Coloring-Based Channel Allocation for Multiple Coexisting Wireless Body Area Networks: A Game-Theoretic Approach

Kai-Ju Wu, Y.-W. Peter Hong, *Senior Member, IEEE*, and Jang-Ping Sheu, *Fellow, IEEE*

Abstract—This paper addresses the coexistence problem among multiple wireless body area networks (WBANs), where co-channel interference may occur among different WBANs if the channels are not allocated properly, leading to performance degradation in both energy efficiency and packet transmission reliability. We formulate the channel allocation problem as a graph coloring problem, and develop a solution to increase the co-channel reuse and the number of WBANs with assigned channels. We propose a distributed two-hop incomplete coloring (DTIC) algorithm that adopts a game-theoretic approach to solve the graph coloring problem. The DTIC algorithm exploits two-hop information to enable high channel reuse among two-hop neighbors and allows for incomplete coloring when the number of colors (or channels) is insufficient to color all vertices without conflict. A distributed message-passing protocol is also proposed to achieve collision-free message exchange, and to ensure that consistent coloring information is shared among WBANs. Simulation results show that our proposed algorithm achieves better co-channel reuse and higher throughput than existing methods.

Index Terms—Wireless body area networks, channel allocation, game theory, potential game, graph coloring

1 Introduction

Wireless body area networks (WBANs) [1], [2] typically consist of multiple on-body or implanted sensors along with a coordinator responsible for collecting real-time signals from the sensor devices using short-distance communication interfaces such as Wi-Fi, ZigBee, or Bluetooth. They have been used to monitor vital signals or fitness information associated with the human body and have seen wide applications in various areas, including sports and healthcare [3]. Different from conventional wireless sensor networks, where sensors are often deployed in large-scale over a wide static environment, each WBAN usually consists of only a small number of sensors distributed over a small volume in space, but many WBANs may coexist in a single area causing strong interference to each other. The resulting packet collision, redundant retransmission, and energy consumption may have a significant impact on the reliability and longevity of these healthcare or on-body human monitoring systems and, thus, should be properly managed through effective channel allocation policies.

Specifically, channel allocation policies help avoid interference in wireless networks by assigning different time or frequency channels to users within the interference range of each other. These polices were adopted in many cellular or wireless local area network applications, e.g., in [4], [5], [6], and [7], where the solutions are often centralized and static among base-stations or access points and, thus, may not be directly suitable for WBAN applications. Many works, e.g., [8], [9], [10], and [11], modelled the channel allocation problem as a graph coloring problem, where users within the interference range of each other are treated as neighboring nodes within a graph, and colors (i.e., channels) are assigned to the nodes so that no conflict occurs between neighbors (i.e., no two neighbors are assigned the same color). In particular, [8] was one of the earliest works to study the channel allocation problem from a graph coloring perspective, but proposed only a centralized solution. The works in [9], [10], and [11] proposed solutions for the distributed setting, but do not explicitly address the need for efficient data exchange in these cases. In fact, these schemes often resort to simple broadcasting which may be inefficient, especially in dense networks.

To avoid interference among WBANs in a distributed and highly dynamic environment, this paper proposes the Distributed Two-hop Incomplete Coloring (DTIC) algorithm using a game-theoretic approach. The DTIC algorithm is a distributed channel allocation algorithm that allows each WBAN to determine its own color (i.e., channel) based on the coloring decisions made by its two-hop neighbors and allows a WBAN to remain uncolored (i.e., remain silent) in a certain time slot if no color can be chosen without conflict at that time. This results in the so-called *incomplete coloring* of the graph, which was first investigated in [11]. Here, we model this problem as a *best-response potential game* [12], whose convergence to the Nash equilibrium can be achieved by a sequence of best response actions by different WBANs in an orderly fashion. A distributed message-passing

- *K.-J. Wu and J.-P. Sheu are with the Department of Computer Science, National Tsing Hua University, Hsinchu 30013, Taiwan. E-mail: lbj39919538@gmail.com, sheujp@cs.nthu.edu.tw.*
- *Y.-W. P. Hong is with the Institute of Communications Engineering, National Tsing Hua University, Hsinchu 30013, Taiwan, and also with the MOST Joint Research Center for AI Technology and AII Vista Healthcare, Taipei 10617, Taiwan. E-mail: ywhong@ee.nthu.edu.tw.*

Manuscript received 28 June 2019; revised 24 Dec. 2019; accepted 9 June 2020. Date of publication 16 June 2020; date of current version 3 Dec. 2021. (Corresponding author: Y.-W. Peter Hong.) Digital Object Identifier no. 10.1109/TMC.2020.3002898

protocol is adopted to provide each WBAN with its two-hop neighbors' coloring information, and the Best-Response Resolution (BRR) algorithm is proposed to ensure that the sequence of best-response updates performed by the WBANs are done in a consistent manner. Our main contributions can be summarized as follows:

- the proposition of a distributed channel allocation algorithm for interference avoidance among multiple WBANs using a game-theoretic incomplete graph coloring solution;
- the use of two-hop coloring information to enhance the local channel reuse among WBANs so that more channels are made available for use in other areas of the network;
- the design of a distributed message-passing protocol and the corresponding BRR algorithm that can be used to achieve collision-free message exchanges among WBANs, and to ensure the construction of a sequential best-response improvement path;
- the proof of existence of a Nash equilibrium (NE) in the proposed graph coloring game and its convergence under the proposed distributed message-passing protocol.

It is worthwhile to note that, even though graph coloring problems have been studied extensively in the past, most variations of the problem require all nodes to be colored, albeit under different coloring constraints. For example, defective coloring examines the coloring of vertices subject to a constraint on the number of neighbors with the same color for each vertex; weak coloring examines the coloring of vertices such that each non-isolated vertex is adjacent to at least one vertex with a different color. To the best of our knowledge, [11] is the only work that addresses the incomplete graph coloring problem (i.e., the problem that allows for uncolored nodes) and, thus, is most relevant to our work. Our proposed DTIC algorithm improves upon the RIC algorithm proposed in [11] by utilizing two-hop information to enhance local channel reuse and, thus, release more channels for use in other areas of the network.

The rest of the paper is organized as follows. Section 2 reviews related work on channel allocation and graph coloring. Section 3 describes the system model, and Section 4 describes the graph coloring formulation and the proposed DTIC algorithm. Section 5 provides methods for efficient data-exchange. Finally, the performance is evaluated in Section 6 and a conclusion is drawn in Section 7.

2 Related Work

The coexistence of multiple WBANs leads to interference problems that must be addressed to ensure reliable communication between the wearable sensors and their coordinator within each WBAN. These issues are often resolved through channel allocation policies that assign different channels to neighboring WBANs, similar to that of graph coloring [8], [9], [10]. Specifically, in [8], a unified framework was proposed based on the graph coloring model for the study of channel allocation problems in time, frequency, and code domains. The performance of three different greedy heuristics were studied under this framework. In [9], the graph coloring problem was studied for secondary users in open spectrum access systems where the availability of colors (i.e., channels) at the users are constrained by the primary users in their vicinity. Three distributed coloring algorithms were proposed with the objective of maximizing the total spectrum utilization. Moreover, [10] examined the utility and fairness of channel allocation schemes in open access systems, and showed that the global optimization problem is NP-hard. An approximation algorithm was provided through vertex labeling, and collaborative coloring methods were proposed to address different notions of fairness.

Traditional graph coloring algorithms assign colors to all vertices in a graph without conflict among adjacent vertices. The minimum number of colors required to achieve this task is referred to as the chromatic number. Finding the optimal solution to this problem (i.e., the chromatic number) is in general NP-hard. However, the time-complexity can be significantly reduced by increasing the number of colors to $\Delta + 1$, where Δ is the maximum degree of the graph. In the context of distributed graph coloring, [13] proposed a simple algorithm similar to Luby's MIS algorithm to obtain a $(\Delta + 1)$-coloring in $O(\log n)$ rounds. The results were improved to $O(\sqrt{\log n})$ rounds in [14] and more recently to $O(\text{poly} \log \log n)$ rounds in [15]. Moreover, [16] examined the more restrictive case where a vertex is only assumed to know its own color and whether or not a conflict occurs with its neighbors at any given time. Their proposed algorithm was shown to converge towards a proper $(\Delta + 1)$-coloring in $O(n\Delta)$ expected recolorings. [17] achieved convergence in $O(n \log n)$ rounds by allowing simultaneous recoloring by all vertices with an appropriately chosen distribution that evolves over time. However, these works did not investigate the message exchange overhead required to establish the communication between local nodes. In [18], a distributed graph coloring algorithm that aims to minimize the number of used colors was proposed by emulating the calling behavior of Japanese tree frogs. Contrary to the study on incomplete graph coloring, the above works on traditional graph coloring assume that no vertex can be left uncolored.

In the context of WBANs, a random incomplete coloring (RIC) algorithm was proposed to mitigate the interference among WBANs in [11]. In the case of incomplete coloring, a vertex may be temporarily left uncolored if all colors have been occupied by its immediate neighbors. The flexibility of this approach allows the coloring to be done more efficiently (i.e., with lower time-complexity), and results in higher channel reuse than the case of complete coloring. In particular, [11] proposed a distributed algorithm that utilizes only one-hop information from its neighbors, and thus is limited in its coloring efficiency (i.e., the number of vertices that can be colored without conflict given the number of available colors). Different from [11], our proposed scheme utilizes information from two-hop neighbors and adopts a game-theoretic approach with provable convergence to achieve the distributed coloring task. In addition, we also specify the message-passing scheme necessary to ensure consistency among the coloring updates throughout the process.

Game theory [19] is a field of mathematics that studies the distributed decision-making of rational players in a game. It has applications in many different research areas such as economics, political science, psychology, and computer science. In fact, several works, e.g., [20], [21], and [22], also adopted these techniques to solve the aforementioned channel allocation problem. Specifically, in [20], the channel allocation problem in cognitive radio networks was modelled as a potential game using two different objective functions to capture the utility of selfish and cooperative users. Deterministic Nash equilibria were shown to exist, but require substantial coordination to achieve. The utility depends on the interference between neighboring nodes, but does not preclude users from choosing the same channel. Alternative no-regret learning approaches were proposed to obtain more efficient solutions for this problem. In [21], an interference-aware channel allocation algorithm was also proposed using a game-theoretic approach. In this case, the utility is also dependent on the co-channel interference, which may be asymmetric due to the nodes' different transmit powers. In [22], a priority-based allocation of time slots was considered for WBANs, and a time allocation algorithm was proposed based on the hawk-dove game model. The coordinators of WBANs transmit their data in assigned time slots to a common anchor node, who then forwards the data to the data center. The main concern in [22] was to avoid the interference among coordinators when transmitting to a common anchor node. This is different from our distributed channel allocation problem where the concern is towards avoiding interference among neighboring WBANs. Furthermore, game-theoretic approaches were also adopted to solve graph coloring problems in [23], [24], [25], and [26]. Specifically, in [23], an efficient graph coloring algorithm was proposed using a game-theoretic local search algorithm. In this work, the utility function of each player was defined as the total number of players in the graph that have chosen the same color as its own. This requires global information of the cardinalities of all colors to be available to each player at any time. In [24], a distributed implementation of the algorithm in [23] was proposed by allowing multiple non-neighboring nodes to update their colors simultaneously. A message-passing protocol was proposed to propagate the required network information to necessary nodes in the network. In [25] and [26], game-theoretic interference avoidance channel selection algorithms were proposed using the negative weighted aggregate interference as the utility function for each node. In these cases, the coloring (or channel selection) was not limited to proper coloring solutions. Similar to traditional graph coloring, the above works assumed that all vertices must be colored and, thus, are different from the incomplete graph coloring solution proposed in our work.

3 System Model

Let us consider a network with multiple WBANs, each consisting of a coordinator and several on-body or implanted sensors, as illustrated in Fig. 1. The sensors are responsible for sensing important body signals and communicate directly to the coordinator following a predefined intra-WBAN MAC protocol. If two WBANs are within the transmission range of each other and are transmitting over the same channel, their transmissions will interfere with each other, causing packet collision and data loss. The transmission range of a WBAN coordinator is typically 3 meters, as indicated in [27], which is sufficient to cover a human body. Here, we assume that a coordinator not only collects data from its sensors but also communicates with other coordinators to determine the channel to use in each time slot. In particular, we consider a time-slotted system with slot-level synchronization among coordinators, and focus on resolving inter-WBAN interference through proper channel allocation among multiple WBANs (but do not address the intra-WBAN MAC protocol design). The problem can be modeled as a graph coloring problem as described below.

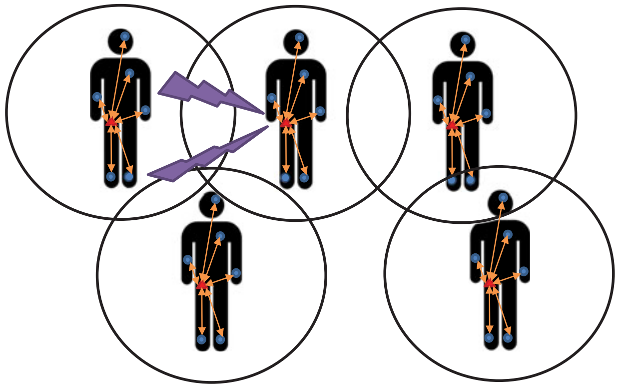
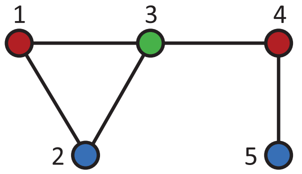


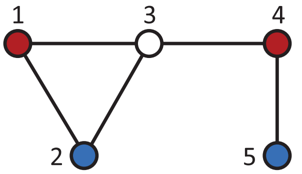
Fig. 1. Illustration of interference among coexisting WBANs.

Specifically, let us consider a graph $G=(V,E)$, where V is the set of vertices and E is the set of edges. Each WBAN is represented by a vertex in V, and two neighboring WBANs, say i and j, are connected by an edge $(i,j)\in E$ if they are within the interference range of each other. We assume that the transmission range is the same as the interference range in this work. The set of channels (or colors) that can be assigned to the vertices is denoted by C. In this work, we consider a graph coloring problem, where each vertex is assigned at most one color and no two adjacent vertices (i.e., vertices connected by an edge) share the same color. In terms of the channel allocation problem, this is equivalent to saying that each WBAN is allocated at most one channel, and no two WBANs within the interference range of each other share the same channel. Conflict is avoided between two adjacent vertices if they are assigned different colors or if at least one of them is left uncolored. Given the graph G and the number of colors $|C|=k$, our goal is to devise an incomplete coloring solution that yields the maximum number of colored vertices. This is referred to as the *incomplete k-coloring problem*. The coloring is incomplete in the sense that some nodes may be left uncolored [11].

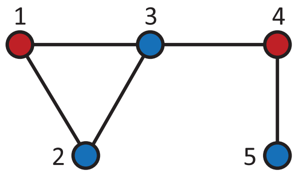
It is worthwhile to remark that, in the traditional graph coloring problem, all vertices should be colored (i.e., no vertex can be left uncolored). The minimum number of colors needed to achieve this task without conflict is referred to as the chromatic number, and is denoted by $\chi(G)$ for graph G. Finding the chromatic number $\chi(G)$ is an NP-hard problem [28]. Different from the traditional graph coloring problem, the incomplete coloring problem considered in this work allows a vertex to be left uncolored (especially when all available colors have been occupied by its neighbors). This corresponds to the practical case where the number of channels is less than the number of mutually interfering WBANs. Examples of traditional graph coloring with $k=3$, incomplete k-coloring with $k=2$, and improper coloring with $k=2$ are given in Fig. 2. The graphs in Fig. 2 correspond to the sample network given in Fig. 1. In the traditional graph coloring



(a) Traditional complete coloring



(b) Incomplete 2-coloring with $C = \{\text{red}, \text{blue}\}$



(c) Improper coloring with $C = \{\text{red}, \text{blue}\}$.

Fig. 2. Examples of complete, incomplete, and improper coloring on a graph with chromatic number $\chi(G) = 3$.

example shown in Fig. 2a, we can see that at least 3 colors are needed to color all vertices in the graph and, thus, the chromatic number of the graph is $\chi(G) = 3$. On the other hand, incomplete graph coloring allows nodes to be left uncolored, resulting in more flexibility in the color assignment and the use of less colors (i.e., less channels). In fact, as shown in Fig. 2b, one of the vertices in the subset $\{1, 2, 3\}$ must be left uncolored if only 2 colors are available. Each of these vertices can be assigned a channel only $2/3$ of the time on the average. The overall throughput of the system is thus reduced due to the number of uncolored vertices. An attempt to color all vertices with only 2 colors can only result in an improper coloring as shown in Fig. 2c, which leads to conflict among neighboring WBANs.

Remark 1. When a WBAN is not colored, it implies that the WBAN is not able to transmit without interfering with other WBANs. In this case, the WBAN is prevented from transmitting and, thus, must wait until a channel is freed up to perform the communication. We would like to note that, due to mobility, the coloring procedure will be repeated every number of time slots, e.g., every 10^4 ms (c. f., Section 6). Hence, the WBANs that were previously uncolored may have the opportunity to be colored later on. In addition, an uncolored WBAN may also be allowed to monitor the usage of channels in its vicinity and assume a certain channel once it becomes idle after a certain amount of time. However, the fairness of the rescheduling over time deserves a more thorough study that is beyond the scope of this work.

4 The Distributed Two-Hop Incomplete Coloring (DTIC) Algorithm

In this section, we propose a distributed incomplete coloring algorithm where two-hop coloring information is utilized by each WBAN to determine its local coloring. The key idea is to have vertices select colors to maximize the channel reuse among its two-hop neighbors, leaving more colors available for other vertices outside of its two-hop radius. In the case of mobility, higher channel reuse among local nodes implies that more colors can be made available locally, allowing nodes to more easily discover available channels once a conflict occurs due to changes in the graph topology. To obtain a distributed solution, we adopt a game-theoretic approach where the WBANs (i.e., vertices) are the players and the colors (or channels) are the actions that can be taken by each player. Instead of looking for the optimal coloring of a graph, which generally requires full knowledge of the graph G and the colors chosen by all other vertices in the graph, we propose a distributed strategy, in which, each vertex utilizes only information from its two-hop neighbors to determine its local action (i.e., color selection). The message-passing required for each player to obtain information from its two-hop neighbors is described in the Section 5.

Specifically, let us consider a multiplayer game $\Gamma = (V, \{A_i\}_{i\in V}, \{u_i\}_{i\in V})$ played on the graph $G = (V, E)$, where V is the set of players, A_i is the set of actions for player i, and $u_i : \prod_{i\in V} A_i \rightarrow \mathbb{R}$ is player i's utility function. The action set $A_i \triangleq C \cup \{\text{uncolored}\}$ is common for all players. Let $\mathbf{a} = (a_1, \ldots, a_n)$ be the action profile, where $a_i \in A_i$ is the action chosen by player i and $n = |V|$ is the number of players. We say that player i prefers action profile $\mathbf{a}$ over another action profile $\mathbf{a}'$ if $u_i(\mathbf{a}) > u_i(\mathbf{a}')$. Since a player can only control its own action, no further change in actions will occur if no player can unilaterally improve its local utility by altering its own action. This results in the so-called Nash equilibrium (NE), which is defined as follows.

Definition 1. *Action profile $\mathbf{a}$ is a Nash equilibrium (NE) if*

$$u_i(a_i, \mathbf{a}_{-i}) \geq u_i(a_i', \mathbf{a}_{-i}), \ \forall a_i' \in A_i \text{ and } \forall i \in V, \tag{1}$$

where $\mathbf{a}_{-i}$ is the action profile excluding the action from player i. That is, $\mathbf{a}$ is an NE if no player can improve its local utility by unilaterally altering its own action.

Notice that, in general, it is necessary for player i to have centralized knowledge of the actions of all players in order to determine its optimal action. However, in this work, we are instead interested in finding a distributed coloring algorithm that allows each player to make its own coloring decisions based only on its two-hop information. This can be done by choosing a utility function for each player that depends only on the coloring decisions of its two-hop neighbors. To do so, let us first introduce the notion of a 2-clique [29] as a subset $Q \subseteq V$ whose members are within two-hops of each other. Then, a subset Q is said to be a maximal 2-clique if it cannot be extended by including more vertices. For example, in Fig. 2, the subset of vertices $Q' = \{1, 2, 3\}$ forms a 2-clique since all vertices in Q' are within two-hops of each other. However, Q' is not a maximal 2-clique since adding vertex 4 into Q' also forms a 2-clique. On the other hand, the subset $Q = Q' \cup \{4\} = \{1, 2, 3, 4\}$ is a maximal 2-clique since vertex 5 is beyond two hops of vertices 1 and 2 and, thus, cannot be added to form a larger 2-clique.

Specifically, let us define the utility function of player i as

$$u_i(\mathbf{a}) \triangleq \sum_{Q\in \mathrm{CL}_2 : i\in Q} \Phi^Q(\mathbf{a}), \tag{2}$$

where $\Phi^Q(\mathbf{a})$ is the local function associated with a maximal 2-clique Q in the graph G, and CL_2 is the collection of all maximal 2-cliques in the graph G. The local function is defined as

$$\Phi^Q(\mathbf{a}) = \sum_{j\in Q\setminus \tilde{Q}(\mathbf{a})} M_j^Q(\mathbf{a}) - \sum_{j\in \tilde{Q}(\mathbf{a})} M_{\max}, \tag{3}$$

where

$$M_j^Q(\mathbf{a}) \triangleq \begin{cases} \left|\{k\in Q : a_k = a_j\}\right|, & \text{if } a_j \in C,\\ 0, & \text{if } a_j = \text{uncolored}, \end{cases} \tag{4}$$

is the number of vertices in the maximal 2-clique Q that have the same color as player j, $\tilde{Q}(\mathbf{a}) = \{i \in Q : \exists j \in Q \text{ such that } (i,j)\in E, a_i = a_j, \text{ and } a_i, a_j \in C\}$ is the subset of vertices in Q whose colors conflict with its one-hop neighbors in Q, and $M_{\max}$ is a constant chosen such that $u_i(\mathbf{a}) < M_{\max}$, for all i and $\mathbf{a}$.[1] The choice of $M_{\max}$ ensures that no player will select a color that conflicts with its one-hop neighbors. Notice that, by the above definition, the utility function $u_i(\mathbf{a})$ indeed depends only on the actions of its one-hop and two-hop neighbors. Hence, for player i to determine its best action, it only needs to know the actions of neighbors within its two-hop radius.

Example: To better understand the notations defined above, let us consider as an example the incomplete graph coloring given in Fig. 2b. Notice that there are two maximal 2-cliques in this graph, namely, $Q_1 = \{1,2,3,4\}$ and $Q_2 = \{3,4,5\}$. For the maximal 2-clique $Q_1 = \{1,2,3,4\}$, we have $M_1^{Q_1}(\mathbf{a}) = M_4^{Q_1}(\mathbf{a}) = 2$ since players 1 and 4 have chosen the same color (i.e., the color "red"). Similarly, we have $M_2^{Q_1}(\mathbf{a}) = 1$. Also, $M_3^{Q_1}(\mathbf{a}) = 0$ since player 3 is uncolored. Since no conflict occurs among one-hop neighbors in this example, we have $\tilde{Q}_1(\mathbf{a}) = \emptyset$. Consequently, the local function of Q_1 can be computed as $\Phi^{Q_1}(\mathbf{a}) = \sum_{j\in Q_1} M_j^{Q_1}(\mathbf{a}) = 2+1+0+2 = 5$. Similarly, the local function of Q_2 is $\Phi^{Q_2}(\mathbf{a}) = \sum_{j\in Q_2} M_j^{Q_2}(\mathbf{a}) = 0+1+1 = 2$. In this case, we have $u_1(\mathbf{a}) = u_2(\mathbf{a}) = \Phi^{Q_1}(\mathbf{a}) = 5$, $u_3(\mathbf{a}) = u_4(\mathbf{a}) = \Phi^{Q_1}(\mathbf{a}) + \Phi^{Q_2}(\mathbf{a}) = 7$, and $u_5 = \Phi^{Q_2}(\mathbf{a}) = 2$. However, if player 3 opts for the color "blue", the graph results in the improper coloring shown in Fig. 2c which yields the conflicting subset $\tilde{Q}_1(\mathbf{a}) = \{2,3\}$ of the maximal 2-clique $Q_1 = \{1,2,3,4\}$. In this case, the local functions of Q_1 and Q_2 become $\Phi^{Q_1}(\mathbf{a}) = \sum_{j\in\{1,4\}} M_j^{Q_1}(\mathbf{a}) - \sum_{j\in\{2,3\}} M_{\max} = 4 - 2M_{\max}$ and $\Phi^{Q_2}(\mathbf{a}) = \sum_{j\in Q_2} M_j^{Q_2}(\mathbf{a}) = 2+1+2 = 5$, respectively. Therefore, the utility function of player 3 becomes $u_3(\mathbf{a}) = \Phi^{Q_1}(\mathbf{a}) + \Phi^{Q_2}(\mathbf{a}) = 9 - 2M_{\max}$. Notice that, by choosing $M_{\max}$ to be sufficiently large (e.g., $M_{\max} = 5^3 = 125$), player 3 will always result in a negative utility when choosing a color that conflicts with one of its one-hop neighbors. Hence, the optimal selfish choice for player 3 would be to remain uncolored.

Furthermore, it is interesting to note that the utility function provides larger payoff to players that share colors with more of its two-hop neighbors (while avoiding conflict with its one-hop neighbors). In fact, given that there is no conflict, the local function can be written equivalently as $\Phi^Q(\mathbf{a}) = \sum_{c\in C} |i \in Q : a_i = c|^2$, which gives a quadratic gain to colors used multiple times within Q. Following the above example, the local functions of Q_1 and Q_2 can be computed alternatively as $\Phi^{Q_1}(\mathbf{a}) = \sum_{c\in C} |i\in Q_1 : a_i = c|^2 = 2^2 + 1^2 = 5$ and $\Phi^{Q_2}(\mathbf{a}) = 1^2 + 1^2 = 2$. We argue that, by allowing the use of colors to be more concentrated locally (within a two-hop neighborhood), more colors can be freed up for use in other areas of the network.

To show the existence of an NE, we first define the notion of a *best-response potential game* [12] as given below.

Definition 2. *The game Γ is a best-response potential game if and only if there exists a function $\Psi : \prod_{i\in V} A_i \to \mathbb{R}$ such that*

$$B_i(\mathbf{a}_{-i}) = \arg\max_{a_i\in A_i} \Psi(a_i, \mathbf{a}_{-i}),\ \forall \mathbf{a}_{-i} \in \prod_{j\in V\setminus\{i\}} A_j, \tag{5}$$

where $B_i(\mathbf{a}_{-i}) \triangleq \{a_i^* : a_i^* \in \arg\max_{a_i\in A_i} u_i(a_i, \mathbf{a}_{-i})\}$ is the set of best-response correspondence for player i. Here, Ψ is referred to as the global potential function.

By the above definition, we can see that, in a best-response potential game, the maximization of a player's local utility function leads to the maximization of the global potential function. Hence, by Theorem 2.1 and Corollary 2.2 of [12], every finite best-response potential game possesses a pure-strategy NE. In the following, we show that the proposed incomplete coloring game is a best-response potential game and, thus, possesses a pure-strategy NE.

Theorem 1. *The coloring game Γ is a best-response potential game with potential function*

$$\Psi(\mathbf{a}) = \sum_{Q\in \mathrm{CL}_2} \Phi^Q(\mathbf{a}). \tag{6}$$

Proof. By (6) and the definition of the utility function for player i given in (2), we have

$$\begin{aligned} \arg\max_{a_i\in A_i} \Psi(a_i, \mathbf{a}_{-i}) &= \arg\max_{a_i\in A_i} \sum_{Q\in \mathrm{CL}_2} \Phi^Q(\mathbf{a}) \\ &\overset{(a)}{=} \arg\max_{a_i\in A_i} \sum_{Q\in \mathrm{CL}_2 : i\in Q} \Phi^Q(\mathbf{a}) \\ &= \arg\max_{a_i\in A_i} u_i(a_i, \mathbf{a}_{-i}) = B_i(\mathbf{a}_{-i}), \end{aligned}$$

where (a) follows from the fact that $\Phi^Q(\mathbf{a})$ depends only on the colors of players in Q. Hence, by Definition 2, the coloring game Γ is a best-response potential game with potential function Ψ defined in (6). □

Interestingly, for best-response potential games with finite action spaces, it was established in Theorem 2.8 of [12] that a sequential best-response improvement path converges

1. In fact, it is sufficient to choose $M_{\max} = n^3$ in our case since $u_i(\mathbf{a}) = \sum_{Q\in \mathrm{CL}_2 : i\in Q} \Phi^Q(\mathbf{a}) \le \sum_{Q\in \mathrm{CL}_2 : i\in Q} \sum_{j\in Q} M_j^Q(\mathbf{a}) \le \sum_{Q\in \mathrm{CL}_2 : i\in Q} \sum_{j\in Q} n \le n^3$, for all i and $\mathbf{a}$.

to an NE in a finite number of steps. In particular, a best-response improvement path is defined as follows.

Definition 3. *A sequence of action profiles $\boldsymbol{\rho} = \{\mathbf{a}[0], \mathbf{a}[1], \mathbf{a}[2], \ldots\}$ is a sequential best-response improvement path if $\mathbf{a}[\ell]$ is obtained from $\mathbf{a}[\ell-1]$ by replacing the action of some player $i[\ell]$ (i.e., the single deviator in the ℓ-th step) with an action from its best-response correspondence $B_{i[\ell]}(\mathbf{a}_{-i[\ell]}[\ell-1])$ (i.e., $a_j[\ell] = a_j[\ell-1]$, for $j \neq i[\ell]$, and $a_{i[\ell]}[\ell] \in B_{i[\ell]}(\mathbf{a}_{-i[\ell]}[\ell-1])$), for $\ell = 1, 2, \ldots$.*

In the following, we demonstrate how a sequential best-response improvement path can be constructed in the distributed scenario under consideration.

5 Distributed Message-Passing Protocol

In this section, we describe how players (i.e., WBANs) can communicate to achieve the sequential best-response improvement path in a distributed fashion. We adopt a distributed message-passing protocol that consists of two phases: a random access and a scheduled access phase. The random access phase is used to provide each player with initial information regarding its one-hop and two-hop neighbors, similar to the commonly adopted neighbor protocol (e.g., in [30]), whereas the scheduled access phase is used for players to exchange and update their actions in a way that enables the construction of a sequential best-response improvement path. The two phases are described in detail below.

5.1 Random Access Phase

In the random access phase, the players exchange information with its neighbors to obtain the initial colors and the two-hop neighborhood information of all players within its two-hop radius.

Let $a_i \in C$ be the initial color chosen by player i, and let $N_1(i)$ and $N_2(i)$ be sets consisting of player i's one-hop and two-hop neighbors, respectively. In this case, we can define the information vector associated with player i as

$$\mathrm{info}_i \triangleq (a_i, N_1(i), N_2(i)). \tag{7}$$

Moreover, each player, say player i, maintains a set $L(i)$ to record the information vectors associated with its one-hop neighbors. This set is initialized as an empty set, i.e., $L(i) = \emptyset$, at the beginning of the random access phase. When player i is given the opportunity to transmit in the random access phase, it broadcasts a control packet that includes both its own information vector info_i and its currently recorded information $L(i)$ to its one-hop neighbors. This control packet is denoted by

$$\mathrm{pkt}_i \triangleq (\mathrm{info}_i, L(i)) = (\mathrm{info}_j, \forall j \in \{i\} \cup N_1(i)). \tag{8}$$

The packet transmission can be performed by contention following the random access protocol proposed in [31]. In fact, it was shown in [31] that, by allowing each player to randomly select a time slot to transmit within a frame of length $1.44 \times \frac{1}{n}\sum_{i=1}^{n} |N_1(i) \cup N_2(i)|$, a packet delivery ratio of 99 percent can be achieved within 7 retransmission frames.

When player i does not transmit in a certain time slot, it listens to pick up control packets sent by its one-hop neighbors, and updates the sets $N_1(i)$, $N_2(i)$, and $L(i)$ accordingly. In particular, suppose that player i receives $\mathrm{pkt}_j = (\mathrm{info}_k, \forall k \in \{j\} \cup N_1(j))$ from its one-hop neighbor player j in a certain time slot. In this case, player i updates its one-hop neighbor set $N_1(i)$ by adding j to the set (i.e., $N_1(i) \leftarrow N_1(i) \cup \{j\}$), and updates its two-hop neighbor set $N_2(i)$ by including the set of one-hop neighbors of player j that are not one-hop neighbors of player i to the set (i.e., $N_2(i) \leftarrow N_2(i) \cup N_1(j) \setminus N_1(i)$). The information regarding player j in the set $L(i)$ is updated accordingly.

Notice that, after a few successful updates from its one-hop neighbors, player i will be able to obtain knowledge of the initial colors of players within its two-hop radius, and also knowledge of the one-hop and two-hop neighbors of these players (i.e., $N_1(k)$ and $N_2(k)$, $\forall k \in N_1(i) \cup N_2(i)$). The latter yields knowledge of the players within a four-hop radius of player i. With the above neighbor information, player i can then determine all of the maximal 2-cliques that it belongs to and the initial colors associated with their members. This provides each player with enough information to determine its best response. However, the color updates must be scheduled carefully in order to obtain the desired sequential best-response improvement path.

5.2 Scheduled Access Phase

In the scheduled access period, the players update their colors in a round-by-round fashion, with each player updating its action exactly once in each round. In each time slot, a player may compute an update of its own action, and compete for transmission of this action to its one-hop neighbors. To avoid collision, only one player within each two-hop neighborhood may be allowed to transmit. Following the neighborhood-aware contention resolution (NCR) method proposed in [32], we allow a player to transmit only if its priority is the highest among its one-hop and two-hop neighbors in the current time slot. The priority of player i in time slot t can be computed as

$$\mathrm{prio}(i, t) = \mathrm{hash}(i \ \mathrm{XOR} \ t) \ \mathrm{concatenate} \ i. \tag{9}$$

The hash value concatenated by the player's index i ensures that its priority is unique. Notice that player i is capable of computing the priority of all of its one-hop and two-hop neighbors (i.e., $\mathrm{prio}(j, t)$, for all $j \in N_1(i) \cup N_2(i)$) since the identities of these players have been obtained in the random access phase and, thus, is capable of determining locally without message exchange whether or not it is allowed to transmit in the current time slot.

Let a_i and r_i be the action of player i and the round that it is in at the beginning of a certain time slot, and let $\hat{a}_j(i)$ and $\hat{r}_j(i)$ be the action and the round information associated with player j that player i is aware of at this time (i.e., the information recorded by player i about player j). Note that $\hat{a}_i(i) = a_i$ and $\hat{r}_i(i) = r_i$. Player i is allowed to update its action and broadcast a coloring packet to its one-hop neighbors in time slot t if it has the highest priority among all nodes within its two-hop radius at this time (i.e., $\mathrm{prio}(i, t) > \mathrm{prio}(j, t)$, for all $j \in N_1(i) \cup N_2(i)$). The coloring packet contains locally recorded information about the actions and rounds of the

Algorithm 1. Best-Response Resolution (BRR) Algorithm (for player i in time slot t)

Input: $N_1(i)$, $N_2(i)$, $\{(\hat{a}_j(i), \hat{r}_j(i)), \forall j \in \{i\} \cup N_1(i) \cup N_2(i)\}$, W_i
Output: W_i.
for each player $k \in N_2(i)$ **do**
  if $\text{prio}(k,t) > \text{prio}(j,t), \forall j \in N_1(k) \cup N_2(k)$ **then**
    if $\hat{r}_k(i) \le r_i$ **then**
      $W_i \leftarrow W_i \cup \{k\}$
    end if
  end if
end for
for each $k' \in N_1(i)$ **do**
  if $\text{prio}(k',t) > \text{prio}(j,t), \forall j \in N_1(k') \cup N_2(k')$ **then**
    $W_i \leftarrow W_i - N_1(k')$
  end if
end for

player itself and also of those within its two-hop neighborhood (i.e., $\{(\hat{a}_j(i), \hat{r}_j(i)), \forall j \in \{i\} \cup N_1(i) \cup N_2(i)\}$). If an update is made, the new action is chosen based on player i's knowledge of the actions of its one-hop and two-hop neighbors, and is given by

$$a_i \leftarrow \arg\max_{a_i \in A_i} u_i(a_i, \hat{\mathbf{a}}_{-i}(i)), \tag{10}$$

where $\hat{\mathbf{a}}_{-i}(i) = (\hat{a}_1(i), \ldots, \hat{a}_{i-1}(i), \hat{a}_{i+1}(i), \ldots, \hat{a}_N(i))$. On the other hand, if player i is not allowed to transmit, then it listens for coloring packets sent from other players in its one-hop neighborhood. Suppose that a coloring packet is received from player $j \in N_1(i)$, which contains the information $\{(\hat{a}_k(j), \hat{r}_k(j)), \forall k \in \{j\} \cup N_1(j) \cup N_2(j)\}$. Then, for $k \in N_1(i) \cup N_2(i)$ such that $\hat{r}_k(j) > \hat{r}_k(i)$, its information at player i is updated as $(\hat{a}_k(i), \hat{r}_k(i)) \leftarrow (\hat{a}_k(j), \hat{r}_k(j))$. That is, the coloring information regarding player k is updated if the packet received from player j contains more up-to-date information about the player.

To ensure fast convergence towards an NE, we propose to have players make updates in a round-by-round fashion. That is, a player, say player i, is allowed to update its own action only if all of its one-hop and two-hop neighbors have at least updated their actions in the previous coloring round and that player i is aware of the most recent updates made by these neighbors (i.e., if $r_j \ge r_i$ and $\hat{r}_j(i) = r_j$, for all $j \in N_1(i) \cup N_2(i)$). Consequently, no two players within two-hops of each other can differ in rounds by more than 1, i.e., $|r_i - r_j| \le 1$ for i and j such that $j \in N_1(i) \cup N_2(i)$. However, since the coloring packets are sent only to one-hop neighbors, player i will not be able to receive immediate updates from its two-hop neighbors. Hence, even when a player has the highest priority and is allowed to update its own action in the current time slot, it may not be capable of making the correct decision until updates made by its two-hop neighbors have been received. To ensure that the players have consistent information about the coloring status of other nodes in the associated round, we propose the following *Best-Response Resolution (BRR)* algorithm, which achieves the task by keeping track of the index of players whose information may potentially be outdated.

In the BRR algorithm, each player, say player i, maintains a waiting vertex set W_i, which records the list of players' updates that it must wait for before a new local decision can be made. Therefore, when player i is given the priority to transmit, it will update its own action before transmitting a coloring packet only if

$$W_i = \emptyset. \tag{11}$$

If W_i is non-empty, player i simply broadcasts a coloring packet containing its old action (along with the updated information from its neighbors). On the other hand, if player $k \in N_2(i)$ is given the priority to transmit instead of player i and that $\hat{r}_k(i) \le r_i$, then player k is added to the wait list W_i, meaning that player i must wait to receive this update from player k before it can perform its next update. A player is removed from the waiting list W_i if player i receives an update of the player's action from one of the received coloring packets. BRR algorithm is run locally by each player in every time slot, and is summarized in Algorithm 1.

Fig. 3 gives an example with three players (i.e., players 1, 2, and 3) using the BRR algorithm to resolve coloring decisions over the color set $C = \{\text{red}, \text{green}, \text{blue}\}$. In the initial time slot after the random access phase, i.e., time slot 0, the waiting vertex sets are initialized as $W_1 = W_2 = W_3 = \emptyset$ and the initial round indices recorded by all players are set to 0 (i.e., $\hat{r}_j(i) = 0$, for all i and j). In time slot 1, player 1 has the highest priority and is allowed to update its action since $W_1 = \emptyset$. In this case, player 3 updates its waiting set as $W_3 = \{1\}$ since player 1 has the highest priority and $\hat{r}_1(3) \le r_3$. In this time slot, player 1 updates its action a_1 with the color blue since the choice yields the highest local utility. A coloring packet is then sent to its one-hop neighbor, which is player 2 in this case. Next, in time slot 2, player 3 has the highest priority but is not allowed to update its action since its waiting set W_3 is not yet empty. In time slot 3, player 2 has the highest priority and $W_2 = \emptyset$. Therefore, player 2 is allowed to update its action and chooses $a_2 = \text{green}$. Note that player 2 could have also chosen red in this case. The coloring packet is again broadcast to its neighbors. Upon receiving the coloring packet from player 2, player 3 removes player 1 from its waiting set W_3 since the received packet contains information regarding player 1's previous action update. In this case, $W_3 = \emptyset$. In time slot 4, player 3 is allowed to update its action since it has the highest priority and $W_3 = \emptyset$. The action $a_3 = \text{blue}$ is chosen since it maximizes its local utility function.

Theorem 2. *The distributed message passing protocol yields a sequential best-response improvement path and, thus, converges to an NE.*

Proof. Let $t_i^{(r)}$ be the time that player i determines its action in the r-th round, and let $a_i^{(r)}$ be the action that is chosen at this time. Recall that, by the design of the scheduled access period and the BRR algorithm, player i is able to make an update in the r-th round only if all of its one-hop and two-hop neighbors have at least updated their actions in round $r-1$ and that player i is aware of any updates made by these neighbors up to this time. This implies that $t_j^{(r-1)} < t_i^{(r)}$, for all $j \in N_1(i) \cup N_2(i)$. Therefore, at time $t_i^{(r)}$, the information $(\hat{a}_j(i), \hat{r}_j(i))$ that player i has acquired about player $j \in N_1(i) \cup N_2(i)$ is given by $\hat{a}_j(i) = a_j^{(r-1)}$ and $\hat{r}_j(i) = r-1$, if $t_j^{(r)} > t_i^{(r)}$, and $\hat{a}_j(i) = a_j^{(r)}$ and $\hat{r}_j(i) = r$, if

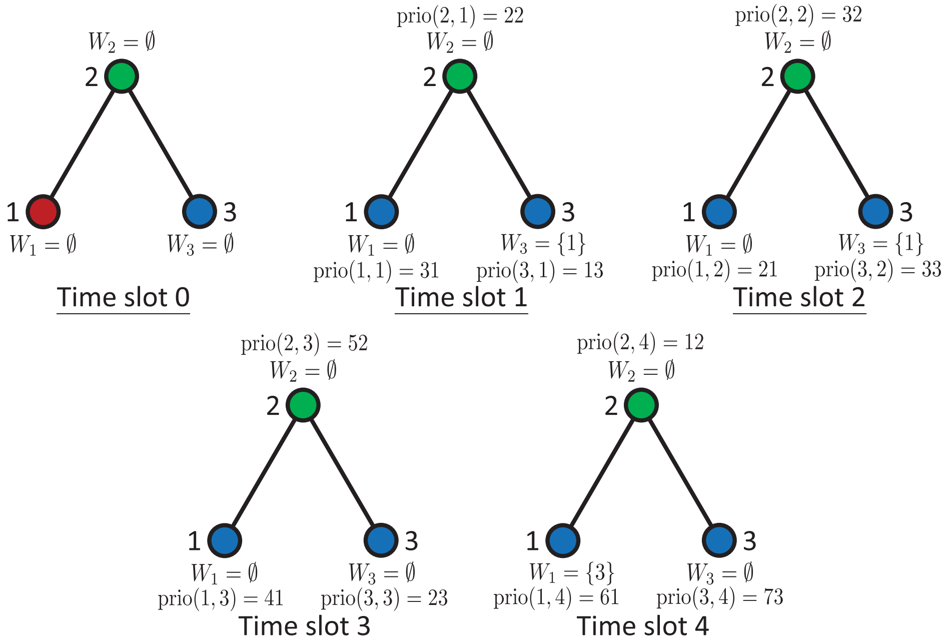


Fig. 3. An example of the best response resolution.

$t_j^{(r)} < t_i^{(r)}$. For $j \notin N_1(i) \cup N_2(i)$, the values of $\hat{a}_j(i)$ and $\hat{r}_j(i)$ can be arbitrary.

Let us define $\pi_r : \{1, \ldots, n\} \to \{1, \ldots, n\}$ as an ordering of vertices in the r-th round such that $t^{(r)}_{\pi_r(1)} \leq t^{(r)}_{\pi_r(2)} \leq \cdots \leq t^{(r)}_{\pi_r(n)}$. In this case, the action $a^{(r)}_{\pi_r(i)}$ made by player $\pi_r(i)$ in the r-th round must belong to the set

$$B_{\pi_r(i)}(\mathbf{a}^{(r)}_{\pi_r(i)}) = \arg\max_{a_{\pi_r(i)} \in A_{\pi_r(i)}} u_{\pi_r(i)}(a_{\pi_r(i)}, \hat{\mathbf{a}}_{-\pi_r(i)}(\pi_r(i))) \tag{12}$$

$$= \arg\max_{a_{\pi_r(i)} \in A_{\pi_r(i)}} u_{\pi_r(i)}(a_{\pi_r(i)}, \{a^{(r-1)}_{\pi_r(j)}\}_{j>i}, \{a^{(r)}_{\pi_r(j)}\}_{j<i}) \tag{13}$$

Hence, by letting $\boldsymbol{\rho}_r = \{\mathbf{a}[r,1], \ldots, \mathbf{a}[r,n]\}$ be the sequence of action profiles in the r-th round obtained by updating the players' actions one-by-one according to the order π_r so that $\mathbf{a}[r,i] = (a^{(r)}_{\pi_r(1)}, \ldots, a^{(r)}_{\pi_r(i)}, a^{(r-1)}_{\pi_r(i+1)}, \ldots, a^{(r-1)}_{\pi_r(n)})$, the sequence of action profiles $\boldsymbol{\rho} = \{\boldsymbol{\rho}_1, \boldsymbol{\rho}_2, \ldots, \boldsymbol{\rho}_r, \ldots\}$ forms a sequential best-response improvement path and, thus by Theorem 2.8 of [12], converges to an NE in a finite number of steps. □

The DTIC algorithm and the distributed message passing protocol can be performed continuously to adapt to changes in the environment. However, in a static network, a player can detect the occurrence of a convergent state by launching the so-called snapshot algorithm [33] whenever the actions of its one-hop and two-hop neighbors do not change in two consecutive time slots.

6 Performance Evaluation

In this section, we evaluate the performance of the DTIC algorithm by computer simulations. The DTIC algorithm is compared with the following four algorithms, all of which models the channel allocation problem as a form of graph coloring that allows for uncolored vertices (i.e., the incomplete coloring problem).

- *Centralized Algorithm:* Each player selects a color that is chosen the most among all vertices in the graph. The players' actions are updated in turn according to a centralized round-robin schedule. This requires global knowledge of the coloring status of all vertices.
- *Greedy Two-Hop Incomplete Coloring (GTIC):* Each player selects the color that is reused the most among its two-hop neighbors (but not used by its one-hop neighbors). Notice that this is in general done according to a centralized round-robin schedule, but can also be implemented using our proposed distributed message-passing protocol. Since the selection is done only once for each player, the GTIC algorithm can be viewed as the implementation of a single round of the DTIC algorithm with a different utility function (namely, a utility function that is equal to the number of two-hop neighbors with the same color).
- *Random Incomplete Coloring (RIC):* The RIC algorithm proposed in [11] is implemented in multiple rounds with each round consisting of the coloring slots period and the winner notification period. In the coloring slots period, each uncolored node i randomly selects a color c_i that is not yet chosen by its one-hop neighbors and generates a random value s_i. Then, it randomly selects a time slot within this period to broadcast its coloring message (c_i, s_i) to its one-hop neighbors. If node i receives a coloring message from node $j \in N_1(i)$ with $s_j \geq s_i$ and $c_i = c_j$, node i remains uncolored and removes the color from its available set of colors. Otherwise, node i marks itself as a winner and updates its color as c_i. Then, in the winner notification period, node i broadcasts a notification message indicating that it has chosen the color c_i. Following [11], the number of rounds is set

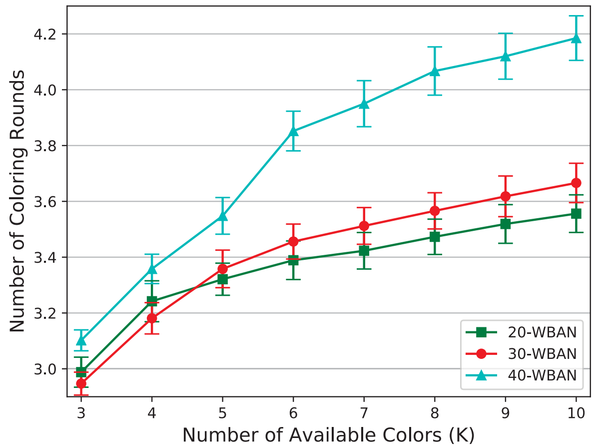


Fig. 4. Number of coloring rounds versus the number of available colors (i.e., channels) for the DTIC algorithm under different number of WBANs

as 5 and the number of time slots for the coloring slots and the winner notification periods are set as $40\frac{k}{5} - k$ and k, respectively, where k is the total number of colors (or channels).

- *Deflation via the Optimal Z3 Solver:* A sequential search of optimal coloring solutions is performed by using the Z3 solver, i.e., a Satisfiability Modulo Theories (SMT) solver [34]. This solver determines the feasibility of a set of colors on a specific graph in the traditional graph coloring problem. Specifically, given the total number of colors, vertices are removed in decreasing order of the sizes of their largest maximal 2-cliques until a feasible coloring is found. The optimal Z3 solver requires high computational complexity and, thus, can only be applied to problems with small size.

It is worthwhile to note that, even though graph coloring problems have been studied extensively in the past, most variations of the problem, to the best of our knowledge, require all nodes to be colored, albeit under different coloring constraints. The RIC algorithm proposed in [11] is the only work we could find that addresses the incomplete graph coloring problem (i.e., the problem that allows for uncolored nodes). Hence, it is adopted as our main comparison scheme. In our experiments, the network topology is generated by randomly deploying WBANs in a 10×10 m^2 area. Each WBAN is assumed to have an interference range of 3 meters. We use Bluetooth as the communication protocol between coordinators of WBANs and the duration of a time slot is 0.25 ms (milliseconds). The simulation curves are averaged over 100 random deployments of the WBANs and are shown with 95 percent confidence interval.

In Fig. 4, we show the number of coloring rounds required to reach the NE of the proposed DTIC algorithm in systems with 20, 30, and 40 WBANs, respectively. The total number of colors (or channels) k is varied from 3 to 10. We can see, in Fig. 4, that the number of coloring rounds increases from around 3 to 4.2 when there are 40 WBANs, and from 3 to 3.6 when there are 20 or 30 WBANs. This shows that the number of coloring rounds does not increase significantly with the increase in network size. In Table 1, we show the number of time slots required to reach convergence and the corresponding elapsed time given slot

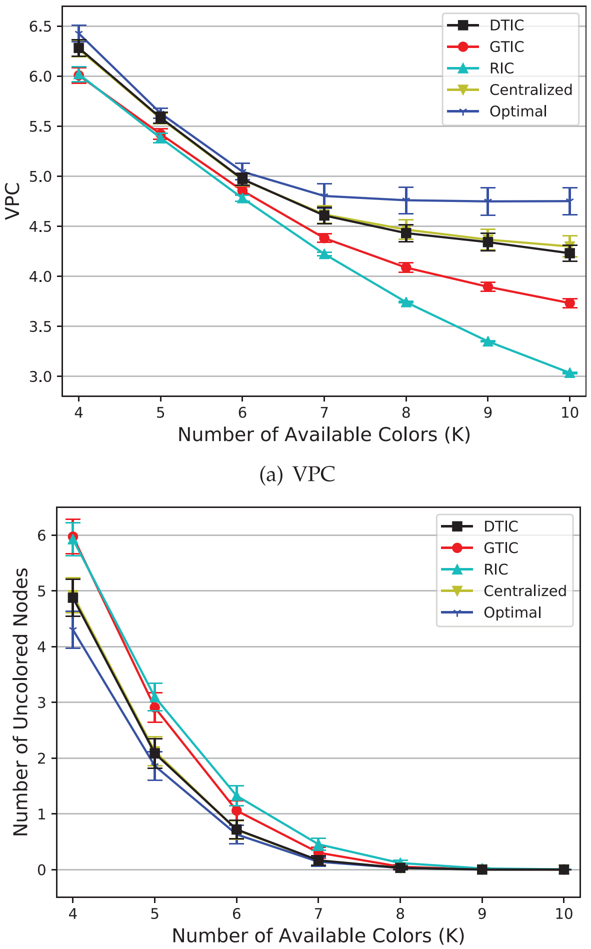
TABLE 1
Comparison of the Overhead Required in Different Algorithms for a System With 30 WBANs

Number of Colors	Number of Time Slots		
	DTIC	GTIC	RIC
3	220.87 slots (55.22 ms)	21.92 slots (5.48 ms)	120 slots (30 ms)
5	237.97 slots (59.49 ms)	21.92 slots (5.48 ms)	200 slots (50 ms)
7	242.63 slots (60.66 ms)	21.92 slots (5.48 ms)	280 slots (70 ms)

duration equal to 0.25 ms. The above values for DTIC is compared with GTIC and RIC, respectively. Notice that the first coloring round typically takes much less time than other rounds since the initial coloring status of all players is assumed to be known globally and, thus, no message exchange is required before local updates can be made. In this case, GTIC requires much less overhead since it performs only a single round of coloring. However, as we show later on in Fig. 8, the number of conflict-free nodes does not decrease significantly over 10^4 ms for players moving at average speed of 1 m/s. Hence, the coloring operation need not be performed frequently and, thus, the overhead required for DTIC to converge can be negligible (e.g., ~ 0.5 percent if the colors are updated every 10^4 ms).

In Fig. 5, we show the efficiency of the co-channel reuse with respect to the number of available colors k in the case with 30 WBANs. Recall that, in the incomplete coloring problem, a node can be left uncolored when no color can be chosen without conflict. Moreover, in the compared RIC algorithm, we ignore the loss due to packet collision, which may cause inconsistency in the nodes' understanding of the winner of each round, and eventually result in color conflict among neighboring nodes. This inconsistency does not occur in the DTIC algorithm due to the design of the proposed message passing protocol.

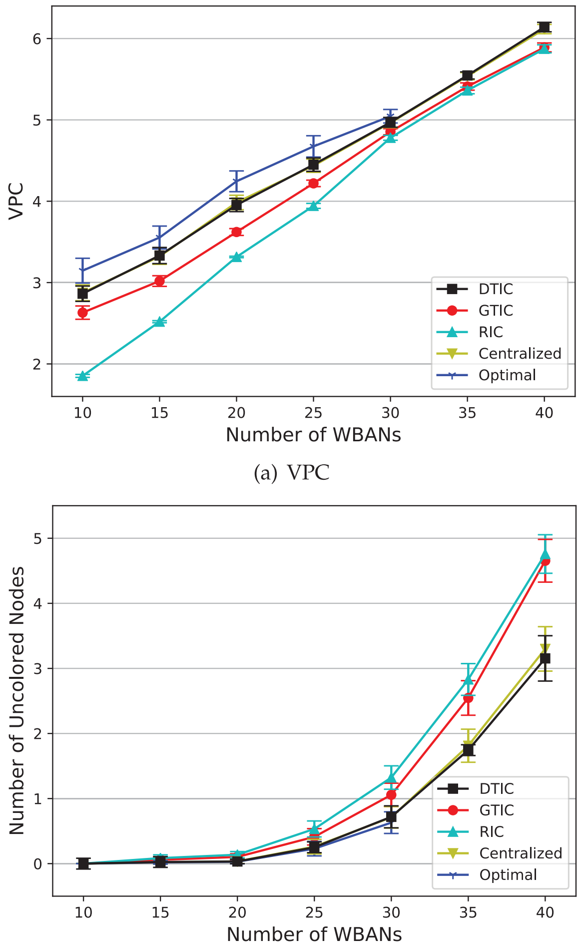
More specifically, in Fig. 5a, we measure the co-channel reuse efficiency in terms of the vertex-per-color (VPC), which is defined as the ratio between the number of colored nodes and the number of used colors. As expected, VPC decreases in all cases as the total number of colors k increases since fewer nodes need to be supported by each color as k becomes larger. We can see that the performance of the DTIC algorithm is close to that of the centralized algorithm, which has global knowledge and control of all nodes in the network. In fact, the curves associated with the two algorithms are so close that they are often indistinguishable. However, there is some loss compared to the optimal solution (obtained by the Z3 solver) when $k > 7$, but is uniformly better than the GTIC and the RIC algorithms. Notice that RIC yields the smallest VPC value since it does not take into consideration the channel reuse situation among its two-hop neighbors. In Fig. 5b, we examine the number of uncolored nodes under a similar setting. Similar to Fig. 5a, the DTIC algorithm performs close to the centralized algorithm, but outperforms the GTIC and the RIC algorithms. In particular, when $k = 4$, the DTIC and the centralized



(a) VPC

(b) Number of uncolored nodes.

Fig. 5. Comparison of different coloring algorithms for a system with 30 WBANs.



(a) VPC

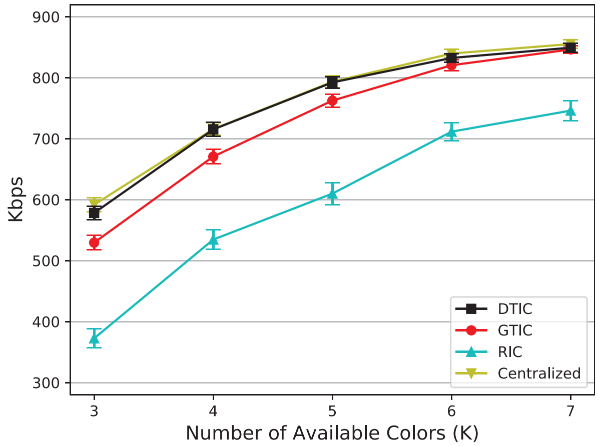
(b) Number of uncolored nodes.

Fig. 6. Comparison of different coloring algorithms for the case with 6 available colors.

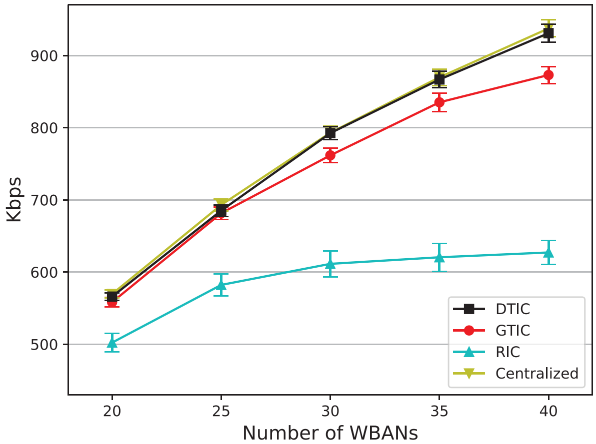
algorithms yield about 4.87 uncolored nodes while the GTIC and the RIC algorithms yield approximately 5.9 uncolored nodes. The optimal Z3 solver yields only 4.3. When $k \geq 8$, the number of colors is more than sufficient to serve the 30 WBANs and, thus, the number of uncolored nodes approaches zero. It is worthwhile to note that the number of uncolored nodes correspond to the number of WBANs that are not able to transmit without interfering with other WBANs. These WBANs will be restricted from transmitting and directly corresponds to the downtime of these nodes. In this experiment, the number of uncolored nodes can reach around 4 to 6 when only 4 colors are available. This constitutes a significant portion of the 30 WBANs in the experiment. Even though the number of uncolored nodes decreases as the number of channels increases, this requires the devices to have the ability to transmit over more channels which directly increases the cost of the devices.

In Fig. 6, we examine the co-channel reuse efficiency and the number of uncolored nodes for different numbers of WBANs. Here, the number of available colors is fixed as $k = 6$. The solution obtained by the optimal Z3-solver is evaluated only up to 30 WBANs since the complexity becomes prohibitive beyond that. In Fig. 6a, we can see that the VPC value increases as the number of WBANs becomes larger. The performances of the DTIC and the centralized algorithms are similar, and is only slightly worse than that of the optimal Z3-solver. The DTIC algorithm again performs better than the GTIC and the RIC algorithms. In Fig. 6b, we can see that, when the number of WBANs is 40, both the DTIC and the centralized algorithms yield an average of 3.15 uncolored nodes whereas the GTIC and the RIC algorithms yield 4.65 and 4.76 uncolored nodes respectively. Moreover, when the number of WBANs is below 20, the number of uncolored nodes approaches zero since 6 colors are sufficient to serve the WBANs in this case.

In Fig. 7, we evaluate the throughput with respect to the number of available colors k and the number of WBANs. Here, we assume that the WBANs may be changing positions over time according to the Gauss-Markov mobility model [35]. The average speed of a WBAN is 1 m/s and the randomness factor is $\alpha = 0.3$. The data rate of a WBAN is set as 30 Kbps according to [27]. The graph topology is updated every 100 ms (i.e., 400 time slots) and the throughput is averaged over a duration of 10^4 ms. We do not compare with the optimal Z3-solver solution here since it cannot be run in practice due to its high computational complexity. In Fig. 7a, throughput is shown with respect to the number of available colors when the number of WBANs is 30. We can see that the DTIC algorithm performs close to the centralized scheme and significantly better than the



(a) Throughput versus the number of available colors.



(b) Throughput versus the number of WBANs.

Fig. 7. Comparison of the throughput between different coloring algorithms.

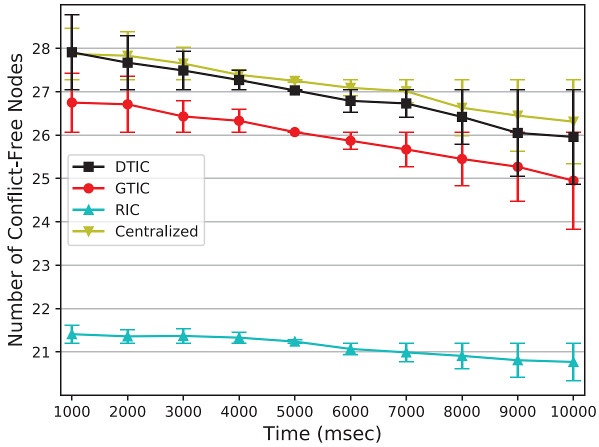


Fig. 8. Number of conflict-free nodes versus time for a system with 30 WBANs and $k = 5$.

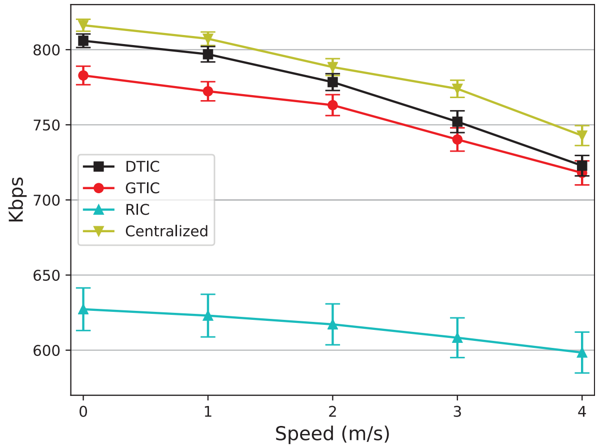


Fig. 9. Throughput versus mobility speed for a system with 30 WBANs and $k = 5$.

GTIC and the RIC algorithms. When $k = 4$, the centralized and the DTIC algorithms achieve approximately 715 Kbps while the GTIC and the RIC algorithms achieve only 671 Kbps and 535 Kbps, respectively. In Fig. 7b, throughput is shown with respect to the number of WBANs for the case where the number of available colors is fixed as $k = 5$. When the number of WBANs is small, the performances of the compared algorithms do not differ significantly. However, as the number of WBANs increases, the disparity between the performances of the RIC, GTIC, and DTIC algorithms becomes more evident. Yet, the DTIC algorithm performs close to the centralized solution in all cases. Notice that packet collision is considered in this case and, thus, RIC suffers more as the number of WBANs increases. When the number of WBANs is 40, the DTIC algorithm achieves approximately 931 Kbps whereas the GTIC and the RIC algorithms achieve only 873 Kbps and 627 Kbps, respectively.

To further examine the impact of mobility on the system performance, we show, in Fig. 8, the number of conflict-free nodes as time elapses for a system with 30 WBANs and $k = 5$ colors. This corresponds to the scenario considered in Fig. 7 where the coloring is performed only initially while the players locations are varied every 100 ms. We can see that, with average mobility of 1 m/s, the graph topology does not vary significantly within the observed time period of 10^4 ms. The proposed DTIC algorithm maintains approximately a fixed advantage over GTIC and RIC, despite the user movement. Hence, it is sufficient to perform the coloring procedure only once every 10^4 ms in this case, which makes the overhead in Table 1 negligible. In Fig. 9, we show the throughput versus different average mobility speeds for a system with 30 WBAN and $k = 5$. We can see that the throughput decreases as the speed increases since the graph topology changes more rapidly over time. In this case, nodes may more easily result in conflict after a certain amount of time, at which point a recoloring is required.

7 Conclusion

In this work, we addressed the coexistence problem between multiple WBANs using a game-theoretic approach. In particular, we modeled the channel allocation problem as a graph coloring problem, and proposed a distributed two-hop incomplete coloring algorithm to solve the coloring problem in a distributed fashion. Incomplete coloring allows nodes to remain uncolored if it is necessary to avoid conflict between neighboring nodes. The proposed DTIC algorithm utilizes coloring information from its two-hop neighbors to determine a local choice that aims to maximize the channel reuse. Moreover, we designed a distributed message-passing protocol that can be used to ensure that the action updates are done in a consistent manner. Numerical results showed that the proposed DTIC algorithm performs close to the centralized algorithm, and improves upon the GTIC and the RIC algorithms.

Acknowledgments

The authors would like to thank Mr. Cheng-Hung He for his assistance in completing the simulations of this work. This work was supported in part by the Ministry of Science and Technology, Taiwan, under Grants 107-2622-8-007-014-TA, 108-2634-F-007-004, and 109-2634-F-007-015.

References

[1] G.-T. Chen, W.-T. Chen, and S.-H. Shen, “2L-MAC: A mac protocol with two-layer interference mitigation in wireless body area networks for medical applications,” in *Proc. IEEE Int. Conf. Commun.*, 2014, pp. 3523–3528.
[2] S. Movassaghi, M. Abolhasan, and D. Smith, “Smart spectrum allocation for interference mitigation in wireless body area networks,” in *Proc. IEEE Int. Conf. Commun.*, 2014, pp. 5688–5693.
[3] M. M. Dhanvijay and S. C. Patil, “Internet of things: A survey of enabling technologies in healthcare and its applications,” *Comput. Netw.*, vol. 153, pp. 113–131, 2019.
[4] L. Tan, Z. Feng, W. Li, Z. Jing, and T. A. Gulliver, “Graph coloring based spectrum allocation for femtocell downlink interference mitigation,” in *Proc. Wireless Commun. Netw. Conf.*, 2011, pp. 1248–1252.
[5] R. Y. Chang, Z. Tao, J. Zhang, and C.-C. Kuo, “A graph approach to dynamic fractional frequency reuse (FFR) in multi-cell OFDMA networks,” in *Proc. IEEE Int. Conf. Commun.*, 2009, pp. 1–6.
[6] E. Pateromichelakis, M. Shariat, A. U. Quddus, and R. Tafazolli, “On the evolution of multi-cell scheduling in 3GPP LTE/LTE-A,” *IEEE Commun. Surveys Tuts.*, vol. 15, no. 2, pp. 701–717, Second Quarter 2013.
[7] A. Mishra, S. Banerjee, and W. Arbaugh, “Weighted coloring based channel assignment for WLANs,” *ACM SIGMOBILE Mobile Comput. Commun. Rev.*, vol. 9, no. 3, pp. 19–31, Jul. 2005.
[8] S. Ramanathan, “A unified framework and algorithm for channel assignment in wireless networks,” *Wireless Netw.*, vol. 5, no. 2, pp. 81–94, Mar. 1999.
[9] W. Wang and X. Liu, “List-coloring based channel allocation for open-spectrum wireless networks,” in *Proc. IEEE Veh. Technol. Conf.*, 2005, pp. 690–694.
[10] C. Peng, H. Zheng, and B. Y. Zhao, “Utilization and fairness in spectrum assignment for opportunistic spec-trum access,” *ACM Mobile Netw. Appl.*, vol. 11, no. 4, pp. 555–76, 2006.
[11] S.-H. Cheng and C.-Y. Huang, “Coloring-based inter-WBAN scheduling for mobile wireless body area networks,” *IEEE Trans. Parallel Distrib. Syst.*, vol. 24, no. 2, pp. 250–259, Feb. 2013.
[12] Q. D. L. A, Y. H. Chew, and B.-H. Soong, *Potential Game Theory - Applications in Radio Resource Allocation*. Berlin, Germany: Springer, 2016.
[13] Ö. Johansson, “Simple distributed $(\Delta + 1)$-coloring of graphs,” *Inf. Process. Letts.*, vol. 70, no. 5, pp. 229–232, Jun. 1999.
[14] D. G. Harris, J. Schneider, and H.-H. Su, “Distributed $(\Delta + 1)$-coloring in sublogarithmic rounds,” *J. ACM*, vol. 65, no. 4, pp. 19:1–19:21, Apr. 2018.
[15] Y.-J. Chang, W. Li, and S. Pettie, “An optimal distributed $(\Delta + 1)$-coloring algorithm?” in *Proc. ACM Symp. Theory Comput.*, 2018, pp. 445–456.
[16] A. Bhartia, D. Chakrabarty, K. Chintalapudi, L. Qiu, B. Radunovic, and R. Ramjee, “IQ-Hopping: Distributed oblivious channel selection for wireless networks,” in *Proc. ACM Int. Symp. Mobile Ad Hoc Netw. Comput.*, 2016, pp. 81–90.
[17] A. Checco and D. J. Leith, “Fast, responsive decentralized graph coloring,” *IEEE/ACM Trans. Netw.*, vol. 25, no. 6, pp. 3628–3640, Dec. 2017.
[18] H. Hernández and C. Blum, “Distributed graph coloring: An approach based on the calling behavior of japanese tree frogs,” *Swarm Intell.*, vol. 6, pp. 117–150, Jun. 2012.
[19] Z. Han, D. Niyato, W. Saad, T. Baar, and A. Hjrungnes, *Game Theory in Wireless and Communication Networks: Theory, Models, and Ap-plications*, 1st ed. New York, NY, USA: Cambridge University Press, 2012.
[20] N. Nie and C. Comaniciu, “Adaptive channel allocation spectrum etiquette for cognitive radio net-works,” *Mobile Netw. Appl.*, vol. 11, no. 6, pp. 779–797, 2006.
[21] J.-P. Sheu, Z.-X. Wu, C. Ma, and R. Jagadeesha, “Interference-aware channel allocation algorithm with game theoretic approach for cognitive radio networks,” in *Proc. IEEE Int. Conf. Parallel Distrib. Syst.*, 2014, pp. 41–46.
[22] S. Misra and S. Sarkar, “Priority-based time-slot allocation in wireless body area networks during medical emergency situations: An evolutionary game-theoretic perspective,” *IEEE J. Biomed. Health Informat.*, vol. 19, no. 2, pp. 541–548, Mar. 2015.
[23] P. N. Panagopoulou and P. G. Spirakis, “A game theoretic approach for efficient graph coloring,” in *Proc. Int. Symp. Algorithms Comput.*, 2008, pp. 183–195.
[24] I. Chatzigiannakis, C. Koninis, P. N. Panagopoulou, and P. G. Spirakis, “Distributed game-theoretic vertex coloring,” in *Proc. Int. Conf. Princ. Distrib. Syst.*, 2010, pp. 103–118.
[25] B. Babadi and V. Tarokh, “GADIA: A greedy asynchronous distributed interference avoidance algorithm,” *IEEE Trans. Inf. Theory*, vol. 56, no. 12, pp. 6228–6252, Dec. 2010.
[26] J. Wang, Y. Xu, A. Anpalagan, Q. Wu, and Z. Gao, “Optimal distributed interference avoidance: Potential game and learning,” *Trans. Emerg. Telecommun. Technol.*, vol. 23, no. 4, pp. 317–326, 2012.
[27] B. Zhen, M. Patel, S. Lee, E. Won, and A. Astrin, “TG6 technical requirements document (TRD),” Tech. Rep. P802.15–08-0644-09-0006, 2008.
[28] M. R. Garey and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*. New York, NY, USA: W. H. Freeman & Co., 1990.
[29] V. B. Foad, M. Pajouh, A. Veremyev, *Examining Robustness and Vulnerability of Networked Systems*. IOS Press, Amsterdam, Netherlands, 2014, ch. Analysis and Design of Robust Network Clusters with Bounded Diameter, pp. 141–160.
[30] V. Rajendran, K. Obraczka, and J. J. Garcia-Luna-Aceves, “Energy-efficient, collision-free medium access control for wireless sensor networks,” *Wireless Netw.*, vol. 12, no. 1, pp. 63–78, 2006.
[31] L. Bao and J. J. Garcia-Luna-Aceves, “Hybrid channel access scheduling in ad hoc networks,” in *Proc. IEEE Int. Conf. Netw. Protoc.*, 2002, pp. 46–57.
[32] L. Bao and J. J. Garcia-Luna-Aceves, “A new approach to channel access scheduling for ad hoc networks,” in *Proc. ACM Int. Conf. Mobile Comput. Netw.*, 2001, pp. 210–221.
[33] K. M. Chandy and L. Lamport, “Distributed snapshots: Determining global states of distributed systems,” *ACM Trans. Comput. Syst.*, vol. 3, pp. 63–75, 1985.
[34] L. D. Moura and N. Bjorner, “Z3: An efficient SMT solver,” in *Proc. Int. Conf. Tools Algorithms Construction Anal. Syst.*, 2008, pp. 337–340.
[35] V. Tolety, “Load reduction in ad hoc networks using mobile servers,” Master’s thesis, Colorado School of Mines, 1999.

Kai-Ju Wu received the BS and MS degrees in computer science from National Tsing Hua University, Hsinchu, Taiwan, in 2016 and 2018, respectively. He is currently an engineer with Phison Electronics Corporation. His research interests include wireless body area network (WBAN) and bluetooth protocols.

Y.-W. Peter Hong (Senior Member, IEEE) received the BS degree in electrical engineering from National Taiwan University, Taipei, Taiwan, in 1999, and the PhD degree in electrical engineering from Cornell University, Ithaca, NY, USA, in 2005. He joined the Institute of Communications Engineering and the Department of Electrical Engineering at National Tsing Hua University (NTHU), Hsinchu, Taiwan, in 2005, where he is currently a full professor. He also currently serves as Director of the Institute of Communications Engineering, and Associate Vice President of Research and Development at NTHU. His research interests include multiuser communications, physical layer security, machine learning for wireless communications, and distributed signal processing for IoT and sensor networks. He received the IEEE ComSoc Asia-Pacific Outstanding Young Researcher Award in 2010, the Y. Z. Hsu Scientific Paper Award in 2011, the National Science Council Wu Ta-You Memorial Award in 2011, the Chinese Institute of Electrical Engineering Outstanding Young Electrical Engineer Award in 2012, the Best Paper Award from the Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC) in 2013, and the Ministry of Science and Technology Outstanding Research Award in 2019. He served as an associate editor of *IEEE Transactions on Signal Processing* and *IEEE Transactions on Information Forensics and Security*, and is currently an editor of *IEEE Transactions on Communications*. He was chair of the IEEE ComSoc Taipei Chapter, in 2017-2018, and co-chair of the Technical Affairs Committee and the Information Services Committee of the IEEE ComSoc Asia-Pacific Board, in 2014-2015 and 2016-2019, respectively.

Jang-Ping Sheu (Fellow, IEEE) received the B.S. degree in computer science from Tamkang University, Taiwan, Republic of China, in 1981, and the M.S. and Ph.D. degrees in computer science from National Tsing Hua University, Taiwan, Republic of China, in 1983 and 1987, respectively. He is currently a chair professor at the Department of Computer Science, National Tsing Hua University. He was a chair of the Department of Computer Science and Information Engineering, National Central University from 1997 to 1999. He was a Director of Computer Center, National Central University from 2003 to 2006. He was a director of the Computer and Communication Research Center from 2009 to 2015, National Tsing Hua University. He was an associate dean at the College of Electrical and Computer Science from 2016 to 2017, National Tsing Hua University. His current research interests include wireless communications, mobile computing, Internet of Things, and UAV-assisted communication systems. He was an associate editor of the IEEE Transactions on Parallel and Distributed Systems and International Journal of Sensor Networks. He is an advisory board member of the International Journal of Ad Hoc and Ubiquitous Computing and International Journal of Vehicle Information and Communication Systems. He received the distinguished research awards of the National Science Council of the Republic of China in 1993-1994, 1995-1996, and 1997-1998. He received the Distinguished Engineering Professor Award of the Chinese Institute of Engineers in 2003. He received the K. -T. Li Research Breakthrough Award of the Institute of Information and Computing Machinery in 2007. He received the Y. Z. Hsu Scientific Chair Professor Award and Pan Wen Yuan Outstanding Research Award in 2009 and 2014, respectively. He received the Academic Award in Engineering from the Ministry of Education and Medal of Honor in Information Sciences from the Institute of Information and Computing Machinery in 2016 and 2017, respectively. He received the TECO awards in 2019. He is a member of the Phi Tau Phi Society.