KICS The Korean Institute of Communications and Information Sciences

Available online at www.sciencedirect.com

ScienceDirect

ICT Express 9 (2023) 1085–1094

www.elsevier.com/locate/icte

# DACODE: Distributed adaptive communication framework for energy efficient industrial IoT-based heterogeneous WSN

Junsuk Oh, Donghyun Lee, Demeke Shumeye Lakew, Sungrae Cho*

*School of Computer Science and Engineering, Chung-Ang University, Seoul, South Korea*

Received 27 September 2022; received in revised form 7 January 2023; accepted 22 February 2023
Available online 10 March 2023

**Abstract**

As the cornerstone of IoT-based systems, WSN connecting a wide range of intelligent sensor nodes is expected to bring significant changes in the near future. Due to the limited battery capacity of the sensor node, WSN considers maximizing network lifetime by minimizing the power consumption to be the most important challenge. To this end, we propose a Distributed Adaptive Communication with On/Off switching and Dual queuing for Energy efficiency (DACODE) as a novel asynchronous duty cycling mechanism. The performance evaluation shows that the proposed mechanism significantly reduces power consumption while maintaining network throughput and guaranteeing data urgency and queue stability.



*Keywords:* Asynchronous duty cycling; Energy efficiency; Data urgency; Queue stability; IoT-based WSN

## 1. Introduction

The Internet of Things (IoT) connects various devices to the Internet to realize ubiquitous computing. By 2030, the number of IoT devices connected to the Internet is expected to be about 500 billion [1]. Thus, IoT, which builds a massive infrastructure, receives considerable attention in various industrial areas due to its flexibility and scalability [2]. In addition, it is expected that the sustainability and stability of industry will be improved as various things are connected to the Internet through IoT [3]. At this point, a wireless sensor network (WSN) connecting a wide range of intelligent sensor nodes is expected to bring significant changes in the near future as the cornerstone of IoT-based systems [4]. Moreover, as most industries become automated, Industry 4.0 relies on IoT-based WSN [5]. Here, WSN is a network consisting of sensor nodes that periodically collect objects, events, etc., generate sensing data, and then transmit it to a sink node [6]. Therefore, IoT attempts to solve various problems in the real world by including WSN as a core technology.

However, due to the limited battery capacity of sensor nodes, maximizing the network lifetime of WSN by minimizing their power consumption is the most critical challenge [6]. To address this challenge, various approaches have been proposed, including clustering-based routing protocols [7,8] and mobile charger-based battery charging strategies [9,10]. However, these approaches are not suitable in some industrial environments, such as a factory, where all sensor nodes are in the range of wireless access points (WAPs). In addition, the battery is mainly depleted by the communication module including the transmitter (TX) and the receiver (RX) rather than the processing module, etc [11]. Therefore, duty cycling mechanisms that turn the radio module on/off to reduce the power consumption of the sensor node are suitable in the above environment. In addition, asynchronous mechanisms (i.e., distributed schemes) are more attractive to energy-efficient IoT deployments than synchronous mechanisms (i.e., centralized schemes) in which sensor nodes require high communication overhead to periodically communicate with a central server or neighboring sensor nodes resulting in high power consumption [12].

As the IoT-based WSN, we consider the environment in which the sensor node periodically senses the factory to generate sensing data and then transmits it to the sink node, and the factory manager (i.e., the sink node) receives it to monitor the factory in real-time. In particular, since the factory

* Corresponding author.
*E-mail addresses:* jsoh@uclab.re.kr (J. Oh), dhlee@uclab.re.kr (D. Lee), demeke@uclab.re.kr (D.S. Lakew), srcho@cau.ac.kr (S. Cho).

Peer review under responsibility of The Korean Institute of Communications and Information Sciences (KICS).

https://doi.org/10.1016/j.icte.2023.02.009

manager moves around the factory or resides at a particular point in the factory, the factory manager can be a mobile sink node. In this environment, if the factory manager is near the sensor node, the sensor node may significantly reduce its power consumption by using a short-range radio module (i.e., direct transmission) instead of a long-range radio module (i.e., indirect transmission via WAPs) to transmit the sensing data [13]. Specifically, we consider the sensor node that determines either the short-range or long-range radio module according to the distance from the sink node.

In this paper, to maximize the IoT-based WSN lifetime, we formulate an optimization problem that minimizes the power consumption of the sensor node by turning the long-range radio module on/off. Consequently, we propose a Distributive Adaptive Communication with On/Off switching and Dual queuing for Energy efficiency (DACODE) as a novel asynchronous duty cycling mechanism to solve the optimization problem. In particular, due to the limited memory and computational capacity of the sensor node, applying complex algorithms to the sensor node is not suitable for energy-constrained IoT [6]. To this end, the proposed mechanism is designed to turn the long-range radio module on/off by observing the trade-off between power consumption and queue size based on Lyapunov drift optimization techniques [14]. That is, the proposed mechanism is designed to determine when to turn the long-range radio module on to transmit the sensing data to the sink node, unlike traditional mechanisms that determine when to turn the radio module on to receive the sensing data from neighboring sensor nodes. More specifically, the main contributions of this paper are summarized as follows:

- If the sensor node belatedly transmits the sensing data, which includes event contents such as gas leakage have occurred, to the sink node, it may bring severe situations such as casualties. Thus, on the optimization problem, we introduce the constraint that data urgency should be guaranteed. To this end, the proposed mechanism is designed to classify the sensing data into either urgent (i.e., the sensing data which includes contents that events have occurred) or non-urgent data, store it in either urgent or non-urgent queues, and then preferentially transmit urgent data over non-urgent data. Here, the proposed mechanism is also designed to turn the long-range radio module on when the sensing data is classified as urgent data.
- If the sensor node waits for some time to be connected to the sink node after turning the long-range radio module on by observing the current queue size of the non-urgent queue, it may bring a severe situation in which the non-urgent queue drops the sensing data generated during this waiting time. Thus, on the optimization problem, we introduce the constraint that queue stability should be guaranteed. To this end, the proposed mechanism is designed to predict the future queue size of the non-urgent queue, observe the trade-off between power consumption and queue size, and then turn the long-range radio module on. Here, the proposed mechanism is also designed to move some sensing data from the non-urgent queue to the urgent queue when the senor node awaits a long time to connect to the sink node than expected.
- To evaluate the performance of the proposed mechanism, we configure a testbed and establish frameworks, which are set up similarly to the real world, and then perform simulations. In addition, we implement a Multi-Priority based QoS MAC (MPQ-MAC) protocol to perform a comparative analysis with the proposed mechanism [15]. Through these simulations, we show that the proposed mechanism significantly reduces the power consumption of the sensor node while maintaining network throughput and guaranteeing data urgency and queue stability.

The remainder of this paper is organized as follows: First, Section 2 outlines the related works for the asynchronous duty cycling mechanism. Section 3 describes a system model for the sensor node. Next, we formulate the optimization problem in Section 4 and propose the DACODE mechanism in Section 5. Finally, we provide the performance evaluation for the proposed mechanism in Section 6, followed by the conclusion in Section 7.

## 2. Related work

To turn the radio module of the sensor node on/off, the asynchronous duty cycling mechanism is based on either transmitter- or receiver-initiated [16,17].

The transmitter-initiated approaches establish communication links between transmitter and receiver nodes by using preamble sampling or Low-Power Listening (LPL) technology [18,19]. The transmitter node transmits a preamble to the receiver node before transmitting the sensing data to initiate communication. The receiver node receives the preamble after awakening and then waits to receive sensing data. However, these approaches increase delay and decrease throughput as it waits for a long time to transmit and receive the preamble between transmitter and receiver nodes.

The receiver-initiated approaches establish communication links between receiver and transmitter nodes by broadcasting beacons [20,21]. In particular, the receiver node notifies transmitter nodes that it may receive sensing data by broadcasting beacons indicating that it is awake to initiate communication. The transmitter node receives beacons after awakening and then transmits the sensing data to the receiver node. Thus. receiver-initiated approaches have shown better performance than the transmitter-initiated approaches in terms of energy efficiency [17].

The asynchronous duty cycling mechanism may be based on receiver-initiated Quality of Service (QoS). The receiver-initiated QoS approaches support the QoS by decreasing the delay for the sensing data with high priority than with sensing data with low priority [22]. The receiver node collects TX beacons containing the address and the priority from transmitter nodes during the waiting timer after awakening, then transmits RX beacons to all transmitter nodes after selecting the transmitter node with the highest priority. The transmitter node transmits the sensing data to the receiver after being selected. However, these approaches increase delay and power

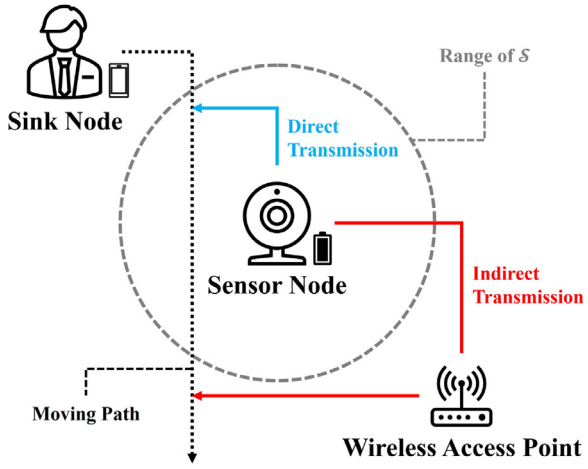



**Fig. 1.** Determining the radio module according to the distance from the sink node.

consumption because it must wait during the waiting timer. Thus, recent receiver-initiated QoS approaches support more than two priority levels in the transmitter node and stop the waiting timer when the TX beacon containing the highest priority is collected in the receiver node [15,23].

In traditional approaches, the sensor node awakes as the receiver node to receive sensing data from neighboring sensor nodes. However, in the considered environment, it is not suitable for energy-efficient IoT for the sensor node to awake as the receiver, since the transmitter node immediately transmits sensing data to the WAP without waiting for beacons from the receiver node. Therefore, a novel approach suitable for the considered environment is required.

## 3. System model

### 3.1. Heterogeneous network model

In the heterogeneous network, we consider the sensor node which includes the short-range radio module $\mathcal{S}$ and the long-range radio module $\mathcal{L}$. Here, $\mathcal{S}$ has lower power consumption than $\mathcal{L}$ under the same conditions. Therefore, as shown in Fig. 1, the sensor node determines either $\mathcal{S}$ or $\mathcal{L}$ to transmit the sensing data according to the distance from the sink node.

For the sensor node in time slot $t$, if $\zeta \in \{\mathcal{S}, \mathcal{L}\}$, an on/off state $\phi_\zeta^t$ and a connection state $\psi_\zeta^t$ are defined as follows:

$$\phi_\zeta^t = \begin{cases} 1, & \text{if } \zeta \text{ is turned on,} \\ 0, & \text{otherwise,} \end{cases} \tag{1}$$

$$\psi_\zeta^t = \begin{cases} 1, & \text{if } \zeta \text{ is connected to the sink node,} \\ 0, & \text{otherwise,} \end{cases} \tag{2}$$

where $\psi_\zeta^t$ is 0 when $\phi_\zeta^t$ is 0. For the switch from 0 to 1 and its opposition, we express it as an on/off in $\phi_\zeta^t$ and an up/down in $\psi_\zeta^t$. Therefore, the sensor node tries the switch up when it completes the switch on. At this point, the sensor node always tries the $\mathcal{S}$ switch up since $\phi_\mathcal{S}^t$ is always 1 to detect the distance from the sink node. In addition, as summarized in Table 1, the sensor node waits for some time until each switch is completed. Here, $c = [c_{\min}, c_{\max}]$.

**Table 1**
Time to wait for each switch to complete.

| State type | $\phi_\mathcal{S}^t$, $\phi_\mathcal{L}^t$ | | $\psi_\mathcal{S}^t$ | | $\psi_\mathcal{L}^t$ | |
|---|---|---|---|---|---|---|
| Switch type | on | off | up | down | up | down |
| Time slots | 0 | 0 | 0 | 0 | $c$ | 0 |

If the sink node is not near the sensor node, power is wasted by $\mathcal{S}$, which is always turned on. In contrast, if the sink node is near the sensor node, power is saved as the sensor node transmits the sensing data using $\mathcal{S}$ instead of $\mathcal{L}$. Therefore, power saving by $\mathcal{S}$ may offset power waste by $\mathcal{S}$. To this end, $r$, the ratio of time near the sensor node by the sink node, is defined as follows:

$$r = \frac{1}{T}\sum_{t=0}^{T-1} \psi_\mathcal{S}^t, \tag{3}$$

where this offset occurs when $r$ is greater than or equal to the particular threshold.

### 3.2. Power consumption model

For the sensor node in time slot $t$, the power consumption $\mathbf{P}^t$ is defined as follows [24]:

$$\mathbf{P}^t = \mathbf{P}_{\text{idle}}^t + \mathbf{P}_{\text{proc}}^t + \sum_\zeta \mathbf{P}_\zeta^t, \tag{4}$$

where $\mathbf{P}_{\text{idle}}^t$ is the idle power consumption on the board. $\mathbf{P}_{\text{proc}}^t$ is the power consumption on the processor, which depends on the processor utility (i.e., the complexity of the algorithm executed on the sensor node). In addition, $\mathbf{P}_\zeta^t$, the power consumption on the radio module, is defined as follows:

$$\mathbf{P}_\zeta^t = \phi_\zeta^t \cdot \left(\mathbf{P}_{\zeta,\text{idle}}^t + \psi_\zeta^t \cdot \mathbf{P}_{\zeta,\text{TX}}^t(b_\zeta^t, w_\zeta^t)\right), \tag{5}$$

where $\mathbf{P}_{\zeta,\text{idle}}^t$ is the idle power consumption on the radio module and is 0 when $\phi_\zeta^t$ is 0. Moreover, $\mathbf{P}_{\zeta,\text{TX}}^t(b_\zeta^t, w_\zeta^t)$ is the transmission power consumption on the radio module and is 0 when $\psi_\zeta^t$ is 0, which depends on the payload $b_\zeta^t = [0, b_{\zeta,\max}]$ (bytes) and the decibel-milliwatts $w_\zeta^t = [w_{\zeta,\min}, w_{\zeta,\max}]$ (dBm).

To increase $r$ by increasing the transmission range of $\mathcal{S}$, the sensor node sets $w_\mathcal{S}^t$ as $w_{\mathcal{S},\max}$. At this point, the sensor node also sets $w_\mathcal{L}^t$ as any constant since the WAP is fixed at a particular point. That is, $\mathbf{P}_\zeta^t$ per byte is affected only by $b_\zeta^t$ and reduced as $b_\zeta^t$ approaches $b_{\zeta,\max}$. Therefore, as shown in Fig. 2, the sensor node determines the $\mathcal{L}$ switch on and the transmission of the sensing data according to the available $b_\mathcal{L}^t$.

### 3.3. Dual queue model

In the monitoring application, for the factory problems such as gas leakage, the basic purpose is to rapidly detect and troubleshoot. At this point, to reduce power consumption, if the sensor node belated transmits sensing data, which can figure out these problems, it may bring severe situations such as casualties or property damage. Therefore, we consider the

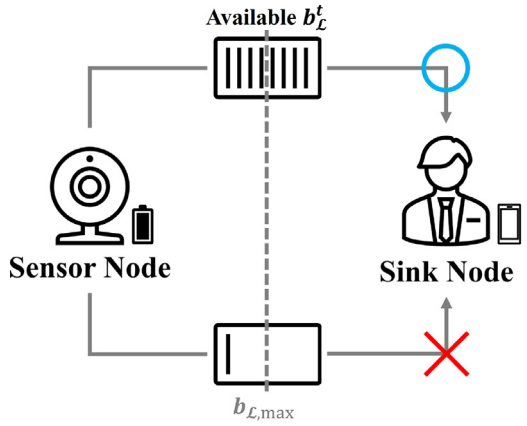



**Fig. 2.** Determining the $\mathcal{L}$ switch on and the transmission of the sensing data according to the available $b^t_{\mathcal{L}}$.

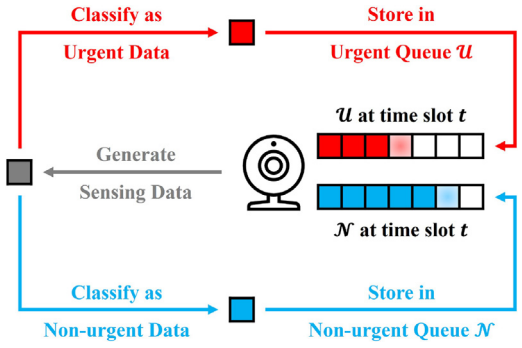



**Fig. 3.** Determining the queue and the $\mathcal{L}$ switch on according to $\lambda^t$.

dual queue in order not to undermine this basic purpose. For the sensor node in time slot $t$, a classifier $\lambda^t$, which classifies the generated sensing data as urgent data if it contains the factory problem and otherwise classifies it as non-urgent data, is defined as follows:

$$\lambda^t = \begin{cases} 1, & \text{if the generated sensing data is} \\ & \text{classified as urgent data,} \\ 0, & \text{otherwise,} \end{cases} \tag{6}$$

where the sensor node stores the generated sensing data in an urgent queue $\mathcal{U}$ if $\lambda^t$ is 1 and otherwise stores the generated sensing data in a non-urgent queue $\mathcal{N}$. Therefore, as shown in Fig. 3, the sensor node determines either $\mathcal{U}$ or $\mathcal{N}$ and the $\mathcal{L}$ switch on to transmit the sensing data according to $\lambda^t$.

For the sensor node in time slot $t$, if $\xi \in \{\mathcal{U}, \mathcal{N}\}$, the update of the queue size $\mathbf{Q}^t_\xi$ is defined as follows:

$$\mathbf{Q}^{t+1}_\xi = \max\left[0, \mathbf{Q}^t_\xi - o^t_\xi\right] + i^t_\xi, \tag{7}$$

where $\mathbf{Q}^t_\xi$ is less than or equal to the maximum queue size $\mathbf{Q}_{\max}$. In addition, $o^t_\xi$ and $i^t_\xi$ are the size of outgoing data and the size of incoming data, respectively, which are defined as follows:

$$o^t_{\mathcal{U}} = \begin{cases} \min\left[\mathbf{Q}^t_{\mathcal{U}}, b_{\mathcal{S},\max}\right], & \text{if } \psi^t_{\mathcal{S}} \text{ is 1,} \\ \min\left[\mathbf{Q}^t_{\mathcal{U}}, b_{\mathcal{L},\max}\right], & \text{if } \psi^t_{\mathcal{L}} \text{ is 1,} \\ 0, & \text{otherwise,} \end{cases} \tag{8}$$

$$o^t_{\mathcal{N}} = \begin{cases} b_{\mathcal{S},\max} - o^t_{\mathcal{U}}, & \text{if } \psi^t_{\mathcal{S}} \text{ is 1,} \\ b_{\mathcal{L},\max} - o^t_{\mathcal{U}}, & \text{if } \psi^t_{\mathcal{L}} \text{ is 1,} \\ 0, & \text{otherwise,} \end{cases} \tag{9}$$

$$i^t_{\mathcal{U}} = \lambda^t \cdot v^t, \tag{10}$$

$$i^t_{\mathcal{N}} = (1 - \lambda^t) \cdot v^t, \tag{11}$$

where urgent data is preferentially transmitted over non-urgent data. $v^t = [v_{\min}, v_{\max}]$ is the volume occupied by the sensing data stored in the queue.

For the sensing data generated at time slot $t$, a queuing delay $\mathbf{D}^t$, which is the difference between the time slot when it is stored and the time slot when it is transmitted, is defined as follows:

$$\mathbf{D}^t \geq 1, \tag{12}$$

where 1 is that the sensing data generated and stored at time slot $t$ can be transmitted from time slot $t+1$.

## 4. Problem formulation

In this paper, we formulate the optimization problem (**P1**) based on the system model as follows:

$$(\mathbf{P1}) \quad \min_{\phi^t_{\mathcal{L}}} \lim_{T\to\infty} \frac{1}{T} \sum_{t=0}^{T-1} \mathbf{P}^t, \tag{13}$$

$$\text{s.t.} \quad \phi^t_\zeta \in \{0,1\}, \forall \zeta \in \{\mathcal{S}, \mathcal{L}\}, \tag{14}$$

$$\psi^t_\zeta \in \{0,1\}, \forall \zeta \in \{\mathcal{S}, \mathcal{L}\}, \tag{15}$$

$$b^t_\zeta = [0, b_{\zeta,\max}], \forall \zeta \in \{\mathcal{S}, \mathcal{L}\}, \tag{16}$$

$$w^t_\zeta = [w_{\zeta,\min}, w_{\zeta,\max}], \forall \zeta \in \{\mathcal{S}, \mathcal{L}\}, \tag{17}$$

where the optimization problem (**P1**) is the non-convex problem by the binary variable $\phi^t_{\mathcal{L}}$. In addition, we introduce a constraint that data urgency should be guaranteed, which is defined as follows:

$$\lambda^t \cdot \mathbb{E}\left[\mathbf{D}^t\right] \leq c_{\max} + 1, \tag{18}$$

where Eq. (18) specifies that urgent data should be transmitted as soon as possible. Moreover, we introduce a constraint that queue stability should be guaranteed, which is defined as follows:

$$\lim_{T\to\infty} \frac{1}{T} \sum_{t=0}^{T-1} \mathbb{E}\left[\mathbf{Q}^t_\xi\right] < \infty, \forall \xi \in \{\mathcal{U}, \mathcal{N}\}, \tag{19}$$

where Eq. (19) indicates the mean rate stability constraint for $\mathcal{U}$ and $\mathcal{N}$. Therefore, the optimization problem (**P1**) is the NP-hard problem by the time-domain objective function equation (13) and the time-domain constraint equations (18) and (19).

## 5. DACODE mechanism

In this paper, as the novel asynchronous duty cycling mechanism, we propose a Distributed Adaptive Communication with On/Off switching and Dual queuing for Energy efficiency (DACODE) to solve the optimization problem (**P1**).

In an on/off switching, we first describe cases for the $\mathcal{L}$ switch on as follows:

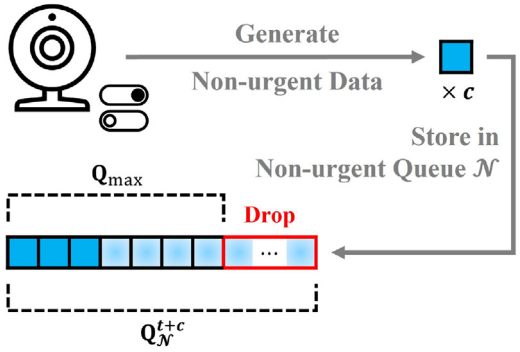



**Fig. 4.** Drop in $\mathcal{N}$ caused by waiting time for $\mathcal{L}$ switch up to complete.

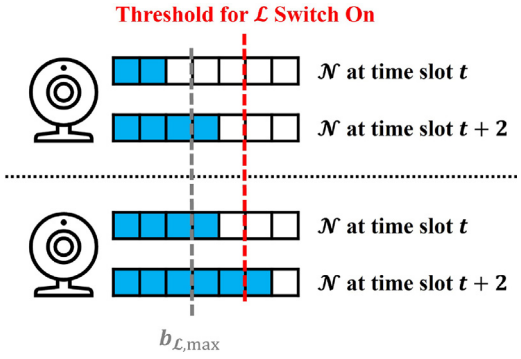



**Fig. 5.** Frequent $\mathcal{L}$ switch on/off caused by not transmitting as much as possible.

- **ON1**: If the generated sensing data is classified as urgent data (i.e., $\mathbf{Q}^t_{\mathcal{U}} > 0$), the sensor node performs the switch on.
- **ON2**: Otherwise, based on the Lyapunov drift optimization technique [14], the sensor node observes the trade-off between $\mathbf{P}^t_{\mathcal{L}}$ and $\mathbf{Q}^t_{\mathcal{N}}$ and performs the switch on.

The sensor node tries the $\mathcal{L}$ switch up when the $\mathcal{L}$ switch on is completed. Here, as shown in Fig. 4, the constraint equation (19) is not satisfied since $\mathbf{Q}^{t+c}_{\mathcal{N}}$ may be larger than $\mathbf{Q}_{\max}$ by the sensing data generated during $c$. To this end, the proposed mechanism is designed to predict $\mathbf{P}^{t+c}_{\mathcal{L}}$ and $\mathbf{Q}^{t+c}_{\mathcal{N}}$, observe the trade-off between them, and then perform the $\mathcal{L}$ switch on. For the sensor node in time slot $t$, the Lyapunov function $\mathbf{F}^t$ is defined as follows:

$$\mathbf{F}^t = \frac{1}{2}\left(\widehat{\mathbf{Q}}^t_{\mathcal{N}}\right)^2 = \frac{1}{2}\left(\mathbf{Q}^t_{\mathcal{N}} + c_{\text{avg}} \cdot v_{\text{avg}}\right)^2, \tag{20}$$

where $c_{\text{avg}}$ and $v_{\text{avg}}$ are the average of $c$ and the average of $v^t$, respectively. Thus, $\widehat{\mathbf{Q}}^t_{\mathcal{N}}$ is the prediction of $\mathbf{Q}^{t+c}_{\mathcal{N}}$. At this point, as $\mathbf{F}^t \geq 0$, the Lyapunov drift $\mathbf{\Delta}^t$ is defined as follows:

$$\mathbf{\Delta}^t = \mathbb{E}\left[\mathbf{F}^{t+1} - \mathbf{F}^t \mid \widehat{\mathbf{Q}}^t_{\mathcal{N}}\right], \tag{21}$$

where drift policy is designed to solve the optimization problem (**P1**) by observing $\widehat{\mathbf{Q}}^t_{\mathcal{N}}$ while minimizing the bound on

$$\widehat{\mathbf{P}}^t_{\mathcal{L}} - G \cdot \mathbf{\Delta}^t, \tag{22}$$

where $\widehat{\mathbf{P}}^t_{\mathcal{L}}$ and $G$ are the prediction of $\mathbf{P}^{t+c}_{\mathcal{L}}$ and the positive constant to control drift policy, respectively. Thus, we transform part of the optimization problem (**P1**) to formulate the optimization problem (**P2**) as follows:

$$(\mathbf{P2}) \quad \min_{\phi^t_{\mathcal{L}}} \left\{\widehat{\mathbf{P}}^t_{\mathcal{L}} + G \cdot \widehat{\mathbf{Q}}^t_{\mathcal{N}} \cdot \left(v_{\text{avg}} - b_{\mathcal{L},\max}\right)\right\}, \tag{23}$$

$$\text{s.t.} \quad \phi^t_{\mathcal{L}} \in \{0, 1\}, \tag{24}$$

$$\psi^t_{\mathcal{L}} \in \{0, 1\}, \tag{25}$$

$$b^t_{\mathcal{L}} = \left[0, b_{\mathcal{L},\max}\right], \tag{26}$$

$$\lambda^t \cdot \mathbb{E}\left[\mathbf{D}^t\right] \leq c_{\max} + 1, \tag{27}$$

$$\lim_{T\to\infty} \frac{1}{T} \sum_{t=0}^{T-1} \mathbb{E}\left[\mathbf{Q}^t_{\mathcal{N}}\right] < \infty, \tag{28}$$

where $v_{\text{avg}}$ and $b_{\mathcal{L},\max}$ are the prediction of $i^{t+c}$ and the prediction of $o^{t+c}$, respectively. At this point, we assume that the solution is 1 or 0 and that the sensor node performs the switch on when the solution is 1. For the sake of simplicity, we assume that $\mathbf{P}^t_{\text{proc}}$ is the power consumption to solve the optimization problem (**P2**). Next, we describe cases for the $\mathcal{L}$ switch off as follows:

- **OFF1**: If the sink node is detected to be near the sensor node (i.e., $\psi^t_{\mathcal{S}} = 1$), the sensor node performs the switch off.
- **OFF2**: Otherwise, when there is no urgent data (i.e., $\mathbf{Q}^t_{\mathcal{U}} = 0$) and there is not enough available $b^t_{\mathcal{L}}$ (i.e., $\mathbf{Q}^t_{\mathcal{N}} < b_{\mathcal{L},\max}$), the sensor node performs the switch off.

If the sensing data is not transmitted as much as possible, the sensor node frequently performs the $\mathcal{L}$ switch on/off as shown in Fig. 5. For the $\mathcal{L}$ switch up, as power is wasted during $c$, frequent $\mathcal{L}$ switch on/off is correlated to frequent power waste. Therefore, the proposed mechanism is designed to transmit the sensing data as much as possible.

In a dual queuing, by Eqs. (8), (9), (10), and (11), the proposed mechanism is designed to classify the sensing data into either urgent or non-urgent data, store it in either $\mathcal{U}$ or $\mathcal{N}$ and then preferentially transmit urgent data over non-urgent data. At this point, to transmit the sensing data using $\mathcal{L}$, the sensor node should first perform the $\mathcal{L}$ switch on. Therefore, $t^\star$, which is a time slot expected for the $\mathcal{L}$ switch up to be completed, is defined as follows:

$$t^\star = \begin{cases} t + c_{\text{avg}}, & \text{if the up-switch is tried,} \\ t + 1, & \text{if the up-switch is not completed at } t^\star, \\ -1, & \text{otherwise,} \end{cases} \tag{29}$$

where the constraint equation (28) is not satisfied if the $\mathcal{L}$ switch up is not completed at $t^\star$ (i.e., the expectation is wrong) as shown in Fig. 6. To this end, the proposed mechanism is designed to move the sensing data stored at index 0 in $\mathcal{N}$ into

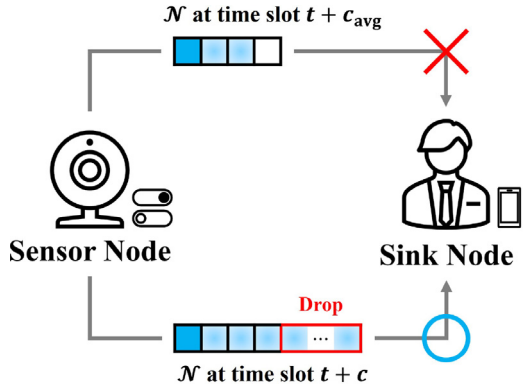



**Fig. 6.** Drop in $\mathcal{N}$ caused by the wrong expectation.

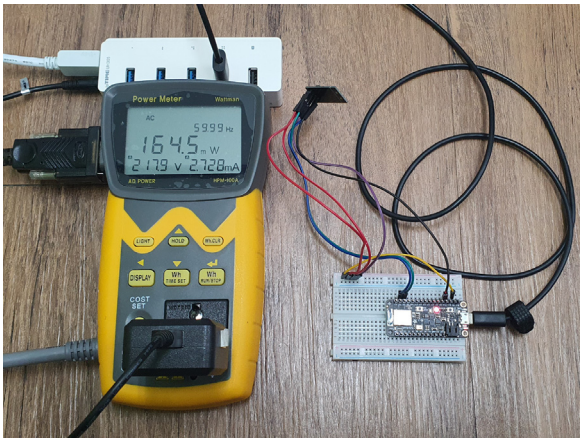


**Fig. 7.** Testbed configured for parameter setup.

$\mathcal{U}$. Therefore, $o^t_{\mathcal{N}}$ and $i^t_{\mathcal{U}}$ are redefined as follows:

$$o^t_{\mathcal{N}} = \begin{cases} b_{\mathcal{S},\max} - o^t_{\mathcal{U}}, & \text{if } \psi^t_{\mathcal{S}} \text{ is 1,} \\ b_{\mathcal{L},\max} - o^t_{\mathcal{U}}, & \text{if } \psi^t_{\mathcal{L}} \text{ is 1,} \\ v_{\mathcal{N}(0)}, & \text{if the expectation is wrong,} \\ 0, & \text{otherwise,} \end{cases} \tag{30}$$

$$i^t_{\mathcal{U}} = \begin{cases} \lambda^t \cdot v^t + o^t_{\mathcal{N}}, & \text{if the expectation is wrong,} \\ \lambda^t \cdot v^t, & \text{otherwise,} \end{cases} \tag{31}$$

where $v_{\mathcal{N}(0)}$ is the volume occupied by the sensing data stored at index 0 in $\mathcal{N}$.

In the proposed mechanism as shown in Algorithm 1, the on/off switching reduces the power consumption of the sensor node while guaranteeing data urgency and queue stability, and the dual queuing supports guaranteeing data urgency and queue stability. Therefore, the sensor node with the proposed mechanism adaptively determines its own action according to its own state and surrounding environment, without control by the central server or cooperation with neighboring sensor nodes.

**Algorithm 1** Distributed Adaptive Communication with On/Off switching and Dual queuing for Energy efficiency (DACODE) Mechanism

$\{\phi^0_{\mathcal{S}}, \psi^0_{\mathcal{S}}, \phi^0_{\mathcal{L}}, \psi^0_{\mathcal{L}}\} \leftarrow \{1, 0, 0, 0\}$
$\{\mathbf{Q}^0_{\mathcal{U}}, \mathbf{Q}^0_{\mathcal{N}}\} \leftarrow \{0, 0\}$
$\{t, t^\star\} \leftarrow \{0, -1\}$
**for** $t = 0, 1, 2, \ldots, T-1$ **do**
▷ *On/Off switching*
  **if** the sink node is nearby **then**
    $\psi^t_{\mathcal{S}} \leftarrow 1$
    $\{\phi^t_{\mathcal{L}}, \psi^t_{\mathcal{L}}\} \leftarrow \{0, 0\}$ by **OFF1**
  **else**
    $\psi^t_{\mathcal{S}} \leftarrow 0$
    **if** $\phi^t_{\mathcal{L}} == 1$ **then**
      **if** $\psi^t_{\mathcal{L}} == 1$ **then**
        **if** $\mathbf{Q}^t_{\mathcal{U}} == 0$ and $\mathbf{Q}^t_{\mathcal{N}} < b_{\mathcal{L},\max}$ **then**
          $\{\phi^t_{\mathcal{L}}, \psi^t_{\mathcal{L}}\} \leftarrow \{0, 0\}$ by **OFF2**
        **end if**
      **else if** the $\mathcal{L}$ switch up is finished **then**
        $\psi^t_{\mathcal{L}} \leftarrow 1$
      **end if**
    **else**
      **if** $\mathbf{Q}^t_{\mathcal{U}} > 0$ **then**
        $\phi^t_{\mathcal{L}} \leftarrow 1$ by **ON1**
        the $\mathcal{L}$ switch up is tried
      **else if** the solution to **P2** is 1 **then**
        $\phi^t_{\mathcal{L}} \leftarrow 1$ by **ON2**
        the $\mathcal{L}$ switch up is tried
      **end if**
    **end if**
  **end if**
  $\{\phi^{t+1}_{\mathcal{S}}, \psi^{t+1}_{\mathcal{S}}, \phi^{t+1}_{\mathcal{L}}, \psi^{t+1}_{\mathcal{L}}\} \leftarrow \{\phi^t_{\mathcal{S}}, \psi^t_{\mathcal{S}}, \phi^t_{\mathcal{L}}, \psi^t_{\mathcal{L}}\}$
▷ *Dual queuing*
  $\{o^t_{\mathcal{U}}, o^t_{\mathcal{N}}\} \leftarrow$ by equations (8) and (30)
  $\{i^t_{\mathcal{U}}, i^t_{\mathcal{N}}\} \leftarrow$ by equations (31) and (11)
  $\{\mathbf{Q}^{t+1}_{\mathcal{U}}, \mathbf{Q}^{t+1}_{\mathcal{N}}\} \leftarrow$ by equation (7)
  $t^\star \leftarrow$ by equation (29)
**end for**

## 6. Performance evaluation

### 6.1. Simulation setup

First, as technologies that realize an industrial IoT include Wi-Fi, WiMAX, LRPAN, Bluetooth, and LoRA, we set $\mathcal{S}$ and $\mathcal{L}$ as Bluetooth Low Energy (BLE) and Wi-Fi, respectively [3]. Thus, the sensor node is set as Adafruit Feather nRF52840 Sense board with ESP8266. As summarized in Table 2, we set the parameters of the system model and proposed mechanism. At this point, we set the length of the time slot to 10 ms and configure a testbed as shown in Fig. 7 to measure some parameters as follows:

- When measured for 1000 s, the maximum payloads per second transmitted over BLE and Wi-Fi are measured at 127e3 and 1072e3 bytes, respectively. Thus, we set

**Table 2**
Parameter setup for the performance evaluation.

| Parameter | Value |
|---|---|
| $T$ | $10^5$ |
| $b_{\mathcal{S},\max}$ | 1270 bytes |
| $b_{\mathcal{L},\max}$ | 10720 bytes |
| $w_{\mathcal{S},\min}$, $w_{\mathcal{L},\min}$ | 2 dBm |
| $w_{\mathcal{S},\max}$, $w_{\mathcal{L},\max}$ | 8 dBm |
| $\mathbf{P}^t_{\text{idle}}$ | 20.6537 μA |
| $\mathbf{P}^t_{\text{proc}}$ | 0.6647 μA |
| $\mathbf{P}^t_{\mathcal{S},\text{idle}}$ | 0.0025 μA |
| $\mathbf{P}^t_{\mathcal{S},\text{TX}}(1270, 2)$ | 4.2334 μA |
| $\mathbf{P}^t_{\mathcal{S},\text{TX}}(1270, 4)$ | 4.7951 μA |
| $\mathbf{P}^t_{\mathcal{S},\text{TX}}(1270, 6)$ | 4.9120 μA |
| $\mathbf{P}^t_{\mathcal{S},\text{TX}}(1270, 8)$ | 4.9283 μA |
| $\mathbf{P}^t_{\mathcal{L},\text{idle}}$ | 37.6941 μA |
| $\mathbf{P}^t_{\mathcal{L},\text{TX}}(1270, 2)$ | 2.3469 μA |
| $\mathbf{P}^t_{\mathcal{L},\text{TX}}(10720, 2)$ | 7.1930 μA |
| $\widehat{\mathbf{P}}^t_{\mathcal{L}}$ | 44.8871 μA |
| $\mathbf{Q}_{\max}$ | 64 KB |
| $c_{\text{avg}}$ | 391 |
| $c_{\min}$ | 222 |
| $c_{\max}$ | 527 |
| $v_{\text{avg}}$ | 35 bytes |
| $v_{\min}$ | 30 bytes |
| $v_{\max}$ | 40 bytes |
| $G$ | $0.8 \times 10^{-7}$ |

$b_{\mathcal{S},\max}$ and $b_{\mathcal{L},\max}$ by converting it into bytes per time slot.

- When measured for 1000 s, each combination of $\mathbf{P}^t$ is measured as shown in Fig. 8. Thus, we set each parameter of $\mathbf{P}^t$ by obtaining the average power consumption per second through differences between combinations and then converting it into the average power consumption per time slot.
- When measured 1000 times, $c_{\min}$ and $c_{\max}$ are measured at 222 and 527, respectively. Thus, we set $c_{\text{avg}}$ by averaging the measurements.
- When measured 1000 times, since the sensing data includes ID, date and time, temperature, and humidity, $v_{\min}$ and $v_{\max}$ are measured at 30 and 40 bytes, respectively. Thus, we set $v_{\text{avg}}$ by averaging the measurements.
- When measured for each $G$, the threshold at which the solution to the optimization problem ((P2)) becomes 1 is measured as shown in Fig. 9. Here, $\mathbf{Q}^{t+c}_{\mathcal{N}}$ is the minimum and maximum queue sizes that $\mathcal{N}$ may have when connected to the sink node. Thus, we set $G$ for guaranteeing queue stability and preventing frequent $\mathcal{L}$ switch on/off.

In addition, we set $w^t_{\mathcal{S}}$ and $w^t_{\mathcal{L}}$ as 8 and 2, respectively. Moreover, $\widehat{\mathbf{P}}^t_{\mathcal{L}}$ is set to 44.8871 $\mu$A and $\mathbf{Q}_{\max}$ is set to 64 KB [25].

Next, for the performance evaluation of the proposed mechanism, we set standards such as power consumption and network throughput and set benchmarks as follows:

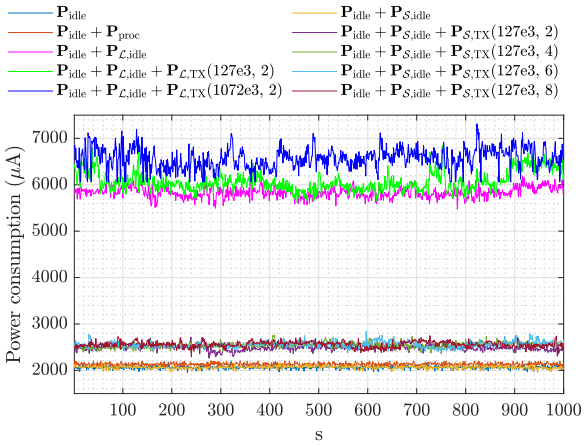



**Fig. 8.** Power consumption per second.

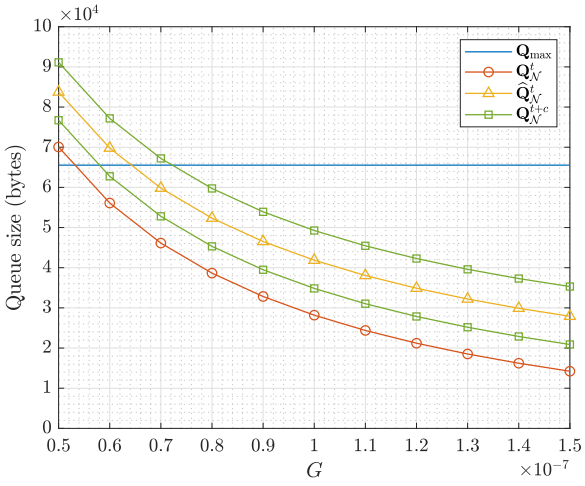


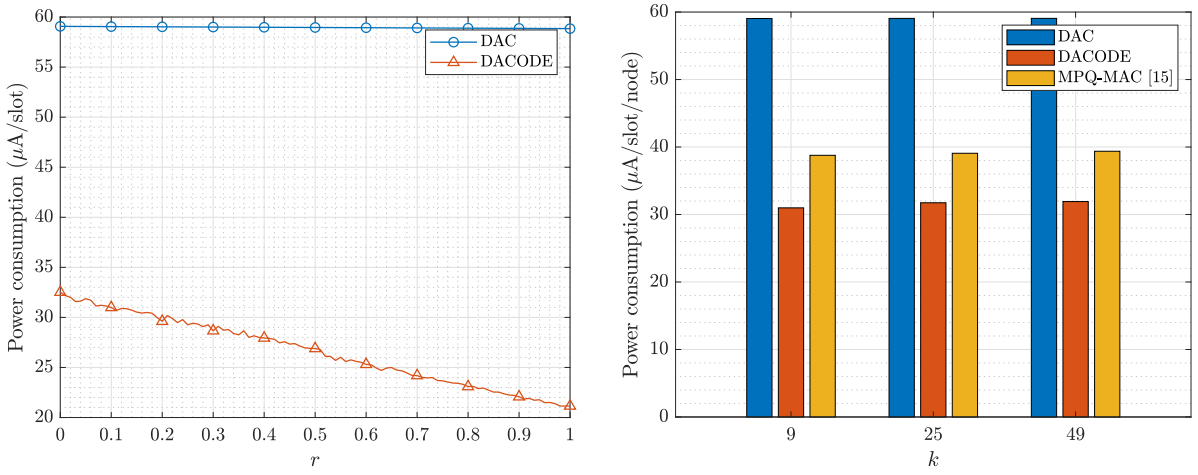

**Fig. 9.** Queue size threshold according to $G$.

- *Distributed Adaptive Communication (DAC)*: In this scheme, the sensor node determines either $\mathcal{S}$ or $\mathcal{L}$ according to the distance from the sink node without any mechanism or protocol.
- *MPQ-MAC* [15]: This is the application of MPQ-MAC protocol to the sensor node. Here, since the sensor node directly transmits and receives sensing data to and from the neighboring sensor node, the sensor node uses only $\mathcal{S}$ without $\mathcal{L}$. In addition, the sensor node turns $\mathcal{S}$ on for every two time slots and requires reception power consumption, which is about 75% of transmission power consumption [26].

In addition, we set DACODE as the application of the proposed mechanism to the sensor node.

Finally, to compare the proposed mechanism with benchmarks, we set up frameworks that implement the considered sensor node and environment as follows:

- *Single Node*: This is a framework with WAP and the sink node and a single sensor node. Here, $\lambda^t$ is 1 when $t$ is $[24000, 25000)$ or $[74000, 75000)$, and $\psi^t_{\mathcal{S}}$ is 1 when $t$ is $[50000 \cdot (1 - r), 50000)$ or $[50000 \cdot (2 - r), 100000)$.




(a) Single node framework

(b) Multiple nodes framework

**Fig. 10.** Power consumption according to $r$ or $k$.

In addition, $r$ is spaced 0.01 from 0 to 1. Thus, since MPQ-MAC requires multiple sensor nodes, we compare DACODE with only DAC.

- *Multiple Nodes*: This is a framework with WAP and the sink node and multiple sensor nodes. Here, it is assumed that the sink node stays at each sensor node for some time to check the sensor node. In addition, it is assumed that each sensor node has up to four neighboring sensor nodes within the range of $\mathcal{S}$. Moreover, we set $k$, which is the number of sensor nodes, to 9, 25, and 49. Thus, we compare DACODE with DAC and MPQ-MAC.

### 6.2. Simulation results

First, as shown in Fig. 10, we express power consumption as $\mu$A/slot in the single node framework and as $\mu$A/slot per node in the multiple nodes framework. Fig. 10(a) shows that the power consumption of DAC and DACODE decreases as $r$ increases. At this point, DACODE reduces power consumption by at least 45% to up to 66% compared to DAC because it prevents unnecessary power consumption by turning $\mathcal{L}$ on/off. In addition, Fig. 10(b) shows that the power consumption of DAC, DACODE, and MPQ-MAC is constant regardless of $k$ because they are all distributed schemes that are not controlled by the central server. Here, DACODE reduces 19% compared to MPQ-MAC because it prevents unnecessary reception power consumption.

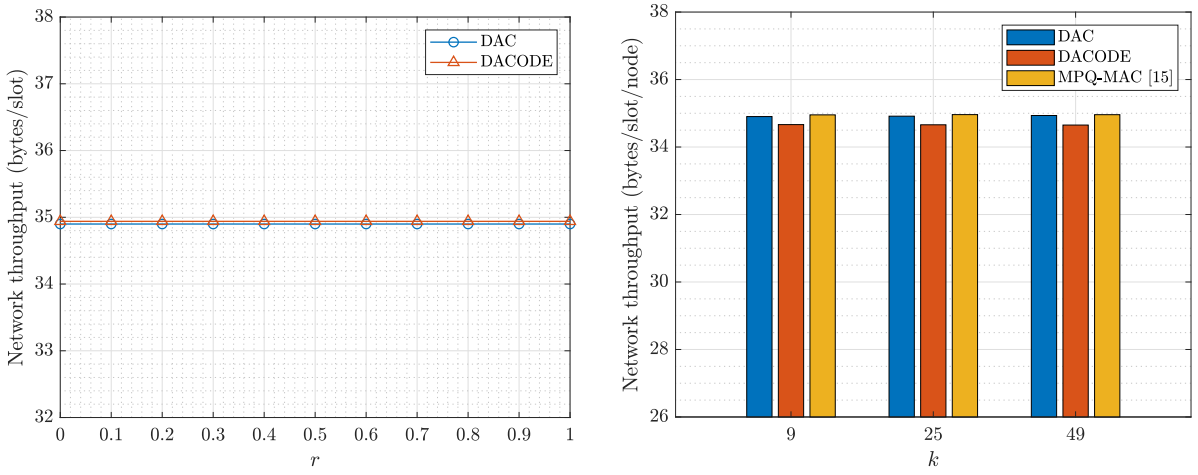
Next, as shown in Fig. 11, we express the network throughput as bytes/slot in the single node framework and as bytes/slot per node in the multiple nodes framework. Here, Fig. 11(a) shows similar network throughput for DAC and DACODE regardless of $r$. In addition, Fig. 11(b) shows similar network throughput for DAC, DACODE, and MPQ-MAC regardless of $k$. Moreover, the network throughput is achieved near $v_{\text{avg}}$ for DAC, DACODE, and MPQ-MAC. At this point, the slight difference in network throughput occurs because the volume of the sensing data generated by the sensor node is slightly different.

Meanwhile, as shown in Fig. 12, for DACODE in the single node framework, we express the queuing delay as slots. Here, $\mathbf{D}_{\text{avg}}$ and $\mathbf{D}_{\text{max}}$ are the average and maximum of $\mathbf{D}^t$, respectively. That is, Fig. 12 shows that DACODE satisfies the constraint that data urgency should be guaranteed since $\mathbf{D}_{\text{max}}$ is smaller than $c_{\text{max}}+1$ regardless of $r$. At this point, if $r$ is greater than or equal to 0.52, the queuing delay is always 1 because the sink node is near the sensor node when urgent data is generated. In addition, the queuing delay fluctuates regardless of $r$ since $c$ is inconsistent.

Finally, as shown in Fig. 13, for DACODE in the single node framework, we express the queue size as bytes. Fig. 13 shows that DACODE satisfies the constraint that queue stability should be guaranteed since $\mathbf{Q}^t_{\mathcal{U}}$ and $\mathbf{Q}^t_{\mathcal{N}}$ are smaller than $\mathbf{Q}_{\text{max}}$ regardless of $t$. In addition, some time slots (2719, 5481, etc.) show that queue stability of $\mathcal{N}$ is guaranteed when the expectation is wrong. Here, time slots between 45000 and 49999 show that $\mathbf{Q}^t_{\mathcal{U}}$ and $\mathbf{Q}^t_{\mathcal{N}}$ are close to 0 because the sink node is near the sensor node. Moreover, since the sensing data generated from time slots between 24000 and 24999 is urgent data, $\mathbf{Q}^t_{\mathcal{U}}$ increases from time slot 24000 and $\mathbf{Q}^t_{\mathcal{N}}$ increases from time slot 25001.

## 7. Conclusion

In this paper, to maximize the IoT-based WSN lifetime, we aim to minimize the power consumption of the sensor node while guaranteeing data urgency and queue stability. To this end, as a novel asynchronous duty cycling mechanism, we propose a Distributed Adaptive Communication with On/Off switching and Dual queuing for Energy efficiency (DACODE). First, the proposed mechanism is designed to classify the sensing data into either urgent or non-urgent data, store it in either urgent or non-urgent queues, and then preferentially transmit urgent data over non-urgent data. Here, it is also designed to turn the long-range radio module on when the sensing data is classified as urgent data. In addition, the proposed mechanism is designed to predict the future queue size of the non-urgent queue, observe the trade-off between power consumption and




(a) Single node framework

(b) Multiple nodes framework

**Fig. 11.** Network throughput according to $r$ or $k$.

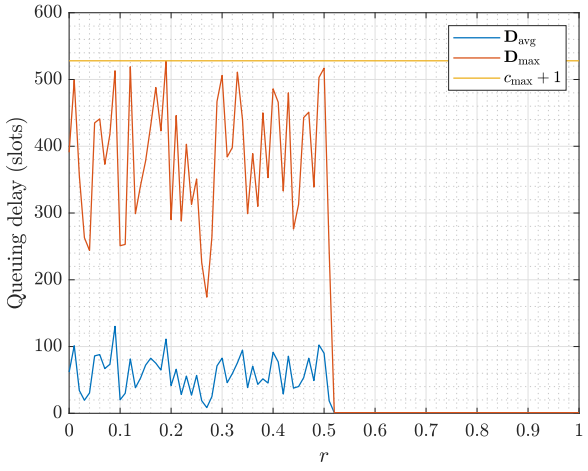



**Fig. 12.** Queuing delay according to $r$.

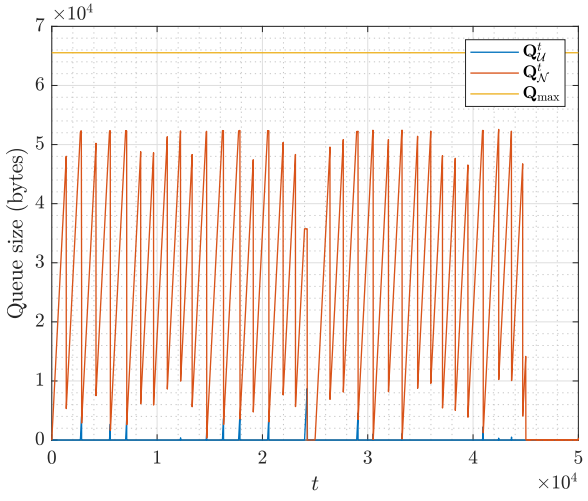



**Fig. 13.** Queuing size according to $t$.

queue size, and then turn the long-range radio module on. Moreover, it is designed to move some sensing data from the non-urgent queue to the urgent queue when the senor node awaits a long time to connect to the sink node than expected. In the performance evaluation, simulation results showed that the proposed mechanism significantly reduces power consumption while maintaining network throughput and guaranteeing data urgency and queue stability.

## CRediT authorship contribution statement

**Junsuk Oh:** Conceptualization, Writing – original draft, Software. **Donghyun Lee:** Methodology, Investigation. **Demeke Shumeye Lakew:** Writing – review & editing, Validation. **Sungrae Cho:** Supervision.

## Declaration of competing interest

The authors declare that there is no conflict of interest in this paper.

## Acknowledgments

This research was supported in part by the Chung-Ang University Graduate Research Scholarship in 2023 and in part by the MSIT (Ministry of Science and ICT), Korea, under the ITRC (Information Technology Research Center) support program (IITP-2023-RS-2022-00156353) supervised by the IITP (Institute for Information and Communications Technology Planning and Evaluation).

## References

[1] Y.B. Zikria, R. Ali, M.K. Afzal, S.W. Kim, Next-generation internet of things (IoT): Opportunities, challenges, and solutions, Sensors 21 (4) (2021) 1174.

[2] E. Sisinni, A. Saifullah, S. Han, U. Jennehag, M. Gidlund, Industrial internet of things: Challenges, opportunities, and directions, IEEE Trans. Ind. Inform. 14 (11) (2018) 4724–4734.

[3] P.K. Malik, R. Sharma, R. Singh, A. Gehlot, S.C. Satapathy, W.S. Alnumay, D. Pelusi, U. Ghosh, J. Nayak, Industrial Internet of Things

and its applications in industry 4.0: State of the art, Comput. Commun. 166 (2021) 125–139.
[4] K. Gulati, R.S.K. Boddu, D. Kapila, S.L. Bangare, N. Chandnani, G. Saravanan, A review paper on wireless sensor network techniques in Internet of Things (IoT), Mater. Today: Proceedings 51 (2022) 161–165.
[5] M. Majid, S. Habib, A.R. Javed, M. Rizwan, G. Srivastava, T.R. Gadekallu, J.C.-W. Lin, Applications of wireless sensor networks and internet of things frameworks in the industry revolution 4.0: A systematic literature review, Sensors 22 (6) (2022) 2087.
[6] J. Amutha, S. Sharma, J. Nagar, WSN strategies based on sensors, deployment, sensing models, coverage and energy efficiency: Review, approaches and open issues, Wirel. Pers. Commun. 111 (2) (2020) 1089–1115.
[7] A. Naeem, A.R. Javed, M. Rizwan, S. Abbas, J.C.-W. Lin, T.R. Gadekallu, DARE-SEP: A hybrid approach of distance aware residual energy-efficient SEP for WSN, IEEE Trans. Green Commun. Netw. 5 (2) (2021) 611–621.
[8] P. Maheshwari, A.K. Sharma, K. Verma, Energy efficient cluster based routing protocol for WSN using butterfly optimization algorithm and ant colony optimization, Ad Hoc Netw. 110 (2021) 102317.
[9] C. Lee, W. Na, G. Jang, C. Lee, S. Cho, Energy-efficient and delay-minimizing charging method with a multiple directional mobile charger, IEEE Internet Things J. 8 (10) (2020) 8291–8303.
[10] D. Lee, C. Lee, G. Jang, W. Na, S. Cho, Energy-efficient directional charging strategy for wireless rechargeable sensor networks, IEEE Internet Things J. 9 (19) (2022) 19034–19048.
[11] J. Singh, R. Kaur, D. Singh, A survey and taxonomy on energy management schemes in wireless sensor networks, J. Syst. Archit. 111 (2020) 101782.
[12] D. Passos, H. Balbi, R. Carrano, C. Albuquerque, Asynchronous radio duty cycling for green IoT: State of the art and future perspectives, IEEE Commun. Mag. 57 (9) (2019) 106–111.
[13] A. Mehto, S. Tapaswi, K. Pattanaik, A review on rendezvous based data acquisition methods in wireless sensor networks with mobile sink, Wirel. Netw. 26 (4) (2020) 2639–2663.
[14] M.J. Neely, Stochastic network optimization with application to communication and queueing systems, Synth. Lect. Commun. Netw. 3 (1) (2010) 1–211.
[15] S. Sarang, M. Drieberg, A. Awang, Multi-priority based QoS MAC protocol for wireless sensor networks, in: 2017 7th IEEE International Conference on System Engineering and Technology, ICSET, IEEE, 2017, pp. 54–58.
[16] J. Polastre, J. Hill, D. Culler, Versatile low power media access for wireless sensor networks, in: Proceedings of the 2nd International Conference on Embedded Networked Sensor Systems, 2004, pp. 95–107.
[17] Y. Sun, O. Gurewitz, D.B. Johnson, RI-MAC: a receiver-initiated asynchronous duty cycle MAC protocol for dynamic traffic loads in wireless sensor networks, in: Proceedings of the 6th ACM Conference on Embedded Network Sensor Systems, 2008, pp. 1–14.
[18] R. Su, Z. Gong, D. Zhang, C. Li, Y. Chen, R. Venkatesan, An adaptive asynchronous wake-up scheme for underwater acoustic sensor networks using deep reinforcement learning, IEEE Trans. Veh. Technol. 70 (2) (2021) 1851–1865.
[19] Z. Ahmed, M.M. Rehan, O. Chughtai, M.W. Rehan, AD-RDC: A novel adaptive dynamic radio duty cycle mechanism for low-power IoT devices, IEEE Internet Things J. 9 (15) (2022) 13376–13389.
[20] M.L. Wymore, D. Qiao, RIVER-MAC: A Receiver-initiated asynchronously duty-cycled MAC protocol for the internet of things, in: 2019 IEEE 43rd Annual Computer Software and Applications Conference, COMPSAC, 1, IEEE, 2019, pp. 860–869.
[21] C. Blondia, Evaluation of the end-to-end response times in an energy harvesting wireless sensor network using a receiver-initiated MAC protocol, Ad Hoc Netw. 136 (2022) 102971.
[22] S.C. Kim, J.H. Jeon, H.J. Park, Qos aware energy-efficient (QAEE) MAC protocol for energy harvesting wireless sensor networks, in: International Conference on Hybrid Information Technology, Springer, 2012, pp. 41–48.
[23] N.T.T. Hang, N.C. Trinh, N.T. Ban, M. Raza, H.X. Nguyen, Delay and reliability analysis of p-persistent carrier sense multiple access for multi-event industrial wireless sensor networks, IEEE Sens. J. 20 (20) (2020) 12402–12414.
[24] F. Kaup, P. Gottschling, D. Hausheer, PowerPi: Measuring and modeling the power consumption of the raspberry Pi, in: 39th Annual IEEE Conference on Local Computer Networks, IEEE, 2014, pp. 236–243.
[25] IBM, WebSphere Application Server Network Deployment traditional 9.0.5.x, https://www.ibm.com/docs/en/was-nd/9.0.5, (accessed: 2023-01-07).
[26] K. Shahzad, M. O'Nils, Condition monitoring in industry 4.0-design challenges and possibilities: A case study, in: 2018 Workshop on Metrology for Industry 4.0 and IoT, IEEE, 2018, pp. 101–106.