Asynchronous and Selective Transmission for DeWiring of Building Management Systems

Qinghua Luo, *Member, IEEE,* Abraham Hang-yat Lam, Dan Wang, *Senior Member, IEEE,* Dawei Pan, *Member, IEEE,* Wai-tin Chan, Yu Peng, *Member, IEEE,* Xiyuan Peng

***Abstract*—In this paper, we show a design and implementation of a (partial) wireless Building Management System (BMS). Compared to the existing wired BMS, a wireless system can be much cheaper and more flexible in deployment. There are existing studies on smart and wireless BMS. Our design differs from others as the latter usually takes a re-arch approach and develops a brand new suite of protocols. However, it can take a considerably long time for re-standardization and adoption by vendors. Our design does not intend to tear down the full suite of upper layer protocols. We thus face difficulties as we need to maintain the upper layer protocols in operation and support their data traffic. The key ideas of our approach are an asynchronous-response framework to maintain the control plane of the upper layer protocols intact, and a modular design to prioritize and schedule data flow to handle link quality and throughput variations. We implemented the proposed design into a real system and evaluated the system by comprehensive experiments with real BMS controllers and software. In addition, we conducted a field deployment by integrating our system with the BMS in FG-building of The Hong Kong Polytechnic Univ. The system operated smoothly during five-hour deployment.**

***Index Terms*—BMS, BACnet, MS/TP, Asynchronous-Response.**

I. INTRODUCTION

After a decade of research, we have a decent understanding on the designs within a wireless sensor network, e.g., OS, programming languages, routing, MAC, etc. People are now actively studying application scenarios so that wireless sensor networks can be more pervasively used.

There is another type of promising applications, i.e., using wireless sensor networks to replace or enhance existing wired urban sensor systems. Wired sensor networks have already been used for ages in such domains as building management system, industry and manufactory, to name but a few. A wireless sensor network has great advantages for its wireless communication, storage and processing power of the smart sensors; all these can lead to a system that is much cheaper, more readily deployable and flexible.

As a concrete example, there are efforts in developing a (partially) wireless Building Management System (BMS) [1]. A BMS acts as the brain of a building in controlling and monitoring its mechanical and electrical equipment (see Fig. 1). A wireless BMS system can be cheaper and more flexible [2]. For instance, when an office room is required to be partitioned into two rooms, the lighting and control systems may need to be re-positioned. If a wired system is used, we may need to redeploy the cables and this may even result in revamping walls. If a wireless system is used, such adjustment is much easier as we only need to redeploy the wireless sensing devices. When we talked to building and service engineers, they are eager to apply new wireless technologies as substantial cost savings can be expected. Another possible good thing is that if wireless smart sensors are used over the previous passive sensing system, people may easily develop new innovations into buildings, such as energy conservation schemes [3]. Even mobile phones can interact with the smart BMS and new top-up services/apps can be developed [4].

As such, many works are done [5][6] for smart wireless BMS. These proposals mainly take a re-arch approach to the current BMS. Note that existing BMS has a full suite of upper-layer standard protocols developed. For example, Building Automation and Control Networks (BACnet) [7] is used as the protocol that specifies the interaction between sensing devices (e.g., lighting, heating, air-conditioning, etc.), the Direct Digital Controllers (DDCs, i.e., the data "relays") and the operation center at the facility office. These past studies re-design or make non-trivial modifications to the upper layers. System re-arch, however, it usually takes a long time for standardization and adoption/deployment. As architecture of the new BMS is needed to be re-designed, and the new BSM is needed re-deployed, including equipments deploying, integrated wiring, system debugging and related works. It also faces possible backward compatibility problems.

In this paper, we look from a new angle by proposing a framework that can convert existing wired sensor network into wireless without changing the upper layer protocols and no

---

Manuscript received January 31, 2016; revised July 15, 2016 and September 1, 2016; accepted September 18, 2016. Paper no. TII-16-0711.

Qinghua Luo (Corresponding author) was with the Department of Computing, The Hong Kong Polytechnic University, Hong Kong, and the School of Information and Electrical Engineering, Harbin Institute of Technology at WeiHai, WeiHai, China. This work was done while Qinghua Luo was a research assistant in the Department of Computing, The Hong Kong Polytechnic University. He is also with GuangXi Key Laboratory of Automatic Detecting Technology and Instruments (GuiLin University of Electronic Technology),Shan Dong Institute of ship building technology, the State Key Laboratory of Geo-information Engineering, and State Key Laboratory of Satellite Navigation System and Equipment Technology. Email: luoqinghua081519@163.com

Abraham Hang-yat Lam was with the Department of Computing, The Hong Kong Polytechnic University, Hong Kong and MEGA Automation Ltd. Email: cshylam@163.@comp.polyu.edu.hk

Dan Wang was with the Department of Computing, The Hong Kong Polytechnic University, Hong Kong. Email:csdwang@comp.polyu.edu.hk

Dawei Pan was with the college of information and communication engineering, Harbin Engineering University,Harbin, China. Email: tianxiapdw@163.com

Yu Peng and Xiyuan Peng were with the School of Electrical Engineering and Automation, Harbin Institute of Technology, Harbin, China. Email: pony911@163.com and pxy@hit.edu.cn

intrusive extension to the existing hardware.

Our approach faces two key difficulties: 1) we need to maintain the control plane of the upper layer protocols in operation. We will show that it cannot be easily realized as protocol commands have time constraints that are difficult to meet merely using wireless links. Besides, such constraints cannot be achieved by simply increasing the bandwidth of wireless communication; and 2) the throughput and quality of wireless communications are worse than that of wires. We need to maintain the data plane of the upper layer protocols so that it can satisfy the application requirements.

In our design, we propose a novel asynchronous-response scheme to maintain the control plane of the upper layer protocol intact. We show that our scheme can achieve the same functionality to that of the wired scheme. We use modular design for wireless data plane to prioritize and schedule data transmission in case of link quality and throughput variation. In principle, we identify critical frames and send them with priority. For regular monitoring frames, we develop a transmission sequence that maximize the throughput while maintaining application fairness. We evaluate our scheme through 1) experiments using real DDCs, connected with Arduino sensors, under real building protocols and software. The experiments show the effectiveness of our asynchronous-response scheme; 2) a comprehensive simulations to show the scalability of our algorithms; and 3) a field deployment of our system, integrated into the existing BMS in FG-building of The Hong Kong Polytechnic University. The system operated smoothly in our 5-hour deployment. We release our program codes as open source in [8]. We want to comment that our scheme does not substitute the efforts on BMS system re-design; we believe these two approaches complement each other.

The remaining part of this paper proceeds as follows. In Section II-A, we discuss some BMS background and taxonomy used in this paper. We give an overview of our design in Section II-B. Section III elaborates the design details on the asynchronous-response module and the wireless data transmission modules. In Section IV, we present implementation details, which are imperative for effective system operations. We evaluate our framework in Section V and a real world deployment is shown in Section VI. In Section VII, we present related work and we conclude our paper in Section VIII.

II. BACKGROUND AND OVERVIEW

A. BMS Background and Wireless Communications in BMS

We first briefly introduce BMS architecture and building protocols (refer to Fig. 1). The BMS acts as the brain of a building in controlling and monitoring the mechanical and electrical equipments of a building. In BMS, physical data are recorded by sensing devices. These sensing devices are passive sensors (e.g., smoke detectors). To make our presentation clear, in the follows, we call them *sensing devices* the passive sensors in BMS; and *sensors* the active smart sensors that have the ability of processing, storage and communication, as widely understood by computer scientists. The sensing devices in BMS are connected to the Direct Digital Controllers (DDC). The DDCs form the hardware backbone of BMS. There are two types of DDCs: system DDCs (usually more powerful) and field DDCs (or in short, DDCs). The physical connection of the DDCs is RS-485, a physical layer standard. On top of RS-485, there is an MS/TP (Master Slave/Token Passing) protocol for DDC connections. The system DDCs are connected to the operation center using Ethernet. The protocol of BMS is BACnet, standardized by ASHRAE [7]. BACnet defines the interaction behavior of the BMS devices. There are variations in BMS architecture. We believe, however, this aforementioned architecture is one of the most widely adopted architectures.

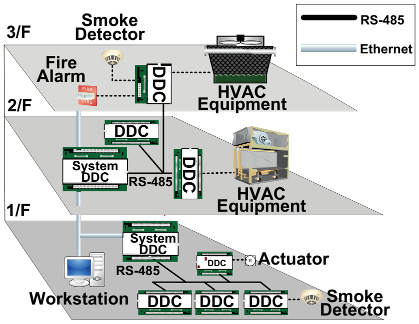


Fig. 1. The BMS architecture

Many parts of the BMS are wire-connected. For example, Ethernet is wire-connected, but it is usually well-planned along with the building construction and requires less flexibility. Converting this part into wireless is thus less fruitful.

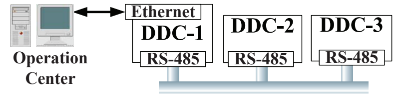
Besides Ethernet, DDCs and the sensing devices are also wire-connected. There is a large number of different sensing devices (e.g., smoke detector, thermostat, etc.). These sensing devices are passive, vendor oriented and the connection is point-to-point. Hence, converting this part into wireless is more specific and if these sensing devices are enhanced by the smart sensors, it is easier to individually convert this part into wireless and integrate into the BMS architecture.

Apart from the above-mentioned, the connections between DDCs are also wire-connected. This part is more flexible than the Ethernet section. The distance between DDCs can be long (between floors as shown in Fig. 1) and the DDCs form a subnet. Hence, converting this part into wireless has large gain, including lower cost, higher efficiency and more flexible in system deploying and integration. But it also faces non-trivial challenges. This is the focus of this paper. The protocol governing this part is MS/TP, developed specifically under BACnet by ASHRAE. Every DDC in this subnet has two roles, master or slave. There is a token in the subnet and a DDC can send data frames or command frames when it holds a token.

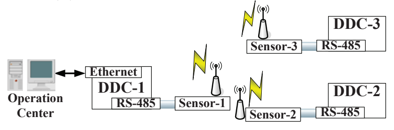
B. An Overview of the Design

The physical change made by our system is illustrated from Fig. 2 (a) to Fig. 2 (b). We attach a sensor to a DDC through RS-485. We leave more details on sensor hardware selection and development to Section IV. Note that we make no modification on the DDC hardware.

In this example, the communication from DDC-1 to DDC-2 is replaced by communication from DDC-1, relayed by Sensor-1 and Sensor-2, to DDC-2. Our experience shows that a straight forward replacement does not work. There are two problems. First, for each frame sent by DDC-1, there is a delay constraint. More specifically, if this frame is not received (or replied) within a certain amount of time, it is considered

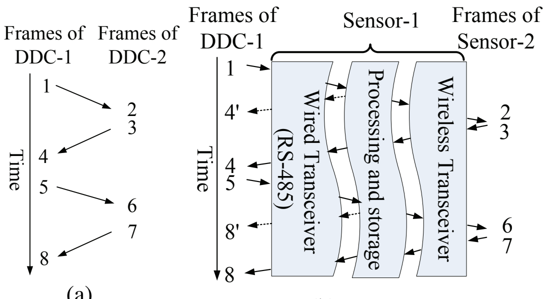


(a) Wired Connection



(b) Wireless Connection

Fig. 2. Converting wired connection to wireless



(a) (b)

Fig. 3. Asynchronous Scheme Illustration

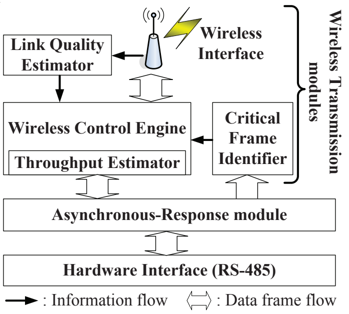


Fig. 4. A Modular Framework

expired. In MS/TP of the BMS system, this time is 10bit propagation-time on RS-485 communication. While this delay can be easily satisfied by a wired link, for a wireless link such as 802.15.4, the processing delay and the propagation delay make it impossible to meet such delay. As a matter of fact, we measured a 1500bit time delay in our experiments. Even worse, this cannot be improved by increasing bandwidth. This introduces a difficulty in maintaining the control plane of the upper layer MS/TP protocol. Second, the transmission of wireless links is slower and more unstable than wired links. Thus, the data throughput from the application layer may exceed the wireless link capacity at a certain time. This affects the data plane and data storage and scheduling are needed.

Our key proposal is a novel asynchronous-response scheme (see Fig. 3). The sensors run MS/TP protocol stack to communicate with their corresponding DDCs. For a command from DDC-1 (refer to Fig. 2), Sensor-1 will send this command to Sensor-2. In the meantime, if it needs to meet the MS/TP timing constraint, Sensor-1 will also send a valid MS/TP protocol response to DDC-1. This response is asynchronous to the request sent/received from Sensor-2. Sensor-2 will send the request to DDC-2 and then respond to Sensor-1 after receiving response from DDC-2. To the best of our knowledge, we are the first to introduce such design in our scenario.

We develop a wireless BMS framework using a modular design (see Fig. 4). The asynchronous-response module reads frames from RS-485. It passes outgoing frames to the wireless transmission modules for further process and transmission.

As discussed, the data traffic may be greater than the capacity of the wireless link at certain times. Therefore, we need to schedule data frames while satisfying the application requirements (e.g., timely update of the readings of the sensing devices, accurate critical event report, etc.). We thus have a set of wireless transmission modules to support data transmission. We have a link quality estimator module which monitors the wireless communication quality. If the link quality deteriorates, more retransmission is allocated. We have a critical frame identification module. With the understanding of application traffic pattern, we can identify critical frames and assign high priority for these traffic in case of need. We have a wireless control engine assisted by a throughput estimator. It monitors the data traffic information from other modules and makes data scheduling and transmission decisions.

III. DESIGN DETAILS

A. The Asynchronous-Response Module

We look into the details of the MS/TP protocol. The frame format is shown in Fig. 5. Each DDC has an address. In our implementation, we also give each sensor associated with the DDC an address. There are eight public commands and some proprietary commands. We do not study the proprietary commands, as they can be handled through individual vendors if necessary. For the public commands, we group them into two categories: 1) MS/TP link and system maintenance frames; and 2) data transmission frames (see Table I).

TABLE I
MS/TP FRAME SPECIFICATION AND CLASSIFICATION

Type	Name	Category
00	Token	Link maintenance
01	Poll For Master	Link maintenance
02	Reply to Poll For Master	Link maintenance
03	Test_Request	System maintenance
04	Test_Response	System maintenance
05	Data Expecting Reply	Data transmission
06	Data Not Expecting Reply	Data transmission
07	Reply Postponed	Data transmission

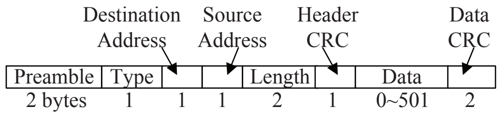


Fig. 5. MS/TP Frame format

We handle these commands in an asynchronous manner. For the sake of concise presentation, we only describe the major commands used in our design.

- **00 Token**: This command is to pass network mastership to the destination node. Only the token holding node can send data.
  In our implementation, when Sensor-1 receives **00 Token** from DDC-1, it checks if there are valid data frames asynchronously received and stored. If there exist, it sends these data frames to DDC-1. Sensor-1 returns **00 Token** to DDC-1 after it finishes sending data or if it does not have any data to send.
  If the token gets lost over the wireless transmissions, there is no token in the wireless MS/TP system, and a new

---

This is a common design for a system to get rid of outdated or lingering packets/frames. For example, in Internet routing, there is a maximum hop number constraint for each packet. A packet should be dropped if the number of hops exceeds this number.

Asynchronous designs have been used in other domains, e.g., Ajax was proposed to improve the response time of web pages by asynchronizing the user-browser communication with the browser-web server communication.

token will generate over 0.5 second according to MS/TP protocol.

- **05 BACnet Data Expecting Reply**: This command is used by master nodes to convey the data frame which expects a reply.
  In our implementation, Sensor-1 sends **07 Reply Postponed** to DDC-1 when it receives **05 BACnet Data Expecting Reply**. In the meantime, it relays this command to the sensor according to the destination address in the frame.
- **07 Reply Postponed**: This command is used by master node to defer sending a reply to a previously received BACnet Data Expecting Reply command.
  In our implementation, when Sensor-1 queries Sensor-2 and Sensor-2 queries DDC-2, Sensor-2 can receive **07 Reply Postponed** when the data of DDC-2 is not ready (For example, DDC-2 does not get data from its associated sensing devices, e.g., a thermometer). A tricky part here is that Sensor-2 will *not* send **07 Reply Postponed** to Sensor-1 as it knows that Sensor-1 has already asynchronously sent a **07 Reply Postponed** to DDC-1. Sensor-2 will also pass the token to DDC-2 by sending **00 Token**; otherwise, DDC-2 does not have the right to send data. In case that DDC-2 is still not ready, DDC-2 will pass the token back to Sensor-2 (see previous explanation on the token). The token passing continues between Sensor-2 and DDC-2 until the data are ready.

Our evaluation in Section V and VI also show that our asynchronous response module successfully supports system operation.

B. The Wireless Transmission Modules

We assume that a wireless link is slower and more unstable than a wired link. Note that we do not say that every wired application can become wireless without modification of the upper layers. If the difference between the wire and wireless communication speed is big and the application requirement is stringent, holding upper layers unchanged can be impossible.

For BMS, the data traffic (especially the averaged data traffic) is moderate. Based on our experience we often see that even the whole traffic is manageable by wireless capacity. We will use scheduling and priority to achieve smooth data transmission under traffic and link quality variation.

We classify two different traffic categories: 1) data traffic for regular monitoring of the sensing devices; and 2) data traffic for emergency report. We present our wireless transmission modules and show how these traffic are supported.

1) Critical Frame Identification: There are emergency reports in BMS. More specifically, the BACnet can define an emergency by setting a threshold for a sensing device. For example, an emergency can be defined as temperature above 140°F (60°C). When an emergency happens, a DDC will detect the emergency by its associated sensing devices. The DDC then generates emergency critical frames to report to the operation center. We develop critical frame identification algorithms (CFI) to identify these reports and these frames will be prioritized in transmission.

We present two CFI methods: determine critical frames by 1) specific data fields; and 2) frame pattern recognition.

Specific Data Fields: In the data field of an MS/TP frame, there is a special “service choice field”. For critical events, this field will be labeled to 1 (i.e., security), 2 (i.e., critical), or 4 (i.e., fire). By inspecting this data field, we can identify critical frames. Using specific data fields is simple yet we admit that this violates framework layering to certain extend as we have to inspect the data content.

Pattern Recognition: In some applications, the data field in the frame may be encrypted or the data is not allowed to open. We thus identify the critical frames using pattern recognition. We found that the data frame pattern in BMS is very regular. We show an example in Fig. 6. The frame length of the critical frames is different from that of the regular frames. This is reasonable as the traffic in BMS is regulated according to specific buildings and monitoring procedure. Thus, we develop a simple pattern recognition scheme as follows.

We use frame length as the criteria to differentiate regular frames and critical frames. Since the data pattern is correlated to individual buildings, we need a first round training for the frame lengths. In the training period, we run the system for a period of time when no critical event happens. During this period, we record the set of all regular frame lengths. In the operation period, whenever a frame has a length that is not in the set, we will mark it as a critical frame.

We will show in our experiments (Section V-B), that both CFI methods can achieve 100% accuracy.

2) Link Quality Estimation: Wireless link management has long been a research topic. We are working on a token passing protocol. Thus, we do not face serious interference and collisions. We need to handle link quality deterioration, however. The main factors that affect the link quality are distance and blockage. Since the BMS system is designed in a building, we believe that the distance can be more or less measured in advance. The blockage is caused by temporary (e.g., a few days to months) room separation, decoration, etc., where walls or Christmas trees are installed. Such blockage should be detected and transmission adjustment is required.

There are many methods to detect link quality change. Based on hardware, there are RSSI [9], LQI, SNR [10], etc. Based on software, there are PRR [11], RNP [12], ETX [13], ETF [14] etc. In our application, we need a light-weight scheme because the sensor CPUs are loaded with many tasks and their processing power is not strong for complicated schemes. We choose ETF mainly to show how this module fits in our framework to provide input for the wireless control module. Other schemes can be used as well. ETF’s advantage is that it does not require additional hardware and much computation.

The ETF is the expected number of transmissions over a forward link. It can be calculated by d_f the packet received rate (PRR) of forward link [14].

$$ETF = 1/d_f \tag{1}$$

We conducted our own indoor field test and showed the correlation between PRR and Distance (see Fig. 7). We handle link quality deterioration by increasing the number of

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TII.2016.2616110, IEEE Transactions on Industrial Informatics

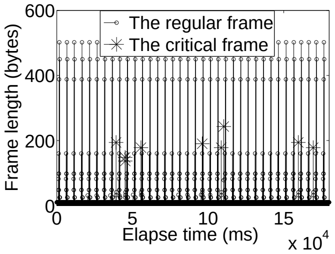


Fig. 6. Critical Frame Identification

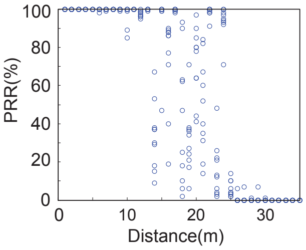


Fig. 7. The Correlation between PRR and Distance

retransmissions. We compute the number of retransmissions as follows. Let r be the number of retransmissions. Let $\mathcal{L}$ be the ETF. The ETF can be estimated mainly via asynchronous broadcast beacons. It is not drastic changes as the link distance in our system does not change unexpectedly. To compute an expected frame success rate Δ, we have

$$1 - (1/\mathcal{L})^r \geq \Delta \Rightarrow r \leq log_{1/\mathcal{L}}(1 - \Delta) \tag{2}$$

We also put a threshold $\mathcal{R}$ on r. If $r > \mathcal{R}$, the sensor indicates that the link is broken by not sending any data on this link. The operation center will not get any data frames from this DDC, and will show a broken icon on the link on the monitoring screen. This threshold is used to protect the sensors from retransmission overloads.

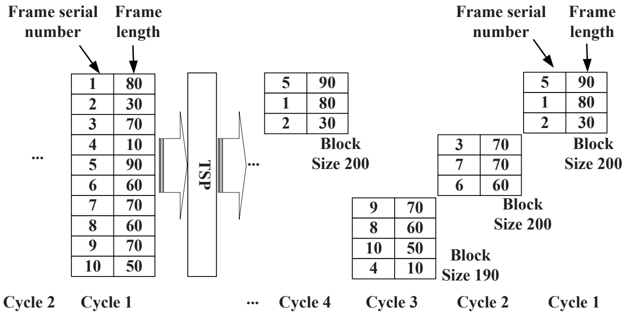
Note that if there is serious wireless link blockage in case of room renovation, even if a wired network is used, system re-deployment may also be needed. Severe building renovation (and initial building deployment) should consider the BMS restructuring. Such planning is out of the scope of this paper and worth a separate study.

3) Wireless Control Engine: The wireless control engine transmits the critical frames and the control frames directly. For the data frames, there is a throughput estimator sub-module which monitors the traffic intensity from the application layer. If the data traffic is less than the residual wireless capacity, all frames will be transmitted.

We consider the case where the data input from RS-485 is greater than the data output to wireless link. Let V_{in} be the data input speed and V_{out} be the data output speed. We delay the details on computing V_{in} and V_{out} in Section IV-C. Since $V_{in} > V_{out}$, some data have to be dropped. In BMS, this means filtering out some regular traffic. From the user application point of view, the refresh interval of the sensing devices will be increased, e.g., we refresh the thermometers every 10 seconds instead of every 2 seconds.

We consider two user requirements: 1) *fairness:* if the refresh interval needs to be increased, all the sensing devices increase equally, and 2) *importance:* there might be certain important rooms/locations that need higher refresh rate, by compromising the refresh rate of other rooms/locations.

We first study the fairness requirement. In BMS, each DDC can connect to tens or even hundreds of sensing devices. The readings of multiple sensing devices can be combined in a data frame. Since the monitoring is regular, each data frame always has the readings of the same sensing devices. Therefore, as long as the data frames are transmitted fairly, we can guarantee fairness between different sensing devices. The transmission is divided into *cycles*. Each data frame has a serial number for a cycle and their frame lengths are different (see Fig. 8 (a)). In wired transmission, each cycle transmits the same sequence. For example, in Fig. 8 (a), each cycle will transmit 10 data frames. In wireless transmission, each cycle may not have enough capacity. Thus, we need to maximally utilize the wireless communication capacity and transmit each frame with equal interval in terms of cycles. For example, in Fig. 8 (b), we cannot transmit all 10 frames in one cycle. We show a transmission schedule that each frame is fairly separated with an interval of 2 cycles.



(a) Data frame sequence of wire transmission

(b) Data Frame sequence of wireless TSP

Fig. 8. The Data Frame Transmission

We next formally show how such schedule should be developed. Let $F_i, i \in \{0, \cdots, N-1\}$ be the frames where N denote the total number of frames. Let $l_i \in \{0, \cdots, N-1\}$ be the length of frame F_i. Let L_C be the maximum bytes that a cycle can send in wireless communication. Let $\mathcal{P}$ be an arbitrary period consists of $\mathcal{C}$ cycles and L_k be the total amount of bytes transmitted in cycle k. Let $L_m = \sum_{k \in \mathcal{C}} L_k$. Let T_i be the total number of times that F_i is transmitted in $\mathcal{P}$. Let $\mathcal{T}$ be a pre-defined threshold to bound the difference of T_i, $\forall i, j, |T_i - T_j|$, i.e., the fairness.

Definition 1: The Transmission Sequence Problem (TSP): Find a transmission sequence for $F_i, i \in \{0, \cdots, N-1\}$ in any arbitrary period $\mathcal{P}$ which can be divided into $\mathcal{C}$ cycles, such that the total transmission $L_m = \sum_{k \in \mathcal{C}} L_k$ is maximized and the difference of the frames F_i transmitted is bounded by $\mathcal{T}$.

Theorem 1: TSP is NP-hard.

The proof of Theorem 1 can be found in [8].

We need an online algorithm for TSP. We first developed an offline algorithm and then extend it into an online version. Our offline algorithm follows the First Fit Decreasing (FFD) algorithm that is used for bin-packing problem [15]. The principle of FFD algorithm is to first sort all the items in a descending order, and then use a greedy method to put the items into bins. Our algorithm follows a similar principle by first putting large frames into cycles. It checks in each iteration the number of times a frame transmitted so that $\mathcal{T}$ is never violated. Our online algorithm applies the offline algorithm for each cycle, which is shown in Algorithm 1 in our technical

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TII.2016.2616110, IEEE Transactions on Industrial Informatics

report [8] .

The fairness requirement is per DDC based. For the importance requirement, if certain sections of the building need a higher refresh rate, we choose to give the DDC associated with this section a longer timeout when the token arrives at it so that it can transmit more. More specifically, let $\mathcal{N}$ be the number of DDCs. Let $p_i, i \in \{1, \mathcal{N}\}, p_i > 0$ denote the priority of the ith DDC. The lower the priority is, the longer timeout the DDC has. Let t_r be the refresh interval, we set the timeout u_i of the ith DDC to be $u_i = \frac{(1/p_i) \times t_r}{\sum_{j=1}^{\mathcal{N}} 1/p_j}$.

Lemma 1: The complexity of algorithm TSP() is O(N^2).

The proof of Lemma 1 can be found in [8].

IV. IMPLEMENTATION DETAILS

A. Sensor Connection with a DDC

DDCs use RS-485 for connection. We choose the Arduino sensors as we can use the I/O Expansion shield [16] developed by DFROBOT community, which directly supports RS-485 communication.

B. Fast Forward of Frame Transmission

For an Arduino sensor, it needs to connect from RS-485 to DDC to process MS/TP frames. In the mean time, it also needs to process wireless data frames. There is a big gap between the speed of wireless interface (which is slow) and the CPU speed (which is fast). As a result, it can take a long time if an Arduino sensor sends a frame of more than 300 bytes, such as 05 BACnet Data Expecting Reply, 06 BACnet Data Not Expecting Reply, 03 Test_Request, 04 Test_Response. During this period of time, its CPU cannot effectively process the MS/TP frame from RS-485. During our experiments, the Arduino sensor can become unstable or even malfunction if we operate data transmission frame-by-frame.

To handle this problem, we use a *fast forward strategy for frame transmission*. More specifically, the CPU sends in the granularity of each byte instead of each frame to the wireless interface. Since CPU process is much faster than wireless interface, the interval of each byte is small. From the wireless transmission point of view, its neighboring sensor still sees an integrated data frame. With fast forward frame transmission, the Arduino sensor operates reliably and effectively.

C. A Wireless Token Ring Network

In wired network, when a DDC sends a frame, it broadcasts in the physical link. After we connect each DDC with a wireless sensor, if we still use broadcast for sensor to sensor communication, we need to specially handle interference and collision. In our implementation, we use unicast between the sensors by constructing a wireless token ring among sensors. A sensor can transmit only when it has a token. We emphasize that this token ring is in a lower layer and should not be confused to the Token Passing protocol among DDCs.

We describe how our implementation computes V_{in} and V_{out} of Section III-B3. V_{out} is computed by the wireless interface speed multiplied with a piece of token time in this wireless token ring network. V_{in} is computed by the total amount of data frames in a cycle divided by cycle length time.

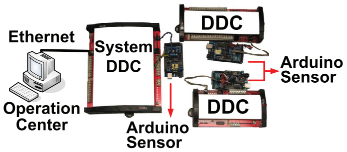


Fig. 9. The experiment environment

V. PERFORMANCE EVALUATION

We evaluate our system through experiments, simulations, and a real deployment. The experiments evaluate system functions that are difficult to simulate, e.g., the asynchronous-response module and also the system performance under real environments. The simulation shows the scalability of our algorithms and many algorithm details [8], e.g., for the wireless control engine. We also conducted field deployment of our system in the FG-building of Hong Kong Polytechnic University.

A. Experiment Setup

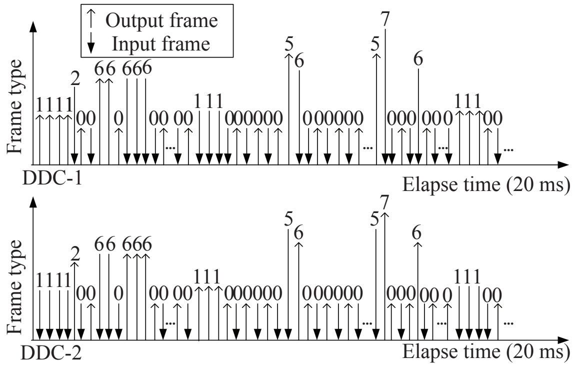
The hardware used in our experiment is shown in Fig. 9. We have three DDCs manufactured by Delta Controls Ltd., one system DDC and two field DDCs. Each DDC is connected with an Arduino Mega 2560 sensor through RS-485 as discussed in Section IV-A. The wireless module adopted in our design is XBee [16], which is based on the IEEE 802.15.4 standard. A PC is used to act as the operation center using real building management software ORCAview 3.33 [17]. The system DDC is connected with the PC using Ethernet, and it communicates with other DDCs using regular MS/TP protocol. Through ORCAview, we can monitor, manage and configure the sensing devices of the DDCs. The traffic injected into and received from system DDCs are from ORCAview, which represents real traffic of BMS. The DDC hardware and software ORCAview are all off-the-shelf products and no modification is made.

In our experiment, we put our operation center (PC) and system DDC at a fixed place. We put the three DDCs with a height of 1 meter and they formed an equilateral triangle, separated with each other by 10 meters. The default transmission power is set to 0dBm. We conducted a preliminary measurement on the link quality using different distances where the distance changes from 5 meters to 40 meters. Especially, we put one field DDC and system DDC out of sight to each other. The results are in Fig. 7 which shows a sharp decrease in the PRR.

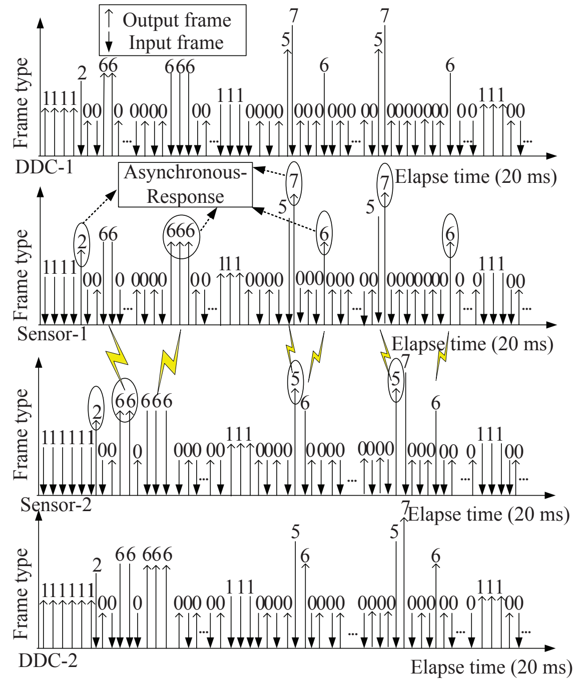
Our objectives are to evaluate the operation of our asynchronous-response module and system performance.

B. Experiment Results

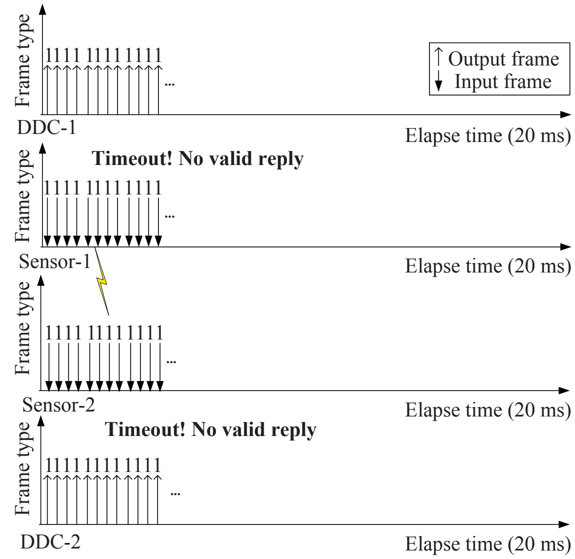
We first evaluate our asynchronous-response module. We draw three figures: 1) command sequence when wired system is used, 2) command sequence when our wireless system (with asynchronous-response) is used, and 3) command sequence when a wireless system without asynchronous-response is used. For all three figures, we perform the same operations. We see the results in Fig. 10. Each command is plotted as



(a) The frame sequence of two DDCs



(b) The frame sequence of asynchronous-response framework



(c) The frame sequence of directly wireless replacement

Fig. 10. The frame sequence of different system.

an arrow and the number shown on top of the arrow is the command type as discussed in Section III.

TABLE II
ASYNCHRONOUS-RESPONSE RATIO

Refresh time	5s	10s	30s	60s
AR ratio	2.5%	1.27%	0.64%	0.32%

In Fig. 10 (a), we show two sub-figures, one shows the commands of DDC-1 and the other one shows the commands of DDC-2. We see that the operations start with a few **01 Poll for Master** followed by obtaining a token **00 Token**, followed by a few data transmission **06 BACnet Data Not Expecting Reply**, followed by token, poll for master and again data transmission **05 BACnet Data Expecting Reply**.

In Fig. 10 (b), we see four sub-figures, each one shows the commands of DDC-1, Sensor-1, Sensor-2 and DDC-2. We label the asynchronous-response frames in circle. For example, we see that when DDC-1 sends **01 Poll for Master**, Sensor-1 replies **02 Reply Poll for Master** asynchronous to relaying this command to DDC-2 (via Sensor-2). We also see that Sensor-1 sends **07 Reply Postponed** to maintain the operation when it does not receive in time data reply from DDC-2 (via Sensor-2). It meets the timing requirement of MS/TP protocol. So the wireless MS/TP system with the proposed asynchronous-response module can operate smoothly according to MS/TP protocol.

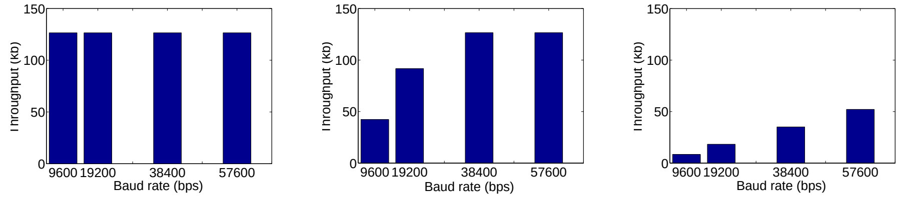
In Fig. 10 (c), we plot a comparison, where we do not use the asynchronous-response module. There are also four sub-figures. We see that the communication breaks after very few command due to no valid reply before timeout.

We also show the ratio between the number of asynchronous-response frames and MS/TP frames in Table II. We see that when the sensing device refresh time increases, the ratio decreases. This indicates that most asynchronous-response frames are used to support data frames.

We next study the performance of our system. Since the application data are generated by the DDCs, we cannot freely manage them. We adjust the wireless speed (physical speed from hardware) to simulate the imbalance between traffic from the wire to the wireless. In Fig. 11, we plot the wireless throughput under different refresh intervals of 15 seconds, 5 seconds and 1 second. Note that different refreshing time intervals represent different traffic intensity from the application layer where 1 second refreshing interval has the highest data traffic. We see from Fig. 11 (a) that the traffic throughput is the same at 126.5Kbits for all different wireless speeds. This means that the data traffic is small (126.5Kbits generated every 15 seconds) and wireless capacities are enough to transmit the data frames. In Fig. 11 (b), we see that for wireless speed 38400bps and 57600bps, the throughput remains 126.5Kbits. However, when we decrease the wireless speed, the throughput decreases. This shows that our scheme starts to adjust if the output capacity is less than the input traffic flow. We can also see that almost all capacity is used. For example, if the physical wireless speed is 19200bps, 95Kbits is transmitted which is the full capacity in a 5-second refresh interval. We see this more clearly in Fig. 11 (c) where the total application layer traffic is the highest.

In Fig. 13, we plot the real wireless throughput rate (Kb/s)

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TII.2016.2616110, IEEE Transactions on Industrial Informatics



(a) Refresh interval 15s (b) Refresh interval 5s (c) Refresh interval 1s

Fig. 11. The throughput under different baud rates

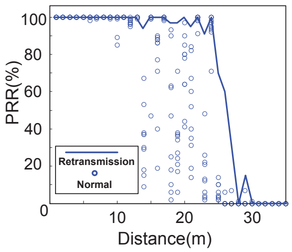
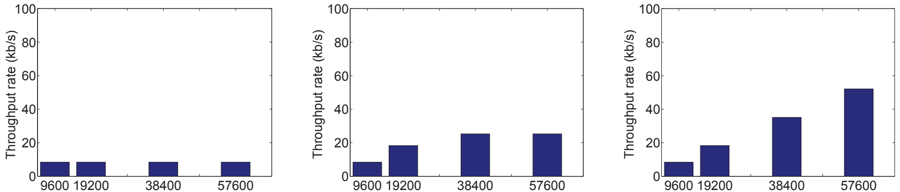


Fig. 12. The PRR under different link quality



(a) Refresh interval 15s (b) Refresh interval 5s (c) Refresh interval 1s

Fig. 13. The throughput rate under different baud rates

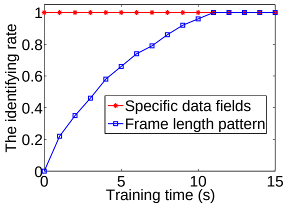


Fig. 14. CFI rate under different training time

as opposed to absolute throughput (Kb) in Fig. 11 and Fig. 13 (a)(b)(c) corresponds to Fig. 11 (a)(b)(c). We see that when there are fewer data to transmit (Fig. 13 (a)), the throughput rates are the same (84300bps) no matter which physical wireless speed is used. The throughput rate increases as the amount of data increases, but will be bounded by the physical wireless speed (Fig. 13 (b)(c)). So the physical wireless speed is an effective way to improve the throughput and throughput rate of the system.

We then study different link quality. Our system conducts retransmission as explained in Section III-B2. From Fig. 12, we see that PRR is improved. Especially, unless the link quality is extremely bad, we achieve 100% of PRR. It is mainly due to the link quality estimator module and wireless control engine module, which can maximum PRR of the wireless MS/TP system.

We next evaluate our system under critical frames. Through ORCAview 3.33, we simulate a temperature sensing device as an AI (Analog Input) port in our real DDC. We set the threshold for an alarm to be 140°F (60°C). The refresh interval is 10 seconds and we change the values of this sensing device randomly. As such, when the value is greater than the threshold, an alarm critical frame will be generated. We run our experiments for five hours and we compare our results with real results from ORCAview 3.33.

Fig. 14 shows the results of our two CFI methods: using specific data fields and pattern recognition. As the critical frame can be identified immediately through specific data field, and training is needed in the pattern recognition method. It is not surprising to see that using specific data fields can achieve 100% identification rate. When using frame length pattern recognition, the identification rate improves when training time increases. After the training time is greater than 10s, the identification rate also achieves 100%. In practice, we believe it is enough if the training time is 2-3 times of the refresh interval.

VI. A DEPLOYMENT EXPERIENCE

Our system was deployed and ran for 5 hour from 11:00am to 4:00pm. We chose this period as it was more representative in reflecting the network traffic, where most of the equipment were in operation to serve the occupants during the working hours. We deploy our system in room FG-417M (a BMS control room) of FG-building of The Hong Kong Polytechnic University. The 4th floor of FG-Building has a Learning Resource Center, a Data Communication Room, a Server Room, 3 Nursing Labs and 2 Mental Health Nursing Labs. There are 8 DDCs each of which controls a VAV (variable air volume) Box and other sensing devices. There is a DDC controlling a PAU (Primary Air-Handling Unit). These 9 DDCs are connected to a system DDC which is then connected to the BMS operation center. The configuration/topology map of the DDCs is shown in Fig. 15. Since we do not have enough hardware, we only attach two Arduino sensors to the DDC controlling the PAU and the system DDC. We show our physical deployment in Fig. 17. There are two DDCs (the other 8 DDCs are spread in other rooms) in this control room (see Fig. 16), and we connect them to our sensors. Our XBee speed is set to be 57600bps. Since the BMS was in operation to control and interact with many other types of building equipment, we were only granted by the University for this 5 hours experiment. Yet, our system ran smoothly without interrupting the normal operation of the whole BMS, and as the traffic pattern of BMS is generally stable in nature, therefore, we proved that our system can be easily integrated with the existing BMS.

In Fig. 18, we show the frame flow we captured every 15

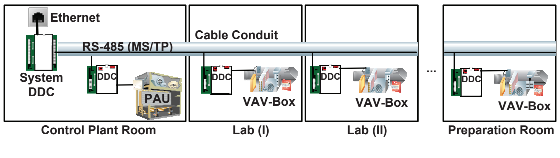


Fig. 15. The deployment of our system

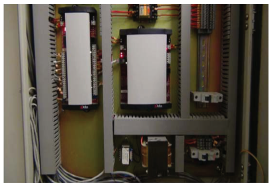
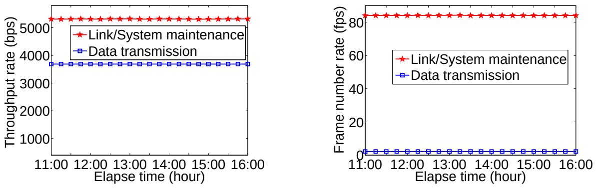


Fig. 16. Original wired M-S/TP system

Fig. 17. Wireless deployment (picture left rotated)



(a) Throughput rate of system (b) Frame rate of system

Fig. 18. Throughput rate and frame rate of the system.

seconds. We detail the traffic into data frames and maintenance frames. Fig. 18 (a) shows the amount of throughput and Fig. 18 (b) shows the number of frames. We see that 1) the traffic is stable and 2) control packets can be dominant. The total number of maintenance frames is 38.2 times greater than data frames and the throughput is 44.07% more, which is determined according to MS/TP protocol.

VII. RELATED WORK

Building system development has been an active research topic recently in wireless sensor system community. There are research focusing on energy conservation [18][1][19], and sensing system development that provides finer monitoring granularity or additional functions (e.g., to improve automation for comfort)[20][21][22].

To understanding the energy consumption and improve energy efficiency, an online questionnaire is distributed to staff, student and interviews, and then identify the trends and patterns in energy use. And the occupancy pattern is considered when control the electrical usage in building [18]. To reduce household energy consumption, a strategy through personalized thermostat recommendations is proposed in [23]. Energy harvest and energy monitoring are considered in [24]. Literature [19] described the performance and operational benefits of a large scale solar system in a building.

Literature [21] surveyed the development of building automation systems (BAS). sMAP is presented in [20] as the architecture and specification of physical information collection, which can be used for sensors, meters and actuators in building environments. There are also standardization efforts [25] for the requirements of future building automation and home automation. These efforts fall into re-arch the current BMS and may take time for standardization and adoption by vendors. Our work, on the other hand, focuses on develop a smart and wireless BMS that supports existing BMS standards.

There are abundant studies to improve wireless network throughput. A key difficulty is an accurate separation of interference, collision and link quality deterioration. A good analysis of throughput of cooperative communication is in [26] and a good related work survey can be found in [27]. More specifically, there are studies to improve throughput by stable link selection [28][9], communication rate control [29] or retransmission according to link quality changes [13]. From routing point of view, ETX [13] is used for muti-hop routing. We believe our framework can benefit from these advanced schemes as well.

VIII. CONCLUSION

In this paper, we developed a scheme which can convert existing wired building management system to a (partial) wireless system without modification on the existing building protocols. This is orthogonal and supplementary to those designs that substantially redevelop the BMS, which may take a long time to standardize and adopt by vendors. The key of our approach is an asynchronous-response scheme that can support the upper layer protocol stack and a modular framework to support data transmission. We present a full set of experiments and deployment experience. We believe such an idea and our experience can be generalized to other application scenarios beyond building management systems.

IX. ACKNOWLEDGMENT

The research presented in this paper is supported by the National Natural Science Foundation of China (61272464, 61571160, 61671174, 61601142), RGC/GRF PolyU 5264/13E, the Natural Science Foundation of Shandong Province of China (ZR2015FM027), science and technology projects of WeiHai, the Guangxi Key Laboratory of Automatic Detecting Technology and Instruments (YQ14205,YQ15203), the State Key Laboratory of Geo-information Engineering(SKLGIE2014-M-2-4), and the State Key Laboratory of Satellite Navigation System and Equipment Technology.

REFERENCES

[1] N. Tolia L. Parolini and et al. A cyber-physical systems approach to energy management in data centers. In *Proc. ACM ICCPS'10*, 2010.

[2] BACnet Unplugged ZigBee and BACnet Connect, ASHRAE Journal 2008, http://www.bacnet.org/Bibliography/AJ-6-2008.pdf.

[3] M. Ortolani A. D. Paola and et al. Intelligent management systems for energy efficiency in buildings: A survey. *ACM Computing Surveys*, 47(1), 2014.

[4] Dawei Pan, Abraham Hang-yat Lam, and Dan Wang. Poster: Carrying my environment with me in iot-enhanced smart buildings. In *Proc ACM MobiSys '13*, 2013.

[5] S. Park J.S. Byun. Development of a self-adapting intelligent system for building energy saving and context-aware smart services. *IEEE Transactions on Consumer Electronics*, 57(1):90–98, January 2011.

[6] G.P. Chiwewe T.M., Mbuya C.F.and Hancke. Using cognitive radio for interference-resistant industrial wireless sensor networks: An overview. *IEEE Transactions on Industrial Informatics*, 11(6):1466–1481, June 2015.

[7] ASHRAE standard 135-2010:A Data Communication Protocol for Building Automation and Control Networks. ASHRAE, 2010.

[8] Open source software for BMS communication using Arduino, available at http://www4.comp.polyu.edu.hk/~csdwang/Projects/wBACnet.htm, 2012.

[9] Tunicliffe M. Entezami F. and C. Politis. Find the weakest link: Statistical analysis on wireless sensor network link-quality metrics. *IEEE Vehicular Technology Magazine*, 9(3):28–38, March 2014.

[10] Fisher Silva B. and et al. Experimental link quality characterization of wireless sensor networks for underground monitoring. *IEEE Vehicular Technology Magazine*, 11 (5):1099 – 1110, 2015.

[11] M. Senel, K. Chintalapudi, D. Lal, A. Keshavarzian, and E.J. Coyle. A kalman filter based link quality estimation scheme for wireless sensor networks. In *Proc. IEEE GLOBECOM '07.*, 2007.

[12] Q. Wang Z.Q. Guo and J.He M.H. Li. Fuzzy logic based multidimensional link quality estimation for multi-hop wireless sensor networks. *IEEE Sensors Journal*, 13(10): 3605 – 3615, 2013.

[13] Douglas S. J. De Couto, Daniel Aguayo, John Bicket, and Robert Morris. A high-throughput path metric for multi-hop wireless routing. In *Proc. ACM MobiCom '03*.

[14] Lifeng Sang, Anish Arora, and Hongwei Zhang. On link asymmetry and one-way estimation in wireless sensor networks. *ACM Trans on Sensor Networks*, 6(2):12:1–12:25, March 2010.

[15] D. S. Johnson. *Near-optical bin-packing algorithms*. MIT Press, cabridge, 1973.

[16] ArduinoBoardMega2560, http://arduino.cc/en/Main/ArduinoBoardMega2560.

[17] Delta Controls Inc., http://www.deltacontrols.com/solutions-products/products.

[18] Patidar S. Gul M. S. Understanding the energy consumption and occupancy of a multi-purpose academic building. *Energy and Buildings*, 87:155 – 165, 2015.

[19] R. DANIEL S. P. Mark and W. Stephen. Performance analysis & energy benefits of a desiccant based solar assisted trigeneration system in a building. *Renewable Energy*, 15:865 – 879, 2016.

[20] S. D. haggerty X. Jiang and D. Culler. smap: simple monitoring and actuation profile. In *Proc. ACM IPSN'10*.

[21] W. Dumidu M. Milos and et al. Building energy management systems: The age of intelligent and adaptive buildings. *IEEE Industrial Electronics Magazine*, 10(1): 25 – 39, 2016.

[22] Y. Yuan H. Lam and D. Wang. An occupant-participatory approach for thermal comfort enhancement and energy conservation in buildings. In *Proc. ACM e-Energy'14*.

[23] M. Newman D. Pisharoty, R. Yang and K. Whitehouse. Thermocoach: Reducing home energy consumption with personalized thermostat recommendations. In *Proc. ACM BuildSys '15*, 2015.

[24] T.T. Ku S.C. Hsieh and et al. Broadcasting control of intelligent air conditioners using power-line-carrier technology. *IEEE Transactions on Industry Applications*, 51(2):1890 – 1896, 2015.

[25] A. Brandt, J. Buron, G. Porcu, and Telecom Italia. Home automation routing requirements in low-power and lossy networks. In *RFC-5826*, 2010.

[26] Y.Zhou and W.H. Zhuang. Throughput analysis of cooperative communication in wireless. *ACM Trans. on Wireless Communications*, 14(1):205–218, 2015.

[27] M. Xiao G. Yang, J.F. Du. Maximum throughput path selection with random blockage for indoor 60 ghz relay networks. *ACM Trans. on Communications*, 63(10):3511 – 3524, 2015.

[28] N. Binglai R. Keivan and et al. Throughput-efficient scheduling and interference alignment for mimo wireless systems. *IEEE Transactions on Wireless Communications*, 13(4):1779 – 1789, 2014.

[29] S. Guo C.Q. Luo, S.Y. Guo and et al. Green communication in energy renewable wireless mesh networks: Routing, rate control, and power allocation. *IEEE Transactions on Parallel and Distributed Systems*, 25(12):3211 – 3220, 2014.

Qinghua Luo (corresponding author) received B.S. and M.Sc. degrees in communication and information systems from Harbin Engineering University (HEU), Harbin, China, in 2005 and 2008, respectively. and the Ph.D. degree in instrument science and technology from Harbin Institute of Technology (HIT), Harbin, China, in 2013. He is currently an Assistant Professor at the school of Information and Electrical Engineering, Harbin Institute of Technology at WeiHai. His research interests include wireless sensor networks, uncertain data processing and diagnostics. He is a member of the IEEE.

Dan Wang (S05-M07) received the B.Sc degree from Peking University, Beijing, China, in 2000, the M.Sc degree from Case Western Reserve University, Cleveland, Ohio, USA, in 2004, and the Ph. D. degree from Simon Fraser University, Burnaby, B.C., Canada, in 2007; all in computer science. He is currently an associate professor at the Department of Computing, The Hong Kong Polytechnic University. His research interests include wireless sensor networks, Internet routing and applications. He is a senior member of the IEEE.

Abraham Lam received his B.Sc degree and Ph.D from the Department of Computing, The Hong Kong Polytechnic University in 2009 and 2015 respectively. He is the founder of Hong Kong MEGA Automation Ltd. and Shenzhen FUSQUARE Technology Ltd. His research interests include Internet of Things, building automation and smart cities.