ASRQ: Automatic Segment Repeat Request for IEEE 802.15.4-Based WBAN

Byoungseon Kim, *Student Member, IEEE*, Ben Lee, and Jinsung Cho, *Member, IEEE*

***Abstract*— In a wireless body area network (WBAN), high data reliability and long operating time are important requirements. The retransmission process of the default Automatic Repeat reQuest (ARQ) mechanism in IEEE 802.15.4 is a suitable method to ensure the data reliability of WBAN communications, where frame loss can occur frequently. However, retransmitting the entire DATA frame is energy inefficient due to the fact that the most of payload data within lost frames are only partially corrupted. Therefore, this paper proposes the automatic segment repeat request scheme for the IEEE 802.15.4-based WBANs. The proposed scheme partitions the data payload into segments when the channel condition is bad, and retransmits only the corrupted segment(s). This reduces the size of the retransmitted frames, which improves frame reception rate and decreases the amount of transmitted traffic, and thus energy consumption. Our experiments using a real IEEE 802.15.4-based WBAN test bed show that the proposed method provides higher transmission reliability and lower power consumption than the default IEEE 802.15.4 ARQ mechanism.**

***Index Terms*— Wireless body area network, IEEE 802.15.4, automatic request, retransmission, energy consumption.**

I. INTRODUCTION

A *WIRELESS Body Area Network* (WBAN) is a wireless communication technology that functions in, on, or around a human body for the purpose of providing medical and Consumer Electronic (CE) services [1]. To provide these services using WBANs, high data reliability, Quality of Service (QoS), and long lifetime via low-power operation are essential. Based on these requirements, IEEE 802.15 Task Group 6 (BAN) developed the IEEE 802.15.6 standard [2], which defines a MAC layer that supports several PHY layers, i.e., narrowband, Ultra-WideBand (UWB), and Human Body Communications (HBC).

Although WBAN has been standardized, it has not yet been commercialized, and thus WBAN applications are developed using existing communication standards that include IEEE 802.11, IEEE 802.15.4, and IEEE 802.15.1. More specifically, IEEE 802.15.4, which is a low-bitrate and low-power communication technology for Wireless Personal Area Networks (WPANs) [3], [4], is being considered for WBANs. A WBAN based on IEEE 802.15.4 operates in the 2.4 GHz band that is shared with other communication technologies, such as IEEE 802.11, IEEE 802.15.1, etc. Therefore, IEEE 802.15.4 suffers from frequent frame losses due to interference among different communication technologies reducing reliability and increasing energy consumption [5], [6].

To provide reliable communication, IEEE 802.15.4 adopts the Automatic Repeat reQuest (ARQ) mechanism to recover lost frames using retransmissions [3]. This retransmission process is repeated for up to *aMaxFrameRetries* defined in IEEE 802.15.4. However, the ARQ mechanism will increase energy consumption because wireless transmission has the most impact on the total energy consumption of a sensor node [7], and transmission power increases as the frame size increases. In addition, retransmission of lost frames may not improve reliability because the frame error rate depends on the frame size [8]–[10], i.e., larger frame sizes increase the likely of errors. As a result, the IEEE 802.15.4-based ARQ mechanism cannot simultaneously satisfy both high reliability and low energy consumption requirements of WBANs.

Although various ARQ techniques exist to reduce energy consumption and/or to improve data reliability, their applications are mostly focused on providing data transmissions for large-scale Wireless Sensor Networks (WSNs). Moreover, some of these techniques require high-performance processors or do not comply with the IEEE 802.15.4 standard. To the best of our knowledge, there is no work on improving the ARQ mechanism itself to reduce the energy consumption as well as to increase the reliability of a WBAN.

This paper proposes an improvement to the IEEE 802.15.4 ARQ scheme, called *Automatic Segment Repeat reQuest* (ASRQ), to reduce energy consumption as well as improve reliability. The basic idea behind ASRQ is to partition the data to be transmitted into segments and retransmit only the data segment(s) that is(are) lost, which reduces both transmission energy and frame error rate by decreasing the size of retransmitted frames. The proposed method is based on the IEEE 802.15.4 ARQ and consists of the following new features: First, a new operation is designed to provide partitioned data transmission and selective retransmission by reflecting the error characteristics of WBANs. Second, two new frame structures are introduced to support the aforementioned operation. Finally, several exception handling

Manuscript received December 16, 2016; revised February 14, 2017; accepted February 15, 2017. Date of publication March 1, 2017; date of current version April 10, 2017. This work was supported by the Basic Science Research Program through National Research Foundation of Korea (NRF) funded by the Ministry of Education under Grant NRF-2013R1A1A2059741. The associate editor coordinating the review of this paper and approving it for publication was Dr. Wan-Young Chung.

B. Kim and J. Cho are with the Department of Computer Science and Engineering, Kyung Hee University, Yongin 17104, South Korea (e-mail: ykbs0903@khu.ac.kr; chojs@khu.ac.kr).

B. Lee is with the School of Electrical Engineering and Computer Science, Oregon State University, Corvallis, OR 97331 USA (e-mail: benl@eecs.orst.edu).

Digital Object Identifier 10.1109/JSEN.2017.2676163

mechanisms are defined to deal with various conditions that can occur with the new frames. Our experimental study shows that the proposed ASRQ scheme reduces energy consumption and improves reliability compared to the default ARQ mechanism of IEEE 801.15.4.

The rest of paper is organized as follows: Section II discusses the related studies on improving reliability and reducing transmission energy. Section III presents a background on IEEE 802.15.4, and frame loss patterns in wireless data communication. The detailed operation of the proposed ASRQ scheme is presented in Section IV. Section V discusses the experimental environment and results. Finally, Section VI concludes the paper and discusses possible future work.

II. Related Works

Various techniques have been proposed to reduce the energy consumption and/or to improve the data reliability of IEEE 802.15.4-based ARQ. However, most of these techniques focus on large-scale WSNs with large amount of data transmissions without taking into account of WBAN requirements. Moreover, some of the techniques do not properly consider the compliance requirement of the IEEE 802.15.4 standard, which directly affects scalability. Therefore, this section discusses the related work on IEEE 802.15.4-based ARQ protocols.

Ganti *et al*. proposed Seda, which is a frame fragmentation technique to reduce the number of retransmissions [11]. The major functions are performed mainly by the sender. The sender transmits a Data frame composed of multiple blocks, where each block contains an evenly segmented data, a sequence number, and a checksum code. After receiving a certain number of Data frames, the receiver broadcasts a response message to request retransmission of lost blocks. The sender receiving a response message transmits the next Data, which is made up of new blocks as well as recovery blocks. Although Seda provides efficient error recovery by reducing the number of retransmissions, it can not be applied to WBAN sensor nodes that deal with periodic traffic as well as emergency traffic because fast recovery of lost blocks is not supported. If the recovery of emergency messages from a sensor is not quickly performed, the user(s) can be in jeopardy. In addition, under good channel conditions where the frame error rate is very low, the additional bytes needed to carry a sequence number and a checksum within each block increases transmission energy.

Hauer *et al.* proposed the RSSI-based bit Error Position Estimation (REPE) ARQ algorithm, which is an RSSI-based partial recovery scheme that only retransmits the sections with errors to reduce the size of retransmitted packets [12]. The receiver samples the RSSI value every 16 μs during frame receptions. After the samples are collected, the receiver estimates the error position(s) by tracing abnormal elevations in the RSSI time series. After the error estimation, a response message that includes information of the sections containing errors is broadcasted. The sender receiving the response message retransmits the recovery frame. This scheme can reduce the energy consumption by reducing the size of retransmitted frames. However, the RSSI sampling operation causes significant computation overhead and the default sampling rate is not sufficient to capture all possible interference sources [13]. In addition, the retransmission procedure does not follow the IEEE 802.15.4 standard because three of the seven reserved bits to indicate the corrupted sections in a retransmitted frame are used differently than the original purpose of the new frame type extensions. As mentioned in Section IV, our proposed scheme only utilizes the 3-bits of the unused bits to efficiently implement the frame structure that complies with the standard.

Guo *et al.* proposed the Link Quality aware ARQ (LQ-ARQ) scheme to reduce the energy consumption caused by frequent retransmissions [14]. The sender periodically transmits a sensed data and waits for a response message from the receiver. If a response is not received, the transmitted packet is stored in a buffer. The sender also estimates the link state based on RSSIs of received response messages. If the link state is good, all the stored packets in the buffer are transmitted at once. This scheme can reduce energy consumption by avoiding frequent retransmissions under a bad link state. However, this scheme is not suitable for WBANs that require real-time connectivity between the sensors and the coordinator. In addition, a lightweight sensor cannot accommodate a large buffer for bad link conditions.

Dong *et al.* proposed the Dynamic Packet Length Control (DPLC) scheme that dynamically adjusts the size of all packets without reducing the size of retransmitted packets to reduce the transmission overhead and improve energy efficiency [15]. This scheme provides two major functions. The sender first measures the transmission overhead metric after transmitting a number of packets. Then, the packet length is empirically determined to avoid errors in a noisy channel state. However, this scheme has a complex communication process and long computational delay under bad channel conditions because the packet handling process is required for individual packets (i.e., packet structure configuration, acknowledgement, and retransmission) and the increased number of transmitted packets. Moreover, it does not consider the loss of some packets among the transmitted packets, which wastes transmission energy. For example, suppose three packets (i.e., Pkt1$\sim$3) representing one message are transmitted. If Pkt2 fails despite retransmissions, the receiver cannot restore the original message, and all the received packets are discarded.

Daghistani *et al.* proposed an adaptive power mechanism called Green-Flag to improve throughput and reduce energy consumption [16]. The sender generates and transmits a block-based Data frame, which basically includes the partitioned data and checksum code. After receiving a certain number of Data frames, the receiver repeatedly broadcasts an ACK frame for a specific duration containing information of corrupted and lost blocks. After receiving a response, the sender prepares the next Data frame containing new blocks as well as recovery blocks, and transmits it with an adjusted transmission power based on the ratio of correctly received blocks and corrupted blocks. Although repeatedly broadcasting ACK frames can improve the reception rate at the sender side, this does not comply with the standard that defines sending an ACK frame only once. This also degrades the performance of neighbor nodes by consuming energy to receive responses, wasting hardware resources to store incoming frames, and decoding the frames.

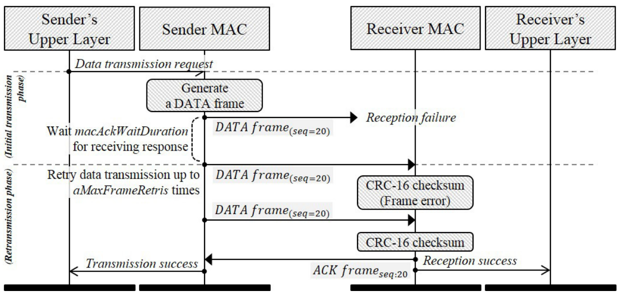

Fig. 1. The example operation of the ACK-based ARQ mechanism in IEEE 802.15.4.

In addition, the complex process of combining the new and lost blocks strains processing capability of the sensor node.

There are also a number of studies related to Hybrid ARQ (H-ARQ) that combines the ARQ and Forward Error Correction (FEC) [17], [18]. The basic idea of H-ARQ is to first try to detect and correct errors on the receiver side. If the receiver cannot correct all the errors, it requests for a retransmission. The H-ARQ can provide higher reliability than the basic ARQ under bad channel conditions, but it requires high computational and memory usage of a sensor node.

Although various IEEE 802.15.4-based ARQ schemes have been proposed to reduce the energy consumption and/or to improve the data reliability exist, they are not applicable to a WBAN environment. Therefore, a scheme that considers the WBAN requirements and is in compliance with the standard is required.

III. Background & Preliminary Experiments

A. The Fundamental Transmission Mechanism in IEEE 802.15.4

IEEE 802.15.4 transmissions are classified into two types: Acknowledgement (ACK) based (i.e., Stop-and-Wait) Automatic Repeat reQuest (ARQ) and No acknowledgement (No-ACK) based [3], [4]. The ARQ mechanism provides reliability by having the receiver send an ACK upon a successful frame reception. On the other hand, a No-ACK transmission does not guarantee data reliability because there is no follow up measures to handle lost frames. An example operation based on the ARQ mechanism in IEEE 802.15.4 is shown in Fig. 1 and described below.

The sender constructs a DATA frame when a data transmission is requested by the upper layer. A DATA frame consists of three fields: MAC frame header (MHR), frame payload, and Frame Check Sequence (FCS). The MHR contains the necessary information to transmit a frame, such as frame type, Data Sequence Number (DSN), source address, destination address, etc. The data to be transmitted is inserted into the frame payload field, and FCS contains a CRC-16 code to verify the frame's integrity. The sender transmits a DATA frame to the receiver during the *Initial Transmission* phase, and then waits for *macAckWaitDuration* to receive an ACK frame. When the sender receives the ACK frame, it reports to the higher layer that the transmission was successful. If the ACK frame is not received within *macAckWaitDuration*, the sender retransmits the DATA frame and increases the transmission attempt count. The retransmission process is repeated up to *aMaxFrameRetries* (default is 3) during the *Retransmission* phase. If the number of transmission attempts including retransmission reaches *aMaxFrameRetries*+1, the sender terminates the transmission process and reports the transmission failure to the upper layer.

The receiver performs CRC-16 checksum to verify the integrity of the received frame. If there are no errors, an ACK frame is broadcasted. On the other hand, if the checksum result is incorrect, the frame is discarded.

B. Preliminary Study on Bit-Error Patterns of Lost Frames

In order to analyze bit-error characteristics of loss frames, a simple IEEE 802.15.4-based environment was set up consisting of a sensor node and a sink node that are 5 m apart from each other. The sensor node periodically transmits a 66-byte frame to the sink node without retransmission for 2,000 times, while a PC causes WLAN interference between the two WBAN nodes by periodically transmitting data to an Access Point (AP) (see Sec. V-A). The operation of the sink node was modified to extract information of the lost frames. After the sink node receives a frame, it is stored in a buffer and checked for errors using the CRC-16 checksum. If the checksum fails, the sink node sends the frame's information to the PC, which includes payload length, payload data, number of bit errors, and error pattern type.

The bit-error pattern can be classified as either *distributed* or *partial*, which is determined as follows: The payload is first subdivided into three equal sections. If all the sections have errors, then the error pattern of this payload is categorized as distributed meaning that the entire frame has to be retransmitted. In contrast, if bit errors are clustered within either one or two sections, the error pattern is classified as partial meaning that the payload can be partially retransmitted.

Fig. 2 shows the frame status and error patterns of the received frames with payload errors, which are averages of four experiments. The X-axis represents the amount of interference traffic (KB) generated for a data transmission rate (data size/period). For example, 2K (128/500) means that an interference traffic of 2 KB is generated by transmitting 128 byte of data every 500 ms.

Fig. 2(a) shows the status of received frames, which includes the percentages of frames with normal payload, frames with payload errors, and frames that are considered lost due to PHY/MAC header errors. Under low interference levels of 2K and 10K, the percentage of frames with normal payload is over 70% while the percentage of frames with payload errors is less than 10%. As the interference traffic increases (i.e., 100K, 1M, and 2M), the percentage of frames with normal payload decreases, while the percentage of frames with payload errors increases. Fig. 2(b) shows the percentages of the two error patterns among the frames with payload error. Most of the error patterns are partial indicating that bit errors are clustered on one or two sections. This means that close to 70% of the frames with payload error can be partially recovered by subdividing the payload into three equal sections and

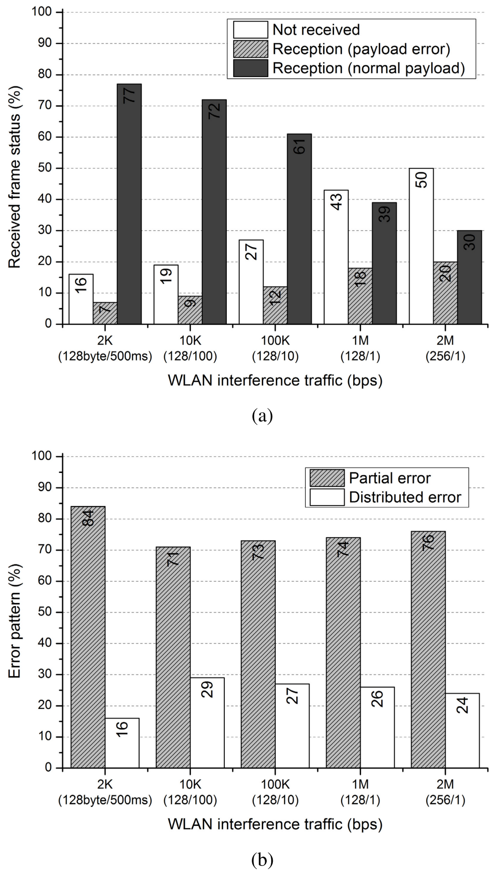

Fig. 2. The status of received frames and their error patterns. (a) Received frame status. (b) Error pattern.

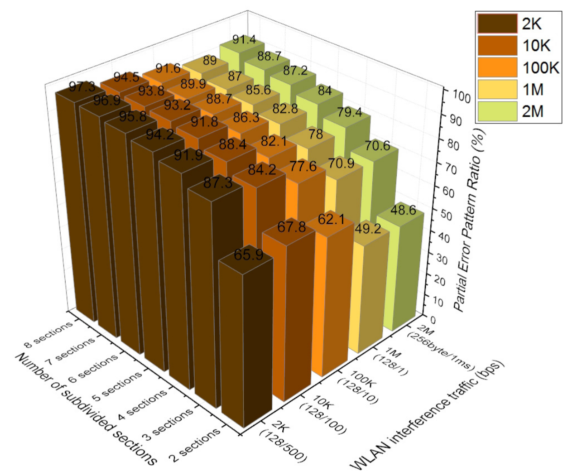

Fig. 3. The ratio of partial error patterns using varying number of sections under interference levels

retransmitting only the lost section(s), which would improve transmission success rate and energy consumption.

Additional simulations were conducted to analyze the error patterns when different numbers of sections are used under various interference conditions. The 1,000 trace data was used as WLAN interference model in the simulation. Fig. 3 shows the ratio of partial error patterns with varying number sections under different interference traffic levels. Most of the error patterns are partial, and their ratios decrease slightly as the level of interference traffic increases. In particular, the partial error pattern ratios of using three sections show on average over 15% improvement compared to using only two sections. However, as the number of sections increases from 3 to 4, their partial error pattern ratios increase slightly. Furthermore, their ratios increase minimally as the number of sections increases beyond 4. These results show that the retransmission scheme using three sections is ideal because it provides sufficient amount of coverage for partial error patterns and, as will be discussed in Section IV, it provides compatibility with the existing standard.

IV. Proposed Scheme

In addition to the Default Data transmission of IEEE 802.15.4, the proposed ASRQ method supports *Partitioned Data* transmission, where data is partitioned into *Data Segments* (DSs) each protected with its own CRC code. The Default Data transmission is used when the channel condition is stable, while the Partitioned Data transmission is performed when the channel condition is unstable. When errors occur, the size of the retransmitted frames can be reduced by only retransmitting DS(s) that had errors. Which transmission is performed is determined using the *Frame Selection* model during the Initial Transmission phase. The detailed operations of ASRQ are explained in the following subsections.

A. The Frame Type Definition

The frame structure of ASRQ is classified into five types: the DATA frame, the Partitioned Data (PD) frame, the Recovery Data (RD) frame, the ACK frame, and the NACK frame. Since these formats are based on the general frame structure of IEEE 802.15.4, ASRQ is fully backward compatible with existing IEEE 802.15.4 devices.

Fig. 4 shows the frame format definitions for the five frames of ASRQ. Fig. 4(a) shows the Frame Control Field (FCF) in the two-byte MHR containing the frame information, which is common for all five frame types of the proposed scheme. The Frame Type (FT) field classifies two groups of frames: data and response frames. The *data frames* are identified by FT = 001, and include DATA, PD, and RD frames containing the data to be transmitted. The *response frames* are indicated by FT = 010, and include ACK and NACK frames for positive or negative reception response, respectively. The IEEE 802.15.4 standard defines bits 7-9 of the FCF as reserved for future extension. The proposed scheme defines these bits as the *Loss Status* (LS) field, which is used by the receiver to request an RD frame and to distinguish between DATA and PD frames. Each bit in the LS field is sequentially mapped to each partitioned DS. Based on this mapping information encoded into NACK frames, the sender can determine which DSs within the PD frame had errors and generate an RD frame to recover these DSs. The LS field is also used together with the FT field to classify a variety of frame types, e.g., if the

	Type \ Bits	0-2 (FT; Frame Type)	3-6	7-9 (LS; Loss Status)			10-15
				DS-1 (7th bit)	DS-2	DS-3	
Data Group	DATA	001	…	0	0	0	…
	PD	001		1	1	1	
	RD	001		0/1	0/1	0/1	
Response Group	ACK	010		0	0	0	
	NACK	010		0/1	0/1	0/1	

(a)

MAC header (MHR)					Frame payload						FCS
Octets:2	1	2	2	2	Variable						2
Frame Control **(FCF)**	Seq	PAN ID	Dest Addr	Src Addr	DS-1		DS-2		DS-3		(802.15.4-based) CRC-16
					Data	CRC-8 code	Data	CRC-8 code	Data	CRC-8 code	

Each DS is mapped to each bit of the LS field

Type \ Bits	0-2 (FT; Frame Type)	3-6	7-9 (LS; Loss Status)			10-15
			DS1 (7th bit)	DS2	DS3	
PD	001	…	1	1	1	…

(b)

Octets:2	1	2	2	2	Variable (Frame payload)	2
FCF	Seq	PAN ID	Dest Addr	Src Addr	Set of DS-#(s) without CRC code	FCS

Type \ Bits	0-2 (FT; Frame Type)	3-6	7-9 (LS; Loss Status)			10-15
			DS-1 (7th bit)	DS-2	DS-3	
RD	001	…	This field cannot have values of 000b or 111b. Allow only from 1(001b) to 6(110b).			…
			0(normal)/1(error)	0/1	0/1	

(c)

Octets:2	1	2
Frame Control	Seq	FCS

(d)

Fig. 4. Frame format definition of the proposed scheme. (a) Composition of Frame Control Field (FCF) in the MAC Header (MHR). (b) PD frame structure. (c) RD frame structure. (d) ACK/NACK frame structure.

receiver receives a frame with FT=001 and LS=111, then this frame is a PD frame. Fig. 4(b) shows the format of a PD frame, which is transmitted during the Initial Transmission phase. This frame is reconstructed to an RD frame during the Retransmission phase depending on the information contained in the received NACK frame. The payload of a PD frame is composed of three DSs, where each DS contains a partitioned data and a 1-byte CRC code. To improve the performance of encoding and decoding, a table-driven CRC-8 method based on the byte-wise operation is adopted, which is faster than polynomial-based CRC-8.

The structure of the RD frame is shown in Fig. 4(c), which is used to recover lost DSs during the Retransmission phase. This frame is distinguished by FT= 001, and the LS field indicates which DS(s) is(are) being recovered. Note that an acceptable range of values for the LS field is 001∼110 because 000 and 111 are reserved for the DATA frame and the PD frame, respectively. If all the DSs in the received PD frame have errors (i.e., FT=001 and LS=111), the receiver does not response with a NACK frame and the frame is discarded. Then, the sender retransmits the frame after a timeout (see Fig. 6(a)). On the other hand, if all the DSs are properly received (i.e., FT=001 and LS=000), the receiver broadcasts an ACK frame indicating a successful transmission. In all other cases, the receiver sends a NACK frame to the sender with the LS field indicating which DSs need to be retransmitted. The sender then includes the lost DSs in an RD frame, and sends it to the receiver. Note that, unlike PD frames, the payload data in an RD frame does not contain checksum code.

The format of response (i.e., ACK/NACK) frames is shown in Fig. 4(d). An ACK frame has basically the same structure as an ACK frame in IEEE 802.15.4 (i.e., FT=010 and LS=000), and it is utilized for the purpose of responding to a successful reception of DATA, PD, or RD frame. In contrast, a NACK frame is utilized to request the retransmission of lost DS(s) using an RD frame. The possible range of values in the LS field for the NACK frames is the same as the LS field of the RD frame (i.e., 001∼110). For example, if the received frame has FT=010 and LS=000, the sender determines that this is an ACK frame. On the other hand, 001∼110 in the LS field indicates a NACK frame. When the LS field is 111, it indicates a special case where a NACK frame is not sent (see Sec. IV-D).

Note that the number of DSs supported during a Partitioned Data transmission is three. This is because the loss/recovery status of three DSs is identified using the existing 3-bit FT and 3-bit LS fields in order to comply with the IEEE 802.15.4 standard. Supporting more than three DSs would require additional data space within the payload of the PD, RD, and NACK frames. This would consume more transmission and reception energy than using three DSs, and increase the functional complexity during the Initial Transmission phase. For example, the IEEE 802.15.4 standard does not allow the ACK frame to have a payload. If more than three DSs are used, the structure and its processing requirement for the NACK frame type to contain additional data have to be newly defined because the existing ACK frame structure cannot be reused. Moreover, using three DSs leads to the size of each DS to be close to the optimal (i.e., 20∼25 bytes) for most channel conditions since the size of typical medical messages is more than 60 bytes [11], [19].

B. The Main Operation of the Proposed Scheme

As mentioned above, the main operations of the proposed ASRQ scheme are Default Data and Partitioned Data transmissions. The Default Data transmission is the same as the default transmission process using both DATA and ACK frames in IEEE 802.15.4 (see Fig. 1). On the other hand, the Partitioned Data transmission is described below.

1) The Sender Operation: The sender performs Partitioned Data transmissions when the channel is considered bad using the Frame Selection model discussed in Sec. IV-C. In order to partition the data into DSs for a PD frame, the size of the i^{th} DS, S_{DS_i}, is calculated based on the following equation:

$$S_{DS_i} = \begin{cases} \lfloor \frac{S_D+S_C}{N} \rfloor, & i = 1, 2 \\ \lfloor \frac{S_D+S_C}{N} \rfloor + ((S_D + S_C)\ mod\ N), & i = 3 \end{cases} \quad (1)$$

where S_C is the CRC size, S_D is the data size, and N is the number of available DSs (i.e., 3). If any surplus bytes occur in the process of partitioning the data, it is added to the last DS (i.e., DS_3) using the remainder calculation. Afterwards, each DS together with its checksum code is inserted into the payload field.

When the sender transmits a PD frame during the Initial Transmission phase, it waits for *macAckWaitDuration* to receive an ACK frame. If the sender receives an ACK frame within *macAckWaitDuration*, it reports the successful transmission to the upper layer. Otherwise, the sender waits for additional *macNAckWaitDuration* for a NACK frame to take into account the extra time required by the receiver to perform CRC calculation of each DS. Since this parameter depends on the hardware capability to perform the CRC calculation, it needs to be set to a minimum value required for the target environment. For example, in order to perform CRC calculations using an ATmega128A core running at 7.37 MHz, the *macNackWaitDuration* needs to be at least 350 μs. In addition, if the proposed scheme utilizes the IEEE 802.15.4-based superframe structure, this parameter has to be adjusted so that the entire transmission process can be completed within the Contention Access Period (CAP).

If the sender receives a NACK frame, it enters the *Retransmission* phase and reconstructs the PD frame to an RD frame. Which of the three DSs need to be retransmitted are determined according to the LS field in the received NACK frame. For example, if LS=010, the sender reconstructs an RD frame consisting only of DS_2. On the other hand, if the sender does not receives a NACK frame, it retransmits the PD frame. In both cases, the total number of transmission attempts is *aMaxFrameRetries*+1. If the transmission of PD frame fails, this is reported to the upper layer.

2) The Receiver Operation: The receiver operation depends on the type of frame received:

(a) In case of a PD frame reception, the receiver performs the CRC-16 calculation to verify its integrity. If the PD frame contain no errors, the receiver broadcasts an ACK frame and informs the successful data reception to the upper layer. If an error is detected, the receiver checks each DS using a CRC-8 checksum. To verify the integrity of each DS, their sizes are first calculated using the equation shown below:

$$S_{DS_i} = \begin{cases} \lfloor \frac{S_P}{N} \rfloor, & i = 1, 2 \\ \lfloor \frac{S_P}{N} \rfloor + (S_P \ mod \ N), & i = 3 \end{cases} \quad (2)$$

where S_P is the payload size of the received PD frame. After the CRC-8 checksum, the status of each DS is represented as either lost ('1') or not lost ('0'), and is indicated in the LS field of the NACK frame. For example, if the receiver detects errors in both DS_1 and DS_2, the value 110 is written to the LS field of the NACK frame and broadcasted. After broadcasting, the properly received DS_3 is temporarily buffered.

(b) In case of an RD frame reception, the receiver verifies the frame using the CRC-16 checksum. If an error is detected, the RD frame is discarded. In contrast, if the frame has no error, the receiver broadcasts an ACK frame and the previously buffered DS(s) is merged with the DS(s) of the received RD frame.

An example sequence of operations for ARSQ is illustrated in Fig. 5. During the Initial Transmission phase, the current channel condition is estimated using the Frame Selection model (see Sec. IV-C). The sender determines that the channel is good and performs a Default Data (DD) transmission. The sender transmits the first DATA frame and waits for *macAckWaitDuration*, but the receiver discards the received frame due to a CRC error. Since an ACK frame was not received before the *macAckWaitDuration* timeout, the sender enters the Retransmission phase and retransmits the frame. However, the receiver detects a frame error again. Both the second and third retransmissions are also not successful due to the same reason, and the number of transmission attempts eventually reaches *aMaxFrameRetries* + 1 (assuming *aMaxFrameRetries*=3). Consequently, the sender reports the failed transmission to the upper layer.

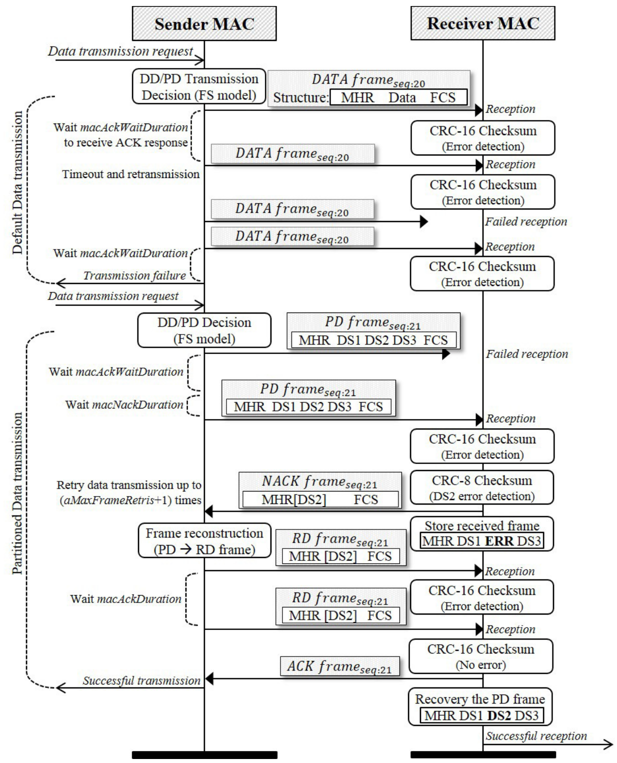

Fig. 5. The example sequence of operations for ARSQ.

Next, the sender receives another transmission request from the upper layer. During the Initial Transmission phase, the sender determines that the channel is poor and performs a Partitioned Data transmission by sending a PD frame. Although the sender waits for both *macAckWaitDuration* and *macNAckWaitDuration*, there is no response frame because the PD frame is lost. The sender then enters the Retransmission phase and retransmits the PD frame. However, the receiver detects an error in this frame. In order to confirm the integrity of DSs, a CRC checksum is performed for each DS, and consequently an error is found in DS_2. The receiver broadcasts a NACK frame that includes the loss information of DS_2 in the LS field and then both DS_1 and DS_3 are buffered. When the sender receives the NACK frame, the PD frame is reconstructed to an RD frame containing DS_2 without the CRC code. After the RD frame is retransmitted for the first time, the receiver detects a frame error, and thus an ACK frame is not broadcasted. When the *macAckWaitDuration* expires for the second retransmission, the sender retransmits the RD frame again. After the receiver confirms that the RD frame has no error, it broadcasts an ACK frame and DS_2 is recovered.

C. The Frame Selection Model

The Frame Selection model selects between Default Data and Partitioned Data transmissions. The Frame Selection model consists of two functions: (1) ACK History Queue to record the received ACK frames and (2) channel state estimation based on a two-state Markov model.

The *ACK History Queue* keeps track of DATA/PD frames transmitted during the Initial Transmission phase. The queue contains 16 (default) entries where each entry is a binary value indicating whether or not a transmitted DATA/PD frame was successful. Therefore, when the sender receives an ACK frame for a transmitted frame, '1' is inserted into the queue. In contrast, if an ACK frame is not received within *macAckWaitDuration*, '0' is inserted into the queue. The initial value in this queue is set to 0xFFFF, where MSB and LSB represent the head and tail entries of the queue, respectively. For example, suppose an ACK frame is not received for the DATA frame transmitted during the n^{th} transmission, then '0' will be inserted into the queue, and its contents will be 0xFFFE. Next, if an ACK frame is received for a PD frame during the $(n+1)^{st}$ transmission, the queue will contain 0xFFFD (i.e., 0b1111111111111101).

The sender estimates the channel state based on the result of the ACK History Queue. This is achieved using a two-state Markov model consisting of *good* (G) state and *bad* (B) state for the channel condition and transition probabilities p and q, where p is the transition probability from G to B and q is the transition probability from B to G. The probabilities for p and q are obtained using the number of transition cases from $(n-1)^{st}$ transmission to n^{th} transmission, which are defined as follows:

- N_{GG} – the number of times that n^{th} ACK was received when $(n-1)^{st}$ ACK was received.
- N_{GB} – the number of times that n^{th} ACK was not received when $(n-1)^{st}$ ACK was received.
- N_{BG} – the number of times that n^{th} ACK was received when $(n-1)^{st}$ ACK was not received.
- N_{BB} – the number of times that n^{th} ACK was not received when $(n-1)^{st}$ ACK was not received.

For example, if the ACK History Queue contains 0b1110111111101000, then N_{GG}, N_{GB}, N_{BG}, and N_{BB} are 8, 3, 2, and 2, respectively.

The transition probabilities p and q can then be calculated based on N_{GG}, N_{GB}, N_{BG}, and N_{BB} using the following equations:

$$p = \frac{N_{GB}}{N_{GG} + N_{GB}} \quad \text{and} \quad q = \frac{N_{BG}}{N_{BB} + N_{BG}}. \tag{3}$$

Then, the stationary probabilities of the state G, $P[G]$, and state B, $P[B]$, can be calculated based on p and q using the following equations:

$$P[G] = \frac{q}{p+q} \quad \text{and} \quad P[B] = \frac{p}{p+q}. \tag{4}$$

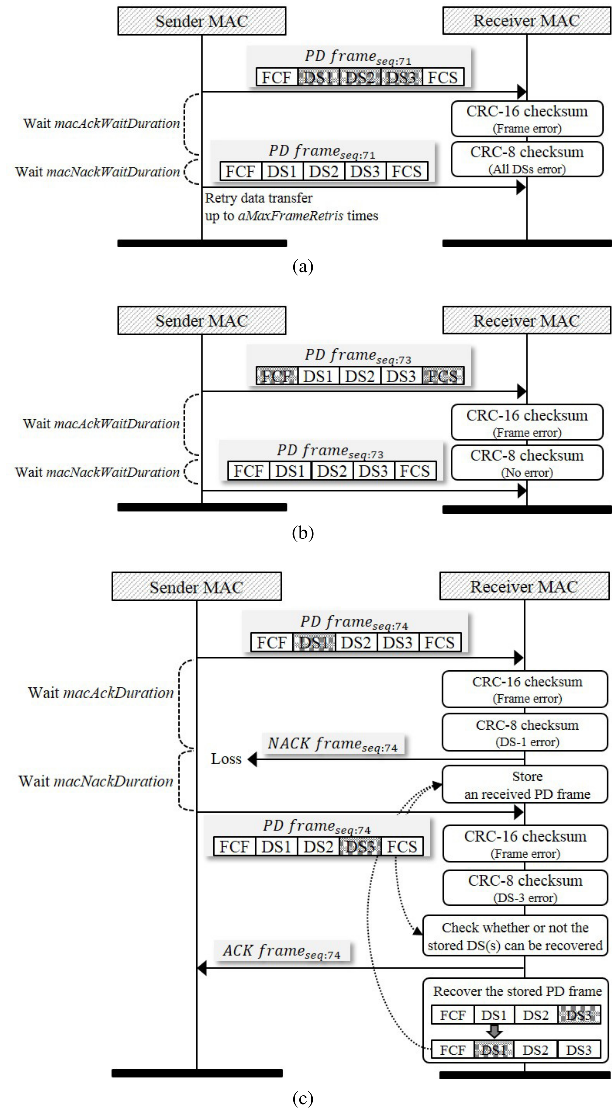

Fig. 6. Three exception handling cases of the proposed scheme. (a) All the DSs in a received PD frame have errors. (b) A received PD frame has a MAC header error, but none of its DSs have errors. (c) Duplicate receptions of erroneous PD frames with DS error(s).

Finally, the Frame Selection model compares the channel states G and B based on the stationary probabilities. If $P[B] > P[G]$, the channel is predicted to be bad and the Partitioned Data transmission is selected; otherwise, the Default Data transmission is selected.

D. Exception Handling for Proposed Frames

When a Partitioned Data transmission is performed in a real network environment, various exception situations can cause unnecessary overhead, such as additional energy consumption and data processing. These exceptions can be classified into three types and how they are handled is illustrated in Fig. 6 and explained below:

Case 1: *All the DSs in a PD frame have errors* (see Fig. 6(a)) - When only one or two DSs in a PD frame have errors, the receiver broadcasts a NACK frame to request

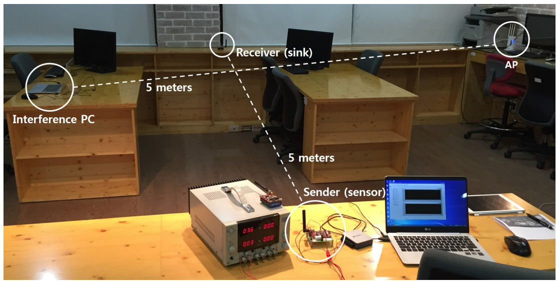

Fig. 7. The experimental environment containing a WBAN and a WLAN.

an RD frame. However, when all three DSs have errors, the receiver does not broadcast a NACK frame, and instead, the sender retransmits the PD frame after a time-out. This process is repeated until either a NACK frame is received or *aMaxFrameRetries*+1 expires. Since both PD and RD frames contain the same data, this exception handling can reduce the overhead required by the sender to reconstruct the PD frame into an RD frame.

Case 2: *An error was detected in a PD frame, but none of the DSs have errors* (see Fig. 6(b)) - In this situation, a MAC header error was detected by the CRC-16 checksum. The basic mechanism for handling an error in a PD frame is to broadcast a NACK frame to request for an RD frame. However, this causes unnecessary processing overhead since the DSs in both PD and RD frames will be identical. This overhead is eliminated by having the sender simply retransmits the PD frame after a time-out.

Case 3: *Duplicate receptions of erroneous PD frames with DS error(s)* (see Fig. 6(c)) - If the sender does not receive a NACK frame because it is lost, the PD frame is retransmitted for up to *aMaxFrameRetries* possibly causing the receiver to receive multiple PD frames with DS errors. This situation is handled by storing the first received PD frame with DS errors into a buffer, and then performing recovery with DSs included in the duplicate PD frame(s). If the DSs in the duplicate PD frame has no errors, then an ACK frame is broadcasted and the DS(s) with errors stored in the buffer is replaced with the newly received DS(s) without errors. As an example, suppose the receiver stored a PD frame that had an error in DS_1, and the NACK frame requesting DS_1 was lost. The receiver receives another PD frame, but an error is detected in DS_3. In this situation, the receiver recovers DS_1 stored in the buffer with DS_1 from the duplicate PD frame. This recovery process reduces the number of retransmission attempts by improving the frame reception rate of the receiver.

V. Performance Evaluation

A. Experimental Setup

Fig. 7 shows the experimental scenario, which consists of a WBAN and a WLAN that act as a source of interference. In the WBAN, a sensor node periodically transmits measurement data to the sink node, which is 5 m away. Meanwhile, a PC acting as interfering device sends data to a remote server (not shown) via the AP. The distance between the PC and the AP is also 5 m. An application was developed based on TCP/IP to periodically transmit data from the client and the server, and the client transmits data according to pre-defined parameters (i.e., data size and transmission period). Both the PC and the AP reside in channel 1, which does not overlap with other channel frequencies (the available channels in WLAN are 1∼ 13). However, this channel interferes with channels 11∼14 of WBAN (the available channels in WBAN are 11∼ 26)).

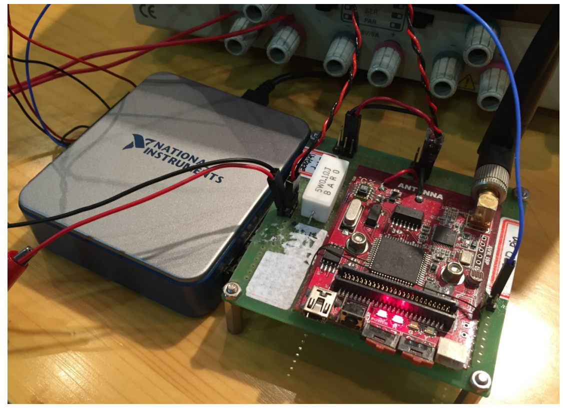

Fig. 8. The DAQ hardware (on the left) and the sensor device (on the right).

TABLE I
Parameters for the Sensor and the Sink Device

	Sensor (Sender)	Sink (Receiver)
TX power	-15 dBm (9.9 mA)	-5 dBm (13.9 mA)
TX period	500 ms	-
Payload size	64 bytes	-

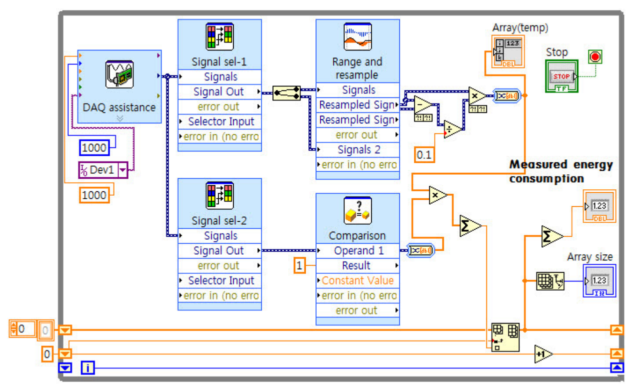

Fig. 9. The block diagram of DAQ software to measure energy consumption.

Fig. 8 shows the sensor node for the WBAN, which is a zigbee mote based on the CC2420 RF transceiver that provides compatibility with IEEE 802.15.4. The WBAN operates in channel 12. The two IEEE 802.15.4-based devices operate as a sensor node and a sink node, and the parameters for these motes are shown in Table I.

Since the sensor node is designed for low power and short range communications, the transmission power is set so that its signal would just reach the sink node. The sink node's

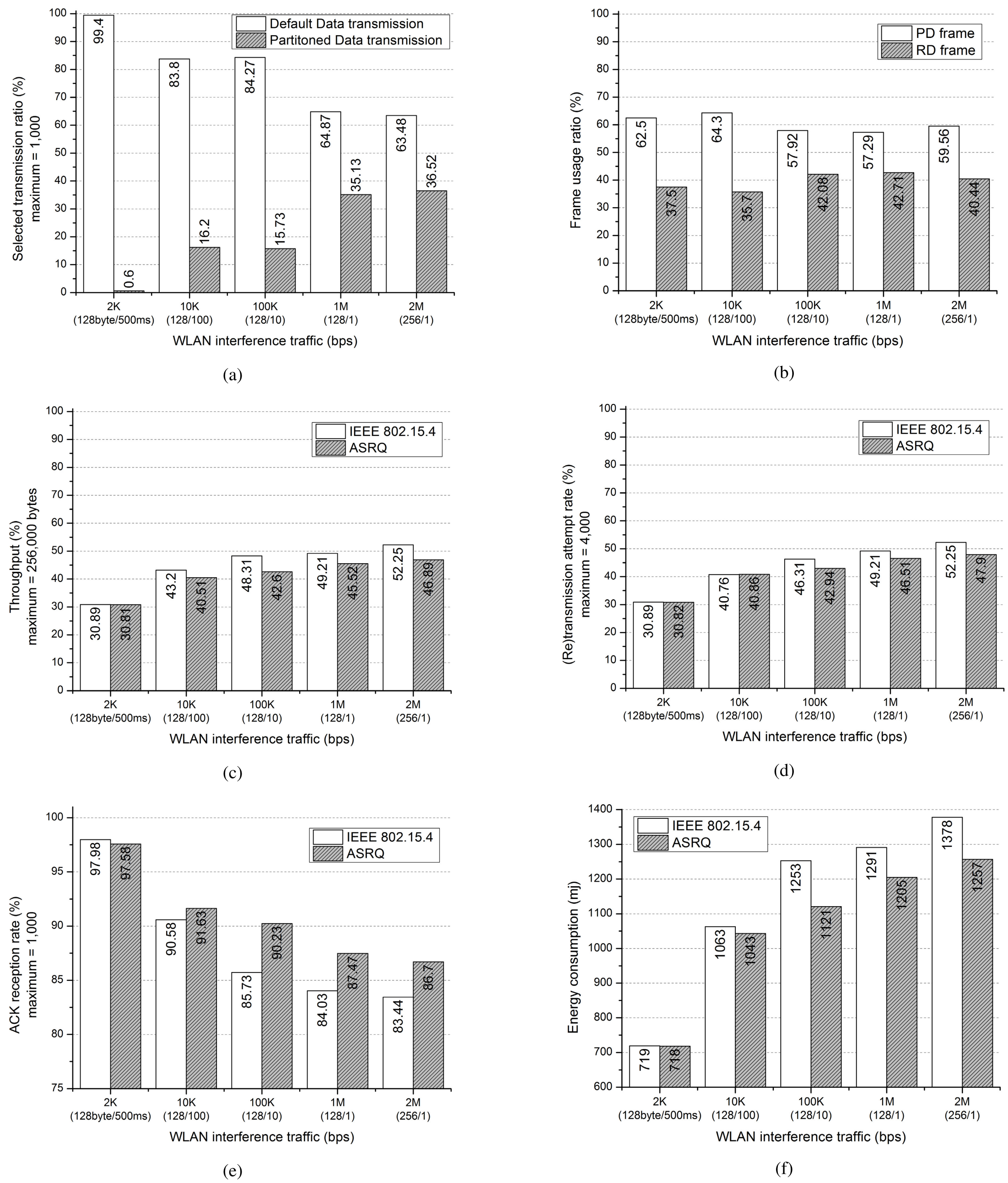

Fig. 10. The measurement results. (a) Selected transmission ratio. (b) Frame usage ratio during retransmission phase. (c) Throughput. (d) Transmission attempt rate of the sensor. (e) ACK reception rate of the sensor. (f) Energy consumption of the sensor.

transmission power is set to be higher than the sensor node assuming that it has no energy constraints, i.e., its battery can be replaced. The two nodes operate in non-beacon mode with unsoltted CSMA/CA channel access mechanism. The transmission period for the sensor node is 500 ms, and the data size is 64 bytes in accordance with anthropometric data size in personal health device standards [20]. The total number of data transmissions is 1,000.

A Data Acquisition (DAQ) hardware shown in Fig. 8 is used to measure the voltage of the sensor node, and the energy consumption is calculated by the DAQ software [21], [22]. Fig. 9 shows the block diagram of the DAQ software to calculate energy consumption.

In order to measure the power consumption (P), the voltage across a 0.1 Ω (R) resistor connected between the power supply and the sensor node is measured using the equation below:

$$V_s - V_d = I \cdot R \quad \text{and} \quad I = \frac{V_s - V_d}{R}, \tag{5}$$

where V_s is the power supply voltage and V_d is the voltage drop across the resistor. Then, the power consumption

$P = I \cdot V_s$ is calculated and stored into a temporary array in LabVIEW.

In order to accurately measure energy consumption of the sensor node, any unnecessary standby energy consumption of other modules not used in this experiment were excluded (i.e., temperature sensor, humidity sensor, bluetooth, etc.).

B. Analysis of Results

Fig. 10 shows the measurement results of the proposed scheme and the IEEE 802.15.4-based ARQ in terms of the selected transmission ratio, the frame usage ratio, throughput, the transmission attempt rate, the ACK reception rate, and the energy consumption. These results are based on the average of four measurements.

Fig. 10(a) shows the selected transmission ratio between Default Data transmission and Partitioned Data transmission using the Frame Selection model of ASRQ. Under low (2K) interference traffic, the proposed scheme operates similar to the IEEE 802.15.4 ARQ because the Partitioned Data transmission is rarely selected. But, as the interference level increases, the percentage of the time the Partitioned Data transmission is selected increases.

Fig. 10(b) shows the frame usage ratio, which represents the percentages of PD and RD frames used during the Retransmission phase. The bit-error pattern experiment discussed in Sec. III-B showed that most error patterns ($> 70\%$) are Partial errors (see Fig. 2(b)), which suggests that a large number of RD frames would be generated. However, RD frames are utilized only $\sim$40% of the time. The reason is due frequent retransmissions of PD frames caused by either loss of ACK/NACK frames or frame reception failures (i.e., MAC header errors).

Fig. 10(c) compares the sensor's throughput. Under low (2K) interference traffic, the throughput of the two methods is similar because the number of Partitioned Data and RD frame transmissions is very low for ARSQ (see Fig. 10(a)). On the other hand, the proposed scheme guarantees lower throughput than the IEEE 802.15.4 ARQ with interference traffic levels of 10K, 100K, 1M, and 2M because RD frames were appropriately utilized during the Retransmission phase.

Figs. 10(d) and 10(e) show the (re)transmission attempt rate and ACK reception rate, respectively. With 2K interference traffic, both methods show similar results because ASRQ is similar to the IEEE 802.15.4 ARQ. Likewise, the (re)transmission attempt rate of the two methods is not that different even though the Partitioned Data transmissions are used during the 10K interference traffic because most of retransmissions in the two methods occur only once. However, the proposed scheme reduces throughput by using RD frames (see Fig. 10(c)). Finally, the proposed scheme shows improved performance in terms of both (re)transmission attempts and ACK receptions for the other interference traffic levels. The reason is that both RD frames and exception handling process for duplicated frame receptions (see Fig. 6(c)) improve the frame reception rate and reduce the number of retransmission attempts.

Fig. 10(f) shows the energy consumption measured by DAQ hardware. With the exception of 2K interference traffic where Partitioned Data transmissions rarely occur, the energy consumption of the proposed scheme is reduced by using RD frames for the other interference traffic levels. In particular, the energy consumption of the proposed scheme is significantly lower for 100K, 1M, and 2M interference traffic levels due to the frequent uses of RD frames.

VI. Conclusion

This paper proposed the ASRQ scheme to improve the successful transmission rate and reduce energy consumption in IEEE 802.15.4-based WBANs. The salient features of ASRQ are (1) a new data transmission operation that partitions the payload into segments and retransmits only the segments that are lost, (2) a frame selection model that predicts when partitioned data transmission should be used, and (3) new frame structures to support the partitioned data transmission. The experimental results show that the proposed scheme significantly outperforms the IEEE 802.15.4-based ARQ mechanism.

References

[1] B. Latré, B. Braem, I. Moerman, C. Blondia, and P. Demeester, "A survey on wireless body area networks," *J. Wireless Netw.*, vol. 17, no. 1, pp. 1–18, Jan. 2011.

[2] *IEEE Standard for Local and Metropolitan Area Networks—Part 15.6: Wireless Body Area Networks*, IEEE Standard 802.15.6-2012, IEEE Computer Society, Feb. 2012.

[3] *IEEE Standard for Local and Metropolitan Area Networks-Specific Requirements—Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specification for Low-Rate Wireless Personal Area Network (LR-WPANs)*, IEEE Standard 802.15.4-2006, IEEE Computer Society, Sep. 2006.

[4] *IEEE Standard for Low-Rate Wireless Personal Area Networks (LR-WPANs)*, IEEE Standard 802.15.4–2015 (Revision of IEEE Standard 802.15.4–2011), IEEE Computer Society, Apr. 2016.

[5] C. Li, H. B. Li, and R. Kohno, "Performance evaluation of IEEE 802.15.4 for wireless body area network (WBAN)," in *Proc. IEEE Int. Conf. Commun. Workshops (ICC)*, Jun. 2009, pp. 1–5.

[6] Z. Zhao, X. Wu, X. Zhang, J. Zhao, and X.-Y. Li, "ZigBee vs WiFi: Understanding issues and measuring performances of their coexistence," in *Proc. IEEE Int. Perform. Comput. Commun. Conf. (IPCCC)*, Dec. 2014, pp. 1–8.

[7] N. Sharma, "Impact of varying packet size on multihop routing protocol in wireless sensor network," *Int. J. Adv. Stud. Comput. Sci. Eng.*, vol. 3, no. 9, pp. 10–16, 2014.

[8] A. Argyriou, A. C. Breva, and M. Aoun, "Optimizing data forwarding from body area networks in the presence of body shadowing with dual wireless technology nodes," *IEEE Trans. Mobile Comput.*, vol. 14, no. 3, pp. 632–645, Mar. 2015.

[9] J. Korhonen and Y. Wang, "Effect of packet size on loss rate and delay in wireless links," in *Proc. IEEE Wireless Commun. Netw. Conf.*, Mar. 2015, pp. 1608–1613.

[10] M. Petrova, J. Riihijarvi, P. Mahonen, and S. Labella, "Performance study of IEEE 802.15.4 using measurements and simulations," in *Proc. IEEE Wireless Commun. Netw. Conf. (WCNC)*, Apr. 2006, pp. 487–492.

[11] R. Ganti, P. Jayachandran, H. Luo, and T. Abdelzaher, "Datalink streaming in wireless sensor networks," in *Proc. 4th Int. Conf. Embedded Netw. Sensor Syst.*, Oct./Nov. 2006, pp. 209–222.

[12] J.-H. Hauer, A. Willig, and A. Wolisz, "Mitigating the effects of RF interference through RSSI-based error recovery," in *Proc. Eur. Conf. Wireless Sensor Netw. (EWSN)*, Feb. 2010, pp. 1–16.

[13] N. Baccour *et al.*, "External radio interference," in *Radio Link Quality Estimation in Low-Power Wireless Networks*. Heidelberg, Germany: Springer, 2013, pp. 21-63.

[14] C. Guo, R. V. Prasad, P. Pawelczak, and R. Hekmat, "Designing energy efficient automatic repeat request protocol in wireless sensor networks," in *Proc. 4th ACM Workshop Challenged Netw.*, Sep. 2009, pp. 35–42.

[15] W. Dong *et al.*, "Dynamic packet length control in wireless sensor networks," *IEEE Trans. Wireless Commun.*, vol. 13, no. 3, pp. 1172–1181, Mar. 2014.

[16] A. Daghistani, A. B. Khalifa, A. Showail, and B. Shihada, "Green partial packet recovery in wireless sensor networks," *J. Netw. Comput. Appl.*, vol. 58, pp. 267–279, Dec. 2015.

[17] H. Chen, R. G. Maunder, and L. Hanzo, "A survey and tutorial on low-complexity turbo coding techniques and a holistic hybrid ARQ design example," *IEEE Commun. Surveys Tuts.*, vol. 15, no. 4, pp. 1546–1566, 4th Quart., 2013.

[18] W. Dong, J. Yu, and X. Liu, "CARE: Corruption-aware retransmission with adaptive coding for the low-power wireless," in *Proc. IEEE 23rd Int. Conf. Netw. Protocols (ICNP)*, vol. 15, Nov. 2015, pp. 235–244.

[19] *Health Informatics-Personal Health Device Communication—Part 20601: Application Profile—Optimized Exchange Protocol*, IEEE Standard 11073-20601-2010, IEEE Engineering in Medicine and Biology Society, May 2010.

[20] *Health Informatics-Personal Health Device Communication—Part 10407: Device Specialization-Blood Pressure Monitor*, IEEE Standard 11073-10407-2008, IEEE Engineering in Medicine and Biology Society, Apr. 2008.

[21] J. Travis and J. Kring, *LabVIEW for Everyone: Graphical Programming Made Easy and Fun*, 3rd ed. Englewood Cliffs, NJ, USA: Prentice-Hall, Jul. 2006.

[22] R. H. King, *Introduction to Data Acquisition With LabVIEW*. New York, NY, USA: McGraw-Hill, 2008.

Byoungseon Kim received the B.S. degree in computer engineering from the BaekSeok University of Science of Technology, South Korea, in 2011, and the M.S. degree in computer engineering from Kyung Hee University, South Korea, in 2013, where he is currently pursuing the Ph.D. degree with the Department of Computer Engineering. His current research interests include wireless sensor networks and IoT device security.

Ben Lee received the B.E. degree in electrical engineering from the Department of Electrical Engineering, State University of New York, Stony Brook, in 1984, and the Ph.D. degree in computer engineering from Department Electrical and Computer Engineering, Pennsylvania State University, in 1991. His research interests include multimedia streaming, wireless networks, embedded systems, computer architecture, multithreading and thread-level speculation, and parallel and distributed systems. He was a recipient of the Loyd Carter Award for Outstanding and Inspirational Teaching in 1994, the Alumni Professor Award for Outstanding Contribution to the College and the University from the OSU College of Engineering in 2005, and the HKN Innovation Teaching Award from Eta Kappa Nu, School of Electrical Engineering and Computer Science in 2008. He has been on the program committees and organizing committee for numerous international conferences including the 2005–2012 IEEE Workshop on Pervasive Wireless Networking, and the IEEE International Conference on Pervasive Computing and Communications (PerCom). He was the Workshop Chair for PerCom 2009. He was a Guest Editor for the Special Issue on *Wireless Networks and Pervasive Computing* of the *Journal of Pervasive Computing and Communications*. He was also an Invited Speaker at the 2007 International Conference on Embedded Software and System and a Keynote Speaker at the 2014 ACM International Conference on Ubiquitous Information Management and Communication. He is currently the TPC Chair at the 15th Annual IEEE Consumer Communications and Networking Conference (CCNC 2018). He is also an Adjunct Faculty Member from the Korea Advanced Institute of Science and Technology (KAIST).

Jinsung Cho received the B.S., M.S., and Ph.D. degrees from Seoul National University, South Korea, in 1992, 1994, and 2000, respectively, all in computer engineering. He was a Visiting Researcher with the IBM T. J. Watson Research Center in 1998, and a member of the research staff at Samsung Electronics from 1999 to 2003. He is currently a Professor with the Department of Computer Engineering, Kyung Hee University, South Korea. His research interests include mobile system security, embedded security, IoT security, and sensor and body networks.