithm aims to cope with the various challenge of the TDMA protocol and to provide the key properties, such as no collision, fairness, scalability, adaptability, flexibility, and convergence. The outline of the paper is as follows. Section II overviews the various bio-inspired algorithms which have been utilized in resource allocation in a wireless network. Section III briefs the previous frogs' calling behavior algorithm. In Section IV, we explain the proposed resource allocation method based on extended frogs' calling behavior algorithm and prove that that we can achieve strict de-synchronization under the proposed algorithm. Section V evaluates the simulation results and Section VI concludes.

## II. RELATED WORKS

Some bio-inspired algorithms have been applied to the design of resource allocation algorithms. Resource allocation based on bio-inspired algorithm falls into three categories: (A) Cucker-Smale flocking model [7], which is inspired by flocking behavior of bird in flight, (B) de-synchronization (DESYNC) [8] [9], which is inspired by the flashing behavior of fireflies, and (C) pulse-coupled-oscillators-based de-synchronization (PCO-D) [5] [10], which is inspired by the pacemaker cells of the human heart.

### A. CUCKER-SMALE FLOCKING MODEL

Cucker-Smale flocking model was emerged from moving behavior of flocking birds in flight, in which each bird adjusts its own position and velocity based on the average position and speed of its neighbors [3]. Cucker and Smale showed that the velocity and moving direction of all birds converge to the same asymptotic velocity and the same direction, respectively. In addition, the distance between each agent is boundary. C-S (Cucker-Smale) model mathematically expressed the flocking behavior of birds as

$$\frac{dx_i}{dt}(t) = v_i(t), \tag{1}$$

$$v_i(t^+) - v_i(t) = \frac{\lambda}{N}\sum_{j=1}^{N}\Psi(|x_j(t) - x_i(t)|)(v_j(t) - v_i(t)), \tag{2}$$

where $x_i(t)$ and $v_i(t)$ are the position and velocity of the $i$-th bird at time $t$ and $\lambda$ is the coupling strength among the bird and $\Psi$ is the communication range as a function of distance between two birds [7]. A communication function is non-negative and non-increasing function of the distance between two agents. Examples for the communication ranges are as follow:

$$\Psi_1(|x_j - x_i|) = 1,$$
$$\Psi_2(|x_j - x_i|) = 1_{|x_j - x_i| \le r},$$
$$\Psi_3(|x_j - x_i|) = \frac{1}{(1 + |x_j - x_i|^2)^\beta},$$

where $r$ is positive real value and $\beta$ is non-negative value. The authors of [11] applied the flocking model to the resource allocation in the vehicle-mounted mobile relay (VMR) network with the purpose of achieving distributed and fair resource allocation across all nearby VMRs. In this work, an VMR address is granted to each VMR which acts as a reference for the resource allocation and the de-synchronization over VMR addresses for all VMRs are obtained by applying Cucker-Smale flocking model with slight modification on the definition of VMR address.

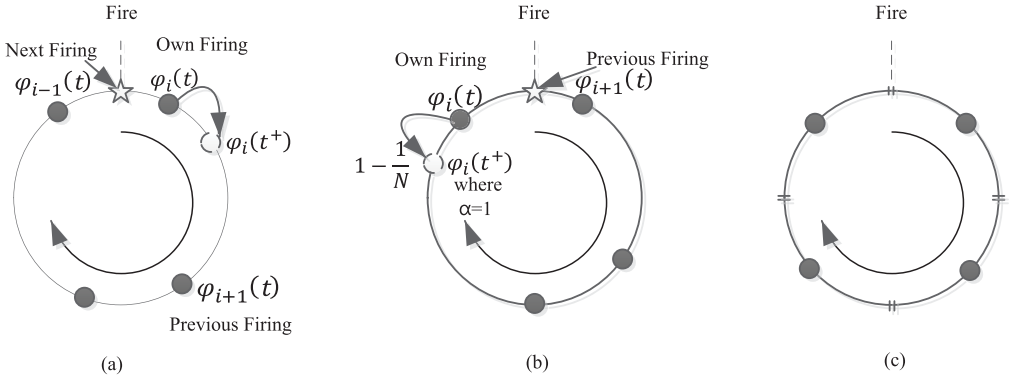


**FIGURE 1.** Phase update procedure of (a) DESYNC, (b) PCO-based desynchronization, and (c) desynchronized state. (a) DESYNC. (b) PCO based desynchronization. (c) Desynchronization state.

### B. DESYNC

De-synchronization is the logical opposite of the fireflies' synchronization, which is inspired by the synchronized flashing at night of thousands of male fireflies. Rather than having nodes attempt to perform tasks at the same time, the algorithm has them perform their tasks at as great a temporal distance as possible from each other. The result is that tasks are spaced evenly in time. In 2007, Nagpal et al. presented the first DESYNC-based TDMA protocol in a fully-connected network [8]. Suppose that there are $N$ nodes and each node performs a task at intervals with the period of $T$. Let $\varphi_i(t) \in [0, 1]$ denote the phase of node $i$ at time $t$, where phases 0 and 1 are identical and $0 \leq i \leq N$. Upon reaching $\varphi_i(t) = 1$, node $i$ fires to indicate the termination of its cycle to the other nodes. Upon firing, the node resets its phase to $\varphi(t^+) = 0$. Node $i$ records the times of the following two firing events: the one that precedes its own firing (previous firing $\varphi_{i+1}(t)$), and the one that occurs just afterwards (next firing $\varphi_{i-1}(t)$, as shown in fig. 1(a). These firing events are called the reference phases for node $i$. Node $i$ calculates the midpoint of its two reference phases as $\varphi_{mid}(t) = \frac{1}{2}(\varphi_{i+1}(t) + \varphi_{i-1}(t))$ and jumps towards it as follows:

$$\varphi_i(t^+) = (1 - \alpha)\varphi_i(t) + \alpha\varphi_{mid}(t), \tag{3}$$

where $\alpha \in [0, 1]$ is a parameter that scales how far node $i$ moves from its current phase toward the desired midpoint. All nodes observe their neighbors' firing-phases, then use this information to jump forwards or backwards in the phase according to (3). Therefore, all oscillators are spaced evenly around the phase ring, as shown in Fig. 1(c). Node $i$ occupies the TDMA slots beginning at the previously computed midpoint between node $i$ and its previous-phase neighbor $i + 1$, and ending at the previously computed midpoint between node $i$ and its next-phase neighbor $i - 1$. In this way, all nodes occupy the non-overlapping time slots that cover $T$ evenly.

### C. PULSE COUPLED OSCILLATOR-BASED DE-SYNCHRONIZATION

In 1975, Peskin devised the PCO synchronization algorithm to model the operation of a pacemaker responsible for regulating the beat of the human heart [12]. PCO comprises elements that pulse individually at constant intervals when separated, but that, when interconnected, change their pulsing periods and phases in response to the signals received from other elements. The PCO-D algorithm, which is based on the inverse phenomenon of the PCO-synchronization process, was proposed in 2008 by Scaglion et al., for collision-free multiple access in a small network of self-organizing devices [10]. In DESYNC, node $i$ records the times of two reference phases (previous and next firings). However, in the PCO-D, node $i$ records the time from a single reference phase that occurs just before it fires, as shown in Fig. 1(b). When node $i + 1$ fires, node $i$ jumps towards it as follows:

$$\varphi_i(t^+) = (1 - \alpha)\varphi_i(t) + \alpha(1 - \frac{1}{N}), \tag{4}$$

where $N$ is the number of nodes in a fully-connected network, and $\varphi \in [0, 1]$ is a parameter that scales how far node i moves from its current phase. All nodes observe their neighbors' firing-phases, then use this information to jump backwards in the phase using equation (4). As a result, all of the oscillators are spaced evenly around the phase ring, as shown in Fig. 1(d).

## III. FROGS' CALLING BEHAVIOR ALGORITHM

As in the previous work, Ikkyu Aihara modeled frogs' calling behavior algorithm as a system of three-phase oscillators with the purpose of achieving strict de-synchronization [13]. This algorithm has modeled from the natural phenomena of tree frogs' calling behavior in Japan, in which caller (usually male frog) always makes the stereotypical sound in a certain period of the time to call female and then it goes quiet before repeating the call. When two or three frogs start to advertise the calling signal at a random time, the calling signal interference may occur, as shown in Fig. 2(a). In this case, the female frogs cannot distinguish between their partners and other. Therefore, each male frog has to shift the time of its calls by recording the calls of the others so as to avoid the overlap. After all, frogs establish this interaction pattern, call alternation without interference is achieved within the group

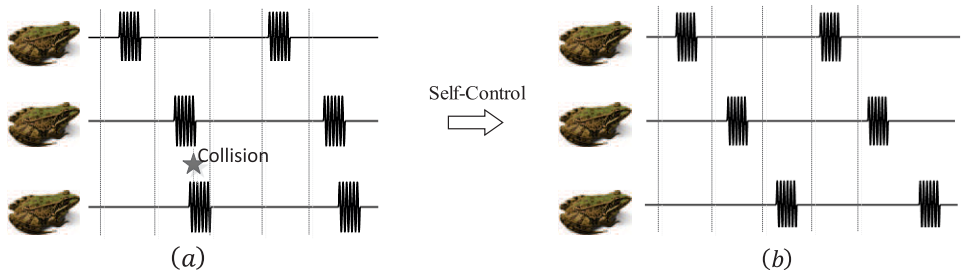


**FIGURE 2.** Three male frogs attempt to control their stereotypical sounds to advertise without collision.

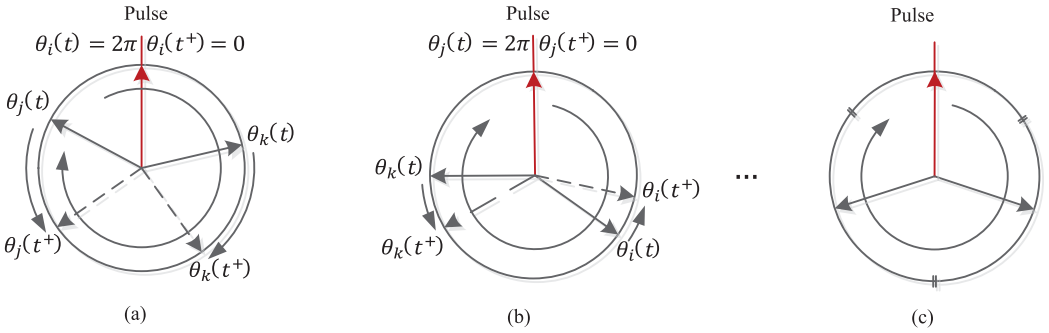


**FIGURE 3.** Phase update procedure of the proposed algorithm.

as depicted in Fig. 2(b). Suppose that frog $i$ calls at time $t_i$ ($i = 1, 2, 3$). When frog $i$ calls at time $t_i$, then the other frogs immediately update their phases. The system of three-phase oscillators which virtually corresponds to a situation where a male frog calls interactively at time $t_i$ is given by

$$\begin{aligned}
\frac{d\theta_1(t)}{dt} &= \omega - K_1 \sin\Big(\theta_2(t) - \theta_1(t)\Big) \\
&\quad - K_3 \sin\Big(\theta_3(t) - \theta_1(t)\Big) \text{ for } t = t_2 \text{ or } t_3, \\
\frac{d\theta_2(t)}{dt} &= \omega - K_1 \sin\Big(\theta_1(t) - \theta_2(t)\Big) \\
&\quad - K_2 \sin\Big(\theta_3(t) - \theta_2(t)\Big) \text{ for } t = t_1 \text{ or } t_3, \\
\frac{d\theta_3(t)}{dt} &= \omega - K_3 \sin\Big(\theta_1(t) - \theta_3(t)\Big) \\
&\quad - K_2 \sin\Big(\theta_2(t) - \theta_3(t)\Big) \text{ for } t = t_1 \text{ or } t_2, \quad (5)
\end{aligned}$$

where $K_i$s ($i = 1, 2, 3$) are the systematically coupled coefficients between frogs $i$ and $i + 1$ for $i = 1, 2$ and frogs 3 and 1 for $i = 3$ [13]. The updating rule of three-phase oscillators can rewrite for the discrete time as

$$\begin{aligned}
\theta_1(t^+) &= \theta_1(t) + \Delta T\Big(\omega - K_1 \sin\big(\theta_2(t) - \theta_1(t)\big) - K_3 \\
&\quad \sin\big(\theta_3(t) - \theta_1(t)\big)\Big) \text{ for } t = t_2 \text{ or } t_3, \\
\theta_2(t^+) &= \theta_2(t) + \Delta T\Big(\omega - K_1 \sin\big(\theta_1(t) - \theta_2(t)\big) - K_2 \\
&\quad \sin\big(\theta_3(t) - \theta_2(t)\big)\Big) \text{ for } t = t_1 \text{ or } t_3, \\
\theta_3(t^+) &= \theta_3(t) + \Delta T\Big(\omega - K_3 \sin\big(\theta_1(t) - \theta_3(t)\big) - K_2 \\
&\quad \sin\big(\theta_2(t) - \theta_3(t)\big)\Big) \text{for } t = t_1 \text{ or } t_2, \quad (6)
\end{aligned}$$

where $\Delta T = t^+ - t$ is a sufficiently small fixed time step. Here, it is noticed that the frogs' calling behavior model in [13] assumed only three frogs and this limitation on the number of frogs (agents) prohibits the wide application of the frogs' model to various systems where the number of agents is usually greater than three.

## IV. PROPOSED RESOURCE ALLOCATION BASED ON FROG CALLING BEHAVIOR ALGORITHM

### A. PROPOSED ALGORITHM

In this subsection, we extend the previous frogs' calling behavior algorithm to a larger system composed of $N$ frogs. The proposed algorithm corresponds to the coupled system that the coupling occurs when a frog calls. Suppose that the phase oscillators of frogs operate in a counter-clockwise order with a fixed period $T$ and frequency $\omega = 1/T$. Let $\theta_i(t) \in [0, 2\pi)$ be the dynamic phase of frog $i$ at time $t$. When $\theta_i(t_i)$ reaches $2\pi$ at time $t_i$, frog $i$ immediately makes a call (send a pulsing message) and resets its phase to zero, as shown in Fig. 3(a). All frogs who hear a pulse message update their phases so as to avoid the overlapped calling period. Let $\theta_j(t_i^+)$ and $\theta_k(t_i^+)$ denote the updated phase of frogs $j$ and $k$ immediately after frog $i$ makes a call at time $t_i$. Frogs $j$ and $k$ instantaneously jump their phases from $\theta_j(t_i)$ to $\theta_j(t_i^+)$ and from $\theta_k(t_i)$ to $\theta_k(t_i^+)$ as depicted in Fig. 3(a). In Fig. 3(b), $\theta_j(t_j)$ reaches $2\pi$ again at time $t_j$ and frog $j$ transmits its pulse message to the other frogs and resets its own phase to zero. The frogs $i$ and $k$ simultaneously jump their phases in order from $\theta_i(t_j)$ to $\theta_i(t_j^+)$ and from $\theta_k(t_j)$ to $\theta_k(t_j^+)$. Over the course of many iterations, all frogs reorganize their phases to achieve

strict de-synchronization, as depicted in Fig. 3(c). The phase updating rule of frog $j$ at time $t_i$ for oscillators is described as

$$\theta_j(t_i^+) = \theta_j(t_i) - F_j(t_i), \tag{7}$$

where $F_j(t_i)$ represents the jumping function of frog $j$ at time $t_i$. $F_j(t)$ will be computed by utilizing the average summation of weighted sine functions, where the weight function is set as a negative power of two-phase difference of exponential function. The two-phase difference between $\theta_k(t)$ and $\theta_j(t)$ is computed by utilizing the *mod* operation that returns all phase values within a range of $[0, 2\pi)$, but does not change phase position on the circle. The function of two-phase differences between frog $k$ and $j$ at time $t$ is determined by

$$\phi_{k,j}(t) = \mod(\theta_k(t) - \theta_j(t), 2\pi), \tag{8}$$

where $\theta_k(t)$, $k = 1, 2, ..., N$, denotes the phase of frog $k$ at time $t$. Alternatively, we can compute the phase differences using the *floor* function which takes a real number $x$ as an input and then gives as outputs the greatest integer less than or equal to $x$, denoted $floor(x) = \lfloor x \rfloor$. The relation between *mod* and *floor* operations is obtained by

$$\mod(x, y) = x - \lfloor x/y \rfloor y, \tag{9}$$

where variables $x$ and $y$ are replaced by $\theta_k(t_i) - \theta_j(t_i)$ and $2\pi$, respectively. From (8) and (9), two phase differences between frog $k$ and $j$ can be rewritten as

$$\phi_{k,j}(t) = \theta_k(t) - \theta_j(t) - \lfloor \frac{\theta_k(t) - \theta_j(t)}{2\pi} \rfloor 2\pi. \tag{10}$$

$W_{k,j}$ represents the weighted function between frogs $j$ and $k$. Here, we set $W_{k,j}$ as

$$W_{k,j}(t) = \begin{cases} e^{-\phi_{k,j}(t)} & \text{if} \quad 0 \le \phi_{k,j}(t) < \pi, \\ e^{-2\pi+\phi_{k,j}(t)} & \text{if} \quad \pi \le \phi_{k,j}(t) < 2\pi. \end{cases} \tag{11}$$

It is noticed that the elements of the weight function are symmetric with respect to $\phi_{k,j}(t) = \pi$ when the phase oscillators reach the balanced distribution. Then, we define the shifting function as a sum of weighted sine functions, which is given by

$$F_j(t) = \frac{K\Delta T}{N} \sum_{k=1}^{N} W_{k,j}(t) \sin\left(\theta_k(t) - \theta_j(t)\right), \tag{12}$$

where $K$ is the coupled factor coefficient. From (7), the angle distance between an updated phase and its previous phase depends on the shifting function, $F_j(t)$. This shifting function includes a coefficient $K\Delta T$, as expressed in (12). In this paper, we will call $K\Delta T$ as the *convergence speed coefficient* because the convergence speed of the proposed model is determined by the value of $K\Delta T$, which will be shown in Theorem 1. From (7) and (12), we can construct the updating rule of the frogs' calling behavior model with $N$ frogs as

$$\theta_j(t_i^+) = \theta_j(t_i) - \frac{K\Delta T}{N} \sum_{k=1}^{N} W_{k,j}(t_i) \sin\left(\theta_k(t_i) - \theta_j(t_i)\right). \tag{13}$$

We now analyze the condition of the convergence speed coefficient $K\Delta T$ required for the proposed frog calling behavior algorithm to achieve strict de-synchronization. The Lyapunov definition is utilized to determine the necessary convergence speed domain for the phase oscillators to reach a balanced distribution. Additionally, the jumping function is obtained in term of two-phase differences $\phi_{k,j}(t)$ with the purpose of finding its maximum and minimum values. From (8), we can obtain the following equations given by

$$\sin\left(\theta_k(t) - \theta_j(t)\right) = \sin\left(\phi_{k,j}(t)\right), \tag{14}$$

$$\cos\left(\theta_k(t) - \theta_j(t)\right) = \cos\left(\phi_{k,j}(t)\right). \tag{15}$$

From (11), (12), and (14), the jumping function of frog $j$ is rewritten by

$$F_j(t) = \frac{K\Delta T}{N} \sum_{k=1}^{N} e^{-\phi_{k,j}(t)} \sin\left(\phi_{k,j}(t)\right) \tag{16}$$

for $0 \le \phi_{k,j}(t) < \pi$, and

$$F_j(t) = \frac{K\Delta T}{N} \sum_{k=1}^{N} e^{-2\pi+\phi_{k,j}(t)} \sin\left(\phi_{k,j}(t)\right) \tag{17}$$

for $\pi \le \phi_{k,j}(t) < 2\pi$. The minimum and maximum values of the jumping function of frog $j$ are determined by taking a derivative with respect to the two-phase differences between frog $k$ and $j$, which are given by

$$\begin{aligned} \frac{dF_j(t)}{d\phi_{k,j}(t)} &= \frac{d}{d\phi_{k,j}(t)}\left(\frac{K\Delta T}{N} \sum_{l=1}^{N} e^{-\phi_{l,j}(t)} \sin\phi_{l,j}(t)\right) \\ &= \frac{K\Delta T}{N} e^{-\phi_{k,j}(t)}\left(\cos\phi_{k,j}(t) - \sin\phi_{k,j}(t)\right), \\ &\quad 1 \le k \le N. \end{aligned} \tag{18}$$

for $0 \le \phi_{k,j}(t) < \pi$. Because we can get the maximum critical point when $\phi_{k,j}(t) = \frac{\pi}{4}$, the maximum value of $F_j(t)$ is calculated by

$$\max_t\{F_j(t)\} = \frac{K\Delta T}{N} \sum_{k=1}^{N} e^{-\frac{\pi}{4}} \sin\frac{\pi}{4} = K\Delta T e^{-\frac{\pi}{4}} \sin\frac{\pi}{4}. \tag{19}$$

In case of $\pi \le \phi_{k,j}(t) < 2\pi$, we have

$$\begin{aligned} \frac{dF_j(t)}{d\phi_{k,j}(t)} &= \frac{d}{d\phi_{k,j}(t)}\left(\frac{K\Delta T}{N} \sum_{l=1}^{N} e^{-2\pi+\phi_{l,j}(t)} \sin\phi_{l,j}(t)\right) \\ &= \frac{K\Delta T}{N} e^{-2\pi+\phi_{k,j}(t)}\left(\cos\phi_{k,j}(t) + \sin\phi_{k,j}(t)\right), \\ &\quad 1 \le k \le N, \end{aligned} \tag{20}$$

Similarly, we have the minimum critical point when $\phi_{k,j}(t) = \frac{7\pi}{4}$ and the minimum value of $F_j(t)$ is given by

$$\min_t\{F_j(t)\} = -K\Delta T e^{-\frac{\pi}{4}} \sin\frac{\pi}{4}. \tag{21}$$

From (19) and (21), we have

$$\max_t\{F_j(t)\} = -\min_t\{F_j(t)\} = K\Delta T e^{-\frac{\pi}{4}} \sin\frac{\pi}{4}. \tag{22}$$

The phase oscillators become stable or unstable depending on the value of convergence speed coefficient $K\Delta T$. In the following theorem, we specify the condition when the proposed $N$ coupled oscillators achieves convergence.

*Theorem 1: For N coupled oscillators in a discrete system, a balanced state (i.e, $\theta(t^+) = \theta(t)\ \forall t \in \mathbb{R}^+$) is approached when $K\Delta T$ satisfies $0 < K\Delta T < 2\pi \left(e^{-\frac{\pi}{4}} \sin\frac{\pi}{4}\right)^{-1}$.*

Here, we use a Lyapunov function for the proof the Theorem 1. Let $V(t)$ is the Lyapunov candidate at time $t$ (just before calling), $V(t^+)$ is the updated function of the Lyapunov candidate at time $t^+$ (just after calling), and $\Delta V(t)$ is the Lyapunov differential equation at time $t$. The stability of the coupling system is verified if the Lypunov function satisfies the following properties [17]:

- the Lyapunov candidate $V(t)$ is positive for $\forall t \in \mathbb{R}^+$,
- the Lyapunov differential equation $\Delta V(t) = V(t^+) - V(t)$ is less than or equals to zero for $\forall t \in \mathbb{R}^+$.

A useful metric in the study of phase oscillators models is the phase centroid, which is utilized to study strict de-synchronization systems. The phase centroid is obtained by

$$R(t) = \frac{1}{N}\sum_{k=1}^{N}\begin{vmatrix}\cos\theta_k(t)\\ \sin\theta_k(t)\end{vmatrix}. \tag{23}$$

Here, to measure the phase coherence, we define $V(t)$ as the square of the magnitude of the phase centroid, which is given by

$$V(t) = \|R(t)\|^2 = \left\|\frac{1}{N}\sum_{k=1}^{N}\begin{vmatrix}\cos\theta_k(t)\\ \sin\theta_k(t)\end{vmatrix}\right\|^2. \tag{24}$$

From the definition, it is clear that $0 \le V(t) \le 1$ for $\forall t \in \mathbb{R}^+$. It is noticed that $V(t) = 1$ for synchronized states and $V(t) = 0$ for desynchronized states. The system converges to strict desynchronization when the Lyapunov differential function satisfies with the condition: $\Delta V(t) = V(t^+) - V(t) < 0$ for $\forall t \in \mathbb{R}^+$. In the following proof, we will show that the Lypunov differential equation $\Delta V(t)$ is less than zero when $K\Delta T$ is in range of $(0, 2\pi \left(e^{-\frac{\pi}{4}} \sin\frac{\pi}{4}\right)^{-1})$, hence, the coupled system achieves a stable equilibrium when Lyapunov differential equation $\Delta V(t)$ approaches to zero.

*Proof:* The Lyapunov candidate $V(t)$ can be rewritten by

$$\begin{aligned}V(t) &= \frac{1}{N^2}\left(\left(\sum_{k=1}^{N}\cos\theta_k(t)\right)^2 + \left(\sum_{k=1}^{N}\sin\theta_k(t)\right)^2\right)\\ &= \frac{1}{N^2}\left(\sum_{k=1}^{N}\cos^2\theta_k(t) + 2\sum_{k=1}^{N}\sum_{j=k+1}^{N}\cos\theta_k(t)\cos\theta_j(t)\right)\\ &\quad + \frac{1}{N^2}\left(\sum_{k=1}^{N}\sin^2\theta_k(t) + 2\sum_{k=1}^{N}\sum_{j=k+1}^{N}\sin\theta_k(t)\sin\theta_j(t)\right)\\ &= \frac{1}{N^2}\left(N + 2\sum_{k=1}^{N}\sum_{j=k+1}^{N}\cos\left(\theta_k(t) - \theta_j(t)\right)\right).\end{aligned} \tag{25}$$

To determine the differential equation of Lyapunov, it is necessary to find the updated function of Lyapunov. For any jumping moment $t$, $V(t^+)$ is written by

$$V(t^+) = \frac{1}{N^2}\left(N + 2\sum_{k=1}^{N}\sum_{j=k+1}^{N}\cos\left(\theta_k(t^+) - \theta_j(t^+)\right)\right). \tag{26}$$

Both Lyapunov equation and its updated function obtained above. Thus, it is sufficient to find the Lyapunov differential equation. $\Delta V(t)$ is derived as

$$\begin{aligned}\Delta V(t) &= V(t^+) - V(t)\\ &= \frac{2}{N^2}\sum_{k=1}^{N}\sum_{j=k+1}^{N}\begin{pmatrix}\cos\left(\theta_k(t^+) - \theta_j(t^+)\right)\\ [5pt] - \cos\left(\theta_k(t) - \theta_j(t)\right)\end{pmatrix}\\ &= \frac{2}{N^2}\sum_{k=1}^{N}\sum_{j=k+1}^{N}\begin{pmatrix}\cos\left(\theta_k(t^+) - \theta_j(t^+)\right)\\ [5pt] - \cos\left(\theta_k(t^+) - \theta_j(t)\right)\end{pmatrix}\end{aligned} \tag{27}$$

$$+ \frac{2}{N^2}\sum_{k=1}^{N}\sum_{j=k+1}^{N}\begin{pmatrix}\cos\left(\theta_k(t^+) - \theta_j(t)\right)\\ [5pt] - \cos\left(\theta_k(t) - \theta_j(t)\right)\end{pmatrix} \tag{28}$$

Here we define $V_1$ and $V_2$ as

$$\Delta V_1(t) := \sum_{k=1}^{N}\sum_{j=k+1}^{N}\begin{pmatrix}\cos\left(\theta_k(t^+) - \theta_j(t)\right)\\ -\cos\left(\theta_k(t) - \theta_j(t)\right)\end{pmatrix}, \tag{29}$$

$$\Delta V_2(t) := \sum_{k=1}^{N}\sum_{j=k+1}^{N}\begin{pmatrix}\cos\left(\theta_k(t^+) - \theta_j(t^+)\right)\\ -\cos\left(\theta_k(t^+) - \theta_j(t)\right)\end{pmatrix}. \tag{30}$$

Then we have

$$\Delta V(t) = \frac{2}{N^2}\left\{\Delta V_1(t) + \Delta V_2(t)\right\} \tag{31}$$

From (31), the differential equation of Lyapunov candidate $\Delta V(t)$ is negative and will converge to zero under the condition of $\left(\Delta V_1(t) < 0\right) \cap \left(\Delta V_2(t) < 0\right)$. Both $\Delta V_1(t)$ and $\Delta V_2(t)$ become negative functions depending on the summation of the cosine function. It is clear that $cos\varphi - cos\theta < 0$ if both $\theta$ and $\varphi$ fall into both conditions below,

**Case 1,** $0 \le \theta < \pi$ and $\theta < \varphi < 2\pi - \theta$,
**Case 1,** $\pi \le \theta \le 2\pi$ and $2\pi - \theta < \varphi < \theta$.

From (7) and (29), $\Delta V_1(t)$ can be rewritten as

$$\begin{aligned}&\Delta V_1(t)\\ &= \sum_{k=1}^{N}\sum_{j=k+1}^{N}\left\{\cos\left(\theta_k(t^+) - \theta_j(t)\right) - \cos\left(\theta_k(t) - \theta_j(t)\right)\right\}\\ &= \sum_{k=1}^{N}\sum_{j=k+1}^{N}\left\{\cos(\theta_j(t) - \theta_k(t^+)) - \cos(\theta_j(t) - \theta_k(t))\right\}\\ &= \sum_{k=1}^{N}\sum_{j=k+1}^{N}\Big\{\cos\left(\theta_j(t) - \theta_k(t) + F_k(t)\right)\\ &\qquad - \cos\left(\theta_j(t) - \theta_k(t)\right)\Big\}\end{aligned}$$

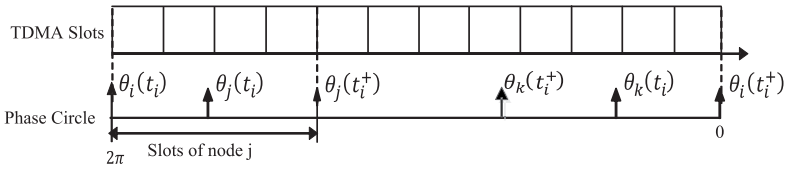


**FIGURE 4.** Mapping relation between TDMA slots and phase of a node.

$$= \sum_{k=1}^{N} \sum_{j=k+1}^{N} \Big\{ \cos(\alpha_{j,k}(t)) - \cos(\beta_{j,k}(t)) \Big\}, \tag{32}$$

where $\beta_{j,k}(t) = \big(\theta_j(t) - \theta_k(t)\big)$ is the two-phase differences between node $j$ and $k$. Also, we have $\alpha_{j,k}(t) = \beta_{j,k}(t) + F_k(t)$.

Considering **Case 1**, $\Delta V_1(t)$ becomes negative if $\alpha_{j,k}(t)$ falls into the conditions below:

$$\begin{aligned} &\beta_{j,k}(t) < \alpha_{j,k}(t) < 2\pi - \beta_{j,k}(t) \\ &\quad \Leftrightarrow \beta_{j,k}(t) < \beta_{j,k}(t) + F_k(t) < 2\pi - \beta_{j,k}(t) \\ &\quad \Leftrightarrow 0 < \max_t\{F_k(t)\} < \max_t\{2\pi - 2\beta_{j,k}(t)\}. \end{aligned} \tag{33}$$

and

$$\begin{aligned} &0 \le \beta_{j,k}(t) < \pi \Leftrightarrow 0 < 2\pi - 2\beta_{j,k}(t) \le 2\pi. \\ &\qquad \Rightarrow \max_t\{2\pi - 2\beta_{j,k}(t)\} = 2\pi. \end{aligned} \tag{34}$$

From (22), (33), and (34), the convergence speed domain is

$$0 < K\Delta T < \frac{2\pi}{e^{-\frac{\pi}{4}} \sin(\frac{\pi}{4})}. \tag{35}$$

Considering **Case 2**, $\Delta V_1(t)$ becomes negative under the conditions below:

$$\begin{aligned} &2\pi - \beta_{j,k}(t) < \alpha_{j,k}(t) < \beta_{j,k}(t) \\ &\quad \Leftrightarrow 2\pi - \beta_{j,k}(t) < \beta_{j,k}(t) + F_k(t) < \beta_{j,k}(t) \\ &\quad \Leftrightarrow \min_t\{2\pi - 2\{\theta_j(t) - \theta_k(t)\}\} < \min_t\{F_k(t)\} < 0. \end{aligned} \tag{36}$$

and

$$\begin{aligned} &\pi \le \beta_{j,k}(t) \le 2\pi \Leftrightarrow -2\pi \le 2\pi - 2\beta_{j,k}(t) \le 0. \\ &\qquad \Rightarrow \min_t\{2\pi - 2\beta_{j,k}(t)\} = -2\pi \end{aligned} \tag{37}$$

From (22), (36), and (37), the convergence speed domain is

$$0 < K\Delta T < \frac{2\pi}{e^{-\frac{\pi}{4}} \sin(\frac{\pi}{4})}. \tag{38}$$

According to the study on the inequality of two cases above, $\Delta V_1(t)$ becomes a negative function if only if the given value of convergence speed, $K\Delta T$, is in range of $(0, \frac{2\pi}{e^{-\frac{\pi}{4}} \sin(\frac{\pi}{4})})$. Then, we need to prove that $\Delta V_2(t)$ is also negative function under the condition of convergence speed $K\Delta T$. From (7) and (30), $\Delta V_2(t)$ can be rewritten as follows:

$$\begin{aligned} &\Delta V_2(t) \\ &= \sum_{k=1}^{N} \sum_{j=k+1}^{N} \Big\{ \cos\big(\theta_k(t^+) - \theta_j(t^+)\big) - \cos\big(\theta_k(t^+) - \theta_j(t)\big) \Big\} \\ &= \sum_{k=1}^{N} \sum_{j=k+1}^{N} \{ \cos\Big(\theta_k(t^+) - \theta_j(t) + F_j(t)\Big) \\ &\qquad - \cos\big(\theta_k(t^+) - \theta_j(t)\big) \} \end{aligned}$$

$\Delta V_2(t) < 0$ under the condition below:

$$\begin{aligned} &0 < F_j(t) < 2\pi - 2\{\theta_k(t^+) - \theta_j(t)\} \\ &\qquad \Leftrightarrow 0 < \max_t\{F_j(t)\} < \max_t\{2\pi - 2\{\theta_k(t^+) - \theta_j(t)\}\}. \end{aligned}$$

From (22), the convergence speed domain is

$$0 < K\Delta T < \frac{2\pi}{e^{-\frac{\pi}{4}} \sin(\frac{\pi}{4})}. \tag{39}$$

As seen in (35), (38), and (39), both $\Delta V_1(t)$ and $\Delta V_2(t)$ are negative for all time, where the convergence coefficient $K\Delta T$ is in the range of $(0, 2\pi(e^{-\frac{\pi}{4}} \sin(\frac{\pi}{4}))^{-1}$. According to (31), $\Delta V(t)$ is also negative within the state range of $K\Delta T$, so the updated function of Lyapunov candidate $V(t^+)$ is always less than $V(t)$ for all time. Therefore, the Lyapunov candidate $V(t)$ will converge to zero at any time $t$, also the vector elements of phase centroid will converge to zeros according to (23), which ends the proof. □

### B. RESOURCE MAPPING BETWEEN TDMA-SLOTS AND PHASE OSCILLATOR

In this subsection, we explain the mapping relation between phase oscillator of a frog and resource allocation of a node in a TDMA-based network. The frogs' calling behavior algorithm permits nodes to automatically regulate time-slot allocation for fully utilizing bandwidth without incurring any collision. We decouple the ring in Fig. 3 into the line segment as depicted in Fig. 4. Node $j$ is expected to emit a firing message after pulsing of node $i$ because it is closer to node $i$ than other nodes. Then a node $j$ is allocated the TDMA-slots by mapping of the two consecutive phase difference from $\theta_i(t_i)$ to $\theta_j(t_i^+)$ as shown in Fig. 4. The firing message is broadcasted via a control channel and includes an ID and phase information as shown in Fig. 5. Nodes in a network that receive the firing message update their phases by using this phase information. Over the course of many iterations, the system reaches strict de-synchronization, in which all nodes are spread out with a even spacing of $T/N$.

## V. PERFORMANCE EVALUATION

In this section, we evaluate the performance of the proposed resource allocation algorithm under various network environments. Fig. 6 shows the phase variations for 10 users.

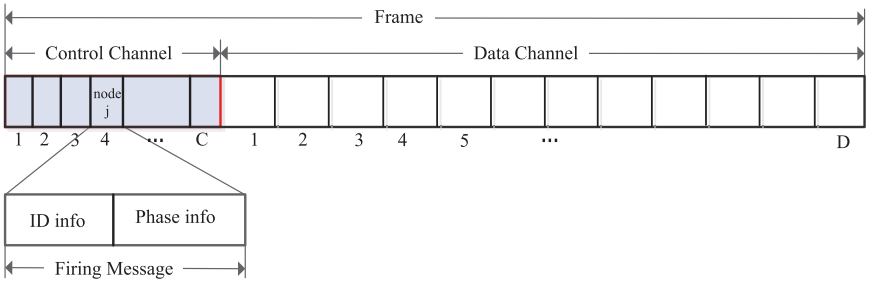


**FIGURE 5.** Frame structure of the firing message.

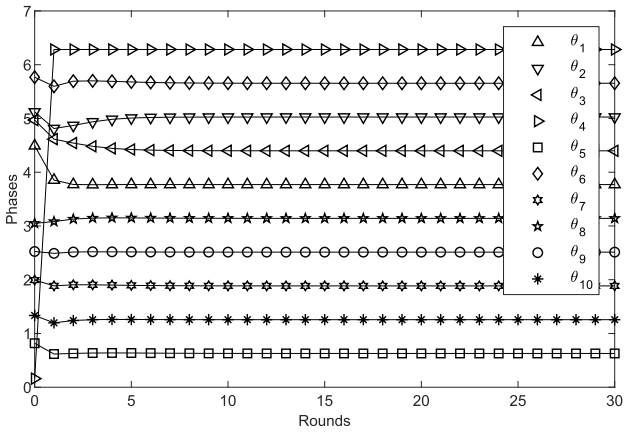


**FIGURE 6.** Phase variation of 10 nodes under the proposed algorithm.

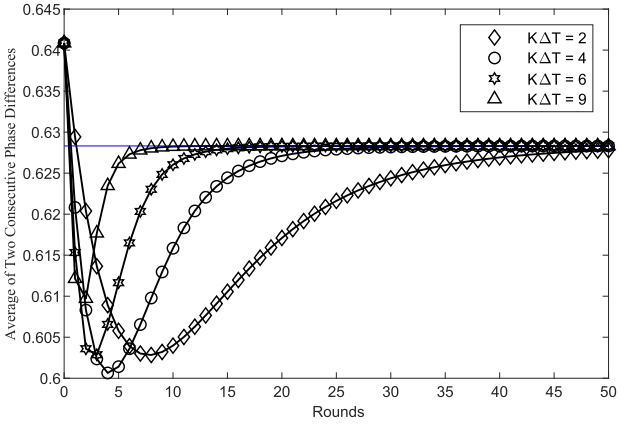


**FIGURE 7.** The phase oscillators reach to de-synchronization state using two various algorithms, such as (a) DESYNCH and (b) PCO-based de-synchronization.

The initial phases of oscillators are randomly located in the interval $[0, 2\pi)$ and the convergence speed coefficient $K\Delta T$ is set to 8. The result shows that the proposed algorithm successfully achieves strict anti-phase synchronization after 5 rounds of iteration.

Fig. 7 compares the convergence speed of the average of two consecutive phase differences for various values of $K\Delta T$ under the same simulation environment to Fig. 6. Here,

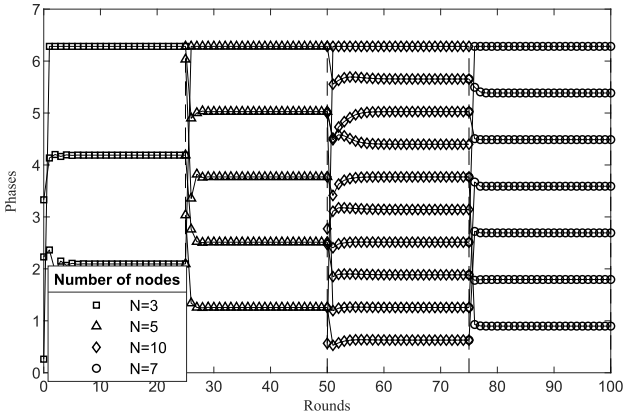


**FIGURE 8.** Stability of the proposed algorithm under dynamic network topology change.

convergence speed coefficient $K\Delta T$ has a value of 2, 4, 6, or 9, where all values lies within the state range $0 < K\Delta T < 2\pi(exp(-\frac{\pi}{4})sin(\frac{\pi}{4}))^{-1}$. Simulation result shows that the averages of consecutive phase differences reach an equilibrium point (blue line). As shown in Fig. 7, the convergence speed become faster as the convergence speed coefficient $K\Delta T$ increases within the convergence interval of $(0, 2\pi(exp(-\frac{\pi}{4})sin(\frac{\pi}{4}))^{-1})$.

Fig. 8 shows the anti-phase convergence of the proposed algorithm with the consideration of dynamic network topology changes. The convergence speed coefficient $K\Delta T$ is set by a value of 8. Initially, we assume three nodes with random initial phases. At the iteration number of 25 and 50, two and five additional nodes are newly connected to the network, respectively. At the iteration number of 75, three nodes leave the network. The result shows that each node control its dynamic phase to maintain proper distance from the other nodes and achieve anti-phase synchronization, irrespective of the change of the number of nodes in the network.

Fig. 9 investigates the maximum/average/minimum number of rounds required to achieve strict de-synchronization as a function of the number of nodes in a network with various convergence speed coefficients $K\Delta T$. The number of nodes increases from 10 to 40 with the granularity of 5 and the convergence speed coefficient are 8, 12 and 17.

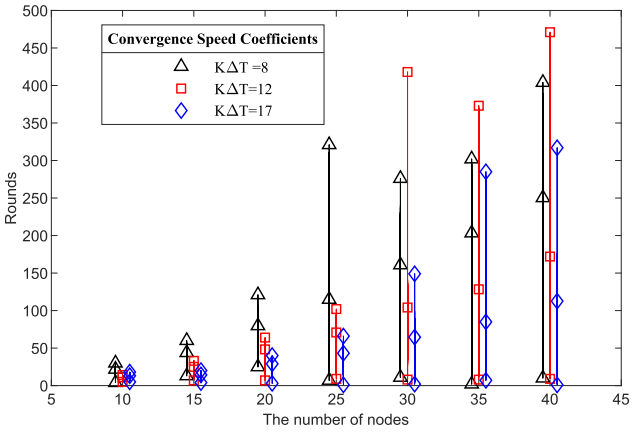


**FIGURE 9.** Avg/max/min number of rounds for stable equilibrium as functions of the numbers of nodes in network and convergence speed coefficient.

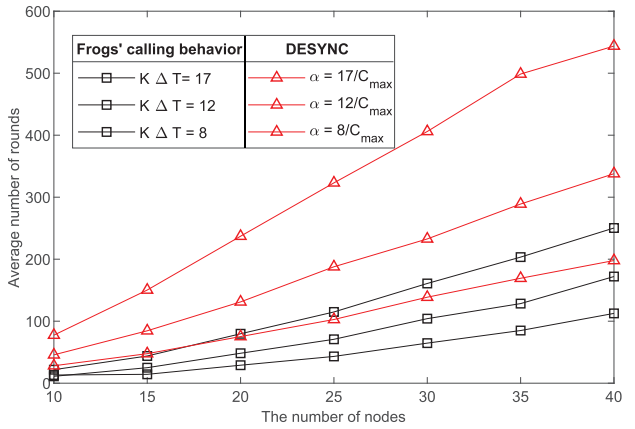


**FIGURE 10.** The average number of rounds required to achieve a stable equilibrium under the proposed algorithm and DESYNC.

The result shows that the number of iterations needed to achieve strict de-synchronization increases as the number of users in a network increases. On the other hand, the system gets to anti-phase synchronization state faster as the value of convergence speed coefficient approaches $(0, 2\pi(exp(-\frac{\pi}{4})sin(\frac{\pi}{4}))^{-1})$.

In the following figure, we compare the convergence performance of the proposed algorithm with that of a comparison algorithm. As a comparison algorithm, we choose DESYNC explained in subsection II-B. It is noted that DESYNC operates in a distributed manner with the purpose of achieving de-synchronization. These characteristics of DESYNC are same to those of the proposed frogs' calling behavior algorithm, which enables the direct performance comparison between DESYNC and the proposed algorithm. The simulation parameters used for DESYNC are the same to those used for the proposed algorithm. Fig. 10 compares the convergence speeds of the proposed algorithm and DESYNC as a function of the number of nodes in a network using the same simulation parameters for both algorithms. Considering the fact that the convergence speeds of the proposed algorithm and DESYNC are affected by the convergence speed coefficient $K\Delta T$ and the weighted factor $\alpha \in [0, 1]$, respectively, we choose the values of $\alpha$s to maintain the ratio of $K\Delta T$ to $C_{max} = 2\pi(exp(-\frac{\pi}{4})sin(\frac{\pi}{4}))^{-1}$. That is, the performance of DESYNC with $\alpha = a/C_{max}$ is compared with that of the proposed algorithm with $K\Delta T = a$. In this simulation runs, we use the convergence speed coefficients $K\Delta T$ of 8, 12 and 17 and $\alpha$ are set to $8/C_{max}$, $12/C_{max}$ and $17/C_{max}$, respectively. The result shows that the number of rounds required to obtain de-synchronization using frogs' calling behavior algorithm is less than that using DESYNC, which verifies that the convergence speed of the proposed algorithm is higher than that of DESYNC.

## VI. CONCLUSION

In this paper, being inspired by frogs' calling behavior model, we proposed the fair and distributed resource allocation method. Proposed algorithm ensures that the phase of each node is set to be evenly spaced across all nodes in a network without the help of any centralized coordinator and thus achieves fair resource allocation among nodes in a distributed manner in a network. Analysis for the convergence criteria showed that the proposed algorithm achieves strict de-synchronization for all phase-coupled oscillators if the convergence speed coefficient satisfies $0 < K\Delta T < 2\pi(exp(-\frac{\pi}{4})sin(\frac{\pi}{4}))^{-1}$. Simulation results showed that the proposed algorithm achieves strict and stable de-synchronization even in the dynamically changing network topology. In addition, the number of iterations need to achieve stable de-synchronization increases as the number of nodes in network system increases, and the proposed algorithm is able to get to the stable de-synchronization state faster as the convergence speed coefficient $K\Delta T$ increases within the interval of $(0, 2\pi(exp(-\frac{\pi}{4})\sin(\frac{\pi}{4}))^{-1})$. Furthermore, the convergence speed of the proposed algorithm is shown to be faster than that of DESYNC. In this study, we proposed the resource allocation algorithm in the context of a fully-connected network. As a further work, we consider extending our algorithm to be suitable for the use in wireless multi-hop network environments.

## REFERENCES

[1] J.-Y. Jung, H.-H. Choi, and J.-R. Lee, "Survey of bio-inspired resource allocation algorithms and MAC protocol design based on a bio-inspired algorithm for mobile ad hoc networks," *IEEE Commun. Mag.*, vol. 56, no. 1, pp. 119–127, Jan. 2018.

[2] A. Rajandekar and B. Sikdar, "A survey of MAC layer issues and protocols for machine-to-machine communications," *IEEE Internet Things J.*, vol. 2, no. 2, pp. 175–186, Apr. 2015.

[3] H.-H. Choi and J.-R. Lee, "Principles, applications, and challenges of synchronization in nature for future mobile communication systems," *Mobile Inf. Syst.*, vol. 2017, Jan. 2017, Art. no. 8932631.

[4] Y.-J. Kim, H.-H. Choi, and J.-R. Lee, "A bioinspired fair resource-allocation algorithm for TDMA-based distributed sensor networks for IoT," *Int. J. Distrib. Sensor Netw.*, vol. 12, no. 4, Apr. 2016, Art. no. 7296359.

[5] R. Pagliari, Y.-W. P. Hong, and A. Scaglione, "Bio-inspired algorithms for decentralized round-robin and proportional fair scheduling," *IEEE J. Sel. Areas Commun.*, vol. 28, no. 4, pp. 564–575, May 2010.

[6] S. Yamanaka, M. Hashimoto, and N. Wakamiya, "An efficient scheduling method based on pulse-coupled oscillator model for heterogeneous large-scale wireless sensor networks," *Procedia Comput. Sci.*, vol. 83, no. 1, pp. 568–575, Jan. 2016.
[7] F. Cucker and S. Smale, "Emergent behavior in flocks," *IEEE Trans. Autom. Control*, vol. 52, no. 5, pp. 852–862, May 2007.
[8] J. Degesys, I. Rose, A. Patel, and R. Nagpal, "DESYNC: Self-organizing desynchronization and TDMA on wireless sensor networks," in *Proc. 6th Int. Conf. Inf. Process. Sensor Netw.* Apr. 2007, pp. 11–20.
[9] J. Degesys and R. Nagpal, "Towards desynchronization of multi-hop topologies," in *Proc. 2nd IEEE Int. Conf. Self-Adapt. Self-Organizing Syst.*, Oct. 2008, pp. 129–138.
[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 *Proc. 1st Int. Symp. Appl. Sci. Biomed. Commun. Technol.* Oct. 2008, pp. 1–5.
[11] H.-H. Choi and J.-R. Lee, "A flocking-inspired algorithm for fair resource allocation in vehicle-mounted mobile relays," *J. Netw. Comput. Appl.*, vol. 85, pp. 134–142, May 2017.
[12] C. S. Peskin, "Mathematical aspects of heart physiology," *Courant Inst. Math. Sci.*, vol. 58, no. 2, pp. 485–494, 2003.
[13] I. Aihara, H. Kitahata, K. Yoshikawa, and K. Aihara, "Mathematical modeling of frogs' calling behavior and its possible application to artificial life and robotics," *Artif. Life Robot.*, vol. 12, nos. 1–2, pp. 29–32, May. 2007.
[14] J. Polastre, R. Szewczyk, C. Sharp, and D. Culler, "The mote revolution: Low power wireless sensor network devices," in *Proc. Symp. High Perform. Chips.*, Aug. 2004.
[15] S. Liu, S. Feng, W. Yea, and H. Zhuang, "Slot allocation algorithms in centralized scheduling scheme for IEEE 802.16 based wireless mesh networks," *Comput. Commun.*, vol. 32, no. 5, pp. 943–953, Mar. 2009.
[16] P. Björklund, P. Värbrand, and D. Yuan, "A column generation method for spatial TDMA scheduling in ad hoc networks," *Ad Hoc Netw.*, vol. 2, no. 4, pp. 405–418, Oct. 2004.
[17] J. Daafouz, P. Riedinger, and C. Iung, "Stability analysis and control synthesis for switched systems: A switched Lyapunov function approach," *IEEE Trans. Autom. Control*, vol. 47, no. 11, PP. 1883–1887, Nov. 2002.

**DARA RON** received the B.S. degree with the Department of Electrical and Energy Engineering, Institute of Technology of Cambodia (ITC), Phnom Penh, Cambodia, in 2017. He is currently pursuing the M.S. and Ph.D. integrated degrees with the School of Electrical and Electronics Engineering, Chung-Ang University, South Korea. His current research interests include bio-inspired algorithms, LoRaWAN protocol, and artificial intelligent-based wireless networks.

**JUNG-RYUN LEE** received the B.S. and M.S. degrees in mathematics from Seoul National University, in 1995 and 1997, respectively, and the Ph.D. degree in electrical and electronics engineering from the Korea Advanced Institute of Science and Technology (KAIST), in 2006. From 1997 to 2005, he was a Chief Research Engineer with LG Electronics, South Korea. From 2006 to 2007, he was a full time Lecturer of electronic engineering with the University of Incheon. Since 2008, he has been a Professor with the School of Electrical and Electronics Engineering, Chung-Ang University, South Korea. His research interests include low energy networks and algorithms, bio-inspired autonomous networks, and artificial intelligence-based networks. He is a member of IEICE, KIISE, and KICS.