Guaranteeing Real-Time Services for Industrial Wireless Sensor Networks With IEEE 802.15.4

Seong-eun Yoo, Poh Kit Chong, *Student Member, IEEE*, Daeyoung Kim, *Member, IEEE*, Yoonmee Doh, Minh-Long Pham, *Student Member, IEEE*, Eunchang Choi, and Jaedoo Huh

***Abstract*—Industrial applications of wireless sensor networks require timeliness in exchanging messages among nodes. Although IEEE 802.15.4 provides a superframe structure for real-time communication, a real-time message-scheduling algorithm is still required to schedule a large number of real-time messages to meet their timing constraints. We propose a distance-constrained real-time offline message-scheduling algorithm which generates the standard specific parameters such as beacon order, superframe order, and guaranteed-time-slot information and allocates each periodic real-time message to superframe slots for a given message set. The proposed scheduling algorithm is evaluated and analyzed extensively through simulations. In addition, a guaranteed time service is implemented in a typical industrial sensor node platform with a well-known IEEE 802.15.4-compliant transceiver CC2420 and ATmega128L to verify the feasibility of the guaranteed time service with the schedule generated by the proposed scheduling algorithm. Through experiments, we prove that the real system runs accurately according to the schedule calculated by the proposed algorithm.**

***Index Terms*—Guaranteed time slot (GTS), industrial wireless sensor network (WSN), real-time communication, WSN.**

I. INTRODUCTION

WIRELESS sensor networks (WSN) have been deployed in many application areas with the help of advances of relevant technology such as low-power wireless communication, low-power microcontrollers, sensors, and embedded software. The scope of the application area is diverse, including agriculture [1], disaster management [2], cultural property management [3], unmanned surveillance, intelligent transport systems [4], [5], and industrial control [6]–[10]. Among them, particularly in industrial applications such as factory automation as well as process automation, WSN systems are beginning to substitute the conventional industrial wired communication systems due to their advantages such as ease of installation and movement and low maintenance cost. Since most industrial applications require low data rate, low power, and timeliness [11], [12], IEEE 802.15.4 is one of the good candidates for industrial WSNs [13]–[17].

The IEEE 802.15.4 standard for low-rate wireless personal area networks (PANs) was published to define the protocol for low-data-rate (250 kb/s at 2.4 GHz), low-power, and low-complexity short-range wireless communication devices [18]. It provides beacon-enabled communication for real-time message exchange, allowing up to seven guaranteed time slot (GTS) allocations in the contention free period (CFP) of a superframe. Since the standard allows including the GTS allocation information only in four consecutive beacon frames, the maximum real-time message in a network is limited to the maximum number of GTS allocations. Considering that there are several tens or hundreds of periodic real-time messages in industrial WSNs, we suggest a modification of the standard to overcome this limitation, i.e., allowing the inclusion of GTS allocation information in every beacon frame to allow every superframe to have different GTS allocation information. With this modification, we need a systematic real-time scheduling algorithm to find the standard specific parameters to define the superframe structure and to allocate the real-time messages to GTS. Therefore, we proposed a distance-constrained offline real-time message-scheduling algorithm to schedule real-time messages in GTS to meet their deadlines [19]. Moreover, it generates the standard specific parameters such as beacon order (BO, defining the beacon interval), superframe order (SO, defining the superframe duration), and GTS information for the superframe structure to meet the real-time constraints of tens or hundreds of periodic real-time messages. We evaluated the algorithm partly by simulation study and demonstrated the guaranteed time service on a prototype evaluation board CC2420DB [20] with an ATmega128L microcontroller and an IEEE 802.15.4-compliant transceiver CC2420 [21] in our previous work [19].

We are extending our previous work [19] in the following aspects. We clarify the main procedure of the scheduling algorithm in more detail. In addition, the scheduling algorithm is evaluated and analyzed extensively with further simulation study. To show the feasibility of the proposed algorithm, we implement a guaranteed time service in a real sensor node platform, namely, T-Sink/Sensor node [4], which was developed to measure the velocity of moving vehicles.

Manuscript received May 29, 2009; revised August 10, 2009 and October 20, 2009; accepted December 11, 2009. Date of publication February 8, 2010; date of current version October 13, 2010. This work was supported by the Ministry of Education, Science and Technology, Korea, under the National Research Laboratory support program supervised by the National Research Foundation of Korea (NRF) under Grant NRF-2010-0018859.

S. Yoo is with the School of Computer and Communication Engineering, Daegu University, Gyeongsan 712-714, Korea (e-mail: seyoo@kaist.ac.kr).

P. K. Chong and D. Kim are with the College of Information Science and Technology, Korea Advanced Institute of Science and Technology, Daejeon 305-701, Korea (e-mail: chongpohkit@kaist.ac.kr; kimd@kaist.ac.kr).

Y. Doh, E. Choi, and J. Huh are with the IT Convergence Technology Research Laboratory, Electronics and Telecommunications Research Institute, Daejeon 305-350, Korea (e-mail: ydoh@etri.re.kr; ecchoi@etri.re.kr; jdhuh@etri.re.kr).

M.-L. Pham is with the Department of Electronics and Telecommunications, Norwegian University of Science and Technology, 7491 Trondheim, Norway (e-mail: longpm@iet.ntnu.no).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIE.2010.2040630

The rest of this paper is organized as follows. Section II presents related works, and Section III summarizes part of the standard. Section IV presents the system model and the problem statement, and Section V describes the proposed scheduling algorithm. Section VI evaluates the algorithm by simulation study, and the implementation results are stated in Section VII. Finally, this paper concludes with a summary of our work and a statement of future work in Section VIII.

II. RELATED WORKS

A. Industrial WSNs

Wireless Interface for Sensors and Actuators [22], [23] is an industrial communication system which provides both wireless communication and wireless power supply which is basically considered in radio-frequency identification but has rarely been considered in WSNs. Based on the physical layer of IEEE 802.15.1, it uses time-division multiple access (TDMA) with frequency hopping for a reliable low-latency transmission for up to 120 nodes in a base station. WirelessHART [6], [24] is the seventh release of the conventional Highway Addressable Remote Transducer (HART) protocol which is a bidirectional industrial field communication protocol combining the analog 4–20-mA control loop with a superimposed digital signal for configuration and diagnostics. Based on IEEE 802.15.4 radio, WirelessHART provides TDMA and frequency hopping with blacklisting, which makes it possible to skip a persistent interfering frequency channel from the frequency hopping sequence. In addition, with full mesh routing, it achieves path redundancy for reliable data transmission. Another standard based on IEEE 802.15.4 radio is ISA 100.11a, which is the first standard of ISA 100 [10], [24]. ISA 100 will support multiple protocols, including its own native protocol and others such as HART, Profibus, and so on, whereas WirelessHART supports only HART commands.

To enhance and add functionality to IEEE 802.15.4-2006 to better support the industrial WSNs, the IEEE 802.15 Task Group 4e [25] is charted to define a Media Access Control (MAC) amendment to the existing IEEE 802.15.4-2006. The proposals can be summarized as the following: 1) network-wide time synchronization; 2) beacon scheduling; 3) enhancing the existing superframe and GTS scheme; and 4) time-slotted channel hopping to support real-time communication under radio interferences. According to the proposals, the IEEE 802.15.4e standard is supposed to support industrial applications better in terms of real-time and reliable communication. However, it still requires a real-time message-scheduling algorithm to schedule periodic real-time messages.

B. GTS Mechanism Improvements in IEEE 802.15.4

Cheng *et al.* [26] proposed a new GTS scheme which utilizes the CFP more efficiently than the standard scheme by dividing the CFP into 16 equally sized subslots. With smaller slots, CFP utilization in the new scheme is better than that in the standard scheme. Song *et al.* [27] proposed a dynamic GTS (D-GTS) allocation algorithm to reduce wasted bandwidth. To reduce the wasted part of GTS allocation, they proposed to allocate a D-GTS in the backoff period unit rather than a superframe slot unit. Huang *et al.* [28] developed an adaptive GTS allocation (AGA) scheme which considers low latency and fairness. The algorithm consists of a priority assignment phase based on recent GTS usage feedbacks and a GTS scheduling phase where a GTS is given to a node with a higher priority. With a series of experiments, the proposed AGA scheme was evaluated and showed better wait time and fairness performance than the standard scheme. Kouba *et al.* [29] proposed an implicit GTS allocation mechanism to overcome the limitations (quick consumption of GTSs and underutilization of GTS bandwidth resources) of the explicit GTS allocation of the standard. The implicit GTS allocation mechanism shares a GTS with different nodes unlike the standard approach with which each node possesses its own GTS slot. For scheduling, they use a traditional round-robin approach, and each node shares the given bandwidth fairly. With an online approach, the authors focus on improving the bandwidth utilization of GTS allocation rather than the message scheduling itself. In [30], the authors proposed an online optimal GTS scheduling algorithm (GSA) which distributes the GTSs of a transaction over as many beacon intervals as possible, meeting the per-transaction delay constraint for the transaction. However, GSA assumes that a superframe size and a duty cycle are given.

Most previous works try to reduce the wastage of GTSs and enhance the GTS utilization or to schedule the GTS itself without providing a way to configure a superframe, but we concentrate on a real-time message-scheduling algorithm which is scheduling periodic real-time messages in CFP and, at the same time, generating standard specific parameters to define a superframe.

III. SUPERFRAME STRUCTURE AND IFS IN IEEE 802.15.4

Before formulating the problem, we overview the necessary parts of the standard required for our work. A more detailed description on the standard is found in [18].

A. Superframe Structure

The optional superframe structure, which is used for the beacon-enabled mode, provides a guaranteed-time message exchange.

The superframe shown in Fig. 1 is bounded by network beacons sent from the coordinator, and the active portion of the superframe is divided into 16 equally sized slots. The beacon frame is transmitted in the first slot of each superframe, and it defines the superframe structure with BO, SO, and GTS descriptors. BO specifies the beacon interval, and SO defines the duration of the active portion of the superframe. For low-latency applications or applications requiring specific data bandwidths, the PAN coordinator may dedicate portions of the active superframe to them. These portions are called GTSs. The GTSs form the CFP, which always appears at the end of the active superframe starting at a slot boundary immediately following the contention access period (CAP), as shown in Fig. 1. The PAN coordinator may allocate up to seven GTSs, and each one may occupy more than one slot period.

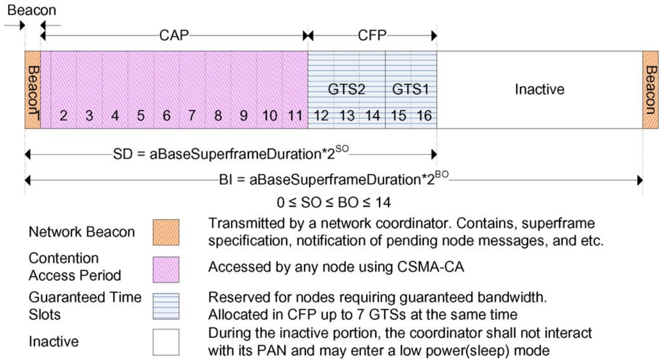

Fig. 1. IEEE 802.15.4 superframe structure.

However, a sufficient and constant portion of the CAP must remain for contention-based access of other networked devices or new devices intending to join the network. All contention-based transactions are completed before the CFP begins. Each device transmitting in a GTS also ensures that its transaction is completed before the time of the next GTS or the end of the CFP.

B. IFS

The MAC sublayer needs a finite amount of time to process data received by the PHY layer. To this end, transmitted frames must be followed by an Inter Frame Spacing (IFS) period. If the transmission requires an acknowledgment, the IFS follows the acknowledgment frame. The length of the IFS period is dependent on the size of the frame that has just been transmitted. Frames [i.e., MAC Protocol data units (MPDUs)] of up to *aMaxSIFSFrameSize* in length are followed by a short IFS (SIFS) period of at least *aMinSIFSPeriod* symbols. MPDUs with lengths greater than *aMaxSIFSFrameSize* are followed by a long IFS (LIFS) of at least *aMinLIFSPeriod* symbols (Fig. 2).

IV. SYSTEM MODEL AND PROBLEM STATEMENT

In this section, we present our system model and the problem statement for the proposed algorithm.

We assume that the network operates in beacon-enabled mode and in a star topology. Therefore, the PAN coordinator participates in all of the packet transmissions. Since most sensed data are transmitted periodically, we consider only periodic real-time messages except for some intermittent MAC command messages. Each periodic real-time message has its deadline assumed to be equal to its period, and it should be scheduled within its period. The message set $\mathbf{M}$ with N periodic real-time messages and its additional information $\mathbf{I}$ are known *a priori*. The notations to be used in this paper are defined in Table I, while throughout this paper, italic fonts represent standard specific constants (starting with "*a*") or variables [18].

The problem statement can be summarized as follows:

> "*finding proper standard specific parameters and beacon information to support periodic real-time messages in the IEEE 802.15.4 standard.*"

Thus, the inputs are the message set $\mathbf{M} = \{\mathrm{M}_1, \mathrm{M}_2, \ldots, \mathrm{M}_N\}$, which includes the period and length of each message, and the additional information set $\mathbf{I}$ for the message set $\mathbf{M}$. The utilization (or density) of the ith message M_i in $\mathbf{M}$ is defined by LS_i/PS_i, and the utilization of $\mathbf{M}$, i.e., $\rho(\mathbf{M})$, is calculated from $\rho(\mathbf{M}) = \sum_{i=1}^{n}(LS_i/PS_i)$. In addition, the maximum number of short and extended pending addresses and the maximum beacon payload length are inputs for calculating slots allocated to the beacon frame. The output of the proposed scheduling algorithm is the beacon table (BT) shown in Table II which includes a BO, a SO, a GTS descriptor for each GTS, and the final CAP slot.

V. SCHEDULING ALGORITHM

We describe the main procedure (Fig. 3) of the scheduling algorithm in detail. The proposed scheduling algorithm determines not only the schedule but also a superframe structure such as (BO, SO) and a GTS descriptor to be used to schedule the given message set $\mathbf{M}$. BO, which defines the beacon interval, is determined by the minimum period of the given messages' periods. SO, which specifies the superframe duration, is determined so that the duty cycle can be minimized to save energy.

In the first step, the *AddOverheads* procedure calculates overheads (i.e., by MAC/PHY layer, ACK, and IFS) that are added via each layer according to the addressing modes and follow the message packets in the air (Fig. 2). It then updates each length of the given message set in symbols considering the overheads. Through the *floor* function, the period of each message is expressed in symbol units. The minimum length of CAP (*aMinCAPLength*) is required in order to transmit the MAC command frames. The number of slots allocated to the beacon frame and the minimum CAP are calculated in the *CalAllocatedSlot* procedure, assuming that the number of GTS descriptors in the beacon frame is equal to the maximum number of GTSs allowed, i.e., seven. SO is initialized to the smallest value considering the smallest duty cycle.

During the *preschedulability* check (step 2 in Fig. 3), the minimum period of the given message set is compared with the smallest beacon interval (i.e., beacon interval when BO is zero) that the standard can support. This step also sets the upper bound for BO from the minimum period of the given message set. From the following steps (3–6), the scheduling algorithm tries to schedule the given message set starting from the upper bound of BO because a bigger BO results in a smaller duty cycle; therefore, the energy consumption is reduced

$$BI \times 2^E \le PS_i < BI \times 2^{E+1}$$
$$\log_2(BI \times 2^E) \le \log_2 PS_i < \log_2(BI \times 2^{E+1})$$
$$E \le \log_2\left(\frac{PS_i}{BI}\right) < E + 1,$$
$$\therefore E = \left\lfloor \log_2\left(\frac{PS_i}{BI}\right) \right\rfloor. \quad (1)$$

The third step harmonizes (which has the same meaning as the specialization in the Sx scheduler [31], [32]) the periods

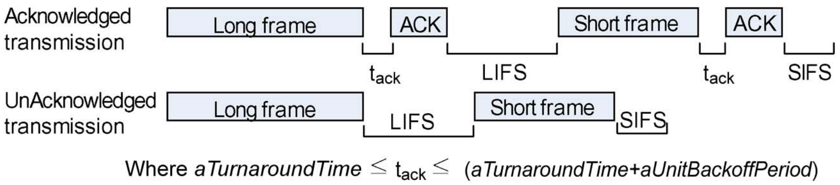

Fig. 2. IFS.

TABLE I
NOTATIONS

Notation	Description
MPDU	MAC Protocol Data Unit
T_S	Symbol duration, 16us@2.4GHz, 200us@915MHz, 400us@868MHz
SS	Slot size(in symbols), SS = SD/16 (Fig. 1)
IFS	InterFrame Spacing (Fig. 2)
ACK	Acknowledgement frame of IEEE 802.15.4
OVH	Overhead set, $\{O_1, O_2, O_3, \ldots, O_N\}$. O_i is the total overhead (in symbols) for M_i added by MAC/PHY layer, ACK, and IFS.
M	Message set, $\{M_1, M_2, M_3, \ldots, M_N\} = \{(P_1, L_1), (P_2, L_2), \ldots, (P_N, L_N)\}$ M_i is the i-th message in **M**
MF	Message set, $\{(PF_1, LF_1), (PF_2, LF_2), \ldots, (PF_N, LF_N)\}$
MS	Message set, $\{(PSS_1, LF_1), (PSS_2, LF_2), \ldots, (PSS_N, LF_N)\}$
N	Number of messages
L_i	length(in bytes) of i-th message, M_i
L	set of lengths(in bytes) of all messages, $\{L_1, \ldots, L_N\}$
LS	set of lengths(in symbols) of all messages, $\{LS_1, \ldots, LS_N\}$
LF_i	length(in slots) of i-th message, M_i
P_i	period(in usec.) of i-th message, M_i
PSS_i	period(in slots) of i-th message, M_i
PS	set of periods(in symbols) of all messages, $\{PS_1, \ldots, PS_N\}$
PH	set of harmonized periods(in symbols) of all messages
PF	set of harmonized periods(in slots) of all messages
PH_i	harmonized periods(in symbols) of i-th message
PF_i	harmonized periods(in slots) of i-th message
A_i	The short address of the device related to i-th message
R_i	The direction of i-th message
$ACKR_i$	The flag for the acknowledgement request
ACKR	The set of the acknowledgement request flags. **ACKR**=$\{ACKR_1, ACKR_2, ACKR_3, \ldots, ACKR_N\}$
I_i	$I_i = (A_i, R_i, ACKR_i)$ is the additional information for M_i.
I	Additional information set to message set **M**, $\{I_1, I_2, \ldots, I_N\}$
S_{SO}	the result of schedulability test for the superframe order, SO
Slot4BNC	Allocated slots for beacon frame and CAP according to SO **Slot4BNC** = { Slot4BNC$_0$, Slot4BNC$_1$, …, Slot4BNC$_{14}$ }

of the given messages. Unlike in the Sx scheduler, this step harmonizes the message set with respect to the BI. The harmonized period of PS_i (the period of ith message in symbols) can be obtained by $PH_i = BI \times 2^E$, where $BI \times 2^E \le PS_i < BI \times 2^{(E+1)}$ and E is an integer in the range of [0, 14]. The value of E in (1) can be derived using the *floor* function as in [31]. Therefore, PH_i is $BI \times 2^{\lfloor \log(PS_i/BI) \rfloor}$.

After transforming the given message set into superframe slots, *schedulability* is checked. The *Schedulable* procedure (Fig. 4) checks two constraints. The procedure checks the in-the-air *utilization constraint* (2), where U_{IP} is the utilization of the inactive portion of the superframe, U_{BCAP} is the utilization of the beacon frame and a minimum CAP, and $\rho(\mathbf{MF})$ is the utilization (or density) of the given message set and is calculated from $\sum_{i=1}^{n}(LF_i/PF_i)$

$$U = U_{\mathrm{IP}} + U_{\mathrm{BCAP}} + \rho(\mathbf{MF}) \le 1. \qquad (2)$$

TABLE II
BT NOTATION

Member			Description
BO			Beacon Order to be used in the minor frame
SO			Superframe Order for the minor frame
NmM			Number of minor frames in the major frame
mf [1:NmM]	Availableslot		Available # of slots in each minor frame
	FinalCAP		Final CAP slot
	numGTS		Number of allocated GTSs
	GTS [1:7]	id	Message id which uses
		startslot	Start slot number for each GTS
		Byte	Size of MAC payload for each GTS
		length	Number of allocated slots for each GTS

As long as the utilization constraint is met, the *maximum GTS number constraint* is checked. Since the standard allows up to seven GTSs in a superframe, the *Schedulable* procedure tries to allocate the given messages to each minor frame (superframe) with the *AllocateSlots* procedure in order to meet the *maximum GTS number constraint*. This procedure, as the most complex step of the main procedure (Fig. 3), is executed in $O(N \cdot J)$, where N is the number of messages in the given message set and J is the number of minor frames in the major frame. The BT (Table II) is configured during this procedure. Only when all these constraints are met does the *Schedulable* procedure set the schedulability result S_{SO} to FEASIBLE. Otherwise, it sets S_{SO} to SHORTGTSORSLOT when *the maximum GTS number constraint* is not satisfied or EXCEEDUBOUND when the *utilization constraint* is violated.

Finally, the main procedure is checking the result of the *Schedulable* procedure. If the result is FEASIBLE, the algorithm finishes with a return value of SUCCESS. Otherwise, the algorithm returns to step 3 or step 4 depending on the result. It is noticeable that whenever BO is decreased, SO is initialized to be zero so that the schedulability is checked beginning from the smallest duty cycle to save energy.

VI. ANALYSIS

We evaluate the performance of the proposed scheduling algorithm and analyze the result of our simulation study on the scheduling algorithm in this section.

```
GTSchedule(M, I, BT)
1.  //Initialize
    1-1. AddOverheads (L, ACKR, LS, OVH);
    1-2. $PS_i \leftarrow$ floor ($P_i/T_S$), for all i;
    1-3.CalAllocatedSlot (PA, BPAYLOAD, Slot4BNC);
    1-4. Initizlize SO to 0;   //to begin with minimum duty-cycle.
2.  //Check pre-schedulability and find the BO based on the minimum PS
    if ($min(PS_i) < aBaseSuperframeDuration \times 2^0$)
        //Unschedulable with the 802.15.4
        return FAIL;
    else if ($min(PS_i) < aBaseSuperframeDuration \times 2^{15}$)
        Set the BO to the integer value E satisfying the following:
        $min(PS_i)/2 < aBaseSuperframeDuration \times 2^E \le \min(PS_i)$, $E \in \{0,1,.., 14\}$
    else
        Set BO to the maximum value, 14; //upper bound
    end if
3.  //Harmonize the periods of messages.
    Harmonize (PS, BO, PH);
4.  //Change the unit of period to slots
    $SD \leftarrow aBaseSuperframeDuration \times 2^{SO}$; $SS \leftarrow SD/aNumSuperframeSlots$;
    for each i th message do
        $PF_i \leftarrow$ floor ($PH_i/SS$); $LF_i \leftarrow$ ceiling ($LS_i/SS$);
    end for
5.  //Check schedulability  and try to allocate the slots to message set
    Schedulable (BO, SO, MF, PH, L, ACKR, Slot4BNC, OVH, $S_{SO}$, BT);
6.  //If the result of the schedulability check is FEASIBLE, return SUCCESS
    if ($S_{SO}$ == FEASIBLE)
        Set BT.SO and BT.BO to SO and BO; return SUCCESS;
    else
        //Determine a SO for which the schedulability is checked.
        if (SO == BO) // SO is initizlized to 0 at STEP1. Always BO $\ge$ SO.
            if (BO == 0)      //Done for all possible (BO, SO) combinations.
                return  FAIL; // but nothing makes the message set schedulable
            else//try to check schedulability for another BO
                decrease BO by 1 and initizlize SO to 0;
                go to the harmonization step (STEP 3);
            end if            //If ( BO == 0)
        else                  //SO < BO
            Increase SO by 1;
            go to the slot-based step (STEP 4);
        end if
    end if

Input:
    M: { ($P_1$, $L_1$), ($P_2$, $L_2$), …, ($P_N$ , $L_N$)}, original message set
    I: Additional information set to message set M
Output:
    BT: beacon table including BO, SO, GTS descriptor list
```

Fig. 3. Main procedure of the proposed scheduling algorithm.

```
Schedulable (BO, SO, MF, PH, L, ACKR, Slot4BNC, OVH, $S_{SO}$, BT)
1. //Total utilization $U_T = U_{BCAP} + \rho(MF) + U_{IP}$
   1-1. //Utilization of beacon frame and CAP
        $BI \leftarrow aBaseSuperframeDuration \times 2^{BO}$;
        $SS \leftarrow aBaseSuperframeDuration \times 2^{SO}/aNumSuperframeSlots$;
        $U_{BCAP} \leftarrow Slot4BNC_{SO}/(BI/SS)$;
   1-2. //Density or utilization of MF,
        $\rho(MF) \leftarrow \sum_{i=1}^{N}(LF_i/PF_i)$;
   1-3. //Utilization of inactive portion, $U_{IP}$
        $U_{IP} \leftarrow (2^{BO} - 2^{SO})/ 2^{BO}$;
   1-4. //Total utilization (or duty cycle)
        $U_T \leftarrow U_{BCAP} + \rho(MF) + U_{IP}$;
2. if ($U_T \le 1$)
       if (AllocateSlots (SS, $Slot4BNC_{SO}$, MF, PH, L, ACKR, OVH, BT)
               != SUCCESS)
           $S_{SO} \leftarrow$ SHORTGTSORSLOT;//Num of required GTS slots > 7
       else
           $S_{SO} \leftarrow$ FEASIBLE;
       end if
   else
       $S_{SO} \leftarrow$ EXCEEDUBOUND;    //exceed utilization bound
   end if

   Input:
     BO, SO: The specified superframe order and beacon order.
     MF: set of the messages in slot (harmonized)
     PH: set of the messages' periods, in symbol (harmonized)
     L : set of length (in bytes) of message set
     ACKR: set of the messages' ACK requests
     Slot4BNC: Allocated slots for beacon frame and CAP
     OVH: the overhead set
   Output:
     $S_{SO}$: the result of schedulability
```

Fig. 4. Schedulable procedure.

A. Schedulability

We evaluate the performance of the proposed scheduling algorithm with the percentage of schedulable message sets (schedulability) in terms of message utilization $(\rho(\mathbf{M}))$ for the given number of messages in a message set.

For the simulation study, we generate random message sets according to the specified utilizations between 1% and 30%. It is noticeable that each utilization requirement only includes those of the original messages from the upper layer (higher than the MAC sublayer) without any overheads incurred by the IEEE 802.15.4 standard. The lengths of messages in a message set are uniformly distributed between the minimum (1 B) and the maximum (*aMaxMACFrameSize* in [18]). The total utilization is randomly distributed for all messages in the message set and is used to compute periods of messages in the message set. The message set generated in this way in order to obtain the specified utilization ρ requires $\rho \times 250$ kb/s (e.g., 10% requires 25 kb/s) excluding the lower layer overheads (i.e., MAC/PHY overheads).

The scheduling procedure in Fig. 3 checks the schedulability for the given randomly generated message set according to the number of messages and the specified utilization. We presented the percentage of schedulable message sets among all message sets for $N = 3, 6, 7, 8, 10, 15, 25, 30, 40, 60, 80, 100,$ and 150 in [19]. Since the schedulabilities for the different N's were similar, we discuss the schedulabilities for the cases of $N = 40, 60, 80,$ and 100 in Fig. 5. The figure shows that the algorithm is capable of scheduling a periodic real-time message set. In other words, it can find the feasible schedule for the given message set. Although the simulation study shows more than 91% schedulability for the utilization of 7%, the schedulability for the utilization of original messages from the upper layer greater than 13% is less than 40%. It is measured that the scheduled message sets with 7% pure

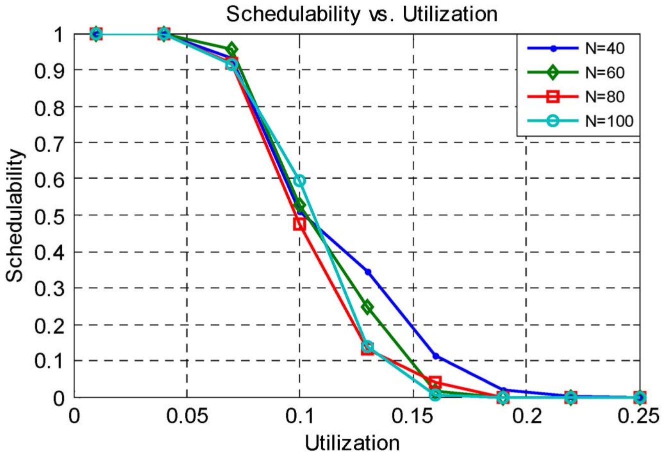

Fig. 5. Schedulability versus utilization ($\rho(\mathbf{M})$). When the utilization of the given message set from the upper layer of the MAC layer is 0.07 (7%), the schedulability is more than 91%.

TABLE III
AVERAGE SLOT UTILIZATION FOR 7% PURE UTILIZATION

N	$\rho(\mathbf{MS})$	$U_{\mathrm{IP}}+U_{\mathrm{BCAP}}$	Total slot utilization
40	0.24	0.45	0.69
60	0.25	0.39	0.64
80	0.27	0.36	0.63
100	0.27	0.33	0.60

utilization ($\rho(\mathbf{M})$) have an average slot utilization ($\rho(\mathbf{MS}) = \sum_{i=1}^{n}(\lceil LS_i/SS\rceil/\lfloor PS_i/SS\rfloor)$ of 24%–27% in this simulation study. $U_{\mathrm{IP}} + U_{\mathrm{BCAP}}$ is measured to be 33%–45%, and the total average slot utilization before harmonization for the scheduled message sets is 60%–69% as in Table III.

We run further simulations to analyze the effect of message length on schedulability. The lengths of messages in a message set are uniformly distributed between the minimum and the maximum lengths specified by message length index (MLI) in Fig. 6. The utilization of each message is generated in the same way as in the previous simulations and is used to compute periods of messages in the message set. Using the scheduling procedure in Fig. 3, a randomly generated message set is checked for schedulability according to the number of messages and the specified utilization and MLI (Fig. 6). For lack of space, only the simulation results for $N = 60$ are shown in Fig. 6. The scheduling algorithm performs with 100% schedulability when the lengths of message sets are distributed from 80 to 102 B (MLI-4) and the utilizations of the sets are less than or equal to 22%. While this shows that the algorithm can schedule 100% of scenarios where the utilization is less than or equal to 22%, it does not mean that it cannot produce feasible schedules for higher utilizations. Another thing to be observed from Fig. 6 is a trend that message sets with lower MLI show lower schedulability. Messages under the lower MLI have relatively shorter periods than those of messages under higher MLI in order to generate the same pure utilization $\rho(\mathbf{M})$. Messages with shorter periods are difficult to fit within the *maximum GTS number constraint*. In contrast, since messages with longer periods can be scheduled over more superframes (i.e., more GTS numbers) of the same length (i.e., have more flexibility), they are mitigated from the *maximum GTS number constraint*. Furthermore, messages with shorter periods limit BO to smaller values. For the smaller values of BO, U_{BCAP} in (2), which is the utilization by the beacon frame and *aMinCAPLength*, gets larger as seen in the next section.

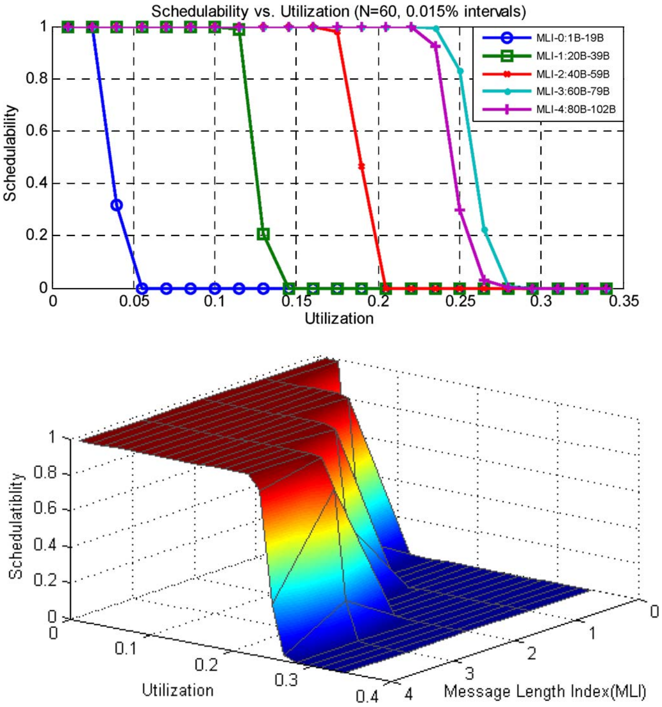

Fig. 6. Schedulability versus utilization ($N = 60$) and MLI.

The schedulable utilization (22% for MLI-4) still seems too low but the reason for this is twofold. The utilization shown in Figs. 5 and 6 is the pure utilization ($\rho(\mathbf{M})$) of the original message, not in-the-air utilization U in (2). The utilization of message sets with MLI-4 for 100% schedulability is only 22%, yet the average slot utilization ($\rho(\mathbf{MS}) = \sum_{i=1}^{n}(\lceil LS_i/SS\rceil/\lfloor PS_i/SS\rfloor)$ for the scheduled message sets with MLI-4 is measured to be 46%. $U_{\mathrm{IP}} + U_{\mathrm{BCAP}}$ is measured to be 27%, and the total average slot utilization before harmonization for the scheduled message sets is $27 + 46 = 73\%$. In addition, there are the following overhead and constraints inherited from the IEEE 802.15.4 standard:

1) overheads added by each layer, IFS, and beacon frame;
2) minimum required CAP region;
3) messages allocated on a superframe slot boundary;
4) harmonization with respect to one of 15 BI values;
5) maximum allocable GTS number in a superframe.

The schedulability of the scheduling algorithm seems to be pessimistic. It is not from the scheduling algorithm itself; rather, it is from the standard, as explained in the following section.

B. Limitations of the Standard

There are several limitations of the standard reducing the schedulability. One of them is beacon frame and the following *aMinCAPLength* period. According to the value of SO, a unit slot size is varied and the allocated slots for them (S_{BC}) is also changed. Assuming that a beacon frame includes one short

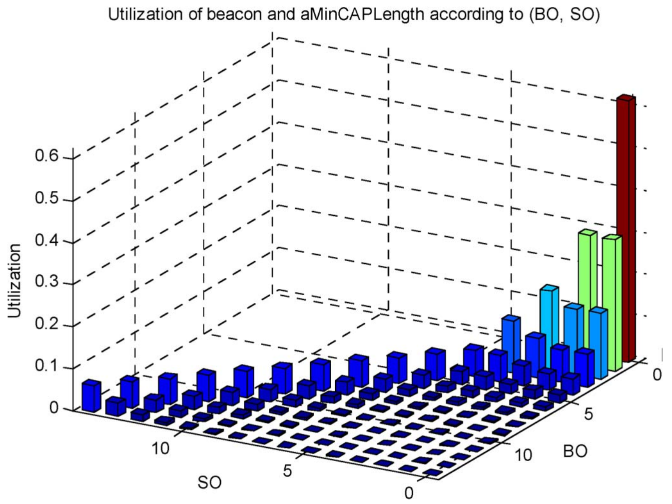

Fig. 7. Utilization of a beacon frame and *aMinCAPLength*. When SO is zero, the beacon frame and *aMinCAPLength* require ten slots. In the worst case $(BO, SO) = (0, 0)$, the utilization is up to 0.625.

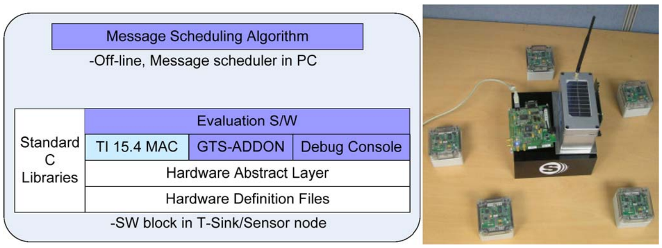

Fig. 8. SW block diagram and guaranteed time service evaluation system. The centered T-Sink node with solar panel is the PAN coordinator. Beside it, a packet sniffer is located. Five T-Sensor nodes are surrounding the T-Sink node and the packet sniffer.

address and one extended address in the pending list and a 4-B beacon payload, S_{BC} is ten slots for $SO = 0$, five slots for $SO = 1$, three slots for $SO = 2$, two slots for $SO = 3$, and one slot for the other SOs between 4 and 14. Fig. 7 shows the utilization by a beacon frame and *aMinCAPLength* period. It is notable that when $(BO, SO) = (0, 0)$, the utilization reaches to almost 0.625. That means only 0.375 of the utilization of the superframe can be used for the real exchanged message set.

Another requirement of the standard is to limit the maximum allowable GTS allocations (i.e., seven) in a superframe. For this requirement, if the maximum GTS allocations are reached in a superframe and there are still available slots in the superframe, the scheduling algorithm cannot utilize the available slots for other messages.

VII. IMPLEMENTATION

We implemented a guaranteed time service in a real industrial sensor node platform, namely, T-Sink/Sensor node [4], which was developed to measure vehicle velocity. Fig. 8 shows a software (SW) block diagram and a guaranteed time service evaluation system consisting of one T-Sink node and five T-Sensor nodes. The proposed offline message-scheduling algorithm was implemented in a PC and used to generate the schedule. We modified the TI 15.4 MAC block to support every beacon frame that includes GTS information. We implemented a GTS-ADDON block to support message transmission and receive in GTS. The schedule (i.e., BT) generated by the offline scheduler is downloaded to the PAN coordinator. When the network starts, the PAN coordinator reads the BT and broadcasts the beacon frame to the network. Whenever each associated device receives the beacon, it parses the beacon information in the beacon frame. If any GTS belongs to it, the device registers the timed callback function for message exchange in the GTS.

TABLE IV
CALCULATED VERSUS MEASURED TIMING

	Start slot	Calculated, C (us)	Measured, M (us)	Diff (us), M-C	Diff (%), (M-C)/C*100
M0_Beacon	0	245760	245730	-30.0	-0.01221
M0_GTS5	9	138240	138222	-18.0	-0.01302
M0_GTS4	11	168960	168937	-23.0	-0.01361
M0_GTS3	12	184320	184295	-25.0	-0.01356
M0_GTS2	13	199680	199654	-26.0	-0.01302
M0_GTS1	14	215040	215012	-28.0	-0.01302
M0_GTS0	15	230400	230370	-30.0	-0.01302
M1_Beacon	0	245760	245730	-30.0	-0.01221
M1_GTS4	11	168960	168937	-23.0	-0.01361
M1_GTS3	12	184320	184295	-25.0	-0.01356
M1_GTS2	13	199680	199654	-26.0	-0.01302
M1_GTS1	14	215040	215012	-28.0	-0.01302
M1_GTS0	15	230400	230370	-30.0	-0.01302

For the case in which the BT specifies $(BO, SO) = (4, 4)$ and two minor frames, we evaluate the guaranteed time service. While the first minor frame includes six GTSs (one device owns two GTSs and the other four devices own one GTS each) whose starting slots are 15, 14, 13, 12, 11, and 9, the second minor frame consists of five GTSs (the five devices own one GTS each) whose starting slots are 15, 14, 13, 12, and 11. The performance of guaranteed time service is evaluated with the calculated beacon interval $(aBaseSuperframeDuration \times 2^{\mathrm{BO}} \times 16\ \mu s)$ and the calculated GTS starting time from the following:

$$\begin{aligned} GTSStartTime &= (Slot\ Duration) \times (Start\ Slot\ Number)\\ &\quad \times (Symbol\ Duration)\\ &= SS \times startslot \times T_S\\ &= aBaseSlotDuration\\ &\quad \times 2^{\mathrm{SO}} \times startslot \times 16\ \mu s. \end{aligned} \tag{3}$$

We used the CC2420DK-based [33] IEEE 802.15.4 packet sniffer from Texas Instruments Incorporated to measure the time stamp of a GTS packet. The first beacon frame specifies the first minor frame for the following six data frames, and the next beacon frame defines the second minor frame for the following five data frames. Those two minor frames make up the major cycle, and they are scheduled to be repeated. The beacon interval was measured for the beacon frames, and the starting time from the beacon frame was measured for the data frames. We measured the timing information for 200 major frames and summarized the average measured results and the calculated GTS starting time in Table IV. The maximum

TABLE V
MEMORY USAGE

Section	SW block in T-Sink/Sensor node
Text	38.4KB
Data and bss	3.2KB
Eeprom	93 B

deviation of the measured values from the average measured value is less than 3 μs in a minor frame. The maximum error is 30 μs and 0.014%. For other combinations of (BO, SO), we evaluated the guaranteed time service. As BO is increased, the error in beacon interval is increased. Likewise, as SO is increased, the error of GTS starting time is increased. Although the error percentage shown is between 0.01% and 0.02%, it can still be reduced further depending on implementation.

Regarding the implementation of the guaranteed time service, there are two remaining issues to be discussed. First, the implemented guaranteed time service does not take much memory space, as shown in Table V. Second, since the proposed scheduling algorithm is designed to schedule a periodic real-time message including IFS which is required to follow every packet according to the IEEE 802.15.4 standard, any jitter less than IFS is tolerable. The jitter is incurred by the time synchronization error of the underlying IEEE 802.15.4 MAC implementation.

VIII. CONCLUSION

Industrial WSNs require real-time data transmission. IEEE 802.15.4 provides GTSs for real-time messages, but it has limitations. We suggested a way to overcome the maximum allowable GTS count in a network. In addition, we proposed a real-time message-scheduling algorithm for IEEE 802.15.4-based industrial WSNs. The scheduling algorithm can schedule a given periodic real-time message set, and the algorithm determines the appropriate standard specific parameters such as BO, SO, and GTS descriptor to meet the timing constraints. The proposed scheduling algorithm was analyzed with an extensive simulation study. The guaranteed time service was implemented in a real sensor node, namely, T-Sink/Sensor node, and we demonstrated, through experiments, that the implemented system runs accurately according to the schedule generated by the proposed algorithm.

Future work includes the extension of the scheduling algorithm for online scheduling and frequency hopping.

REFERENCES

[1] S. Yoo, J. Kim, T. Kim, S. Ahn, J. Sung, and D. Kim, "A^{2}S: Automated agriculture system based on WSN," in *Proc. IEEE Int. Symp. Consum. Electron.*, Jun. 2007, pp. 1–5.

[2] J. Lee, J. Kim, D. Kim, P. K. Chong, J. Kim, and P. Jang, "RFMS: Real-time flood monitoring system with wireless sensor networks," in *Proc. IEEE 5th Int. Conf. Mobile Ad Hoc Sensor Syst.*, Sep. 2008, pp. 527–528.

[3] J. Sung, S. Ahn, T. Park, S. Jang, D. Yun, J. Kang, S. Yoo, P. Chong, and D. Kim, "Wireless sensor networks for cultural property protection," in *Proc. IEEE 22nd Int. Conf. Adv. Inf. Netw. Appl. Workshops*, Mar. 2008, pp. 615–620.

[4] S. Yoo, P. K. Chong, T. Park, Y. Kim, D. Kim, C. Shin, K. Sung, and H. Kim, "DGS: Driving guidance system based on wireless sensor network," in *Proc. IEEE 22nd Int. Conf. Adv. Inf. Netw. Appl. Workshops*, Mar. 2008, pp. 628–633.

[5] S. Yoo, P. K. Chong, and D. Kim, "S3: School zone safety system based on wireless sensor network," *Sensors*, vol. 9, no. 8, pp. 5968–5988, Aug. 2009.

[6] J. Yick, B. Mukherjee, and D. Ghosal, "Wireless sensor network survey," *Comput. Netw.*, vol. 52, no. 12, pp. 2292–2330, Aug. 2008.

[7] W. Hu, G. Liu, and D. Rees, "Event-driven networked predictive control," *IEEE Trans. Ind. Electron.*, vol. 54, no. 3, pp. 1603–1613, Jun. 2007.

[8] G. Liu, Y. Xia, J. Chen, D. Rees, and W. Hu, "Networked predictive control of systems with random network delays in both forward and feedback channels," *IEEE Trans. Ind. Electron.*, vol. 54, no. 3, pp. 1282–1297, Jun. 2007.

[9] X. Cao, J. Chen, Y. Xiao, and Y. Sun, "Building-environment control with wireless sensor and actuator networks: Centralized versus distributed," *IEEE Trans. Ind. Electron.*, vol. 57, no. 11, pp. 3596–3605, Nov. 2010.

[10] A. Willig, "Recent and emerging topics in wireless industrial communications: A selection," *IEEE Trans. Ind. Informat.*, vol. 4, no. 2, pp. 102–124, May 2008.

[11] V. C. Gungor and G. P. Hancke, "Industrial wireless sensor networks: Challenges, design principles, and technical approaches," *IEEE Trans. Ind. Electron.*, vol. 56, no. 10, pp. 4258–4265, Oct. 2009.

[12] K. Al Agha, M.-H. Bertin, T. Dang, A. Guitton, P. Minet, T. Val, and J.-B. Viollet, "Which wireless technology for industrial wireless sensor networks? The development of OCARI technology," *IEEE Trans. Ind. Electron.*, vol. 56, no. 10, pp. 4266–4278, Oct. 2009.

[13] D. Choi and D. Kim, "Wireless fieldbus for networked control systems using LR-WPAN," *Int. J. Control Autom. Syst.*, vol. 6, no. 1, pp. 119–125, Feb. 2008.

[14] G. Cena, A. Valenzano, and S. Vitturi, "Hybrid wired/wireless networks for real-time communications," *IEEE Ind. Electron. Mag.*, vol. 2, no. 1, pp. 8–20, Mar. 2008.

[15] A. Flammini, D. Marioli, E. Sisinni, and A. Taroni, "Design and implementation of a wireless fieldbus for plastic machineries," *IEEE Trans. Ind. Electron.*, vol. 56, no. 3, pp. 747–755, Mar. 2009.

[16] B. Lu and V. C. Gungor, "Online and remote motor energy monitoring and fault diagnostics using wireless sensor networks," *IEEE Trans. Ind. Electron.*, vol. 56, no. 11, pp. 4651–4659, Nov. 2009.

[17] F. Chen, N. Wang, R. German, and F. Dressler, "Simulation study of IEEE 802.15.4 LR-WPAN for industrial applications," *Wireless Commun. Mobile Comput.*, vol. 10, no. 5, pp. 609–621, May 2010.

[18] *IEEE Standard for Information Technology—Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (LR-WPANs)*, IEEE Std. 802.15.4-2003, Oct. 2003.

[19] S. Yoo, D. Kim, M. Pham, Y. Doh, E. Choi, and J. Huh, "Scheduling support for guaranteed time services in IEEE 802.15.4 low rate WPAN," in *Proc. 11th IEEE Int. Conf. Embedded RTCSA*, Aug. 2005, pp. 400–406.

[20] *CC2420DBK Demonstration Board Kit User Manual (Rev.1.3)*, Texas Instrum., Dallas, TX, Jan. 2006.

[21] *2.4 GHz IEEE 802.15.4/ZigBee-ready RF Transceiver (Rev. B)*, Texas Instrum., Dallas, TX, Mar. 2007.

[22] G. Scheible, D. Dzung, J. Endresen, and J. Frey, "Unplugged but connected—Design and implementation of a truly wireless real-time sensor/actuator interface," *IEEE Ind. Electron. Mag.*, vol. 1, no. 2, pp. 25–34, Summer 2007.

[23] J. Kjellsson, A. E. Vallestad, R. Steigmann, and D. Dzung, "Integration of a wireless I/O interface for PROFIBUS and PROFINET for factory automation," *IEEE Trans. Ind. Electron.*, vol. 56, no. 10, pp. 4279–4287, Oct. 2009.

[24] A. N. Kim, F. Hekland, S. Petersen, and P. Doyle, "When HART goes wireless: Understanding and implementing the WirelessHART standard," in *Proc. IEEE Int. Conf. Emerging Technol. Factory Autom.*, Sep. 2008, pp. 899–907.

[25] *IEEE 802.15.4e Task Group*. [Online]. Available: http://www.ieee802.org/15/pub/TG4e.html

[26] L. Cheng, X. Zhang, and A. G. Bourgeois, "GTS allocation scheme revisited," *Electron. Lett.*, vol. 43, no. 18, pp. 1005–1006, Aug. 2007.

[27] J. Song, J.-D. Ryoo, S. Kim, J. Kim, H. Kim, and P. Mah, "A dynamic GTS allocation algorithm in IEEE 802.15.4 for QoS guaranteed real-time applications," in *Proc. IEEE Int. Symp. Consum. Electron.*, Jun. 2007, pp. 1–6.

[28] Y. Huang, A. Pang, and H. Hung, "An adaptive GTS allocation scheme for IEEE 802.15.4," *IEEE Trans. Parallel Distrib. Syst.*, vol. 19, no. 5, pp. 641–651, May 2008.

[29] A. Kouba, M. Alves, E. Tovar, and A. Cunha, "An implicit GTS allocation mechanism in IEEE 802.15.4 for time-sensitive wireless sensor networks: Theory and practice," *Real-Time Syst.*, vol. 39, no. 1–3, pp. 169–204, Aug. 2008.

[30] C. Na, Y. Yang, and A. Mishra, "An optimal GTS scheduling algorithm for time-sensitive transactions in IEEE 802.15.4 networks," *Comput. Netw.*, vol. 52, no. 13, pp. 2543–2557, Sep. 17, 2008.

[31] M. Y. Chan and F. Chin, "Schedulers for larger classes of pinwheel instances," *Algorithmica*, vol. 9, no. 5, pp. 425–462, May 1993.

[32] C. C. Han, K. J. Lin, and C. J. Hou, "Distance-constrained scheduling and its applications to real-time systems," *IEEE Trans. Comput.*, vol. 45, no. 7, pp. 814–826, Jul. 1996.

[33] *CC2420DK Development Kit User Manual (rev. 1.0)*, Texas Instrum., Dallas, TX, Jan. 2006.

Seong-eun Yoo received the B.E. degree in electronics and computer engineering from Hanyang University, Seoul, Korea, in 2003 and the M.S. degree in computer science and engineering from the Information and Communications University [now merged with Korea Advanced Institute of Science and Technology (KAIST)], Daejeon, Korea, in 2005. He received the Ph.D. degree in information and communications engineering, KAIST, Daejeon, Korea, in 2010.

Since September 2010, he has been a faculty member with School of Computer and Communication Engineering, Daegu University, Kyungsan, Korea. He was a Software Engineer with Fine Digital, Inc., from July 1999 to January 2002 and a Research Engineer with Sensor Networks Research, Inc., from March 2006 to February 2009. He was a Postdoctoral Researcher with Information and Electronics Research Institute, KAIST from February to August in 2010. His research interests include real-time communication in wireless sensor networks and real-time embedded systems.

Poh Kit Chong (S'99) received the B.Eng. degree from Universiti Teknologi Malaysia, Johor Bahru, Malaysia, in 2001 and the M.Eng.Sc. degree from Multimedia University, Cyberjaya, Malaysia, in 2003. He received the Ph.D. degree from KAIST, Daejeon, Korea, in 2010.

He was a Network Software Engineer with Intel's Communication Infrastructure Group from 2003 to 2005. His research interests include cross-layer design and optimization for wireless sensor networks and embedded systems.

Daeyoung Kim (M'98) received the B.S. and M.S. degrees in computer science from Pusan National University, Busan, Korea, in 1990 and 1992, respectively, and the Ph.D. degree in computer engineering from the University of Florida, Gainesville, in 2001.

He was a Research Staff Member with the Electronics and Telecommunications Research Institute, Daejeon, from January 1992 to August 1997. From September 2001 to January 2002, he was a Research Assistant Professor with the Arizona State University, Tempe. He was an Associate Professor with the Department of Computer Science and Engineering, Information and Communications University, Daejeon, from 2002 to 2009. Since March 2009, he has been an Associate Professor with the Department of Computer Science, Korea Advanced Institute of Science and Technology, Daejeon, Korea. He is a Director of Auto-ID Lab Korea (www.autoidlabs.org) and a Director of the Global USN National Research Laboratory. His research interests include sensor networks, real-time and embedded systems, and robotics.

Yoonmee Doh received the B.S. and M.S. degrees in electrical engineering from Kyungpook National University, Daegu, Korea, in 1989 and 1991, respectively, and the Ph.D. degree in computer engineering from the University of Florida, Gainesville, in 2003.

Since September 2004, she has been working in the ubiquitous sensor network research field with the Electronics and Telecommunications Research Institute, Daejeon, Korea. She was a Research Professor with the Information and Communications University [now merged with Korea Advanced Institute of Science and Technology (KAIST)], Daejeon, from June 2003. Her research interests include wireless sensor networks, smart grid networks, real-time embedded systems, and power-aware and low-power computing.

Minh-Long Pham (S'07) received the B.E. degree in electronics engineering from Hanoi University of Technology, Hanoi, Vietnam, in 1999 and the M.S. degree in computer science and engineering from Information and Communications University (ICU, now merged with Korea Advanced Institute of Science and Technology), Daejeon, Korea, in 2005. He is currently working toward the Ph.D. degree in the Signal Processing Group, Norwegian University of Science and Technology (NTNU), Trondheim, Norway.

He is also a Research Fellow with NTNU. He was with CoCo Software (1999–2002), Real-Time Embedded Systems Laboratory, ICU (2002–2004), iSphere Software (2005–2006), and Center for Embedded Software Systems, Denmark (2006–2007). His research interests include optimization and multimedia communication for wireless sensor networks.

Eunchang Choi received the B.E. and M.S. degrees in electronics engineering from Kyungpook National University, Daegu, Korea, in 1990 and 1992, respectively, and the Ph.D. degree in mobile and information system engineering from Kyungpook National University, Daegu, Korea, in 2006.

He joined the Electronics and Telecommunications Research Institute (ETRI), Daejeon, in 1993. He was a Senior Member of Research Staff in the field of switch fabric technology development until 2003. He is currently working in u-Computing Technology with ETRI as a Principal Member of the Engineering Staff with the Sensor Networking Application Research Team. His current interests include sensor networking, ZigBee, ultrawideband, and wireless personal area network standardization.

Jaedoo Huh received the B.E. and M.S. degrees in electronics engineering, and the Ph.D. degree in information and communications engineering from Kyungpook National University, Daegu, Korea, in 1987, 1990, and 2000, respectively.

He joined the Electronics and Telecommunications Research Institute (ETRI), Daejeon, in 1987. He was a Principal Member of Research Staff in the field of network research until 1999. He was a Project Leader in the field of broadband convergence networks until 2003. He is currently working in u-Computing Technology with ETRI as the Project Leader of the Sensor Networking Application Research Team. His current interests include sensor networks, sensor network management, sensor networking protocols, context-aware middleware, and intelligent agent technology.