

# Ultra Low Power Data Aggregation for Request Oriented Sensor Networks

Kwang-il Hwang\* and In Jang\*

**Abstract**—Request oriented sensor networks have stricter requirements than conventional event-driven or periodic report models. Therefore, in this paper we propose a minimum energy data aggregation (MEDA), which meets the requirements for request oriented sensor networks by exploiting a low power real-time scheduler, on-demand time synchronization, variable response frame structure, and adaptive retransmission. In addition we introduce a test bed consisting of a number of MEDA prototypes, which support near real-time bidirectional sensor networks. The experimental results also demonstrate that the MEDA guarantees deterministic aggregation time, enables minimum energy operation, and provides a reliable data aggregation service.

**Keywords**—Data Aggregation, Energy Efficient, Low Power Listening, Medium Access Control, Request Oriented, Sensor Networks

## 1. INTRODUCTION

Over the last decade, the rapid advances in wireless sensor networks have enabled the development of a variety of sensor network applications. However, in spite of their diversity, most sensor network applications are designed based on an event-driven or periodic report model. The event-driven model performs a periodic sensing function. Furthermore, only if a sensing target is detected or sensed value is above the predetermined threshold value will each sensor transmit the sensed data to the server through the sensor network. This model is mainly used in military, security, or surveillance applications. On the other hand, in the periodic report model, each sensor node transmits the sensed data to the server at a predetermined interval. Many applications for monitoring environmental changes in a certain area are mainly utilizing the periodic report model. However, since these two representative sensor network models are based on the unidirectional data transfer model, it is hard to deal with various queries or real-time requests from a server. Recently, demands for more flexible and extensible sensor network applications that are based on bidirectional communications, such as the service-centric sensor network model [1], in which a variety of tasks (services) are executed cooperatively according to various service requests or missions from a server, are arising. In order to support such applications, a request oriented sensor network model is required, instead of an event-driven or periodic report model.

The request oriented sensor network model has the specific requirements, in addition to the

---

※ This work was supported by the Incheon National University Research Grant in 2013.  
Manuscript received September 11, 2013; accepted December 11, 2013; online first August 25, 2014.

**Corresponding Author: Kwang-il Hwang** ([hkwangil@incheon.ac.kr](mailto:hkwangil@incheon.ac.kr))

\* Department of Embedded Systems Engineering, Incheon National University, Incheon, 406-772, Korea  
([hkwangil@incheon.ac.kr](mailto:hkwangil@incheon.ac.kr), [myskyeve@gmail.com](mailto:myskyeve@gmail.com))

common requirements, for an event-driven and periodic model.

- 1) **Minimum energy operation.** Energy efficiency is one of the most critical requirements in sensor networks. In particular, a request oriented model makes it more difficult for sensor nodes to operate in an energy efficient manner than with the event-driven or periodic models.
- 2) **Bidirectional communications.** For a request oriented model, bidirectional communication between the server and sensor nodes is a basic requirement. Furthermore, requests from a server should be able to be disseminated to the sensor network in near real-time.
- 3) **Reliable data aggregation.** A header node (e.g., coordinator) should be able to aggregate data from correctly designated nodes. Furthermore, the aggregation method should include a functionality that can reliably aggregate all of the responses from the designated nodes in a harsh environment.
- 4) **Deterministic response time.** The total response time it takes from the moment a service or query request is made until all responses arrive should be deterministic. A non-deterministic response time may make it impossible to estimate the waiting time for all responses.
- 5) **Support for various requests or services.** A sensor network should be able to process a variety of requests or services. To achieve this, variable target node selection (one, selective, or all) should be supported, and also communications among selected nodes should rarely influence the other nodes that are not participating in the communication at that moment.

These strict requirements make it more difficult to apply conventional network models to a request oriented sensor network. Thus, even though there have been numerous amounts of research [2-19] conducted on low power sensor network protocol design, more specifically medium access control (MAC) protocols, none of these, which are based on the event-driven or periodic model, fulfills the requirements for a request oriented sensor network.

Therefore, in this paper we propose a minimum energy data aggregation (MEDA) method to meet the specific requirements for a request oriented sensor network model. The proposed idea basically utilizes a low power listening (LPL) to minimize idle listening. However, unlike other LPL protocols, we exploited novel ideas such as on-demand time synchronization, variable response frames (VRFs), and adaptive retransmission. In addition, to achieve minimum energy operation, we also designed and implemented a low power real-time scheduler.

The remainder of this paper is organized as follows: in Section 2, several protocols for sensor networks, more specifically low power MAC protocols, are investigated. Section 3 presents the architecture and operational details of the proposed scheme. Section 4 presents the experimental results that are based on the developed prototype and performance evaluations. Finally, Section 5 provides some concluding remarks.

## 2. RELATED WORK

Low power or energy efficient protocol design is considered to be one of the critical issues in sensor networks. In particular, the MAC protocol coupled with power management is of the most importance in that all upper layer protocols have to utilize MAC layer services. So far there

have been numerous amounts of research carried out on low power MAC protocols. The study on low power MAC protocols can be largely categorized as a standard approach based on 802.15.4 and a non-standard approach based on traditional sensor networks.

IEEE802.15.4 [2] is a representative standard based low power MAC protocol, which aims at constructing a low power wireless personal area network (WPAN) by maintaining a superframe structure that is based on the beacon frame of a coordinator. Nevertheless, the lifetime of WPAN devices, which have been already deployed in real fields, is not guaranteed as much as we expected. This is due to inherent characteristics using carrier sense multiple access/collision avoidance (CSMA/CA) random access within a superframe. Recently, TG4e [3] proposed two different approaches: coordinated sampled listening and receiver initiated transmission in order to improve energy efficiency by overcoming the limitation of 802.15.4. The former approach is a method for a sender to perform data communication after waking up a receiver that is periodically performing a channel sampling. It does so by sending a long wakeup sequence to the receiver. The latter approach utilizes a method for a receiver to periodically check whether the sender has data to transmit.

Non-standard based low power MAC protocols can be also classified as random-based, slot-based, time division multiple access (TDMA)-based, and random/TDMA hybrid, and LPL. The brief characteristics of each protocol are clearly described in Table 1. Random-based protocols [4-7] utilize a contention-based MAC protocol, as seen in IEEE802.15.4. Slot-based approaches [8-10] propose various schedule algorithms to perform listening at a dedicated slot. TDMA-based approaches [11-14] utilize a repetitive frame structure between a master and slaves based on global synchronization. Hybrid approaches [15,16] propose several hybrid methods that combine a random approach and TDMA MAC. In particular, LPL approaches [17-20], which are based on a preamble sensing to minimize idle listening at the receiver, show a better performance than other approaches in terms of energy efficiency and algorithm complexity.

**Table 1.** Summary of low power MAC protocols

Category	Title	Brief characteristic
Randombased	STEM [4]	- Utilizes separate RF for wake up plane and data plane - Wake up Plane is only used to wake up receiver
	RI-MAC [5]	- Receiver initiates communications - Receiver transmits periodic beacon, and a sender start to transmit data as soon as it listens a beacon
	RATE-EST [6]	- Utilizes additional RF to wake up a receiver like STEM - Schedules wake-up sequences to avoid waking up nodes which are not participated in the communication
	SMAC [7]	- Classic CSMA style MAC protocol - Schedules nodes within virtual cluster, and utilizes a fixed duty cycle to reduce energy consumption
Slot-based (scheduled)	SMAC/AL [8]	- Enhanced version of SMAC including adaptive listening - Expands active period using adaptive duty cycle as in T-MAC
	DMAC [9]	- Data gathering MAC protocol - Reduces delay overhead by applying convergecast - Schedules slots among descendants as soon as a message is received
	SCP-MAC [10]	- Scheduled channel polling MAC - Applies an efficient channel polling method to SMAC/AL - All nodes are synchronized at a common slot, a sender contends with other nodes by transmitting busy tone, and transmits data waking up the receiver at the beginning of a slot

TDMA-based	LMAC [11]	<ul style="list-style-type: none"> <li>- Lightweight MAC</li> <li>- Self-organizing TDMA</li> <li>- All nodes possess its own slots in a fixed length of frame</li> <li>- Broadcasts in an occupied slot, and a new node selects unique two hop slots so that the node can transmit data without collisions</li> </ul>
	AI-MAC [12]	<ul style="list-style-type: none"> <li>- Adaptive information-centric and lightweight MAC</li> <li>- Enhanced version of LMAC</li> <li>- Allows each node to occupy multiple slots</li> </ul>
	TRAMA [13]	<ul style="list-style-type: none"> <li>- Traffic adaptive medium access</li> <li>- TDMA MAC based on a distributive slot selection algorithm</li> <li>- Exchanges two hop neighbor's information to support multi-hop communications</li> </ul>
	CoRe-MAC [14]	<ul style="list-style-type: none"> <li>- Cooperative relaying MAC</li> <li>- Utilizes RTS/CTS for channel reservation, and performs a relay selection through the channel</li> </ul>
Random/TDMA hybrid	ZMAC [15]	<ul style="list-style-type: none"> <li>- Zebra-MAC</li> <li>- Starts from CSMA, but changes to TDMA if communication load is increased</li> <li>- Performs a distributed slot selection algorithm</li> </ul>
	PMAC [16]	<ul style="list-style-type: none"> <li>- Pattern MAC</li> <li>- Distributed and adaptive TDMA</li> <li>- Exchanges traffic information to identify node wake-up time, and performs CSMA/CA communications in the corresponding slot</li> </ul>
Low power listening/preamble sensing based	BMAC [17]	<ul style="list-style-type: none"> <li>- Berkeley MAC</li> <li>- Perform application level low power listening, and utilizes check time, back-off window size, sleep policy</li> <li>- Perform advanced clear channel assessment to remove random noise</li> </ul>
	XMAC [18]	<ul style="list-style-type: none"> <li>- Enhanced version of B-MAC</li> <li>- Long preamble is divided into two periods: micro preamble period and receiver address period.</li> <li>- Resolve unnecessary wake-up of all nodes due to preamble sensing</li> </ul>
	WiseMAC [19]	<ul style="list-style-type: none"> <li>- Improve low power listening by remembering polling schedules of neighboring nodes</li> <li>- Transmits short preamble at the exact time</li> </ul>
	CL-MAC [20]	<ul style="list-style-type: none"> <li>- Cooperative low power MAC</li> <li>- Proposes proactive and reactive method</li> <li>- Utilizes inter-preamble and random back-off to avoid preamble collisions</li> </ul>

MAC=medium access control, TDMA=time division multiple access, CSMA=carrier sense multiple access, CA=collision avoidance.

Although these protocols have contributed to constructing an energy efficient sensor network, most of them are designed based on the event-driven or periodic model. Thus, they cannot fulfill the requirements for a request oriented model. Therefore, a new MAC protocol that can satisfy the strict requirements for the request oriented sensor network is required.

### 3. MEDA FOR REQUEST ORIENTED SENSOR NETWORKS

In this section we present a MEDA that meets the strict requirements for request oriented sensor networks. The MEDA essentially utilizes the LPL, which is based on a preamble transmission and preamble sensing. However, it enables fully bidirectional communications and requires a minimum amount of energy to operate. It does so by exploiting the asynchronous source trigger, VRF structure, and adaptive retransmission based on a low power real-time scheduler.



### 3.1 Low Power Real-Time Scheduler

For LPL based MAC protocols, power saving associated with low power scheduling is one of the most essential capabilities. The MEDA conducts aggressive power savings by using dynamic slot management that is based on on-demand source trigger function, which is also enabled by a low power real-time scheduler.

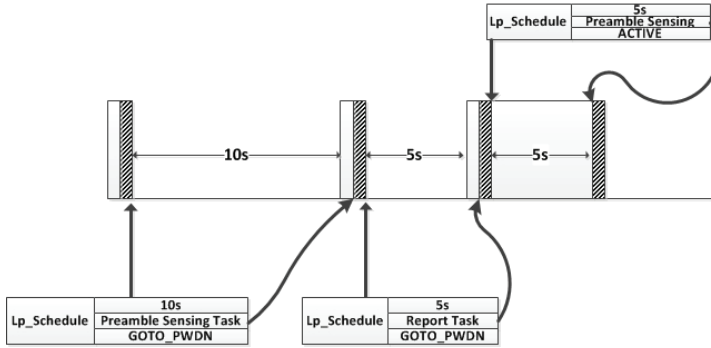
In general, a task scheduler in embedded systems utilizes a general purpose timer, which is sourced by a main clock. This type of scheduler is capable of providing correct and elastic schedule operations during active periods. However, during microcontroller unit (MCU)'s sleep mode, the MCU timers do not run, since the main clock stops. So, a low power scheduling in which a scheduler continuously runs not only in runtime but also in sleep mode, is not possible. On the other hand, a real-time clock (RTC), which is sourced by a separate crystal oscillator (32.768 kHz), can keep on running while only consuming a minimum amount of power even during MCU's sleep mode. Currently, most commercial MCUs have a built-in RTC and they provide a calendar mode that includes an alarm function and interval timer mode. However, since these two modes of a calendar and timer mode cannot be used concurrently, the built-in RTC is generally utilized as a real-time watch in the calendar mode. If the use of an interval timer is required, an external RTC should be added to the system. However, the use of an additional component results in an increase in costs and energy.

Therefore, we developed a low power real-time scheduler, which enables both the operation of the calendar mode and interval RTC concurrently. This does not require the use of additional components. We first set the RTC output pin provided for a clock accuracy test to 1 Hz output, and connected the output pin again to the input of clock source of Timer 3. Eventually, it results in using Timer 3 like a real-time interval timer. The developed *Lpscheduler* enables a sensor node to continuously run not only during the runtime but also during the sleep mode without regard to a main clock operation. The scheduler basically has the following three parameters:

$$Lpschedule(time, handler, p\_state)$$

Where time presents a scheduling time and the unit of time is 1 second, *handler* means a task (function) to be executed at the scheduled time, and *p\_state* determines the MCU power management mode (Active, GOTO\_PWDN) until the scheduled time.

Fig. 1 illustrates a simple example of *Lpscheduler* use. Right after scheduling the first task, the node goes into the sleep mode and maintains the sleep state for 10 seconds. After 10 seconds, the node wakes up, executes the scheduled task (*Preamble\_sensing\_task*), and then goes into sleep mode again as soon as it schedules the next task (*Report\_task*). After 5 seconds, the node wakes up to execute the *Report\_task*, and then schedules the next task (*Preamble\_sensing\_task*) again. Note that the *p\_state* of the task is scheduled as ACTIVE, unlike previous tasks scheduled as GOTO\_PWDN. So, the node maintains an active state for 5 seconds, as shown in Fig. 1. As shown in the example, the *Lpscheduler* operates simply but still is able to provide powerful, low power real-time scheduling by exploiting a RTC clock and chain-based scheduling method.

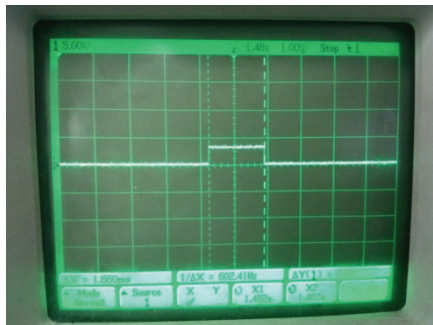


**Fig.1.** An example of a low power real-time scheduler.

### 3.2 Periodic Preamble Sensing

A preamble sensing is an essential functionality for LPL based protocols to minimize idle listening overheads. For such LPL based protocols, one of the critical performance factors is an accuracy for the preamble detection and sensing duration.

Therefore, to provide accurate preamble detection and minimize sensing duration, we sought the optimized preamble sensing duration and preamble quality threshold (PQT) value through experiments. Fig. 2 shows the measured value of the preamble sensing duration used in the MEDA. It is important to note that the optimized preamble sensing duration is very small (only 1.6 ms). Thus, considering periodic preamble sensing interval (PPSI)=3 seconds, 0.0005% duty cycle is obtained. That is, the MEDA enables a sensor node to perform ultra low PPS, and this also lead to minimum energy operation.



**Fig.2.** Preamble sensing duration.

### 3.3 Asynchronous Source Trigger

In order to trigger the nodes that are performing PPS, a sender has to transmit a preamble prior to data transmission. The MEDA has the capability of efficiently triggering all of the nodes that are performing PPS asynchronously (shown in Fig. 3) by transmitting a preamble with PPSI+1 in length. To trigger nodes performing PPS at a configured PPSI, the preamble length should be greater than the PPSI value. Also, considering RTC clock drift and oscillator error, we determined the preamble length to be PPSI+1 (second).

### 3.4 On-Demand Time Synchronization

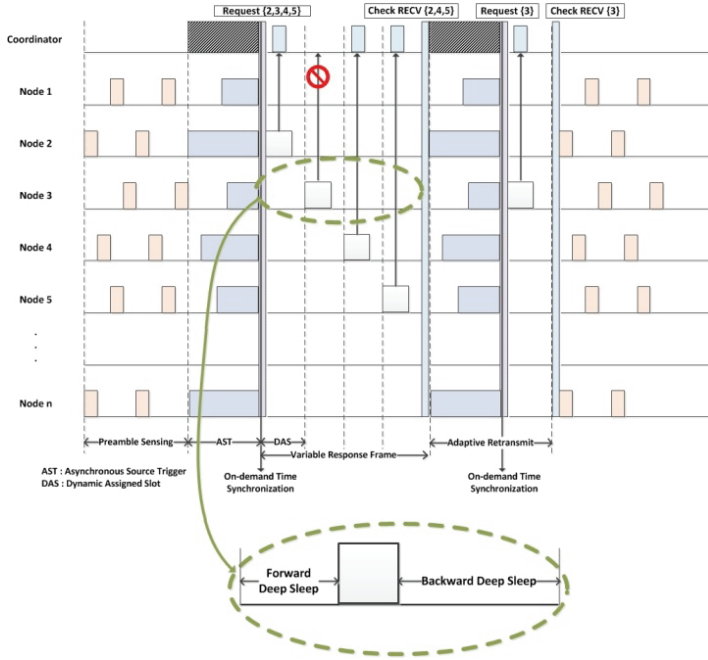
Devices that detect a preamble during PPS have to maintain an active state until they receive the request packet, followed by the preamble. However, each node might detect a preamble at different times, and thus the waiting time delay of each node might be different. In other words, each node might not be synchronized with one another and with the coordinator at that moment. However, this does not matter since the MEDA is capable of doing on-demand time synchronization.

At the end of a preamble, a request packet is broadcasted, and pending nodes, which are triggered by the preamble of the sender, are synchronized with the sender as soon as they receive a request packet. It is important to note that every node does not need to try to keep the current synchronization for further communications. The synchronization in this round is effective only in this round. So, the on-demand synchronization of the MEDA can remove the overhead to maintain time synchronization. Moreover, slot management of the MEDA does not require synchronization as tightly as in TDMA. As such, low power real-time scheduling enables the on-demand synchronization.

### 3.5 Variable Response Frame

One of the outstanding features of the MEDA is that it can accommodate various types of requests that originate from a server or coordinator. It can also aggregate various responses from selective target nodes with respect to a single request. In other words, a single node, selective nodes, or all nodes can respond to a single request within a single data aggregation round. A participation list field in a request packet represents the information about the target nodes. As shown in Fig. 3, all of the nodes that receive the request packet after preamble detection first synchronize with the coordinator. Then, they form a VRF based on the information from the participation list and the total number of participants in a request packet. A VRF is composed of a number of equal sized dynamic assigned slots (DASs), and the number of DASs varies according to the number of participants and target nodes. The procedure for a node to determine its own DAS in the VRF with respect to a new request is as follows: upon the receipt of a request packet, each node first checks whether its ID exists in the participant list in the packet. If so, the node stores the index number of the ID within the list. This index value eventually becomes its own DAS number. After completing the VRF formation all nodes go into sleep mode, and the only the nodes that obtain their own DAS wake up to transmit their own responses to their own DAS. The nodes then go into sleep mode, and perform PPS again at the end of VRF in this round. On the other hand, if the ID of a node does not exist in the list, it means that node does not need to attend the data aggregation process in this round. Instead it has to calculate the entire frame length from the total number of participant fields in a request packet, and then the node goes into sleep mode. This deep sleep function saves much more energy by removing even unnecessary PPSs and also not interfere with other communications during the VRF process in that round.

Since the VRF operates like a TDMA, the MEDA is capable of performing a MEDA. However, the slot configuration and management are more flexible than TDMA and these are performed in a distributed way. Furthermore, the more requested nodes there are, the more energy in each node is saved. This is because the deep sleep duration, in which nodes go into sleep mode without performing PPS, is lengthened. This occurs as the amount of participants increases.



**Fig.3.** Minimum energy data aggregation structure and operational example.

This VRF process is efficiently implemented based on *Lpscheduler*. Fig. 4 describes a pseudo-code for the VRF process that includes two major events: request packet reception and DAS slot notification. First, Lines 2-7 are for the node participating in this round. Lines 3-5 show the DAS determination process with respect to the request packet. Line 6 is a process to

**Event: Request packet Reception with <R>**

<R> is a set of the nodes which should attend data aggregation this time.

(e.g.,  $R = \{2, 4, 5, 6, 10\}$ )

```

1  NumofTotalDevice = length(R);
2  if(My id is in the <R>)
3      for (i = 0; i < NumofTotalDevice; i++)
4          if(R[i] == myId) break
5      MyDAS = i + 1 // Number of forward slots
6      NumofBackwardSlots = NumofTotalDevice - MySlotNum
7      LpSchedule(MySlotNum * BaseSlotDuration, DAS_handler, Goto_PWDN)
8  else
9      LpSchedule(NumofTotalDevice * BaseSlotDuration, PPS_handler, GoTo_PWDN)

```

**Event: My DAS**

```

1  Response Task
2  MakePacket(Data)
3  Send(Response packet)
4  LpSchedule(NumofBackwardSlots * BaseSlotDuration, PPS_handler, GoTo_PWDN)

```

**Fig. 4.** Pseudo-code for variable response frame operation. DAS=dynamic assigned slot.

calculate forward deep sleep duration in which the node goes into sleep mode and only wakes up to perform a response task at the end of the forward deep sleep duration. Line 7 is a process to schedule the DAS, and Lines 8 and 9 are for the nodes that are not participating in this round, and in this case the node registers the PPS handler with *Lpscheuler* to perform PPS again at the end of the VRF process.

The scheduled DAS task is performed as soon as the node wakes up from *Lpscheduler*. It responds to the requester within the DAS. At the end of the response, a node reschedules the PPS task to go into sleep mode. It does so by using the backward deep sleep duration, which is already calculated in the packet reception process. Finally, after finishing the VRF, all nodes perform PPS again, and are ready for a new request.

### 3.6 Adaptive Retransmission

To ensure a reliable data transfer between the coordinator and each node, each response from the nodes should be acknowledged by a coordinator. However, applying the general Data and ACK model to a DAS brings a significant increase in the duration of the entire VRF process. This is because it requires an extension of a DAS duration to accommodate a number of retransmissions, as well as additional timer handling to wait for the ACK from the coordinator. Therefore, the MEDA resolves this problem by exploiting a novel adaptive retransmission method using the group ACK approach. For adaptive retransmission, a coordinator does not acknowledge each response during VRF, and instead removes the corresponding ID from the participation list whenever a response packet is received. At the end of VRF, the coordinator checks the current participation list. If the list is vacant, which means that all responses have been completely received, it terminates this round. If the list is not vacant, which means some of the responses are not completed, the coordinator substitutes the original participation list for the current list, and then starts a preamble transmission for re-request. Since all nodes are performing PPS at that moment, they all are triggered by the preamble. However, since the participation list field in the request packet contains only the IDs of the nodes that the previous response transmission is failed, after preamble detection all the nodes which are not engaged in the list, which means response are successfully completed, goes into deep sleep mode. Therefore, only failure nodes attend the retransmission round. As shown in Fig. 3, in the case where the coordinator fails to receive the response from Node 3 in the original VRF, upon the receipt of the re-request packet followed by a preamble, only Node 3 attends the retransmission round and the others go into sleep mode. As shown in the figure, the retransmission of Node 3 is successfully completed. Also, all of the nodes, including Node 3, perform PPS again at the end of the retransmission round.

The main advantage of adaptive retransmission is that only a coordinator and not general devices is responsible for retransmission processing. The coordinator does so by managing a participation list. Since other nodes regard re-requests as a normal request without any additional retransmission processing, the MEDA cannot only maximize the retransmission efficiency by simple but powerful operation, but can also minimize the overhead or additional delay by doing a retransmission.

## 4. EXPERIMENTAL RESULTS

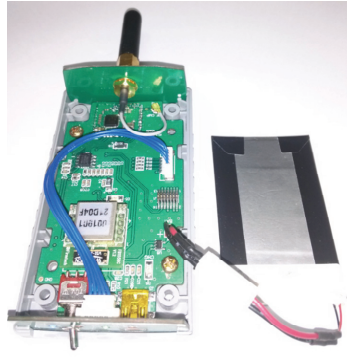
We constructed a request oriented sensor network test bed based on near real-time

bidirectional communications. In this section, the developed MEDA prototype and a test bed consisting of the MEDA nodes are introduced. We then present the performance evaluation of the proposed MEDA.

## 4.1 Experimental Environments

### 4.1.1 MEDA prototype

As shown in Fig. 5, we developed a MEDA prototype based on the MSP430F6736 (Texas Instruments, Dallas, TX, USA) ultra low power 16-bit MCU and CC1120 RF Transceiver (Texas Instruments) to construct a near real-time bidirectional sensor network. The prototype is basically composed of a minimal amount of hardware components to minimize power consumption. The full functions of the MEDA containing hardware abstract layer, an event-driven task manager, and a *Lpscheduler* were also implemented. To deal more efficiently with preamble operations, we also controlled the RF transceiver directly, utilized minimum preamble sensing duration (1.6 ms), and optimized the PQT.



**Fig. 5.** Minimum energy data aggregation prototype.

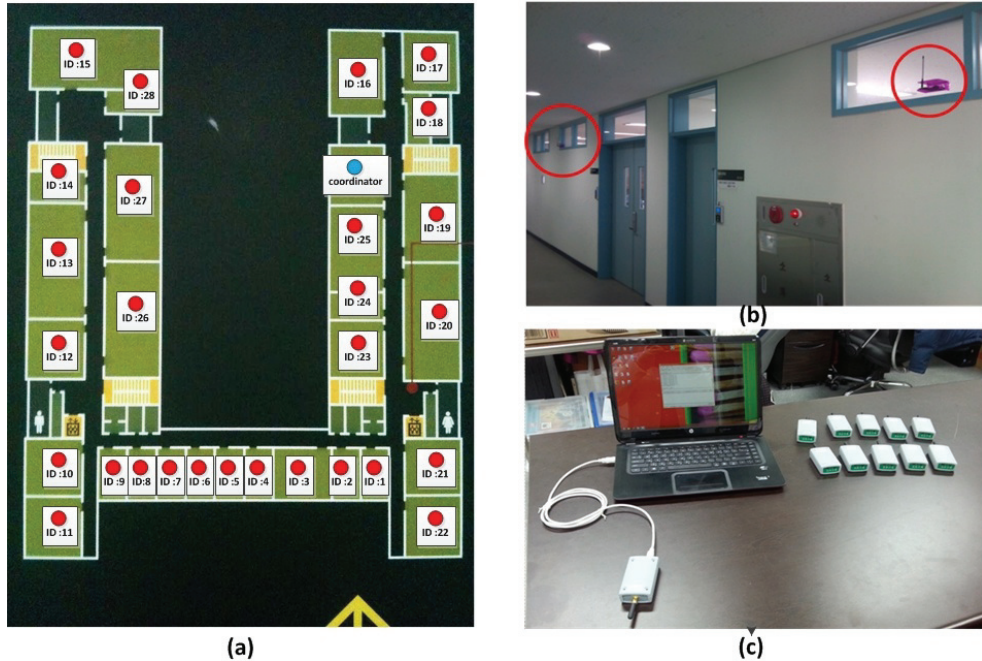
**Table 2.** System characteristics

Parameter	Value
Supply voltage	3.3 V
Current consumption	
Tx	50.58 mA
Rx	21.04 mA
Sleep	19.91 $\mu$ A
Battery capacity	12,000 mAh
Tx power	15 dBm
Periodic preamble sensing interval	1–5 (configurable)
Preamble duration	2–6 (configurable)
Number of retry	3 (max)
Periodic request rate	1 hr
Sensors	Temperature, humidity, CDS

CDS=cadmium sulfide.



Table 2 presents the basic characteristics of the MEDA prototype that we developed. The MEDA node is divided into the coordinator and sensor nodes. In particular, a coordinator node is capable of transmitting data collected from sensor nodes to a server, as well as disseminating server requests to sensor networks.



**Fig. 6.** Minimum energy data aggregation test bed.

#### 4.1.2 MEDA test bed environments

Fig. 6 shows a deployment map of the MEDA test bed, which was constructed on the 3rd floor of our university building. As shown in Fig. 6, 1 coordinator node was connected to a server located in our lab, and 28 sensor nodes were deployed. Each sensor node, which we installed in each room, as shown in Fig. 6(b), can collect environmental data, such as temperature, humidity, and illumination intensity in real-time. To facilitate one hop communication, we set the transmission power to 15 dBm. This allowed the coordinator to aggregate the latest sensed data of each sensor node by request from a server.

### 4.2 Performance Evaluations

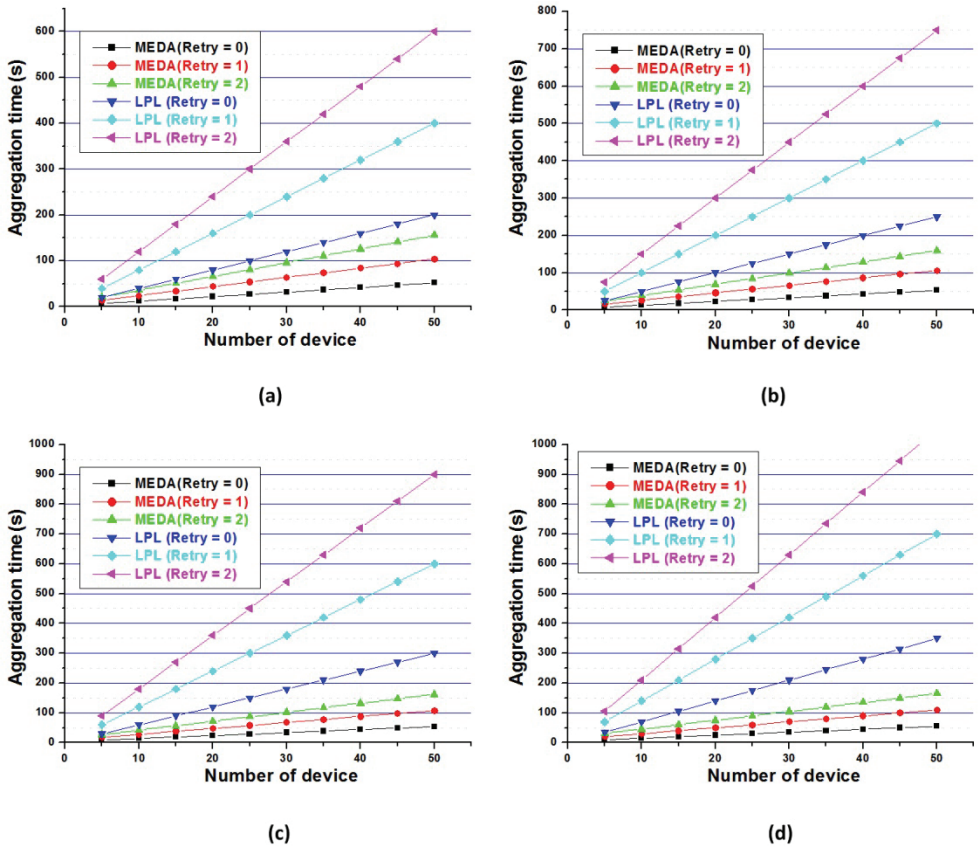
Apart from the test bed, we experimented with 1 coordinator that was connected to a laptop and a number of sensor nodes. We did this in order to evaluate the various performances of the MEDA protocol and system, as shown in Fig. 6(c). The experiment was to measure aggregation time and average throughput with respect to varying the number of nodes from 5 to 50, at different PPSI values. In addition, based on the measurement result of various system parameter values, the lifetime of the node was analyzed.



We also implemented the general LPL algorithm on the same prototype, and the MEDA was compared with the LPL under the same conditions and environment in each experiment. In the general LPL method, a coordinator transmits a unicast request to a requested node at the end of a preamble. If the response failed to transmit, the responder performed retransmissions three times.

#### 4.2.1 Aggregation time

First, we observed the aggregation time, which is the total time required from the request of a server until all responses are aggregated. In this experiment, we compared the aggregation times of MEDA and LPL under three conditions: 1 retry (retransmission), 2 retransmissions, and no retransmission. Fig. 7 shows the result of the aggregation time measurement at PPSI=1, 2, 3, and 4, respectively. The result shows that the aggregation times of MEDA and LPL were both influenced by the PPSI value. Since the preamble duration to trigger the sensor nodes that are performing PPS is lengthened as much as the PPSI increases, this eventually results in an increase of aggregation time. The aggregation time also increases proportionally to increase with the number of sensor nodes. However, it is important to note that as shown in Fig. 7, while the LPL shows that the magnitude of the slope is considerably large, the MEDA shows a smaller



**Fig. 7.** Aggregation time vs. the number of nodes. (a) PPSI = 1, (b) PPSI = 2, (c) PPSI = 3, and (d) PPSI = 4. PPSI=periodic preamble sensing interval, MEDA=minimum energy data aggregation, LPL=low power listening.

rate of increase than the LPL. The LPL, which has no retry condition, was even larger than the MEDA in the case of 2 retransmission conditions. This is because while the LPL performs data aggregation through making an individual request to each node, the MEDA can aggregate all of the requested data within a VRF by making a single request. The result also demonstrates that the VRF and adaptive retransmission structure enables a coordinator to have a deterministic aggregation time that is proportional to the number of requested nodes, which is accomplished by on-demand time synchronization.

#### 4.2.2 Lifetime analysis

Energy efficiency is one of the most important requirements in sensor networks. Therefore, in this subsection, we estimate the lifetime of a MEDA sensor node and coordinator, and perform comparative analysis with the lifetime of a node where the LPL is applied. Actually, since it is hard to observe the actual lifetime of a system, we numerically analyzed the life expectancy of a system by using various parameters that we measured through experiments. Table 3 presents the parameters and measured values that we used in the analysis. The lifetime of a node can generally be obtained by Eq. (1), but each individual lifetime of the MEDA and LPL is varied according to a different active period. The TA (MEDA) and TA (LPL) are obtained by Eqs. (3) and (4), respectively. Fig. 8(a) and (b) show the result of a lifetime analysis. The result shows that the lifetime of a MEDA sensor node can be guaranteed for more than 15 years without regard to the number of requested nodes at all PPSI values. Whereas, the lifetime of a LPL sensor node does not exceed 5 years, except for 7 years lifetime at PPSI=1. The result also reveals that on average, the lifetime of a MEDA sensor node is eight times longer than LPL. It is because unlike the LPL, which performs data request individually, the MEDA can minimize energy to aggregate data from all requested nodes on a single aggregation round, as well as to remove the unnecessary preamble detection. In addition to the sensor node lifetime, Fig. 8(c) and (d) show the coordinator lifetime of MEDA and LPL, respectively. A coordinator consumes

**Table 3.**Notation for lifetime analysis

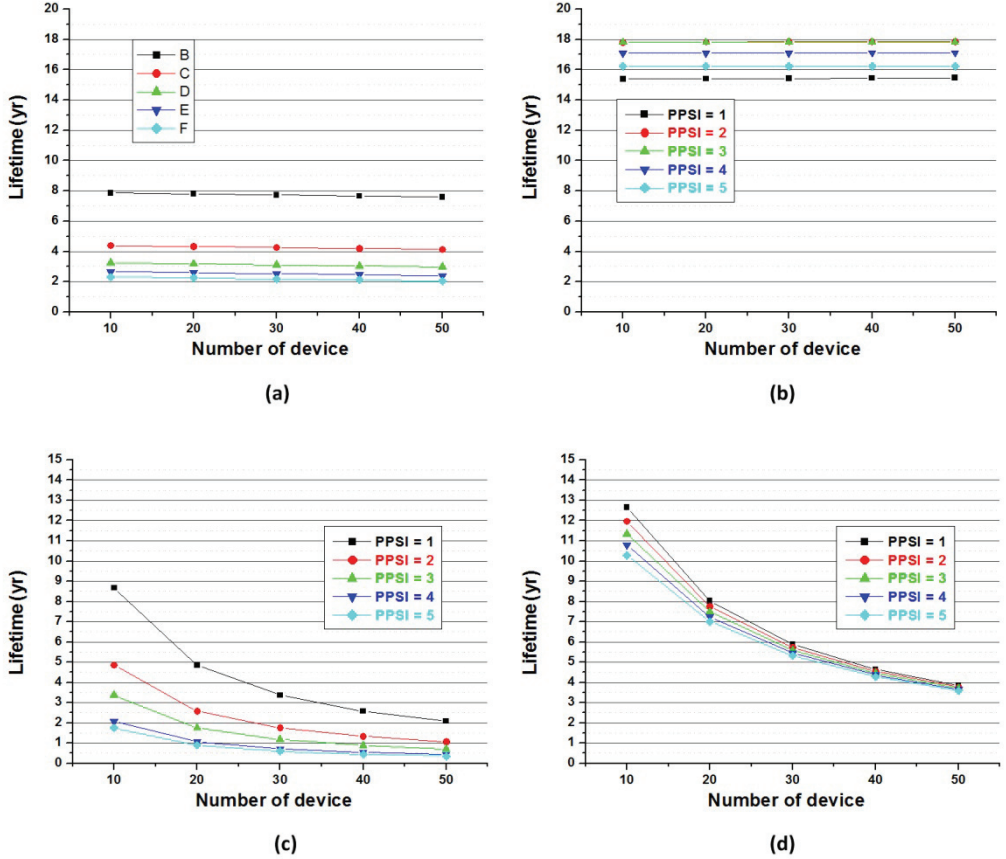
Notation	Description	Value
$C$	Battery capacity	12,000 mAh
$T_A$	Active period	
$T_{PWND}$	Sleep period	
$T$	Period	1 hr (3,600 sec)
$T_{AST}$	Maximum asynchronous source trigger period	Depends on the periodic preamble sensing interval
$T_{resp}$	Transmission time of response packet	8.6 ms
$T_{req}$	Transmission time of response packet	7.8 ms
$T_{DAS}$	Dynamic assigned slot duration	1 sec
$T_{PSD}$	Preamble sensing duration	1.65 ms
$I_A$	Current consumption (active state)	21.04 mA
$I_{PD}$	Current consumption (sleep state)	19.91 $\mu$ A
$n$	Number of nodes	5–50

$$Lifetime = \frac{C}{T_A \times I_A + T_{PWDN} + I_{PD}} \quad (1)$$

$$T_{PWDN} = T - T_A \quad (2)$$

$$T_{A(MEDA)} = T_{AST} + T_{resp} + \left( \frac{T - (T_{AST} + T_{req} + T_{DAS} \times n)}{PPSI + T_{PSD}} \right) \times T_{PSD} \quad (3)$$

$$T_{A(LPL)} = T_{AST} + T_{req} + T_{resp} + \{n \times (T_{AST} + T_{req})\} + \left( \frac{T - (T_{AST} + T_{req}) \times n + T_{rep}}{PPSI + T_{PSD}} \right) \times T_{PSD} \quad (4)$$



**Fig.8.** Lifetime analysis: sensor node lifetime (a, LPL; b, MEDA) and coordinator lifetime (c, LPL; d, MEDA). LPL=low power listening, MEDA=minimum energy data aggregation, PPSI=periodic preamble sensing interval.

more energy than sensor nodes since it needs to maintain an active state during the entire data aggregation process. Nevertheless, it also shows that the lifetime of a MEDA coordinator can be guaranteed for more than 5 years, whereas, the lifetime of a LPL coordinator does not exceed 4 years in the cases of 30 nodes environment.

### 4.3 Average Throughput

The last observation we want to present is about the average throughput, which is used to measure the time required to complete reliable data transfer under various error environments. We measured the average throughput of MEDA and LPL, respectively, with respect to the response packets of a fixed length of 128 bytes at different packet error rates (PERs) of 70%, 80%, and 90%, and the maximum retransmission was extended by 5. Fig. 9 shows the average throughput of response packets with respect to a total of 1,000 requests. The result shows that the MEDA, even under a 70% PER environment, maintains higher throughput than LPL. However, it is noticeable that unlike the LPL, in which the number of nodes rarely influences the average throughput, the MEDA shows a sharp degradation as the number of nodes increases. This is because the length of the VRF increases proportionally to the number of nodes. Also, this is because the total duration is lengthened due to retransmissions starting after checking the responses received from requested nodes at the end of the VRF. However, since the MEDA does not use an individual ACK but instead uses a group ACK based adaptive retransmission, the normal response transmission of non-problematic nodes is not influenced by retransmissions. Therefore, in spite of throughput degradation occurring as the number of nodes increases, the MEDA outperforms LPL from the viewpoint of average throughput.

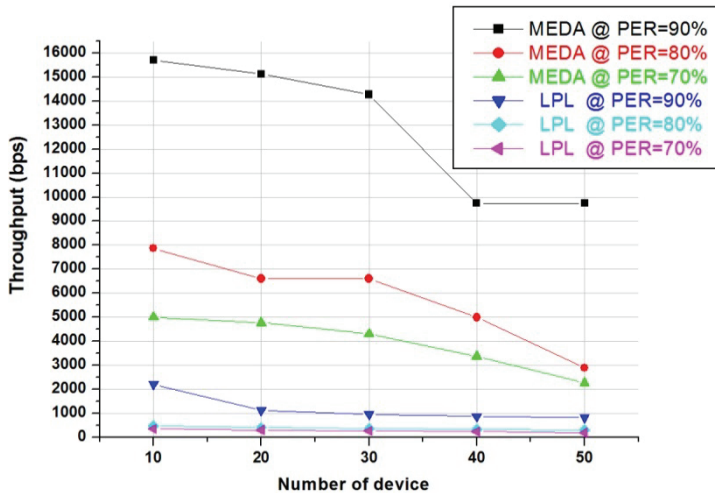


Fig.9. Average throughput. LPL=low power listening, MEDA=minimum energy data aggregation.

## 5. CONCLUSIONS

In this paper, we have dealt with a request oriented sensor network, which is different from

the conventional event-driven or periodic report models. In particular, specific requirements, such as minimum energy operation, bidirectional communications, reliable data aggregation, deterministic aggregation time, and various request supports, make it more difficult that existing low power protocols based on even-driven or periodic models are applied to request oriented service model. Therefore, we proposed a MEDA scheme, which is composed of a low power real-time scheduler, an on-demand time synchronization, VRF structure, and adaptive retransmission.

In addition, we evaluated the feasibility and efficiency of the proposed MEDA by using the near real-time bidirectional sensor network test bed, which consisted of the MEDA prototype. The experimental results also demonstrate that the MEDA guarantees deterministic aggregation time, enables minimum energy operation, and provides a reliable data aggregation service.

Finally, it is expected that the MEDA will be able to contribute to the design and implementation of new sensor networks based on a request oriented model.

## REFERENCES

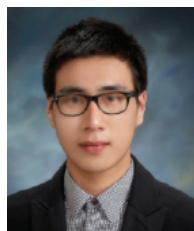
- [1] D. Gracanin, M. Eltoweissy, A. Wadaa, and L. A. DaSilva, "A service-centric model for wireless sensor networks," *IEEE Journal on Selected Areas in Communications*, vol. 23, no. 6, pp. 1159-1166, Jun. 2005.
- [2] *Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (WPANs)*, IEEE Std 802.15.4, IEEE Standard for Information Technology, 2006.
- [3] *Amendment 5: Amendment to the MAC Sub-layer for Low-Rate Wireless Personal Area Networks (WPANs) Amendment 5*, IEEE P802.15.4e Draft, IEEE Standard for Information Technology, 2011.
- [4] C. Schurgers, V. Tsiatsis, S. Ganeriwal, and M. Srivastava, "Optimizing sensor networks in the energy-latency-density design space," *IEEE Transactions on Mobile Computing*, vol. 1, no. 1, pp. 70-80, Jan. 2002.
- [5] Y. Sun, O. Gurewitz, and D. B. Johnson, "RI-MAC: a receiver-initiated asynchronous duty cycle MAC protocol for dynamic traffic loads in wireless sensor networks," in *Proceedings of the 6th ACM Conference on Embedded Network Sensor Systems*, Raleigh, NC, November 4-7, 2008, pp. 1-14.
- [6] M. J. Miller and N. F. Vaidya, "A MAC protocol to reduce sensor network energy consumption using a wakeup radio," *IEEE Transactions on Mobile Computing*, vol. 4, no. 3, pp. 228-242, May 2005.
- [7] Y. Wei, J. Heidemann, and D. Estrin, "An energy-efficient MAC protocol for wireless sensor networks," in *Proceedings of the 21st Annual Joint Conference of the IEEE Computer and Communications Societies*, New York, NY, June 23-27, 2002, pp. 1567-1576.
- [8] Y. Wei, J. Heidemann, and D. Estrin, "Medium access control with coordinated adaptive sleeping for wireless sensor networks," *IEEE/ACM Transactions on Networking*, vol. 12, no. 3, pp. 493-506, Jun. 2004.
- [9] G. Lu, B. Krishnamachari, and C. S. Raghavendra, "An adaptive energy-efficient and low-latency MAC for data gathering in wireless sensor networks," in *Proceedings of the 18th International Parallel and Distributed Processing Symposium*, Santa Fe, NM, April 26-30, 2004, p. 224.
- [10] W. Ye, F. Silva, and J. Heidemann, "Ultra-low duty cycle MAC with scheduled channel polling," in *Proceedings of the 4th International Conference on Embedded Networked Sensor Systems*, Boulder, CO, November 1-3, 2006, pp. 321-334.
- [11] L. F. M. van Hoesel and P. J. M. Havinga, "A lightweight medium access protocol (LMAC) for wireless sensor networks: reducing preamble transmissions and transceiver state switches," in *1st International Workshop on Networked Sensing Systems*, Tokyo, Japan, June 22-23, 2004.
- [12] S. Chatterjea, L. F. W. van Hoesel, and P. J. M. Havinga, "AI-LMAC: an adaptive, information-centric and lightweight MAC protocol for wireless sensor networks," in *Proceedings of the Intelligent*

- Sensors, Sensor Networks and Information Processing Conference*, Melbourne, Australia, December 14-17, 2004, pp. 381-388.
- [13] V. Rajendran, K. Obraczka, and J. J. Garcia-Luna-Aceves, "Energy-efficient, collision-free medium access control for wireless sensor networks," *Wireless Networks*, vol. 12, no. 1, pp. 63-78, Feb. 2006.
  - [14] H. Adam, W. Elmenreich, C. Bettstetter, and S. M. Senouci, "CoRe-MAC: a MAC-protocol for cooperative relaying in wireless networks," in *Proceedings of the IEEE Global Telecommunications Conference*, Honolulu, HI, November 30-December 4, 2009, pp. 1-6.
  - [15] I. Rhee, A. Warrier, M. Aia, and J. Min, "Z-MAC: a hybrid MAC for wireless sensor networks," in *Proceedings of the 3rd International Conference on Embedded Networked Sensor Systems*, San Diego, CA, November 2-4, 2005, pp. 90-101.
  - [16] T. Zheng, S. Radhakrishnan, and V. Sarangan, "PMAC: an adaptive energy-efficient MAC protocol for wireless sensor networks," in *Proceedings of the 19th IEEE International Parallel and Distributed Processing Symposium*, Denver, CO, April 4-8, 2005.
  - [17] J. Polastre, J. Hill, and D. Culler, "Versatile low power media access for wireless sensor networks," in *Proceedings of the 2nd International Conference on Embedded Networked Sensor Systems*, Baltimore, MD, November 3-5, 2004, pp. 95-107.
  - [18] M. Buettner, G. V. Yee, E. Anderson, and R. Han, "X-MAC: a short preamble MAC protocol for duty-cycled wireless sensor networks," in *Proceedings of the 4th International Conference on Embedded Networked Sensor Systems*, Boulder, CO, November 1-3, 2006, pp. 307-320.
  - [19] A. El-Hoiydi and J. D. Decotignie, "WiseMAC: an ultra low power MAC protocol for the downlink of infrastructure wireless sensor networks," in *Proceedings of the 9th International Symposium on Computers and Communications*, Alexandria, Egypt, June 28-July 1, 2004, pp. 244-251.
  - [20] A. B. Nacef, S. M. Senouci, Y. Ghamri-Doudane, and A. L. Beylot, "A cooperative low power Mac protocol for wireless sensor networks," in *Proceedings of the IEEE International Conference on Communications*, Kyoto, Japan, June 5-9, 2011, pp. 1-6.



### Kwang-il Hwang

He has received M.S. and Ph.D. in Electronics and Computer Engineering from Korea University, Seoul, Korea, 2004 and 2007, respectively. Since 2007, he has been with the Department of Embedded Systems Engineering, at Incheon National University (formerly known as University of Incheon), Korea, where he is currently an associate professor. He served vice chair of World Embedded Software Contest 2013. He has also worked as an advisory director of Leotek Inc., since 2007. From 2013, he is also serving ICT standard map executive committee member, TTA. He is a member of IEEE, KICS, KMMS, KIPS, and KISS. His research interests include smartgrid networking, the internet of things, embedded networked systems, and machine to machine (M2M) networks.



### In Jang

He has received his B.S. Degree in the Department of Embedded Systems Engineering, School of Information Technology, from Incheon National University, Korea, 2013 and he is currently a M.S. student in the Department of Embedded System Engineering, University of Incheon, Korea. His research area is basically embedded networks and also he is interested in smart utility networks and the internet of things.