An Energy-Efficient and Delay-Aware Wireless Computing System for Industrial Wireless Sensor Networks

Citation:

Katsuya Suto, Hiroki Nishiyama, Nei Kato, and Chih-Wei Huang, "An Energy-Efficient and Delay-Aware Wireless Computing System for Industrial Wireless Sensor Networks," IEEE Access, vol. 3, pp. 1026-1035, Jul. 2015.

URL:

http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=7123559

An Energy-Efficient and Delay-Aware Wireless Computing System for Industrial Wireless Sensor Networks

Katsuya Suto[§], Hiroki Nishiyama[§], Nei Kato[§], Chih-Wei Huang[‡]
[§]Graduate School of Information Sciences (GSIS), Tohoku University, Japan
[‡]Department of Communication Engineering, National Central University, Taoyuan, Taiwan
E-mails: [§]{suto, bigtree, kato}@it.ecei.tohoku.ac.jp, [‡]cwhuang@ce.ncu.edu.tw

***Abstract*—Industrial wireless sensor networks have attracted much attention as a cornerstone to making the smart factories real. Utilizing industrial wireless sensor networks as a base for smart factories makes it possible to optimize the production line without human resources since it provides industrial Internet of Things (IoT) service, where various types of data are collected from sensors and mined to control the machines based on the analysis result. On the other hand, a fog computing node, which executes such real-time feedback control, should be capable of real-time data collection, management, and processing. To achieve these requirements, in this paper, we introduce Wireless Computing System (WCS) as a fog computing node. Since there are a lot of servers and each server has 60 GHz antennas to connect with other servers and sensors, WCS has high collecting and processing capabilities. However, in order to fulfill a demand for real-time feedback control, WCS needs to satisfy an acceptable delay for data collection. Additionally, lower power consumption is required in order to reduce the cost for factory operation. Therefore, we propose an Energy-Efficient and Delay-Aware Wireless Computing System (E2DA-WCS), which jointly controls the sleep schedule and the number of links to minimize the system power consumption while satisfying an acceptable delay. Furthermore, the effectiveness of our proposed system is evaluated through extensive computer simulations.**

***Index Terms*—Industrial wireless sensor network, wireless computing system, low power consumption, delay-aware data collection.**

I. Introduction

Recent advances in wireless sensor network technology indicate the realization of smart factories, which opens up a new dimension for factory management [1], [2]. A smart factory based on a wireless sensor network provides industrial Internet of Things (IoT) service [3], [4]. In this service, a computing system collects various kinds of data from machines and sensors and mines a large amount of collected data (i.e., industrial big data) to obtain valuable information for factory operation. Machines are automatically controlled by using the obtained information to make an efficient production line (i.e., adequate production speed, low power consumption, failure prediction, and so forth). Therefore, the smart factory makes it possible to optimize the factory operation without human resources [5], [6].

In this paper, we focus on a fog-based industrial wireless sensor network [7], in which a fog computing node gathers data from sensors and mines the collected data. In comparison

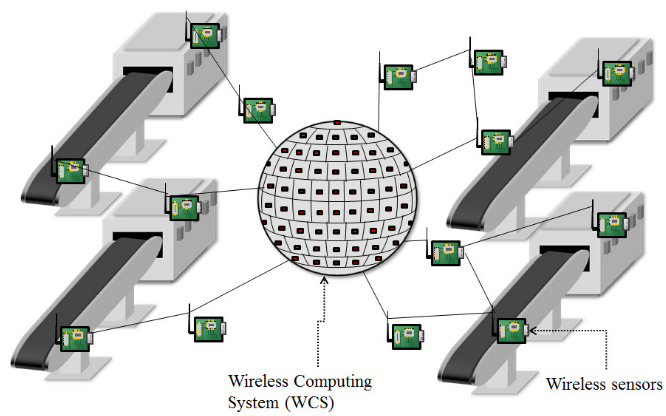


Fig. 1. Our envisioned industrial wireless sensor network using wireless computing system.

with the cloud-based industrial wireless sensor network [8], where the cloud servers in the data center execute the industrial IoT application, the fog-based industrial wireless sensor network can shorten feedback latency. Moreover, in order to achieve enough capability for providing such service, a fog computing node should satisfy the following requirements: (i) high processing performance in order to support real-time big data mining, (ii) concurrent data collection from a lot of sensors, (iii) high service availability, (iv) low system power consumption for a low cost factory operation.

In order to satisfy the aforementioned requirements, we focus on a Wireless Computing System (WCS) [9]–[11], as shown in Fig. 1. While this system is originally used for rack architecture in data center networks, we introduce this system as a fog computing node in industrial wireless sensor networks. Since WCS effectively accommodates a lot of servers, it can achieve high processing capability even if the space is limited. Additionally, each server has two 60 GHz antennas, where one of the antennas, referred to as outside antenna, is used to communicate with the sensors and the other antenna, referred to as inside antenna, is used to communicate with the other servers. Since the 60 GHz antenna has high directivity with high data rate (4-15 Gbps) [12], [13] and there exist multiple outer antennas, WCS can collect data from a lot of sensors concurrently. Although conventional computing systems based

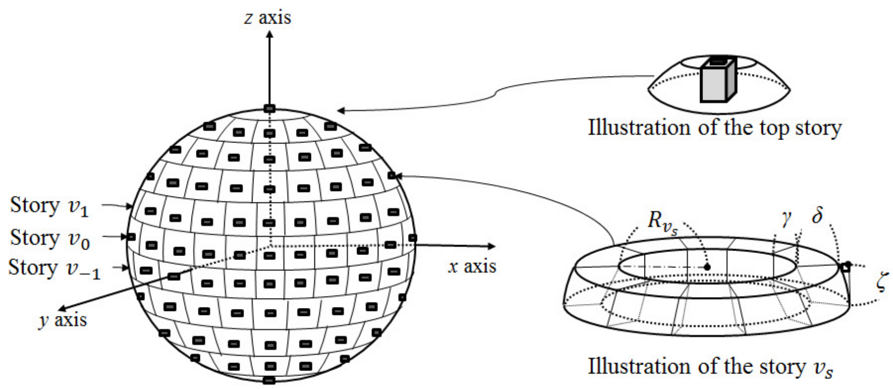


Fig. 2. Architecture of our envisioned spherical wireless computing system.

on wired cables have a drastic decrease in service availability if switches cease to function, WCS achieves high service availability since it keeps the network connectivity even if some servers cease to function. Moreover, WCS achieves lower system power consumption in comparison with wired computing systems. For instance, the collective power consumption of antennas in a WCS consisting of 100 servers is 6 W [10], [13] while the collective power consumption of the routers in a wired computing system based on a two-tier fat tree [14] is 264 W [15].

In this paper, we aim to investigate a system operation scheme for industrial IoT service. Since IoT service is sensitive to the delay, we need to satisfy an acceptable delay for data collection. Additionally, we need to reduce the power consumption of our envisioned system in order to achieve an energy-efficient factory. Consequently, both delay for data collection and system power consumption should be considered. To tackle this research issue, we first construct a mathematical model to evaluate the system power consumption and delay for data collection, and thereby show a trade-off relationship between these values. Furthermore, based on the trade-off relationship, we propose an Energy-Efficient Delay-Aware Wireless Computing System (E2DA-WCS). Our proposed system minimizes the system power consumption while satisfying the acceptable delay of each datum, by controlling the number of servers in a sleep state and the number of links (i.e., the degree of servers).

The remainder of this paper is organized as follows. In section II, we describe the architecture, multiple access scheme, and sleep mode in our envisioned computing system. Additionally, in this section, we show a trade-off relationship between power consumption and delay required for data collection. Our proposed energy-efficient delay-aware system operation scheme is explained in Section III. We present the performance evaluation in Section IV. Finally, concluding remarks are provided in Section V.

II. Envisioned Computing System

In this section, we introduce our envisioned computing system for industrial wireless sensor networks. First, we explain the architecture of our envisioned computing system and its network topology. Then, the time slot allocation scheme for our envisioned system is presented. Furthermore, we describe the sleep mode, which is used to reduce the system power consumption. Finally, we show the trade-off relationship between the system power consumption and data collection delay in our envisioned system.

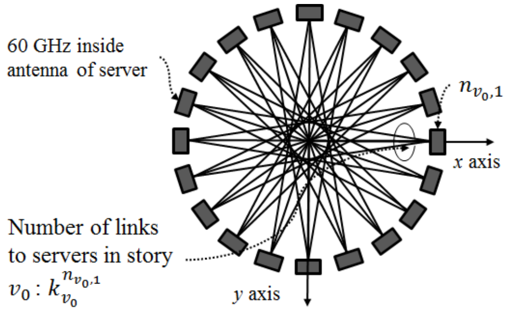
A. Architecture

In this paper, we use a spherical WCS whose form is a spherical shape as shown in Fig. 2. In our envisioned WCS, servers are circularly arranged in both the vertical and horizontal direction, which makes it possible to ensure a low path loss (i.e., high data rate) between any servers even when the degree of servers increases [11].

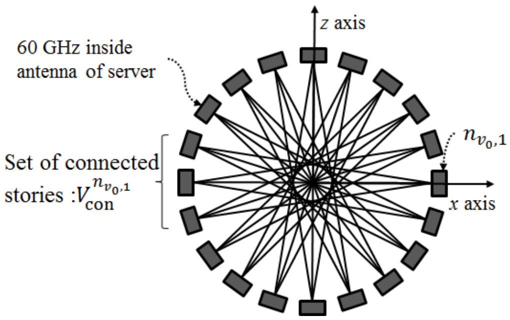
Our envisioned spherical WCS consists of multiple stories, which are shaped like a partial hemisphere to make the envisioned system sphere-shaped. Additionally, each story has multiple containers, which have dimensions of γ meters wide, δ meters deep, and ζ meters high, and a blade server is put in each container. We assume that a set of all servers is defined as $N = \{n_1, n_2, \dots, n_{|N|}\}$, where $|N|$ is the total number of servers, and a set of stories is defined as $V = \{v_{-S}, \dots, v_{-1}, v_0, v_1 \dots, v_S\}$, where $S = (|V| - 1)/2$ and $|V|$ denotes the total number of stories. Additionally, the set of servers in story v_s, N_{v_s}, is given by $\{n_{v_s,1}, n_{v_s,2}, \dots, n_{v_s,|N_{v_s}|}\}$. The number of servers in stories v_0 and v_S, $|N_{v_0}|$ and $|N_{v_S}|$ are set to $2(|V| - 1)$ and 1, respectively. Additionally, since the number of servers in story v_s, $|N_{v_s}|$, is decided based on the inner-radius of the smaller surface of story v_s, R_{v_s}, and the width of the container, γ, the value of $|N_{v_s}|$ is expressed as

$$|N_{v_s}| = \begin{cases} 2(|V| - 1), & \text{if } s = 0, \\ 1, & \text{if } s = S, \\ (2\pi R_{v_s})/\gamma, & \text{otherwise.} \end{cases} \quad (1)$$

Fig. 3 shows the intra-system network topology of our envisioned WCS. In this system, each server transmits data to servers diagonally opposite to it and the number of links from transmitter $n_{v_s,i}$ in story v_s to receivers, $k_{n_{v_s,i}}$, can be changed by controlling the radiation angle of the antenna. The set of stories that contain neighbor servers of server $n_{v_s,i}$, $V_{\text{con}}^{n_{v_s,i}}$,



(a) Network topology in story v_0 from top view



(b) Network topology from side view.

Fig. 3. Intra-system network topologies of our envisioned system.

is expressed as $V_{\text{con}}^{n_{v_s,i}} = \{v_{(-s-Q)}, \dots, v_{-s}, \dots, v_{(-s+Q)}\}$, where $Q = (|V_{\text{con}}^{n_{v_s,i}}| - 1)/2$. Assuming that $k_{v_q}^{n_{v_s,i}}$ is the number of links from server $n_{v_s,i}$ to its neighbor servers in story $v_q \in V_{\text{con}}^{n_{v_s,i}}$, the value of $k_{n_{v_s,i}}$ is expressed as the sum of $k_{v_q}^{n_{v_s,i}}$.

$$k_{n_{v_s,i}} = \sum_{v_q \in V_{\text{con}}^{n_{v_s,i}}} k_{v_q}^{n_{v_s,i}}. \tag{2}$$

Moreover, in this paper, we assume that, for all servers, $|V_{\text{con}}^{n_{v_s,i}}|$ and $k_{n_{v_s,i}}$ are set to the same values, $|V_{\text{con}}|$ and $\langle k \rangle$, respectively. Therefore, intra-system network topologies from the top and side views become a Cayley graph as shown in Fig. 2. This network topology facilitates the design of communication and networking protocols (e.g., multiple access and routing protocols).

B. Space and time division multiple access scheme

Here, we introduce an efficient time slot allocation procedure for our envisioned system. In order to use time slots efficiently, we consider both space- and time-division multiplexing. Our scheme uses time slot cycles in order to fairly allocate time slots to servers and each time slot cycle is divided into multiple time slots. Each time slot is allocated to servers that do not mutually interfere with each other.

Since the network topology of our envisioned system is constructed in a regular pattern (i.e., diagonally opposite servers are connected with the same degree), we can simplify the time slot allocation procedure by dividing it into a story selection phase and a server selection phase. The story selection phase selects the stories whose servers do not interfere with servers in other simultaneously selected stories. On the other hand, the server selection phase chooses multiple servers that do not interfere with each other in each story in order to allocate an independent time slot to servers without interference.

Procedure 1 demonstrates the time slot allocation procedure. A centralized server executes this procedure.

First, the centralized server executes the story selection phase in order to choose *non-interference story groups*, $\bar{v}_i$, in which the inter-story servers do not interfere with each other, from the set of stories, V. Thus, this phase creates a set of $\bar{v}$, $\bar{V} = \{\bar{v}_1, \bar{v}_2, \dots, \bar{v}_{|\bar{V}|}\}$. As shown in Fig. 3(b), the servers $|V_{\text{con}}|$ distant stories apart do not interfere with each other. Assuming that $|V_{\text{con}}| \leq |V|$, the value of $|\bar{V}|$ can be given as

$$|\bar{V}| = |V_{\text{con}}|. \tag{3}$$

Following the story selection phase, the centralized server executes the server selection phase for each group $\bar{v}_i$ in order to allocate the time slots to servers without mutual interference. Servers in each story $v_s \in \bar{v}_i$ are divided into multiple *non-interference server groups*, $\bar{n}_{\bar{v}_i,j}$, in which the intra-story servers do not interfere with each other. In other words, since each group $\bar{n}_{\bar{v}_i,j}$ consists of *non-interference server groups* in each story $v_s \in \bar{v}_i$, $\bar{n}_{v_s,j}$, $\bar{n}_{\bar{v}_i,j}$ is expressed as

$$\bar{n}_{\bar{v}_i,j} = \bigcup_{v_s \in \bar{v}_i} \bar{n}_{v_s,j}. \tag{4}$$

Therefore, the centralized server creates sets of *non-interference server groups* in each story $v_s \in \bar{v}_i$, $\bar{N}_{v_s} = \{\bar{n}_{v_s,1}, \bar{n}_{v_s,2}, \dots, \bar{n}_{v_s,|\bar{N}_{v_s}|}\}$. Then, based on the created $\bar{N}_{v_s}$, it creates a set of *non-interference server groups* in group $\bar{v}_i$, $\bar{N}_{\bar{v}_i} = \{\bar{n}_{\bar{v}_i,1}, \bar{n}_{\bar{v}_i,2}, \dots, \bar{n}_{\bar{v}_i,|\bar{N}_{\bar{v}_i}|}\}$. By using the set $\bar{N}_{\bar{v}_i}$, the centralized server allocates independent time slots to the servers. Since the value of $|\bar{N}_{\bar{v}_i}|$ is the maximum value of $|\bar{N}_{v_s}|$ for all $v_s \in \bar{v}_i$, we have

$$|\bar{N}_{\bar{v}_i}| = \max_{v_s \in \bar{v}_i} |\bar{N}_{v_s}|. \tag{5}$$

Furthermore, since the number of intra-story servers that do not interfere with other intra-story servers in story v_s is given by $\lfloor |N_{v_s}|/k_{v_s,v_{-s}} \rfloor$, where $k_{v_s,v_{-s}}$ is the number of links from any server in story v_s to servers in the diagonally opposite story v_{-s}, the value of $|\bar{N}_{v_s}|$ can be expressed as

$$|\bar{N}_{v_s}| = \left\lceil \frac{|N_{v_s}|}{\lfloor |N_{v_s}|/k_{v_s,v_{-s}} \rfloor} \right\rceil. \tag{6}$$

Finally, the centralized server calculates the amount of time slots, $|T|$, which is expressed with the sum of $|\bar{N}_{\bar{v}_i}|$, as follows.

$$|T| = \sum_{i=1}^{|\bar{V}|} |\bar{N}_{\bar{v}_i}| \tag{7}$$

Additionally, since each time slot has constant length [s], λ, the time slot cycle, Λ, is given as

$$\Lambda = |T|\lambda \tag{8}$$

Since a channel with bandwidth W is divided into $|T|$ time slots, the data transmission rate between servers n_i and n_j,

Procedure 1 Time slot allocation procedure

Create a set of *non-interference story groups*, $\bar{V} \leftarrow \{\bar{v}_1, \bar{v}_2, \dots, \bar{v}_{|\bar{V}|}\}$
for $i = 1$ to $|\bar{V}|$ **do**
  Create a set of *non-interference server groups*, $\bar{N}_{\bar{v}_i} \leftarrow \{\bar{n}_{\bar{v}_i,1}, \bar{n}_{\bar{v}_i,2}, \dots, \bar{n}_{\bar{v}_i,|\bar{N}_{\bar{v}_i}|}\}$, from servers in all stories v_s, $v_s \in \bar{v}_i$,
  Allocate independent time slots to servers in group $\bar{n}_{\bar{v}_i,j}$, for all j, $1 \leq j \leq |\bar{N}_{\bar{v}_i}|$
  $i \leftarrow i + 1$
end for
Calculate the amount of time slots, $|T|$, based on (7)
return the value of $|T|$

$\theta_{n_{i,j}}$ is expressed as

$$\theta_{n_{i,j}} = \frac{W}{|T|} \log_2 \left(1 + \frac{P_{n_i} g_{n_{i,j}}}{\sigma^2}\right). \tag{9}$$

Here, P_{n_i} is the transmission power of server n_i, σ^2 represents the noise power level and $g_{n_{i,j}}$ denotes the channel gain between servers n_i and n_j.

C. Sleep mode

Since low system power consumption is required, we use a sleep mode for the servers in our envisioned system. Therefore, we introduce a summary of sleep mode and demonstrate mathematical expressions to evaluate the system power consumption and data collection delay in multi-hop communication.

The sleep mode aims to reduce system power consumption by letting some servers enter the sleep state. During the sleep state, the power consumption of the server is low. Assuming that ρ percent of the servers enter sleep state, the system power consumption [W], E_{system}, is expressed as

$$\begin{aligned} E_{\text{system}} = &\rho(E^{\text{server}}_{\text{sleep}} + E^{\text{antenna}}_{\text{sleep}}) \\ &+(1-\rho)(E^{\text{server}}_{\text{active}} + E^{\text{antenna}}_{\text{active}}) \end{aligned} \tag{10}$$

where $E^{\text{server}}_{\text{sleep}}$ and $E^{\text{antenna}}_{\text{sleep}}$ are the power consumption of the server and the antenna in sleep state, and $E^{\text{server}}_{\text{active}}$ and $E^{\text{antenna}}_{\text{active}}$ denote the power consumption of the server and the antenna in active state.

However, servers cannot act as router while in sleep state in this system. In other words, servers in sleep state are removed from the intra-system network and the original degree distribution of the network, p_k, is changed. Here, we assume that the servers in sleep state are randomly selected regardless of their degree and location. The degree distribution when some servers enter the sleep state, p'_k, can be expressed based on (10) in [16], as follows.

$$p'_k = \sum_{i=k} \binom{i}{k} (f_i)^{i-k} (1-f_i)^k p_{\text{active},i} \tag{11}$$

where f_i is the probability that a server with degree k enters the sleep state and $p_{\text{active},i}$ denotes the degree distribution of servers in active state. Additionally, $p_{\text{active},i}$ is given as

$$p_{\text{active},i} = \frac{(1-f_i)p_i}{1-\sum_j f_j p_j}, \tag{12}$$

Since all servers have the same degree in our envisioned system, the degree distribution of the intra-system network, p_k, can be defined as

$$p_k = \begin{cases} 1, & \text{if } k = \langle k \rangle, \\ 0, & \text{otherwise.} \end{cases} \tag{13}$$

Additionally, since ρ percent of the servers with degree $\langle k \rangle$ enter the sleep state, f_k is given as

$$f_k = \begin{cases} \rho, & \text{if } k = \langle k \rangle, \\ 0, & \text{otherwise.} \end{cases} \tag{14}$$

Therefore, p'_k can be rewritten as

$$p'_k = \binom{\langle k \rangle}{k} \rho^{\langle k \rangle - k} (1-\rho)^k p_{\langle k \rangle}. \tag{15}$$

Moreover, the average degree when ρ percent of the servers enter the sleep state, $\langle k_\rho \rangle$, can be described based on (15).

$$\langle k_\rho \rangle = \sum_{k=0} k p'_k. \tag{16}$$

Based on (16), we can derive the average hop count between any servers when ρ percent of the servers are in sleep state, $\langle h_\rho \rangle$. Let $|n_{h_l}|$ be the number of l-hop distant servers from a server. Since the 0-hop distant server is the source server and the source server has $\langle k_\rho \rangle$ neighbors, $|n_{h_0}|$ and $|n_{h_1}|$ are 1 and $\langle k_\rho \rangle$, respectively. Although 1-hop distant servers also have $\langle k_\rho \rangle$ neighbors, some of their neighbors are the same server. Therefore, the number of 2-hop distant servers, $|n_{h_2}|$ is given by $\mu \langle k_\rho \rangle^2$, where μ is the ratio of common servers. We suppose that μ is a constant value regardless of hop count. Therefore, $|n_{h_l}|$ is represented as $|n_{h_l}| = \mu^{l-1} \langle k_\rho \rangle^l$. Additionally, since the total number of active servers, $|N_{\text{active}}|$, is given as the sum of the number of h-hop distant servers as h goes from 0 to maximum hop count, H_ρ, we have

$$\begin{aligned} |N_{\text{active}}| &= \sum_{l=0}^{H_\rho} |n_{h_l}| \\ &= 1 + \langle k_\rho \rangle + \mu \langle k_\rho \rangle^2 + \cdots + \mu^{H_\rho - 1} \langle k_\rho \rangle^{H_\rho} \\ &= 1 + \frac{\langle k_\rho \rangle - \mu^{H_\rho} \langle k_\rho \rangle^{H_\rho + 1}}{1 - \mu \langle k_\rho \rangle}. \end{aligned} \tag{17}$$

Therefore, the value of H_ρ can be formulated as

$$\begin{aligned} H_\rho = &\frac{1}{\ln(\mu \langle k_\rho \rangle)} \times \\ &\ln \left\{ \left(|N_{\text{active}}| + \frac{\langle k_\rho \rangle}{\mu \langle k_\rho \rangle - 1} - 1 \right) \left(\frac{\mu \langle k_\rho \rangle - 1}{\langle k_\rho \rangle} \right) \right\}. \end{aligned} \tag{18}$$

Hence, the average hop count between any servers when ρ percent of the servers are in sleep state, $\langle h_\rho \rangle$ can be given as

$$\langle h_\rho \rangle = \frac{1}{\rho |N|} \sum_{l=0}^{H_\rho} l |n_{h_l}|. \tag{19}$$

In the multi-hop communication, since each server waits to transmit its data until its own time slot, the average wait time on each hop is given by $\Lambda/2$. Ignoring queuing delay, the delay from a server which has received data from a sensor to the destination server (referred to as internal delay) when ρ percent of the servers are in sleep state, D_ρ^{int}, can be expressed as

$$D_\rho^{\mathrm{int}} = \langle h_\rho \rangle \frac{\Lambda}{2}. \quad (20)$$

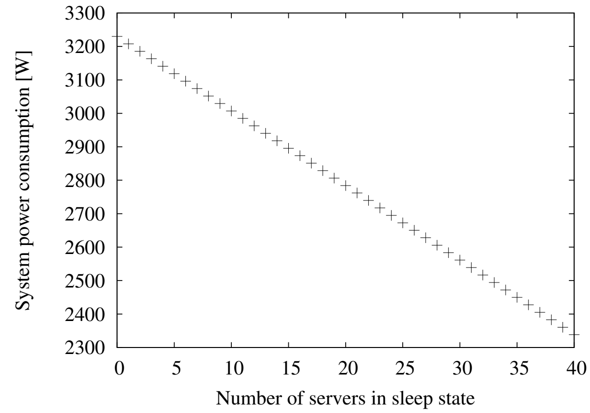
D. Trade-off relationship between power consumption and delay

The rest of this section describes the trade-off relationship between system power consumption and internal delay, which is derived from the aforementioned mathematical expressions.

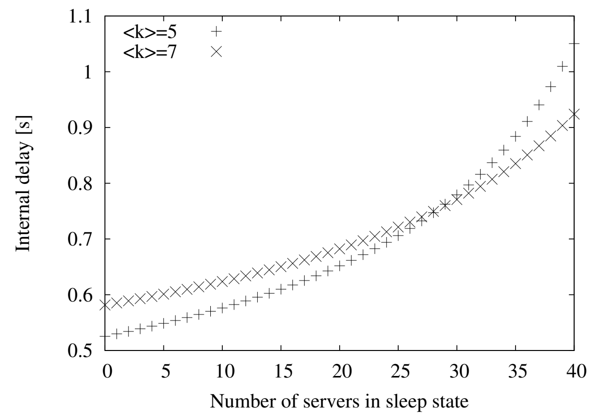
Fig. 4 shows the effects of the number of servers in sleep state on the system power consumption and average internal delay. This result is calculated based on (10) and (20) and the parameters are set as follows: $E_{\mathrm{sleep}}^{\mathrm{server}} = 10[W]$ [17], $E_{\mathrm{active}}^{\mathrm{server}} = 32[W]$ [18], $E_{\mathrm{sleep}}^{\mathrm{antenna}} = 0.008[W]$ [19], $E_{\mathrm{active}}^{\mathrm{antenna}} = 0.3[W]$ [10], $|N| = 100$, $\mu = 0.4$, $\lambda = 0.02[s]$.

As shown in Fig. 4(a), the system power consumption can be reduced in inverse proportion to the number of servers in sleep state. On the other hand, as shown in Fig. 4(b), the average internal delay increases with the increase of the number of servers in sleep state because the average hop count between servers becomes higher in such case. From these results, we can show a trade-off relationship between the system power consumption and internal delay. Consequently, we need to investigate an optimization scheme that considers both system power consumption and internal delay to find an optimal number of servers in sleep state.

Moreover, as shown in Fig. 4(b), the average internal delay changes depending on the degree of servers (i.e., $\langle k \rangle = 5$ and $\langle k \rangle = 7$). Since a network with lower degree has a shorter time slot cycle, the delay is lower when the number of servers in sleep state is low. However, since the hop-count increases in an exponential fashion and growth rate is higher in case of lower degree, the internal delay in the network with lower degree becomes higher when the number of servers in sleep state is large. Consequently, we need to jointly control the number of servers in sleep state and the degree of servers.



(a) Effect of the number of servers in sleep state on the system power consumption.



(b) Effect of the number of servers in sleep state on the internal delay.

Fig. 4. Trade-off relationship between the system power consumption and internal delay.

III. Proposed Energy-Efficient and Delay-Aware Wireless Computing System

In this section, we describe the proposed operation scheme for our envisioned computing system. First, we introduce an optimization problem to minimize the system power consumption while satisfying acceptable delay. Then, based on the optimization problem, we propose a system operation procedure that jointly controls the sleep mode and the degree of servers in a dynamic scenario.

A. Assumed acceptable delay and optimization problem

In order to execute the real-time feedback control, the computing system collects various kinds of data (e.g., ambient information, image, video, log, and so forth) from sensors that are deployed in the whole factory, where we define $G = \{g_1, g_2, \ldots, g_{|G|}\}$, which denotes the set of data types. Additionally, since the computing system periodically collects data and the acceptable internal delay of each data type is previously defined, the computing system has an acceptable internal delay of data type g_i at time t, $D_{g_i}^{\mathrm{int}}(t)$. Additionally, a minimum acceptable internal delay at time t, $D_{\mathrm{int}}(t)$, is given by $D_{\mathrm{int}}(t) = \min_{g_i \in G} D_{g_i}^{\mathrm{int}}(t)$.

Our optimization problem aims to minimize the system power consumption while satisfying an acceptable internal delay at each time slot. As shown in Fig. 4, the ratio of servers in sleep state affects the system power consumption and the actual internal delay, and the degree of server also affects the actual internal delay. Therefore, we can find an optimal ratio of servers in sleep state, ρ^*, and an optimal degree for the servers, $\langle k^* \rangle$, at time t by solving the following integer programing.

$$\begin{aligned} \underset{\langle k \rangle, \rho}{\text{minimize}} \quad & E_{\mathrm{system}} \\ \text{subject to} \quad & D_\rho^{\mathrm{int}} < D_{\mathrm{int}}(t). \end{aligned} \quad (21)$$

B. System operation procedure in dynamic scenario

Here, we propose a system operation procedure for our computing system. In our proposed procedure, the adequate ratio of servers in sleep state and the degree of servers at each time slot t, $\rho(t)$ and $\langle k \rangle(t)$, are dynamically controlled in order to minimize the system power consumption while satisfying an acceptable internal delay. In this regard, $\rho(t)$ and $\langle k \rangle(t)$ are

Procedure 2 System operation procedure at time slot t

```
Calculate ϵ and t_ϵ by using (22) and (23)
Calculate ρ*(t_ϵ) and ⟨k*⟩(t_ϵ) by using (21)
Calculate ρ*(t + T_su) and ⟨k*⟩(t + T_su) by using (21)
if ρ(t − 1) < ρ*(t_ϵ) then
    ρ(t) ← ρ*(t_ϵ) and ⟨k⟩(t) ← ⟨k*⟩(t_ϵ)
    let {ρ*(t_ϵ) − ρ(t − 1)}|N| servers in active state enter sleep state
else if ρ(t − 1) > ρ*(t + T_su) then
    ρ(t) ← ρ*(t + T_su) and ⟨k⟩(t) ← ⟨k*⟩(t + T_su)
    let {ρ(t − 1) − ρ*(t + T_su)}|N| servers in sleep state enter active state
else
    ρ(t) ← ρ(t − 1) and ⟨k⟩(t) ← ⟨k⟩(t − 1)
end if
```

decided based on the acceptable internal delay at time $t+T_{\rm su}$, $D_{\rm int}(t+T_{\rm su})$, since each server in sleep state needs time to start-up $T_{\rm su}$ [20].

Procedure 2 describes our proposed system operation procedure at time t. Similar to Procedure 1, the centralized server executes this procedure.

First, the centralized server calculates the minimum acceptable delay for the interval from time t to $t+T_{\rm su}$, ϵ, which is given by

$$\epsilon = \min_{t \le a \le t+T_{\rm su}} D_{\rm int}(a). \tag{22}$$

Additionally, the centralized server also derives instant with the minimum acceptable delay for the interval from time t to $t+T_{\rm su}$, t_ϵ, which is given by

$$t_\epsilon = \mathop{\arg\min}_{t \le a \le t+T_{\rm su}} D_{\rm int}(a). \tag{23}$$

Furthermore, the centralized server finds the optimal ratio of servers in sleep state and the optimal degree for the servers at time t_ϵ and $t+T_{\rm su}$ by solving (21).

Then, it tries to compare the ratios of servers in sleep state at $(t-1)$, t_ϵ, and $t+T_{\rm su}$, $\rho^*(t-1)$, $\rho^*(t_\epsilon)$, and $\rho^*(t+T_{\rm su})$, in order to decide the next action. There exist three cases: (i) when $\rho(t-1) < \rho^*(t_\epsilon)$, $\{\rho^*(t_\epsilon) - \rho(t-1)\}|N|$ servers in active state can enter the sleep state while satisfying the acceptable delay. In this case, the current ratio and degree, $\rho(t)$ and $\langle k\rangle(t)$, are set to $\rho^*(t_\epsilon)$ and $\langle k^*\rangle(t_\epsilon)$, respectively. (ii) when $\rho(t-1) > \rho^*(t+T_{\rm su})$, the centralized server should let $\{\rho(t-1)-\rho^*(t+T_{\rm su})\}|N|$ servers in sleep state enter the active state in order to satisfy the acceptable delay at $(t+T_{\rm su})$. In this case, the current ratio and degree, $\rho(t)$ and $\langle k\rangle(t)$, are set to $\rho^*(t+T_{\rm su}))$ and $\langle k^*\rangle(t+T_{\rm su}))$, respectively. (iii) otherwise, the centralized server sets the values of $\rho(t)$ and $\langle k\rangle(t)$ as the previous values, $\rho(t-1)$ and $\langle k\rangle(t-1)$.

IV. Performance Evaluation

In this section, we evaluate the performance of our proposed energy-efficient and delay-aware operation scheme in our envisioned system by using extensive computer simulations. Furthermore, we confirm the effectiveness of our proposal in comparison with conventional schemes.

TABLE I
Evaluation settings.

Parameter	Value
Simulation time	100 s
Number of servers	100
Wireless bandwidth	59.5 to 62.5 GHz
Transmission power	10 dBm
Antenna gain	10 dB
Noise power level	-70 dBm
Length of time slot	0.02 s
Degree of server	5∼13
Power consumption of server in active state	32.3 W
Power consumption of server in sleep state	10.008 W
Start-up time of server	20 s
Receive rate	1000 packets/s
Packets size	1500 bytes
Number of servers in sleep state	0∼40

A. Parameter Settings

Table I describes the settings of our simulations. The simulations are executed for 100 seconds. In these simulations, a wireless computing system consisting of 100 servers is considered as a system configuration. The inside antenna uses a single channel from 59.5 to 62.5 GHz, which is divided into multiple time slots based on our multiple access scheme, where the length of each time slot is 0.02 second. As a channel model, the transmission power, antenna gain, and noise power level are set to 10 dBm, 10 dB, and -70 dBm. Additionally, the path loss is calculated by using (8), (10), and (11) in [11]. Due to the limitation of radiation angle [21], the degree of servers varies from 5 to 13. The power consumption of the servers with 60 GHz antennas in active state and sleep state are set to 32.3 W [10], [18] and 10.008 W [17], [19], respectively. Additionally, servers in sleep state need 20 seconds to start-up [20].

The computing system receives 1000 packets per second, with 1500 bytes per packets, from the sensors. In order to periodically vary the minimum acceptable internal delay of these packets, $D_{\rm int}(t)$ is given by

$$D_{\rm int}(t) = \frac{\sin(t/A)}{B} + C, \tag{24}$$

where A denotes the weight factor for length of cycle, B is the weigh factor for range of acceptable internal delay, and C denotes the average acceptable internal delay. Additionally, we consider two functions of $D_{\rm int}(t)$ in order to evaluate the performance with different ranges of acceptable internal delay. In scenario 1, A, B and C are set to 8, 8, and 0.75. In this case, the range of acceptable internal delay is small. On the other hand, in scenario 2, A, B and C are set to 8, 5, and 0.75. These settings can model a wider range of acceptable internal delay. Fig. 5 demonstrates the change of minimum acceptable delay in each scenario.

In order to verify the effectiveness of our proposal, we compare it with the conventional schemes. As one of the conventional schemes, we use a scheme that dynamically controls the sleep scheduling but uses a constant degree, referred to as dynamic sleep with constant degree (DSCD), where the degree is set to 7. The other scheme, referred to as constant sleep with constant degree (CSCD), consistently uses

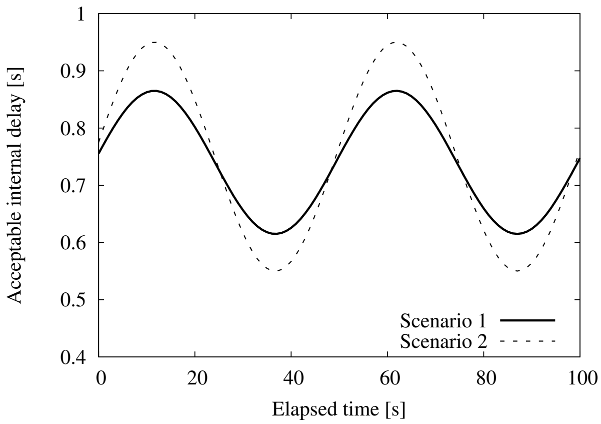
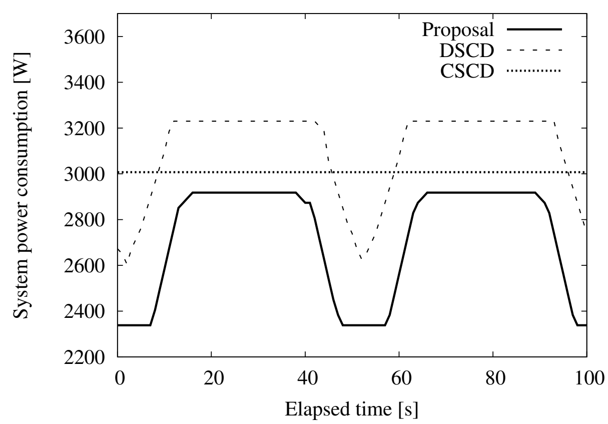
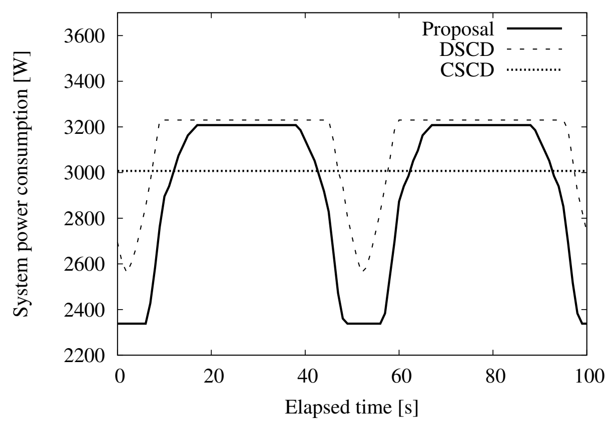


Fig. 5. Change in minimum value of acceptable internal delay.



(a) Change in system power consumption in scenario 1.



(b) Change in system power consumption in scenario 2.

Fig. 6. Change in system power consumption in different scenarios.

a constant number of servers in sleep state and the degree of servers, where these values are set to 10 and 7.

B. Evaluating the improvement in power consumption

Fig. 6 demonstrates the changes in the system power consumption for a 100 seconds period in different scenarios. Since our proposal and DSCD select an adequate number of servers in sleep state, the power consumption of our proposal and DSCD change depending on the value of minimum acceptable delay while the power consumption of CSCD is constant. Additionally, since our proposal decides an adequate degree of the servers, the power consumption of our proposal is lower than that of DSCD. On the other hand, in the case of scenario 2, our proposal consumes much more power compared with scenario 1. This is because the acceptable delay in scenario 2 is strict and our proposal tries to satisfy the acceptable delay by increasing the number of servers in active state.

Fig. 7 shows the amount of system power consumption for a 100 seconds period. Here, the amount of system power consumption is approximately calculated as the sum of the system power consumption at time t. As shown in Fig. 7, our proposal can achieve the lowest system power consumption in comparison with other approaches. Indeed, our proposal results in an approximate 13% drop in system power consumption when compared with DSCD, and an approximate 10% drop when compared with CSCD in scenario 1. Although the reduction ratio decreases in scenario 2, our proposal still achieves lower system power consumption. Consequently, we can confirm the effectiveness of our proposal in terms of system power consumption.

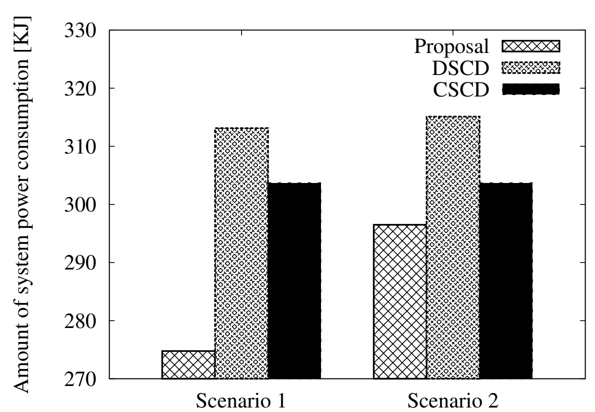


Fig. 7. Performance comparison in terms of the amount of system power consumption in the different scenarios.

C. Evaluating the improvement in delay satisfaction ratio

The rest of this section demonstrates the delay satisfaction ratio in the different scenarios. The delay satisfaction ratio is expressed as $P_{\mathrm{satisfaction}}/P_{\mathrm{total}}$, where $P_{\mathrm{satisfaction}}$ is the number of packets that are received within the acceptable delay and P_{total} denotes the total number of packets. Since the packet rate is set to 1000 packets/s, the value of P_{total} is 1000000 packets. Fig. 8 shows the delay satisfaction ratio in the different scenarios. From this result, it is clear that our proposal can achieve the maximum ratio (i.e., 100%) in our simulation environment. On the other hand, DSCD achieves around 40% of satisfaction ratio in scenario 1 and the ratio further decreases in scenario 2. From this phenomenon, we can notice that the constant degree cannot satisfy the acceptable delay and joint control of the sleep scheduling and the degree

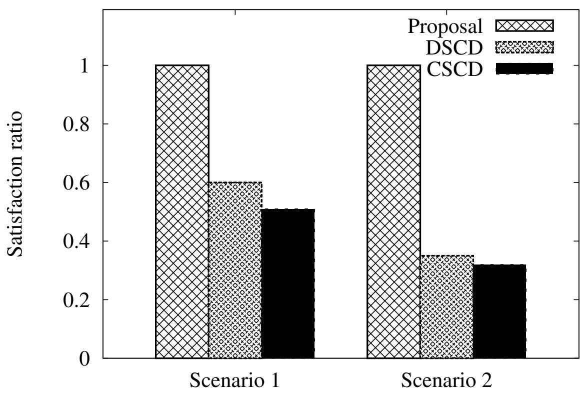


Fig. 8. Performance comparison in terms of the delay satisfaction ratio in the different scenarios.

of servers has great impact on the satisfaction ratio. Therefore, we can conclude that our proposal is effective in reducing the system power consumption while satisfying the acceptable delay.

V. Conclusion

Towards the realization of smart factories based on wireless sensor networks, in this paper we investigated a novel computing system and its operation scheme. Since conventional computing systems lack the capabilities required for providing industrial IoT applications (i.e., high processing and communication capability, high durability, and low power consumption), we first introduced a computing system based on wireless technology. Additionally, we derived a mathematical model to evaluate the performance of our envisioned system based on complex network theory. This model showed existence of a trade-off relationship between power consumption and delay required for data collection. Consequently, we proposed an energy-efficient and delay-aware system operation scheme. Our proposed scheme jointly controls the sleep scheduling and network connectivity to reduce the system power consumption while satisfying an acceptable delay, which is decided based on the requirement of industrial IoT applications. Simulation results showed that by appropriately selecting the number of servers in sleep state and the degree of servers, the proposal can improve both the system power consumption and satisfaction ratio of delay compared to other approaches.

Acknowledgment

This research was supported by Strategic International Research Cooperative Program, Japan Science and Technology Agency (JST).

References

[1] S. Savazzi, V. Rampa, and U. Spagnolini, "Wireless cloud networks for the factory of things: connectivity modeling and layout design," *IEEE Internet of Things Journal*, vol. 1, no. 2, pp. 180-195, Mar. 2014.

[2] E. Shellshear, R. Berlin, and J. S. Carlson, "Maximizing smart factory systems by incrementally updating point clouds," *IEEE Computer Graphics and Applications*, vol. 35, no. 2, pp. 62-69, Mar.-Apr. 2015.

[3] M. R. Palattella, N. Accettura, L. A. Grieco, G. Boggia, M. Dohler, and T. Engel, "On optimal scheduling in duty-cycled industrial IoT applications using IEEE802.15.4e TSCH," *IEEE Sensors Journal*, vol. 13, no. 2, pp. 3655-3666, Jun. 2013.

[4] Accenture, "Driving Unconventional Growth through the Industrial Internet of Things," 2014 [Online], Available: http://www.accenture.com/SiteCollectionDocuments/PDF/Accenture-Driving-Unconventional-Growth-through-IIoT.pdf

[5] F. Salvadori, M. Campos, P. S. Sausen, R. F. Camargo, C. Gehrke, C. Rech, M. A. Spohn, and A. C. Oliveira, "Monitoring in industrial systems using wireless sensor network with dynamic power management," *IEEE Transactions on Instrumentation and Measurement*, vol. 58, no. 9, pp. 3104-3111, Jul. 2009.

[6] A. A. Kumar S., K. Ovsthus, and L. M. Kristensen, "An industrial perspective on wireless sensor networks – A survey of requirements, protocols, and challenges," *IEEE Communications Surveys & Tutorials*, vol. 16, no. 3, pp. 1391-1412, Jan. 2014.

[7] F. Bonomi, R. Milito, J. Zhu, and S. Addepalli, "Fog computing and its role in the Internet of Things," *Proc. of the first edition of the MCC workshop on Mobile cloud computing*, pp. 13-16, New York, USA, 2012.

[8] K. Ahmed, and M. Gregory, "Integrating wireless sensor networks with cloud computing," *Proc. of MSN 2011*, pp. 364-366, Beijing, China, 2011.

[9] J-Y. Shin, E. G. Sirer, H. Weatherspoon, and D. Kirovski, "On the feasibility of completely wireless datacenters," *Proc of the eighth ACM/IEEE symposium on Architectures for networking and communications systems*, pp. 3-14, New York, USA, Oct. 2012.

[10] J-Y. Shin, E. G. Sirer, H. Weatherspoon, and D. Kirovski, "On the feasibility of completely wirelesss datacenters," *IEEE/ACM Trans. on Networking*, vol. 21, no. 5, pp. 1666-1679, Oct. 2013.

[11] K. Suto, H. Nishiyama, N. Kato, T. Nakachi, T. Sakano, and A. Takahara, "A failure-tolerant and spectrum-efficient wireless data center network design for improving performance of big date mining," *Proc. of VTC 2015-Spring*, Glasgow, Scotland, to appear.

[12] A. M. Niknejad, "Siliconization of 60 GHz," *IEEE Microwave Magazine*, vol. 11, no. 1, pp. 78-85, Feb. 2010.

[13] S. Pinel, P. Sen, S. Sarkar, B. Perumana, D. Dawn, D. Yeh, F. Barale, M. Leung, E. Juntunen, P. Vadivelu, K. Chuang, P. Melet, G. Iyer, and J. Laskar, "60 GHz single-chip CMOS digital radios and phased array solutions for gaming and connectivity," *IEEE J. on Selected Areas in Communications*, vol. 27, no. 8, pp. 1347-1357, Oct. 2009.

[14] Z. Guo and Y. Yang, "On nonblocking multicast fat-tree data center networks with server redundancy," *IEEE Trans. on Computers*, vol. 64, no. 4, pp. 1058-1073, Apr. 2014.

[15] Cisco, "Cisco Catalyst 3750 Series Switches," 2009 [Online], Available: http://www.cisco.com/c/en/us/products/collateral/switches/catalyst-3750-series-switches/product_data_sheet0900aecd80371991.pdf

[16] K. Suto, H. Nishiyama, N. Kato, K. Mizutani, O. Akashi, and A. Takahara, "An overlay-based data mining architecture tolerant to physical network disruptions," *" IEEE Transactions on Emerging Topics in Computing (TETC)*, vol. 2, no. 3, pp. 292-301, Sep. 2014.

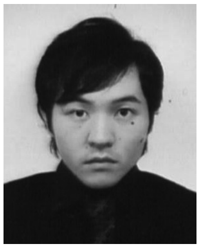
[17] V. K. M. Raj and R. Shriram, "A study on server sleep state transition to reduce power consumption in a virtualized server cluster environment," *Proc. of COMSENETS 2012*, 6 pages, Bangalore, India, Jan. 2012.

[18] Y. Nakagawa, T. Shimizu, Y. Koyanagi, O. Shiraki, S. Kobayashi, K. Hyoudou, T. Miyoshi, Y. Ogata, Y. Umezawa, T. Horie, and A. Hattori, "A single-chip, 10-gigabit ethernet switch LSI for energy-efficient blade servers," *Proc. of GreenCom 2010*, pp. 401-411, Hangzhou, China, Dec. 2010.

[19] R. Matsuo, T. Tandai, T. Tomizawa, T. Adachi, and H. Kasami, "Energy-efficient standby mode algorithm in short-range one-to-one millimeter wave communications," *Proc. of IEEE ICC 2012*, pp. 4033-4038, Ottawa, Canada, Jun. 2012.

[20] A. Gandhi, M. Harchol-Balter, and M. Kozuch, "Are sleep states effective in data centers?," *Proc. IGCC 2012*, 10 pages, San Jose, USA, Jun. 2012.

[21] M. Ohira, A. Miura, and M. Ueba, "60-GHz wideband substrate-integrated-waveguide slot array using closely spaced elements for planar multisector antenna," *IEEE Trans. on Antennas and Propagation*, vol. 58, no. 3, pp. 993-998, Mar. 2010.



Katsuya Suto received M.S. degree in Information Science from the Graduate School of Information Sciences (GSIS), Tohoku University, Japan, in 2013. Currently, he is pursuing the Ph.D. degree in the GSIS at Tohoku University. His research interests are in the areas of big data mining architecture, resilient network design, and wireless networking. He was a recipient of the prestigious Deans Award from Tohoku University in March 2013. He received the Best Paper Award at the IEEE 79th Vehicular Technology Conference (VTC'2013-spring) and the IEICE Communications Society Academic Encouragement Award. He is a student member of IEEE and IEICE.

Hiroki Nishiyama is an Associate Professor at the Graduate School of Information Sciences (GSIS), Tohoku University, Japan. He received his M.S. and PhD in Information Science from Tohoku University, Japan, in 2007 and 2008, respectively. He has published more than 100 peer-reviewed papers including many high quality publications in prestigious IEEE journals and conferences. He was awarded Best Paper Awards from many international conferences including IEEE's flagship events, such as the IEEE Global Communications Conference in 2013 (GLOBECOM'13), GLOBECOM'10, and the IEEE Wireless Communications and Networking Conference in 2012 (WCNC'12). He was also a recipient of the IEEE Communications Society Asia-Pacific Board Outstanding Young Researcher Award 2013, the IEICE Communications Society Academic Encouragement Award 2011, and the 2009 FUNAI Foundation's Research Incentive Award for Information Technology. He has served as a Co-chair for Cognitive Radio and Networks Symposium of IEEE International Conference on Communications 2015 (ICC'15), a Co-chair for Selected Areas in Communications Symposium of IEEE ICC'14, an Associate Editor for IEEE Transactions on Vehicular Technology, an Associate Editor for Springer Journal of Peer-to-Peer Networking and Applications, and the Secretary of IEEE ComSoc Sendai Chapter. His research interests cover a wide range of areas including satellite communications, unmanned aircraft system (UAS) networks, wireless and mobile networks, ad hoc and sensor networks, green networking, and network security. One of his outstanding achievements is Relay-by-Smartphone, which makes it possible to share information among many people by using only WiFi functionality of smartphones. He is a Senior Member of the IEEE, as well as a member of Institute of Electronics, Information and Communication Engineers (IEICE).

Nei Kato received his Bachelor Degree from Polytechnic University, Japan, in 1986, M.S. and Ph.D. Degrees in information engineering from Tohoku University, in 1988 and 1991 respectively. He was promoted to full professor position with Graduate School of Information Sciences, Tohoku University in 2003. He currently serves as a Member-at-Large on the Board of Governors, IEEE Communications Society, the Chair of IEEE Ad Hoc & Sensor Networks Technical Committee, the Chair of IEEE ComSoc Sendai Chapter, the Associate Editor-in-Chief of IEEE Internet of Things Journal, an Area Editor of IEEE Transactions on Vehicular Technology, an editor of IEEE Wireless Communications Magazine and IEEE Network Magazine. He has served as the Chair of IEEE ComSoc Satellite and Space Communications Technical Committee (2010-2012). His awards include Minoru Ishida Foundation Research Encouragement Prize (2003), Distinguished Contributions to Satellite Communications Award from the IEEE ComSoc, Satellite and Space Communications Technical Committee (2005), the FUNAI information Science Award (2007), the TELCOM System Technology Award from Foundation for Electrical Communications Diffusion (2008), the IEICE Network System Research Award (2009), the IEICE Satellite Communications Research Award (2011), the KDDI Foundation Excellent Research Award (2012), IEICE Communications Society Distinguished Service Award (2012), five Best Paper Awards from IEEE GLOBECOM/WCNC/VTC, and IEICE Communications Society Best Paper Award (2012). Besides his academic activities, he also serves on the expert committee of Telecommunications Council, Ministry of Internal Affairs and Communications, and as the chairperson of ITU-R SG4 and SG7, Japan. Nei Kato is a fellow of IEEE and IEICE.

Chih-Wei Huang received the B.S. degree from National Taiwan University, Taipei, in 2001, the M.S. degree from Columbia University, New York, in 2004, and the Ph.D. degree from University of Washington, Seattle, in 2009, all in electrical engineering. He joined the Department of Communication Engineering, National Central University, Taoyuan, Taiwan, in 2010. He is currently an assistant professor heading the Information Processing and Communications (IPC) Laboratory. From 2006 to 2009, he was intern researcher at Siemens Corporate Research and Microsoft Research. He is the author of papers in a broad range of areas, including wireless networking, multimedia communications, digital signal processing, and information retrieval.