ISRMC-MAC: Implementable Single-Radio, Multi-Channel MAC Protocol for WBANs

Kunryun Cho¹, Seokhee Jeon², Jinsung Cho³ and Ben Lee⁴
¹-³Dept. Of Computer Engineering, Kyung Hee University
Yongin, Gyeonggi-do - Korea
[e-mail: whrjsfbs@gmail.com]¹
[e-mail: jeon@khu.ac.kr]²
[e-mail: chojs@khu.ac.kr]³
² School of Electrical Engineering and Computer Science, Oregon State University
Corvallis, OR 97330 - USA
[e-mail: benl@eecs.orst.edu]
*Corresponding author: Jinsung Cho

Received March 3, 2015; revised June 5, 2015; accepted October 15, 2015;
published March 31, 2016

Abstract

Wireless Body Area Networks (WBANs) have received a lot of attention as a promising technology for medical and healthcare applications. A WBAN should guarantee energy efficiency, data reliability, and low data latency because it uses tiny sensors that have limited energy and deals with medical data that needs to be timely and correctly transferred. To satisfy this requirement, many multi-radio multi-channel MAC protocols have been proposed, but these cannot be implemented on current off-the-shelf sensor nodes because they do not support multi-radio transceivers. Thus, recently single-radio multi-channel MAC protocols have been proposed; however, these methods are energy inefficient due to data duplication. This paper proposes a TDMA-based single-radio, multi-channel MAC protocol that uses the Unbalanced Star+Mesh topology to satisfy the requirements of WBANs. Our analytical analysis together experiments using real sensor nodes show that the proposed protocol outperforms existing methods in terms of energy efficiency, reliability, and low data latency.

Keywords: WBAN, multi-channel MAC, Unbalanced Star+Mesh topology, GTS allocation

A preliminary version of this paper appeared in ACM ICUIMC 2014, Siem Reap, Cambodia (best paper award). This version includes the modification of the algorithm and its concrete analysis with implementation results on ZigbeX sensor nodes. This work was supported by the Basic Science Research Program through the National Research Foundation of Korea within the Ministry of Education under Grant NRF-2013R1A1A2059741.
1. Introduction

Life expectancy has increased due to advances in medical technology. As such, many countries are faced with an aging society, and thus, using Wireless Body Area Networks (WBANs) in health care has gained a great deal of attention [1]. Fig. 1 shows an example of a WBAN, which consists of a sink node and sensor nodes. A WBAN provides continuous monitoring of body condition and real-time feedback for medical diagnosis and consultations. This is realized using tiny sensors positioned in, on, or around a human body to measure biological data and communicate with other sensor nodes operating on ISM (2.4 GHz) and MICS (402-405 MHz) bands.

There are three important requirements for WBANs, which are energy efficiency, reliability, and low latency. These requirements also apply to Wireless Sensor Networks (WSNs), and thus, many protocols developed for WSNs have been applied to WBANs. For example, WBANs initially relied on contention-based single-channel MAC protocols that were originally proposed for WSNs [2-9]. However, these techniques do not fully satisfy the requirements of WBANs because WSNs consider communications over a large space, whereas WBANs consider communications over a small space with densely deployed sensor nodes. This leads to frequent collisions resulting in high latency and energy consumption. To reduce collisions, TDMA-based single-channel MAC protocols were proposed [10-12], but they cannot guarantee low latency because nodes have to wait for their allocated time slots to communicate. The multi-channel approach mitigates this inefficiency by enabling sensor nodes to communicate with each other through multiple channels. Several multi-channel MAC protocols for WBANs have been proposed that employ either single radio or multi-radio. Fig. 2a shows a multi-radio approach that utilizes multiple channels and antennas to communicate concurrently and thus increase channel efficiency. However, this approach cannot be realized using off-the-shelf sensor devices since they do not support multiple radios. On the other hand, single-radio with multiple channels shown in Fig. 2b is already supported by off-the-shelf sensor devices. However, special topology and collision prevention mechanism are needed to trade-off between latency and energy efficiency.

In order to understand this trade-off, Fig. 3 compares the existing single-radio MAC protocols that can be categorized into TDMA-based single-channel star topology and
TDMA-based multi-channel tree topology. In these figures, all of the sensor nodes are assumed to transmit one unit data to the sink node. When a node transmits data to the sink node during its allocated time slot, the other nodes go into a sleep state, and all the nodes perform this process sequentially. However, latency increases since the nodes have to wait for their allocated time slots. On the other hand, the TDMA-based multi-channel tree topology shown in Fig. 3b can decrease latency. This is because when a node communicates with the sink node during its allocated time slot, other nodes aggregate data to their parent nodes. This reduces the number of time slots used, but data duplication occurs leading to energy inefficiency.

This paper proposes the Implementable Single-Radio Multi-Channel MAC (ISRMC-MAC) protocol that uses a TDMA-based single-radio, multi-channel approach to resolve the aforementioned problems. As mentioned before, previous single-radio, multi-channel MAC protocols trade off between energy efficiency and latency. In contrast, the proposed ISRMC-MAC protocol provides not only low latency but also high energy efficiency. This is achieved by employing a new topology called Unbalanced Star+Mesh topology, which reduces the total number of utilized time slots leading to lower latency. Moreover, it results in minimum hop transmission reducing data duplication. Thus, energy efficiency increases...
compared with existing single-radio, multi-channel MAC protocols. The proposed ISRMC-MAC protocol has been implemented on off-the-shelf sensor nodes, called ZigeX.

The rest of this paper is organized as follows. Section 2 briefly discusses the related work on multi-channel MAC protocols and Section 3 proposes the ISRMC-MAC protocol. Performance evaluation is presented in Section 4 and Section 5 concludes the paper.

2. Related Work

Multi-channel MAC protocols have received a lot of attention as a promising solution to increase throughput. However, most of the existing multi-channel based protocols have been proposed for WSNs [13], and a few of them are suitable for WBANs [14,15]. These protocols can be categorized into single-radio and multi-radio as shown in Fig. 2. Both methods provide multiple channels, but single-radio cannot provide concurrent communications because the sink node has only one transceiver. To overcome this limitation, single-radio, multi-channel MAC protocols require special topology and algorithm. The following subsections present several existing single-radio and multi-radio multi-channel MAC protocols.

2.1 Single-Radio, Multi-Channel MAC Protocols

Kim et al. proposed Y-MAC [16] that uses a frequency hopping mechanism based on broadcast and unicast periods. Each node transmits and receives data at its own time slot. However, if a node cannot transmit all of its data in a single time slot, frequency hopping is performed to transmit the remaining data. Incel et al. proposed the Multi-Channel Lightweight MAC Protocol (MC-LMAC) [17], which is based on the single-channel LMAC [18]. If the distance between sensor nodes is shorter than three hops, they are assigned to different time slots and channels. However, Y-MAC and MC-LMAC are not suitable for WBANs because they depend on a distributed architecture. Multi-frequency MAC protocol for wireless Sensor Network (MMSN) [19] improves the channel efficiency using frequency assignment and media access mechanism based on a tree topology. Hybrid MAC (HyMAC) [20] and Low-overhead Multi-channel MAC protocol (MuChMAC) [21] also employ the tree topology to solve the bottleneck problem at the sink node. If nodes have the same parent, they are assigned different time slot, and if nodes are assigned the same time slot, they are allocated to different frequencies. MMSN, HyMAC, and MuChMAC can all guarantee high throughput but the aggregation and redundant transmission of data will lead to energy inefficiency as shown in Fig. 3b.

2.2 Multi-Radio, Multi-Channel MAC Protocols

Single-radio, multi-channel MAC protocols improve channel efficiency, but they suffer from energy inefficiency because the tree topology they require leads to data duplication. To solve this problem, some protocols rely on multiple transceivers. Tree-based Multi-Channel Protocol (TMCP) [22] constructs a tree topology with the sink node representing the root. The sink node has multiple transceivers and each child node of the sink node can transmit data using a different channel. Cluster On-demand Multi-channel MAC (COMMAC) [23] is a cluster-based protocol that is organized as a three-tier architecture consisting of the sink node, aggregators, and sensor nodes. The aggregators receive data from the sensor nodes using multiple transceivers and transmit them to the sink node. Lee et al. proposed a MAC protocol [24] that uses two kinds of nodes, outbody and inbody nodes. An outbody node assigns channels to inbody nodes and receives data from inbody nodes using multiple transceivers. Kirbas et al. proposed isMAC [25] that uses a single-radio by default, but the coordinator node
can change to the multi-radio mode to improve channel efficiency when an emergency situation occurs. However, isMAC requires multiple transceivers, and thus, it cannot be realized using off-the-shelf sensor devices.

In summary, existing multi-channel MAC protocols have the following shortcomings: For single-radio approach, a trade-off exists between energy consumption and throughput. On the other hand, multi-radio approach is impracticable using commercial off-the-shelf sensor devices. In order to overcome these problems, the proposed ISRMC-MAC protocol takes the single-radio approach and employs the *unbalanced Star+Mesh topology* and TDMA-based transmission mechanism.

3. The proposed ISRMC-MAC Protocol

There are three kinds of nodes in ISRMC-MAC: the sink node, Data Aggregation nodes (*DA_nodes*), and Leaf nodes (*L_nodes*). The sink node and *DA_nodes* form a mesh topology and have *L_nodes* as their leaves in a star topology. This hybrid topology is referred to as ‘*unbalanced Star+Mesh*’ because the number of leaf nodes that *DA_nodes* may have can be different. The topology construction algorithm is described in Subsection 3.1. As for the transmission mechanism, *L_nodes* first transmit data to their parents, i.e., *DA_nodes*. Then, *DA_nodes* aggregate the received data with their own data and transmit them to the sink node through the mesh network. Fig. 4 illustrates the overall process of ISRMC-MAC assuming that each sensor node transmits one unit of data, and the number of channels and the number of sensor nodes are 4 and 10, respectively. In addition, the circled numbers indicate the starting time slots for the messages and different arrows represent different channels. Note that the number of channels and the number of nodes in Fig. 4 are equivalent to those in Fig. 3b. Therefore, the total amount of data and the number of time slots required for transmission in Fig. 4 are reduced compared with Fig. 3b. This indicates that ISRMC-MAC improves communication throughput and energy efficiency.

The following subsections discuss the construction of the *Unbalanced Star+Mesh topology* and the Guaranteed Time Slots (GTS) allocation algorithm.

3.1 Unbalanced Star+Mesh topology construction
Table 1 defines the notations used throughout this subsection. Fig. 5 shows the flowchart for the construction of the Unbalanced Star+Mesh topology. Moreover, it is assumed that all the nodes can communicate with each other using their lowest transmission power since they are placed in/on/around human body.

The network consists of the sink node $n_0$, $N$ sensor nodes ($n_1$, $n_2$, ..., $n_N$), and $C$ channels ($ch_1$, $ch_2$, ..., $ch_C$). Furthermore, the sink node's receiving channel ($rc_0$) is $ch_1$. The first step involves the selection of $DA_nodes$. For each $n_i$, where $i$ is less than $C$, its node type ($t_i$) becomes $DA_node$ and the number of child nodes it has ($cn_i$) is initialized to 0. In addition, its receiving channel ($rc_i$) and sending channel ($sc_i$) are set to $ch_{i+1}$ and $ch_1$, respectively. Finally, its destination node ($des_i$) becomes $n_0$ since the destination node for $DA_nodes$ is the

**Table 1. Notation used in Fig. 5.**

<table>
<thead>
<tr>
<th>Notation</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>$C$</td>
<td>Number of channels</td>
</tr>
<tr>
<td>$N$</td>
<td>Number of nodes</td>
</tr>
<tr>
<td>$n_0$</td>
<td>Sink node</td>
</tr>
<tr>
<td>$n_i$</td>
<td>Sensor node $i$ ($n_1$–$n_N$)</td>
</tr>
<tr>
<td>$ch_j$</td>
<td>Channel $j$ ($ch_1$–$ch_C$)</td>
</tr>
<tr>
<td>$t_i$</td>
<td>Node type of $n_i$</td>
</tr>
<tr>
<td>$pm_i$</td>
<td>Parent node of $n_i$</td>
</tr>
<tr>
<td>$cn_i$</td>
<td>Number of child nodes for $n_i$</td>
</tr>
<tr>
<td>$sc_i$</td>
<td>Sending channel of $n_i$</td>
</tr>
<tr>
<td>$rc_i$</td>
<td>Receiving channel of $n_i$</td>
</tr>
<tr>
<td>$des_i$</td>
<td>Destination of $n_i$</td>
</tr>
</tbody>
</table>
The second step involves the selection of \( L_{nodes} \). This is done for each node \( n_i \), where \( i \) greater than or equal to \( C \) and less than \( C + MMAXC \). \( MMAXC \) represents the maximum multiple number of channels, which is less than \( N - (C - 1) \), and expressed as \( \lceil (N - (C - 1))/C \rceil \times C \). For each of these nodes, its node type (\( t_i \)) is set to \( L_{node} \) and its parent node (\( p_{ni} \)) is set to \( n_i \% C \). These \( L_{nodes} \) then join the mesh network that consists of the sink node and leaf nodes \( n_1 \sim n_{C-1} \). As each \( L_{node} \) joins the mesh network, the number of child nodes its parent node has (\( c_{n_{ni}} \)) is increased by one. In addition, its sending channel (\( s_{ci} \)) and destination (\( d_{si} \)) are set to the receiving channel of its parent node (\( r_{ci} \)) and its parent node (\( n_{pni} \)), respectively. Note that if there are more than one \( DA_{nodes} \) ready to transmit data to the sink node at the same time, data transmission needs to be performed sequentially causing latency to increase. In this case, \( DA_{nodes} \) collect data from each other via the mesh topology. Thus, their sending channel (\( sc_i \)) and destination (\( des_i \)) need to be changed, which will be explained in detail in Section 3.2.

The final step is the selection of the rest of \( L_{nodes} \). This is done for each node \( n_i \), where \( i \) is larger than or equal to \( C + MMAXC \) and less then or equal to \( N \). For each these nodes, its node type (\( t_i \)) is set to \( L_{node} \) and its parent node (\( p_{ni} \)) is set to \( n_i \% (C-2)+1 \). The remaining nodes join the \( DA_{nodes} \) except the last \( DA_{node} \) (i.e., \( n_{C-1} \)). This is done to reduce data aggregation and to exploit parallelism, and will be explained shortly. The rest of the information \( c_{pni}, sc_i, \) and \( des_i \) are determined the same way as the second step.

**Fig. 6** illustrates the reason why the last \( DA_{node} \) is excluded from the selection of the remaining \( L_{nodes} \). If the remaining \( L_{nodes} \) are assigned to the last \( DA_{node} \) as shown in **Fig. 6a**, the sink node will be idle while nodes \( n_1, n_2, \) and \( n_3 \) receive data from nodes \( n_8, n_9, \) and \( n_{10} \), respectively. More specifically, during the first time slot, \( n_4, n_5, n_6, \) and \( n_7 \) transmit their data during the first time slot in each channel to \( n_0, n_1, n_2, \) and \( n_3, \) respectively. During the second time slot, \( n_8, n_9, n_{10} \) transmit to \( n_1, n_2, n_3, \) respectively. During the third time slot, \( n_4 \) delivers the aggregated data from \( n_1, n_5, \) and \( n_8 \) to the sink node, and \( n_2 \) delivers the aggregated data from \( n_2, n_6, \) and \( n_9 \) to \( n_3 \). During the final time slot, \( n_3 \) delivers the aggregated data from \( n_2, n_3, n_6, n_7, n_9, \) and \( n_{10} \) to the sink node. Therefore, a total 19 units...
of data are transmitted. However, in our proposed topology shown in Fig. 6b, \( n_3 \) can deliver the aggregated data to \( n_7 \) during the second time slot, while \( n_1 \) and \( n_2 \) aggregate data from their \( L_nodes \). Thus, the total data size can be reduced to 16 units in the proposed method.

Fig. 7 shows the construction of an example Unbalanced Star+Mesh topology for \( C = 4 \) and \( N = 10 \). In the first step, sensor nodes \( n_1 \) through \( n_3 \) are selected as \( DA_nodes \). This is done by setting the receiving channels for nodes \( n_1 \), \( n_2 \), and \( n_3 \) to \( ch_2 \), \( ch_3 \), and \( ch_4 \), respectively. Moreover, the sending channels for the \( DA_nodes \) are set to \( ch_1 \), and the destination is set to \( n_0 \). In the second step, the sink node and \( DA_nodes \) are each assigned the same number of \( L_nodes \). Thus, the sensor nodes \( n_4 \) is assigned to the sink node \( n_0 \), and sensor nodes \( n_5 \), \( n_6 \), and \( n_7 \) are assigned to \( DA_nodes \) \( n_1 \), \( n_2 \), and \( n_3 \), respectively. In addition, the sending channels for nodes \( n_4 \), \( n_5 \), \( n_6 \), and \( n_7 \) are set to \( ch_1 \), \( ch_2 \), \( ch_3 \), and \( ch_4 \), respectively. In the last step, the remaining sensor nodes \( n_{C+MMNC} \) through \( n_N \) are sequentially assigned to \( DA_nodes \) except the last \( DA_node \) (i.e., \( n_{C-1} \)), which is done to reduce data aggregation and exploit parallelism.

The proposed method also considers the residual energy of nodes. As can be seen in Fig. 6, \( DA_nodes \) consume more energy than \( L_nodes \) since they aggregate data from \( L_nodes \) and transmit it to the sink node. In contrast, \( L_nodes \) only transmit data to a \( DA_node \). This imbalance in energy consumption requires the role between \( DA_node \) and \( L_node \) to be swapped, and thus, the topology construction in Fig. 5 is performed periodically. The sink node collects the residual energy of all the nodes and the node number in Fig. 5 is given in descending order of its residual energy.

### 3.2 GTS Allocation Algorithm

The proposed GTS allocation algorithm can be applied to any time slot TDMA-based protocols; however, our discussion is based on IEEE 802.15.4 packet frame and superframe...
structure [26]. The superframe structure is shown in Fig. 8, which is composed of Active and Inactive periods. The Active period is subdivided into beacon period, Contention Access Period (CAP), and Contention Free Period (CFP). During the beacon period, the sink node broadcasts a beacon signal to synchronize all the sensor nodes. After the beacon period, all the nodes except the sink node communicate data during the CAP based on the slotted CSMA/CA. During the CFP, nodes use their Guaranteed Time Slots (GTSs) to transmit data without collisions. The Inactive period is the same as the sleep period. The CFP can reduce unnecessary collisions and retransmissions, which are the causes of energy inefficiency. Thus, the proposed ISRMC-MAC protocol employs only the beacon period and the CFP in the superframe structure.

The overall process of the GTS allocation is as follows: The first part of the GTSs in each channel is allocated to $L_{nodes}$ for transmitting data to their parent nodes (i.e., $DA_{nodes}$). The next part of the GTSs is used for $DA_{nodes}$ to deliver the aggregated data either to the sink node in the tree topology or to the other $DA_{nodes}$ in the mesh topology. Table 2 shows the notations used in the GTS allocation algorithm.

Fig. 9 shows the flowchart of the GTS allocation algorithm for $L_{nodes}$. First, the time slots from the beginning of the CFP are allocated to $L_{nodes}$ that were selected during the second step of the Unbalanced Star+Mesh topology construction shown in Fig. 5. This is

Fig. 9. GTS allocation for $L_{nodes}$.

Table 2. Notation used in Fig. 9 and Fig. 10.

<table>
<thead>
<tr>
<th>Notation</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>$t_s$</td>
<td>GTS time slot</td>
</tr>
<tr>
<td>$s_i$</td>
<td>GTS time slot number of $n_i$</td>
</tr>
<tr>
<td>$DA_S$</td>
<td>A set of $DA_{nodes}$ for GTS allocation</td>
</tr>
<tr>
<td>$n_{HCN}$</td>
<td>Node in $DA_S$ with maximum number of child nodes</td>
</tr>
<tr>
<td>$n_{LCN}$</td>
<td>Node in $DA_S$ with minimum number of child nodes</td>
</tr>
</tbody>
</table>
done for each \( n_i \), where \( i \) is greater than equal to \( C \) and less than \( C + MMNC \). For each node that satisfies this requirement, the 1st time slot (\( ts \)) is assigned as the time slot number for \( n_i \) (\( s_i \)). Although \( ts \) starts with 0, it is incremented by one (i.e., the first time slot) in the first iteration because \( i \) starts with \( C \) (i.e., \( i = C \)). After these \( L \) nodes have been assigned to the 1st time slot, \( ts \) is incremented by one. This process repeats until there are less than \( C \) nodes remaining. Next, the remaining \( L \) nodes that were selected during the last step in Fig. 5 are allocated to the time slot. Since the remaining \( L \) nodes are assigned to \( DA \) nodes except the last \( DA \) node (i.e., nodes \( n_1, n_2, \ldots, n_{C-2} \)), both \( n_{C+MMNC} \) and \( n_{C+MMNC+C-2} \) are assigned to \( n_1 \). Therefore, \( ts \) is increased by one when \( i \) is equal to either 0 or \( C - 2 \).

The flowchart for the GTS allocation for \( DA \) nodes is given in Fig. 10. The algorithm starts with \( ts \) initialized to \( s_{C+MMNC} \) and an empty set of \( DA \) nodes (\( DA_S \)). In the first step, a check is made to determine whether each \( DA \) node has aggregated all the data from its \( L \) nodes. This is done by checking whether the number of child nodes it has (\( cn_i \)) is equal to the previous time slot \( ts - 1 \). If so, the \( DA \) node has aggregate data from all of its \( L \) nodes, and the node \( n_i \) is added to the set of \( DA \) nodes (\( DA_S \)). This process is repeated as long as \( i \) is less than \( C \).

In the second step, if \( DA_S \) is not empty, it means one or more \( DA \) nodes are ready to send data to the sink node at time slot \( ts \), thus conflicts may occur because the sink node uses just a single channel. Therefore, the \( DA \) node that has the largest number of child nodes among
The third step involves the GTS allocation on the mesh topology. If there are two or more nodes in DA_S, communication parallelism among the DA_nodes can be utilized on the mesh topology. In other words, DA_nodes in DA_S may aggregate data among them. To do this, \( n_{HCN} \) and \( n_{LCN} \) are first selected, and then \( n_{LCN} \) delivers its data to \( n_{HCN} \). Therefore, the destination node \( d_{L} \) and the sending channel \( s_{L} \), which were determined by the topology construction algorithm, need to be changed to \( n_{HCN} \) and \( r_{HCN} \), respectively. This is achieved by the following set of operations: The time slot number for the node with the minimum number of child nodes (\( s_{L} \)) is set to the current time slot (\( ts \)), the destination for \( n_{LCN} \) (\( d_{L} \)) is set as \( n_{HCN} \), the sending channel of \( n_{LCN} \) (\( s_{L} \)) is set to the receiving channel of \( n_{HCN} \) (\( r_{HCN} \)), and the maximum number of child nodes for \( n_{HCN} \) (\( c_{HCN} \)) is increased by the number of child nodes for \( n_{LCN} \) (\( c_{LCN} \)) plus one. Finally, \( n_{HCN} \) is moved to \( temp \) and both \( n_{HCN} \) and \( n_{LCN} \) are removed from DA_S. This process is repeated until the number of elements in DA_S is less than 2. When this occurs, the node in \( temp \) is added to DA_S and the time slot is incremented by one. Then, the entire process is repeated until DA_S becomes empty.

Fig. 11 illustrates an example of data transmission of DA_nodes in the Unbalanced Star+Mesh topology. In this figure, DA_S = \{n_1, n_2, n_3\} after the first step, \( n_3 \) is selected as \( n_{HCN} \) in the second step, and its destination is the sink node. In the third step, \( n_3 \) and \( n_2 \) are

1 When there are ties, the DA_node that has the smallest number wins.
selected as $n_{HCN}$ and $n_{LCN}$, respectively, and then $n_2$ transmits its data to $n_3$ while $n_4$ delivers its data to the sink node.

Fig. 12 depicts an example of GTS allocation process for the Unbalanced Star+Mesh topology constructed in Fig. 7. Fig. 12a shows the GTS allocation for the $L$ nodes. This information can also be used to determine when a $DA$ node has aggregated data from its $L$ nodes. For example, when $ts = 2$, only $n_3$ has aggregated all the data from its $L$ nodes since $cn_3$ is 1, and thus, $DA_S = \{n_3\}$ after the first step. Fig. 12b shows the GTS allocation for the $DA$ nodes. For example, when $ts = 2$, the time slot is allocated to $n_3$ based on step (2) of Fig. 10 because $DA_S = \{n_3\}$. After the second step, $DA_S$ is empty and thus the first step of the algorithm repeats with $ts = 3$. Similarly, the third and forth time slots are allocated to $n_2$ and $n_1$, respectively, as shown in Fig. 12b.

Fig. 12 however does not cover the case of step (3) in Fig. 10. In order to explain this step, another example shown in Fig. 13 will be used. After the GTS allocation for $L$ nodes ($n_4 \sim n_{11}$), the algorithm in Fig. 10 starts. In step (1), $DA_S$ becomes $\{n_1, n_2, n_3\}$ and the third time slot is allocated to $n_1$ in step (2). In step (3), $n_3$ is selected as $n_{LCN}$ and $n_2$ as $n_{HCN}$, so that $n_3$ can transmit its data to $n_2$ while $n_1$ delivers data to the sink node. Next, based on steps (1) and (2) in Fig. 10, $n_2$ delivers aggregated data to the sink node at $ts = 4$.

4. Performance Evaluation

Our performance evaluation is based on both mathematical analysis and implementation. For the mathematical analysis, ISRMC-MAC is compared with HyMAC [19] and the base MAC protocol, which is the TDMA-based single channel MAC protocol. Section 2 discussed several single-radio multi-channel MAC protocols, and among them HyMAC showed the best performance. For the implementation, the performance of ISRMC-MAC is compared with the IEEE 802.15.4 MAC protocol. The parameters for the performance evaluation are as follows: The number of available channels is 1 ~ 16 and the number of sensor nodes is 1 ~ 64. The Beacon Order and Superframe Order of the superframe structure is set to 10, and thus, the duration of a time slot is 62 ms. In addition, each sensor node is assumed to concurrently transmit data during each superframe after the beacon transmission completes. Finally, the data size is assumed to be 5 bytes.
4.1 Analytical Analysis

This subsection verifies through mathematical analysis that ISRMC-MAC reduces latency compared with the base MAC protocol and increases energy efficiency compared with HyMAC.

4.1.1 Data Latency

All of the compared MAC protocols use the TDMA-based approach, and thus, the latency as function of \( N \) and \( C \) \((T_{(N,C)})\) is defined by

\[
T_{(N,C)} = G_{(N,C)} \times \text{Duration of a GTS (in ms)},
\]

where \( G_{(N,C)} \) represents the maximum number of time slots required for \( N \) sensor nodes given \( C \) channels.

\( G_{(N,1)}^{\text{Base}} \) for the base MAC protocol is defined by the following equation:

\[
G_{(N,1)}^{\text{Base}} = N.
\]
The base MAC protocol uses the star topology with a single-channel, and thus, the number of time slots required is the same as the number of sensor nodes \( N \).

In contrast, ISRMC-MAC and HyMAC reduce data latency compared to the base MAC protocol by concurrently transmitting data using multi-channel communication. Therefore, \( \lceil \frac{N}{C} \rceil \) time slots are required in each channel when parallel transmissions are fully exploited.

Since deriving exact equations for \( G_{(N,C)}^{HyMAC} \) and \( G_{(N,C)}^{ISRMC} \) is impractical, they are manually calculated for \( 1 \leq N \leq 64 \) and \( 1 \leq C \leq 64 \). These results shown in Fig. 14 indicate that (1) both ISRMC-MAC and HyMAC have similar performance on data latency, and (2) multi-channel protocols are superior to single-channel protocols in terms of data latency.

### 4.1.2 Energy Efficiency

Most of the energy consumption occurs during communication. Since the multi-channel approach relies on data duplication to reduce data latency, the number of packets and the total data size affect energy consumption. The analytical analysis in Subsection 4.1.1 reveals that the number of packets for HyMAC and ISRMC-MAC are similar because the required number of time slots for the two methods are almost the same. However, this subsection shows that the proposed ISRMC-MAC outperforms HyMAC in terms of energy efficiency because the total data size is decreased.

Assuming that the packet size is constant, the total data size, \( D_{(N,C)} \), is defined as the total number of data duplications required for all the transmissions of \( N \) sensor nodes using \( C \) channels including original packets from the sensor nodes. In fact, \( D_{(N,C)} \) is equivalent to \( N \) original packets plus the total number of hops for all the transmissions. The total data size \( D_{(N,C)} \) can be expressed as follows:

\[
D_{(N,C)} = D_{(N,C)} \times \text{Packet size (in bytes)}.
\]  

Since there is no data duplication in the base MAC protocol, \( D_{(N,1)}^{Base} \) simply increases linearly as function of \( N \).

HyMAC employs the binary tree topology; therefore, its total data size \( D_{(N,C)}^{HyMAC} \) is given by the following equations:
\[ D^{\text{HyMAC}}_{(L,C)} = 1 \]
\[ D^{\text{HyMAC}}_{(N,L,C)} = D^{\text{HyMAC}}_{(N-1,L,C)} + 1 + \lceil \log_2 (N + 1) \rceil, \text{for } N \geq 2 \]  
(4)

where \( \lceil \log_2 (N + 1) \rceil \) indicates the level of the binary tree.

Finally, the total data size for the proposed ISRMC-MAC, \( D^{\text{ISRMC}}_{(N,L,C)} \), can be expressed as follows:

\[ D^{\text{ISRMC}}_{(N,L,C)} = \lceil \log_2 N \rceil, \text{for } 1 \leq N \leq C \]
\[ D^{\text{ISRMC}}_{(N,L,C)} = D^{\text{ISRMC}}_{(N-1,L,C)} + 1 + \lceil \log_2 (C - 1) \rceil, \text{for } N \geq C \]  
(5)

In the proposed Unbalanced Star+Mesh topology, there are no \( L \_\text{nodes} \) when \( N < C \), i.e., only the sink node and \( DA \_\text{nodes} \) exist. In this case, \( N \ DA \_\text{nodes} \) aggregate data among each other resulting in maximum of \( \lceil \log_2 N \rceil \) data duplications.

When \( N \geq C \), \( L \_\text{nodes} \) exist and one data duplication for each \( L \_\text{node} \) occurs at \( DA \_\text{nodes} \), and additional data duplications occur among \( DA \_\text{nodes} \) in the mesh topology. Since the number of \( DA \_\text{nodes} \) is \( C - 1 \), the upper bound on the number of additional duplications is \( \lceil \log_2 (C - 1) \rceil \).

Comparison of Eqs. 4 and 5 shows that the proposed ISRMC-MAC protocol is superior to HyMAC in terms of energy consumption because \( D^{\text{HyMAC}}_{(N,L,C)} \) depends on \( N \) but \( D^{\text{ISRMC}}_{(N,L,C)} \) depends on \( C \). Fig. 15 shows the calculated values of \( D^{\text{HyMAC}}_{(N,L,C)} \) and \( D^{\text{ISRMC}}_{(N,L,C)} \) for \( 1 \leq N \leq 64 \) and \( 1 \leq C \leq 64 \), which reveals that the above analysis holds.

4.2 Implementation

The proposed ISRMC-MAC protocol has been implemented and tested on ZigbeX shown in Fig. 16. ZigbeX consists of an ATMega128L microcontroller and a CC2420 transceiver, which supports a single-radio with multiple channels. The number of channels \( C \) supported is 4 and the number of nodes \( N \) varies from 1 to 15. The nodes are randomly deployed in \( 1 \text{m} \times 0.5 \text{m} \) space representing a human body. All the nodes utilize the same power level 7, which is the lowest power available on the CC2420 module. Our experiments consider data latency and energy efficiency to verify the performance of the proposed method.

4.2.1 Data Latency

Fig. 17a shows the data latency versus \( N \) when \( C \) is fixed to 4. The latency for the base MAC protocol significantly increases as the number of sensor nodes increases. For example, when \( N = 15 \), the latency is 1,032 ms. On the other hand, the latency for ISRMC-MAC increases
slowly due to the multi-channel approach. The gap between the two methods also increases as $N$ increases. When $N = 15$, the latency of ISRMC-MAC is 292 ms. These results verify that the proposed ISRMC-MAC protocol significantly reduces data latency compared with existing single-channel approaches.

Note that the analytical results of ISRMC-MAC derived in Subsection 4.1.1 are the upper bounds on data latency, thus they are higher than the experimental results as shown in Fig. 17(a).

4.2.2 Energy Efficiency

Fig. 17b shows the total data size versus $N$ when $C$ is fixed to 4. As can be seen, ISRMC-MAC transmits more data than the base MAC protocol. Again, this is because the base MAC protocol uses a TDMA-based single-channel star topology, and thus, there is no data duplication. On the other hand, the Unbalanced Star+Mesh topology of ISRMC-MAC requires data aggregation, which increases the total data size. However, the results from our implementation are very similar to the mathematical analysis, which verifies that our proposed method improves energy efficiency compared to other single-radio multi-channel MAC protocols.

5. Conclusion

WBAN is a promising wireless technology to satisfy the requirements of healthcare applications. However, high density of sensor nodes in a WBAN can cause energy inefficiency and high data latency. Many MAC protocols have been proposed to alleviate these problems, but none of these methods fully satisfy all the requirements of WBANs. Some of them employ multi-radio technique, but they are not feasible using off-the-shelf radios. Others employ single-radio and data aggregation, but energy consumption increases due to data duplication. The proposed ISRMC-MAC protocol uses a single-radio, multichannel approach, which can be implemented on off-the-shelf sensor devices. Moreover, the Unbalanced Star+Mesh topology together with channel and GTS allocation strategies reduces data duplication by reducing the number hops. Therefore, the proposed approach improves throughput using the multi-channel and reduces energy consumption by decreasing data duplication.
The superiority of ISRMC-MAC over the state-of-the-art protocol HyMAC was validated through mathematical analyses and experiments using ZigbeX. Our results showed that the proposed method reduces data latency and improves energy efficiency compared to HyMAC by achieving more than two-fold reduction in data duplication.

References


Kunryun Cho received the B.S. and M.S. degree in computer engineering from Kyung Hee University, Korea, in 2012 and 2014. He is currently pursuing the Ph.D. degree with the Department of Computer Engineering, Seoul National University, Korea. His research interests include wireless sensor network, wireless body area network, wireless local area network, and indoor localization.

Seokhee Jeon received the BS and PhD degrees in computer science and engineering at the Pohang University of Science and Technology (POSTECH) in 2003 and 2010, respectively. From 2010 to 2012, he was a postdoctoral researcher in the Computer Vision Laboratory at ETH Zurich. In 2012, he joined as an assistant professor in the Department of Computer Engineering, Kyung Hee University, where he is currently directing the Haptics Laboratory. His research focuses on haptic rendering in an augmented reality environment, applications of haptics technology in medical training, and usability of augmented reality applications.

Jinsung Cho (M’15) received the B.S., M.S, and Ph.D. degrees in computer engineering from Seoul National University, Korea, in 1992, 1994, and 2000, respectively. He was a Visiting Researcher with the IBM T. J. Watson Research Center, in 1998, and a member of the Research Staff with Samsung Electronics, from 1999 to 2003. He is currently a Professor with the Department of Computer Engineering, Kyung Hee University, Korea. His research interests include mobile networking and computing, embedded systems and software, and sensor and body networks.

Ben Lee received the B.E. degree in electrical engineering from the Department of Electrical Engineering, State University of New York (SUNY) at Stony Brook, in 1984, and the Ph.D. degree in computer engineering from the Department of Electrical and Computer Engineering, Pennsylvania State University, in 1991. He is currently a Professor of School of Electrical Engineering and Computer Science with Oregon State University. His research interests include wireless networks, embedded systems, computer architecture, multimedia systems, and parallel and distributed systems. He received the Loyd Carter Award for Outstanding and Inspirational Teaching and the Alumni Professor Award for Outstanding Contribution to the College and the University from the OSU College of Engineering, in 1994 and 2005, respectively. He also received the HKN Innovative Teaching Award from Eta Kappa Nu, School of Electrical Engineering and Computer Science, in 2008. He has been on the Program and Organizing Committees for numerous international conferences, including the 2003 International Conference on Parallel and Distributed Computing Systems, the 2005–2011 IEEE Workshop on Pervasive Wireless Networking, and the 2006, 2007, and 2009 IEEE International Conference on Pervasive Computing and Communications. He was also a Keynote Speaker at the 2014 International Conference on Ubiquitous Information Management and Communication. He is currently the Chair of the Social, P2P and Multimedia Networking, Services and Applications track for the 2016 IEEE Consumer Communications and Networking Conference.