AUT Journal of Modeling and Simulation

AUT J. Model. Simul., 56(1) (2024) 19-32
DOI: 10.22060/miscj.2024.23295.5363

BlueMIC: Coordination for Mitigating Inter-Piconet Interference in BLE5 Networks

Hamed Khanmirza [1]*, Peyman Teymoori [2]

[1] IoT Lab, Department of CE, K. N. Toosi University of Technology, Tehran, Iran
[2] School of Business, University of South-Eastern Norway, Drammen, Norway

ABSTRACT: The proliferation of Bluetooth Low Energy (BLE) in diverse applications has led to a dense clustering of piconets in confined spaces. BLE operates within the congested 2.4GHz ISM band, shared by numerous short-range wireless protocols. Frequency hopping and Adaptive Frequency Hopping (AFH) mechanisms have been introduced in Bluetooth to reduce interference with other protocols in the same band. However, research shows these mechanisms are ineffective in reducing inter-piconet interferences. These studies have demonstrated the relationship between the number of piconets and the level of interference. In this paper, we investigate the relationship between the similarity of channel maps and the number of neighbouring piconets in interference intensity. Then, we present a light and practical coordination framework to address this challenge. Our solution employs a BLE gateway to detect its surrounding piconet masters and computes a better channel map to reduce the impact of inter-piconet interferences. We also introduce the Isolated Channels (ICA) algorithm for channel allocation of neighbouring piconets with controlled channel overlaps for BLE5. Simulation results show a 20 to 60% reduction of interference level in environments with high to moderate inter-piconet interferences. To the best of our knowledge, this is the first practical BLE5-compatible solution for mitigating the inter-piconet interference problem and does not require modifying the standard stack.

Review History:

Received: Jun. 23, 2024
Revised: Aug. 112, 2024
Accepted: Sep. 02, 2024
Available Online: Sep. 05, 2024

Keywords:

Interference
BLE5 Inter-Piconet Interference
Frequency Hopping
IoT Networks

1- Introduction

The Internet of Things (IoT) is increasingly used in fields like medicine, industry, entertainment, shopping, automation, and smart homes. Among various low-power protocols developed for IoT, Bluetooth Low Energy (BLE) stands out for its widespread use and features [1,2]. The fifth version, BLE5, offers enhanced speed, range, security, and efficiency [3]. BLE shares the crowded 2.4 GHz ISM band with other protocols like WiFi, making optimal usage challenging.

In BLE, devices form a "piconet" through a master-slave network. The cornerstone of Bluetooth's interference mitigation strategy is Frequency Hopping (FH), a technique that orchestrates synchronized channel changes among communicating devices over time. As the time spent in each channel is very short, and the hop sequence of each link is different, the possibility of interference in Bluetooth links is reduced. To further mitigate interference, BLE embeds the Adaptive Frequency Hopping (AFH) mechanism, included since Bluetooth 1.2, which uses a 40-bit channel map to guide devices away from channels with poor conditions [4].

While Adaptive Frequency Hopping (AFH) is generally effective at reducing interference, its performance drops significantly in densely populated environments [5-9] like stadiums [10] or body area networks [11]. In such settings, as interference increases, master nodes change their channel maps to avoid problematic channels. However, due to the random selection of channels, different masters in various piconets can end up choosing the same channels simultaneously, marking them as bad and repeatedly oscillating back to previous channels. This "back-and-forth" selection process degrades the efficiency of channel usage and the quality of communication links. Several studies [5-7, 9] have shown the noticeable throughput degradation of piconets operating within a close range even using the latest versions of Bluetooth. For instance, [9] reports a 60% throughput reduction with only five piconets.

This paper aims to address the issue of inter-piconet channel selection in densely-deployed environments. To begin with, we present a comprehensive analysis highlighting the significant correlation between the similarity of piconet channel maps and the severity of inter-piconet interference. We demonstrate that the default channel selection algorithm employed in BLE5 effectively mitigates interference even when there are only a few differing channels among the piconets. However, in scenarios where the number of piconets passes a specific limit or the channel maps exhibit a high degree of similarity, which is common among closely located piconets in confined areas, the interference level increases

*Corresponding author's email: h.khanmirza@kntu.ac.ir

exponentially.

To address the inter-piconet problem, this paper proposes the *BlueMIC* framework, a fast, and lightweight solution for masters to choose low-interference connection channels smartly. In BlueMIC, masters regularly advertise their channel map to a BLE gateway. The gateway acts as a coordinator of the surrounding masters. It receives the channel map of masters and then calculates and sends globally low-interferred channel maps for each.

This paper presents and assesses two algorithms for generating low-interference channel maps in BLE5 for gateway devices. The first approach, Iterative Channel Removal (ICR), involves masters sharing access address information to enable the gateway to create a Collision Map using the CSA#2 algorithm. Channels with high collisions are identified and excluded until interference falls below a specified threshold. The second method, Isolated Channels Algorithm (ICA), defines a set of preferred channels for each master and restricts the use of others, facilitating controlled channel overlap to enhance link throughput and accommodate more piconets. This strategy optimizes channel selection to reduce interference effectively, even when access address information is not directly accessible at the application layer in many SDKs. This paper makes several key contributions to the field, including:

1- It elucidates the relationship between the quantity of good and distinct channels within the channel map of piconets and the intensity of inter-piconet interference. Through empirical analysis and experimentation, we provide compelling evidence to showcase how variations in the number of such parameters directly influence the severity of interference between piconets.

2- While most related works have only demonstrated the existence of the inter-piconet problem, we have presented a practical solution for the issue in BLE5.

3- The proposed framework uses the built-in mechanisms of BLE5 and does not rely on any external network or mechanism to reduce inter-piconet interference.

4- The proposed channel assignment algorithm is practical and can be implemented in the application layer without changing the lower layers.

The paper is organized as follows. Section 2 reviews the related literature, and section 3 describes the new channel selection algorithm in BLE5. In section 4, we provide our interference model, and in section 5, we present the architecture of our proposal and channel map selection algorithms. Section 6 demonstrates the evaluation results, and finally, section 7 concludes the paper.

2- Literature Review

The coexistence of WLAN, WPAN and LPWAN [12] technologies in the unlicensed shared ISM band has a long history of research [1, 13-16]. This line of research focuses on investigating the impact of these protocols on each other and designing new adaptive and interference-aware mechanisms to reduce the effect of interference [17, 28].

Some Bluetooth-related coexistence solutions [18-20] have also addressed the inter-piconet interference issue. Zhang et al. [21] divide Bluetooth channels into multiple independent sets. Each piconet randomly selects and changes a set in case of intense interference. However, their method loses its capability in dense environments. Sun et al. in [20] specifically focus on *multi-piconet coexistence* issues and present a channel selection algorithm within a cooperative framework like [19]. By manipulating the 48-bit unique address and CLK (a 28-bit binary counter) of piconet masters, a dedicated coordinator parallelizes their hopping sequence. The coordinator also marks the WiFi-occupied channels as "used" so that piconets avoid them. These methods are designed for earlier versions of Bluetooth and need intense modifications to the standard stack.

Hu et al. [22] use a central controller to reduce the interference between piconets. The controller communicates with masters through the LTE network infrastructure and recommends the best channel map for each master based on neighbourhood interference. This solution necessitates having master nodes equipped with Bluetooth and LTE interfaces, making them expensive and energy-consuming. However, it can be useful in scenarios where piconets are located in specific, out-of-reach geographical locations, or there is a need for remote control.

Another branch of studies investigates the performance of close piconets. Mazzenga et al. [5,6] present a packet loss model for a large number of piconets in a limited area. They report up to 60% packet loss in a $20 \times 20 m^2$ area with 40 piconets. Authors of [7] developed a model to measure the probability of selecting a channel. Then, they use this model to predict the collision probability in the presence of communicating pairs. Their model predicts up to 65% collision with 40 communicating pairs.

Hallez et al. [23] compare the performance of channel selection algorithms 1 and 2 (CSA#1 and CSA#2) in different interfering environments. They show that although the collision probability with the new channel selection algorithm is almost halved compared to CSA#1, nevertheless, with only ten active piconets, the probability of collision in a channel is more than 70%. Similar results were achieved in the work of Bocker et al. [8] for BLE5. They concluded that with a packet error rate (PER) of 1% and 1000 events in a day, only ten piconets could be active in one neighbourhood. Ancans et al. [9], through an experiment with only five active piconets, have measured a 60% throughput loss in large connection intervals (300-400 ms). Meanwhile, the presence of a WiFi network had a constant 30% negative impact on the throughput. All of these simulations and experiments emphasize that the shared ISM band is not managed well, and it is necessary to improve the methods of interference detection and management and channel selection [24, 27].

3- BLE5 Channel Selection Algorithm

Alongside significant improvements provided by the BLE5 standard compared to version 4, this version has introduced a new channel selection algorithm which, according to research, has more than doubled the probability

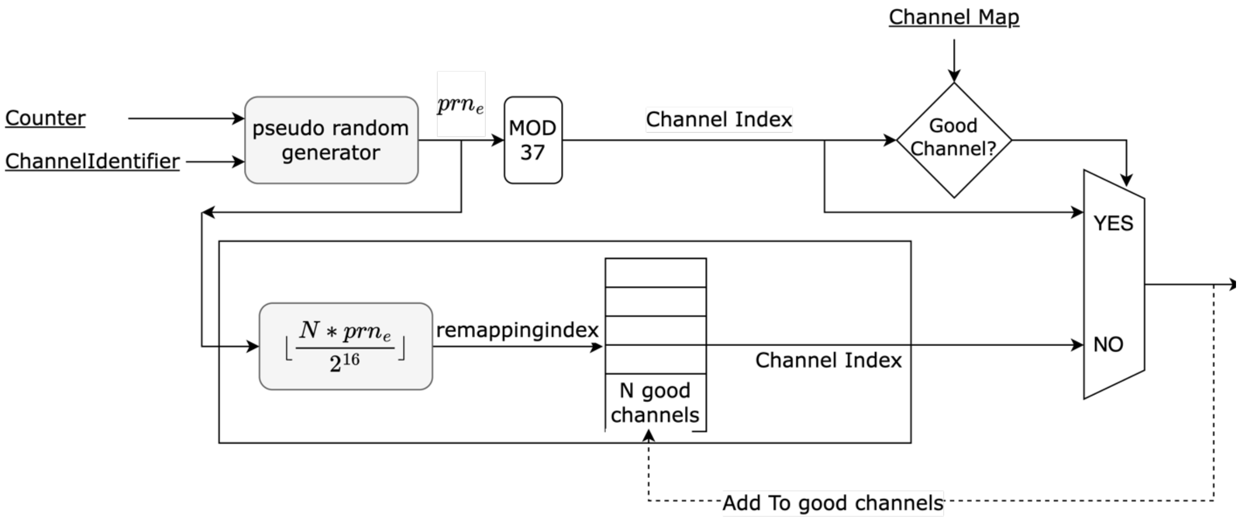


Fig. 1. BLE5 Channel Selection Algorithm #2 Block Diagram (adopted from [3])

of avoiding interference [23]. Channel Selection Algorithm 2 (CSA#2) is also a significantly intricate and challenging method to find or estimate the next hopping channel index [3]. Figure 1 shows the steps of this algorithm. CSA#2 uses the following three inputs to generate the channel index sequence:

1- Channel Map

2- *AccessAddress* : is a unique random identifier generated for each connection that makes a connection distinguishable from other connections. After establishing a connection, a BLE radio only has to listen for packets that match this address.

3-ChannelIdentifer: it is calculated from Access Address with the equation:

$ChannelIdentifier = AccessAddress_{(31-16)} \oplus AccessAddress_{(15-0)}$

4- *Counter* : a 16-bit integer number, updated in each connection event

CSA#2, at the first step, generates a 16-bit pseudo-random number, prn_e , using Counter and *ChannelIdentifier* parameters. If $prn_e \bmod 37$ is an index of a good channel, it is selected as the next hopping channel, and the process is finished. If prn_e does not refer to a good channel, the next index in the good channels set is calculated by $remappingindex = {N * prn_e}/{2^{16}}$, where N is the number of good channels on the current channel map.

4- Interference Model

This section introduces the interference model and how we measure the interference severity. The definition of parameters related to the probability of interference, in this section, is inspired by the model presented in [22].

Probability of System Interference (*PoI*) computed by Eq. (1) measures the probability of inter-piconet interference. This parameter shows the probability of interference of a piconet with other piconets. In this equation, n is the number of piconets in the range, L refers to the frequency hopping length, and $N(p)$ is the number of time segments the piconet, p, interferes with other piconets.

$$\Pi = \frac{1}{n \times L} \sum_{1}^{n} N(p) \tag{1}$$

The Interference Probability of a channel in a piconet is obtained from:

$$\Pi_c(c,p) = \frac{N(p,c)}{L}, c \in [0,39] \tag{2}$$

Where *N(p,c)* refers to the number of time segments the piconet, p, interferes with other piconets in channel c. The total interference probability of a channel denoted by $I_c(c)$ is given by Eq. (3), which effectively is the sum of the $Đ_c(p,c)$ in all piconets.

$$\Pi_c(c) = \frac{1}{n} \sum_{p=1}^{n} \Pi_c(c,p) \tag{3}$$

Assuming *G(p)* is the set of good channels of a piconet, the interference probability of a piconet is computed as follows:

Table 1. Summary of Notations

Notation	Description
M	the channel map bit sequence
$\|M\|$	number of usable channels in a channel map
N	number of piconets
L	total number of frequency hoppings
$G(p)$	set of good channels for piconet
Parameters computed using piconets' hopping sequences	
$N(p)$	number of hoppings piconet p collides with other piconets
$N(p,c)$	number of hoppings piconet p collides with other piconets in channel c
$C(p,l)$	number of piconets, piconet p collides with them in hopping of l
Π	Interference probability of the system
$\Pi_c(c,p)$	Interference probability of a channel in a piconet
$\Pi(c)$	Interference probability of a channel
$\Pi(p)$	Interference probability of a piconet
$S(p)$	Interference severity of a piconet
S	Interference severity of the system
Parameters computed using piconets' channel maps	
$N^M(p)$	number of common good channels between piconet p and other piconets
$N^M(p,c)$	if channel c is in good channels of piconet p and other piconets
$C_c^M(p,c)$	number of piconets, channel c is also in their good channel
$C^M(p,c)$	average number of piconets that have common good channels with piconet p
Π^M	Interference probability of the system
$\Pi_c^M(c,p)$	Interference probability of a channel in a piconet
$\Pi_c^M(c)$	Interference probability of a channel
$\Pi_c^M(p)$	Interference probability of a piconet
S^M	Interference severity of the system

$$\Pi_p(p) = \sum_{c \in G(p)} \Pi_c(c,p) \tag{4}$$

In essence, the set of interference probability parameters shows whether a piconet should expect interference on at least one of its channels and how probable it is. However, they do not reflect the intensity of the interference. Let $C(p,l)$ show the number of conflicts for a piconet in its l^{th} hopping. We introduce piconet interference severity as:

$$S(p) = \frac{1}{L}\sum_{l=1}^{L} C(p,l) \tag{5}$$

The Severity of System Interference (*SoI*) is the mean of all piconets' interference severity.

$$S = \frac{1}{n}\sum_{i=1}^{n} S(p_i) \tag{6}$$

A lower *SoI* value indicates a reduction in the number of piconets that experience interference with one another in hopping sequences, which means a more balanced distribution of channels among the piconets within a given area. Decreased *SoI* correlates with an increase in throughput because it is possible that the interfering piconets operating on the same channel are not situated too close and can maintain minimal data transfer rates.

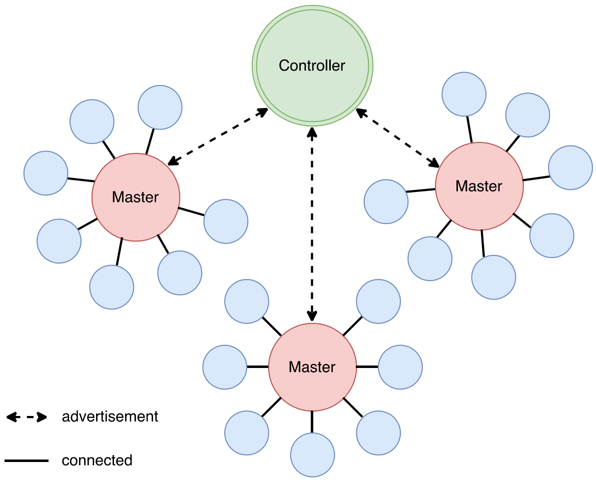


Fig. 2. Architecture of BlueMIC. A gateway managing interference level of surrounding piconets

5- BlueMIC

Inter-piconet interference arises due to the absence of a coordination mechanism among neighbouring piconets. To mitigate this issue, we introduce BlueMIC, a lightweight coordination framework designed to manage channel usage among nearby piconets. The BlueMIC architecture, illustrated in Figure 2, includes a central gateway and multiple piconets. The gateway, which can be any BLE-enabled device with adequate processing power, coordinates with all masters within its reach by exchanging data through predefined channels via advertising. Masters use BLE-connected mode to communicate with their piconet slaves [25,26]. During operation, masters periodically scan their environment to build local channel maps and send these to the gateway. The gateway listens to this channel for consecutive slots, collects channel map packets, and then calculates a new channel map for each piconet to minimize surrounding interference. It communicates these new maps to masters using special advertising packets, enabling them to update their channel maps and initiate new hopping sequences.

BlueMIC and its associated channel selection algorithm are built using the latest BLE5 stack and CSA#2 algorithm, meeting specific BLE requirements. This framework is lightweight enough to be implemented at the application layer, eliminating the need to modify lower-level protocols. The only overhead of BlueMIC for the network is the periodic transmission of channel map packets and the controller's response, which occurs only in situations where there has been a significant increase in congestion. The required parameters of the BlueMIC are also fitted in BLE 31-byte advertisement packets.

Algorithm 1. Pseudo code for Iterative Channel Removal Algorithm

```
procedure ITERATIVE_CHANNEL_REMOVAL
    while true do
        C ← {} // set of channels maps
        A ← {} // set of access addresses
        {C, A} ← RECEIVE_MASTER_INFO()
        if |A| < MIN_PICOS then
            continue
        end if
        CM ← EXECUTE_CSA2_FOR_MASTERS(C, A)
        Π ← COMPUTE_INTERFERENCE(C, CM)
        while Π ≥ MIN_INTERFER OR (no change) do
            c_max ← arg max Π_c(c)
            p_max ← arg max Π_c(c, p) / 2^G
            if |G| > MIN_CHANNELS then
                REMOVE_CHANNEL_FOR_PICONET(p_max, c_max)
                Π ← COMPUTE_INTERFERENCE(C, CM)
                CM ← EXECUTE_CSA2_FOR_MASTERS(C, A)
            end if
        end while
    end while
end procedure
```

5- 1- Iterative Channel Removal Algorithm

The Iterated Channel Removal (ICR) algorithm, which is shown in Algorithm 1 and adopted from [22], reduces the inter-piconet interference by removing the contended channels from highly-interferred piconets. Based on ICR, each master has to advertise its access address and channel map. No action is taken if the number of advertised piconets in a specific period is less than a threshold (line 6) Otherwise, ICR emulates CSA#2 on behalf of all masters and builds a *Collision Map* by tracing the masters' exact hop sequence (line 9). Then, interference probability is calculated by Eq. (1). If Đ becomes greater than a threshold (line 11), ICR finds the channel with the highest interference and the piconet with the highest interference in that channel using the following equations.

$$c_{max} = argmax\ \Pi_c(c) \tag{7}$$

$$p_{max} = argmax \frac{\Pi_c(c,p)}{2^G} \tag{8}$$

Then, the channel index of c_{max} is marked as a bad channel in the channel map of the piconet, p_{max} (line 15). This procedure is repeated for a specified round or until Đ is reduced below a specified threshold. When ICR is terminated, the updated channel maps are advertised to the corresponding masters. Once the masters receive and adjust the new channel maps, the slaves are quickly notified and adjust their connection according to the new channel map.

ICR requires $n(n-1)L$ comparisons to compute $Đ(c,p)$, as it must evaluate the channel conflicts across all piconets for each channel. Following this, ICR identifies the highest interfered channel from a list of $40n$, which entails operations in $O(nlogn)$. In the worst-case scenario, if only two channels remain available for the piconets, the ICR removal loop would be executed $38n$ times. Consequently, this leads to a total of $38n$ sorting operations and computations of $Đ(c,p)$, indicating that the running time of the ICR algorithm is on the order of $O(n^3L)$.

5- 2- Isolated Channels Algorithm

The main problem with the ICR method is its dependency on the *AccessAddress* to generate frequency hopping sequences. However, most operating systems or SDKs keep the *AccessAddress* secret and do not expose it to upper layers for security reasons.

Unlike ICR, the Isolated Channels Algorithm (ICA) is a more practical solution without the *AccessAddress*. However, this also means that ICA cannot determine the exact hop sequence of masters and thus loses the precision of ICA in estimating the actual interfered channels. A possible workaround is to have the masters transmit their hop sequence in the advertised packet, but this has two major drawbacks: First, the packet size can be too large, and second, the hop sequence, which is supposed to be secret because of the inaccessibility of *AccessAddress*, is revealed, posing a serious threat to security integrity.

Without having the hopping sequence, interference formulas must be rewritten according to the channel maps. $N^M(p)$ denotes the number of good channels in the channel map of p, which are also good in the channel map of other piconets. $N^M(p,c)$ shows the same concept in a particular channel index.

$$N^M(p) = len\left(G(P_{-p}) \cap G(p)\right) \tag{9}$$

$$N^M(p,c) = c \in G(P_{-p}) \cap G(p) \tag{10}$$

In these equations, P_{-p} refers to the set of all piconets without piconet p. The channel map-based interference probability of a channel in a piconet and the interference probability of a system are computed using the parameters above.

$$\Pi_c^M(p,c) = \frac{1}{|M|} N^M(p,c), c \in [0,39] \tag{11}$$

$$\Pi^M = \frac{1}{n|M|}\sum_{1}^{n} N^M(p) \tag{12}$$

To compute the interference severity of a piconet, we count the number of common good channels instead of counting conflicting hops. In this regard, $C_c^M(p,c)$ shows the number of piconets with channel c in their good channels, and $C^M(p)$ is the average number of such piconets on all channels.

$$C_c^M(p,c) = \sum_{p_i \in P_{-p}} c \in G(p_i), \forall c \in G(p) \tag{13}$$

$$C^M(p) = \frac{1}{|M|}\sum_{c \in G(p)} C^M(p,c) \tag{14}$$

Using these parameters, the interference severity of a system is the average number of piconets with the same good channels.

$$S^M = \frac{1}{n}\sum_{i=1}^{n} C^M(p_i) \tag{15}$$

Equation 16 indicates that in order to have less interference, piconets should have isolated channels as much as possible. Accordingly, ICA tries to assign distinct channels to piconets to ensure they do not collide at any time, even by chance, as shown in Algorithm 2.

Using ICA, masters advertise their channel map to the gateway. After receiving enough advertisements, the gateway executes ICA. First, ICA calculates $Đ^{M}$ for all channels and piconets based on the channel maps. From then on, ICA was executed in rounds. As BLE requires at least two good channels to establish a connection [24], ICA starts assigning two unique channels for each master in the first round. These channels must not be previously assigned to any piconet; they must be picked from the good channels and have the least $C_c^M(p,c)$. If ICA cannot find at least two distinct channels for all master, it returns the original channel maps without modification. ICA tries to assign more isolated channels to masters in the subsequent rounds and terminates when no channel map can be improved.

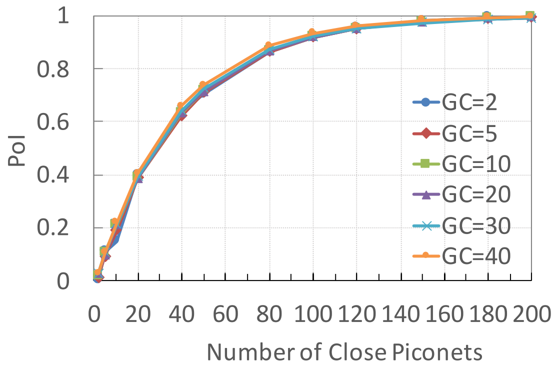
With the proposed method, a piconet network can use ICA when nearby networks are less than twice the number of unique good channels. Given this constraint, ICA can allocate channels to a maximum of 20 networks using all 40 channels of BLE (Algorithm 2, line 8). Since having a common channel increases the probability of interference to support a larger number of piconets and enhance the throughput, the ICA method allows overlapping several piconets in one channel.

As we will observe in the simulations, overlap increases the probability and intensity of interference. However, when the number of good channels is limited, the possibility of reducing the effects of interference without any overlap does not exist.

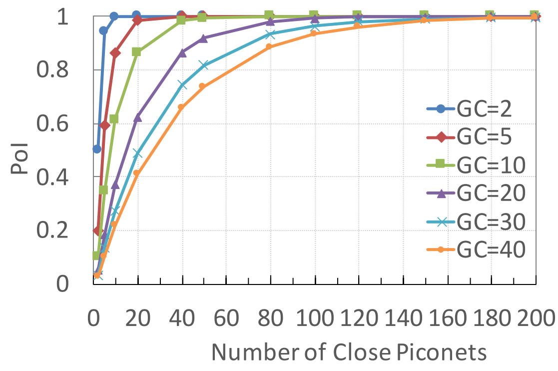
Similar to ICR, ICA in the first step requires finding the channel conflicts and sorting them with running time in the order of $n(n-1)$. In the next step, if the number of piconets is less than 40, the maximum number of loop iterations will be $\frac{40}{n}-1$. Otherwise, the number of loop iterations in the ICA algorithm mostly depends on the overlap amount and the

Algorithm 2. Pseudo code for Isolated Channels Algorithm

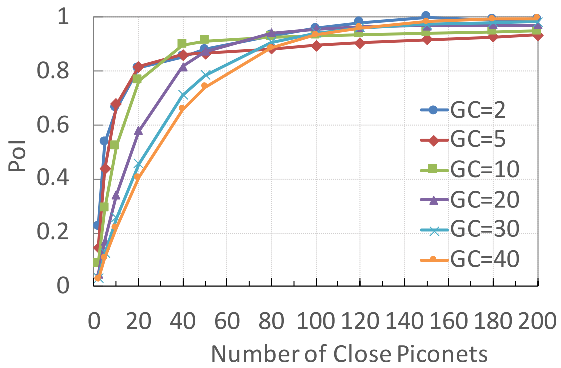
```
1:  procedure ISOLATED_CHANNELS (overlap)
2:      while true do
3:          C ← RECEIVE_MASTERS_INFO() // set of channels maps
4:          if |C | < MIN_PICOS then
5:              continue
6:          end if
7:          dc ← COUNT_DISTINCT_CHANNELS(C)
8:          if dc < |C|/2 AND overlap = 0 then
9:              continue
10:         end if
11:         while changed do
12:             CM ← BUILD_COLLISION_MAP_FOR_MASTERS(C)
13:             Π ← COMPUTE_INTERENCE(C)
14:             changed ← FLASE
15:             for C_i ∈ C do
16:                 if first round then
17:                     {ch_1, ch_2} ← FIND_TWO_GOOD_CHANNELS(CM_i, C_i)
18:                     if ch_1& ch_2 then
19:                         ENABLE_CHANNELS(C_i, ch_1, ch_2)
20:                         changed ← TRUE
21:                     end if
22:                 else
23:                     ch_1 ← FIND_ONE_GOOD_CHANNEL(CM_i, C_i, overlap)
24:                     if ch_1 then
25:                         ENABLE_CHANNELS(C_i, ch_1)
26:                         changed ← TRUE
27:                     end if
28:                 end if
29:             end for
30:         end while
31:     end while
32: end procedure
```



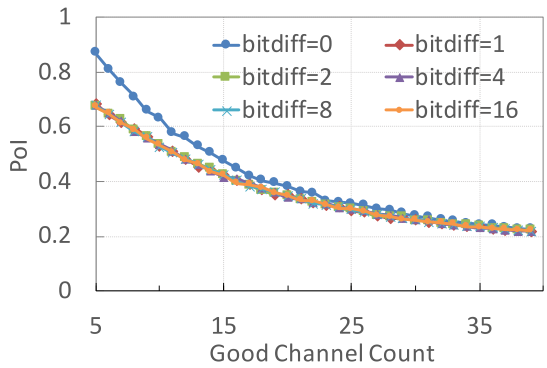
a) Π for randomly-generated channel maps



b) Π for equal channel maps



c) Π for channel maps with only 1 bit-diff



d) Π for a network of 10 piconets with varying numbers of good channels. In each experiment, the channel maps are generated with various bit-diffs

Fig. 3. Probability of System Interference (PoI-Π) of a network of piconets with varying interference severities and channel map patterns. The length of the frequency hopping sequence in all simulations is 500.

number of channels. Given the limited number of channels and the overlap, the execution time of this section of ICA is of linear order. Therefore, in the worst case, the execution time of ICA will be of order $O\left(n^2\right)$.

6- Simulation Results

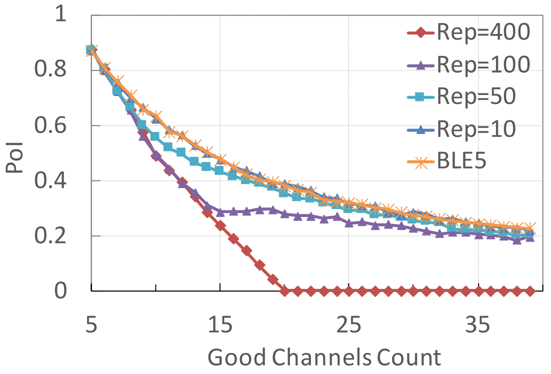
In this section, we first provide insight into how the inter-piconet interference gets intensified in BLE5. Then, we compare the BLE5, ICR, and ICA methods in terms of the probability of system interference (Đ) and the severity of system interference (S) based on our simulation results. We use the channel maps of the worst-case scenarios obtained in the first subsection for comparison.

6- 1- Inter-piconet Interference in BLE5

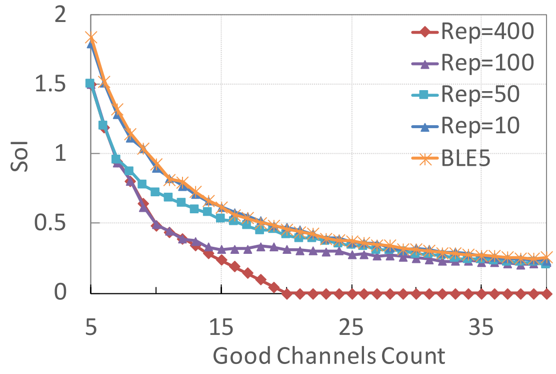
In the first step, we generated completely random channel maps with different interference intensities between 30 and 95%. That means, in the channel maps generated with 80% interference intensity, on average, 32 channels are among the bad channels. Figure 3a shows the effect of the number of nearby piconets on the probability of system interference in a relatively long frequency hopping sequence of 500 hops and a severity of 80%. Based on the figure, we can infer the following results:

1. The number of good channels (equivalent to the severity of interference) does not significantly affect the system interference. The CSA#2 algorithm is so good that it can provide effective bandwidth for piconets even with a limited number of good channels. In other words, BLE efficiently uses its limited number of channels.

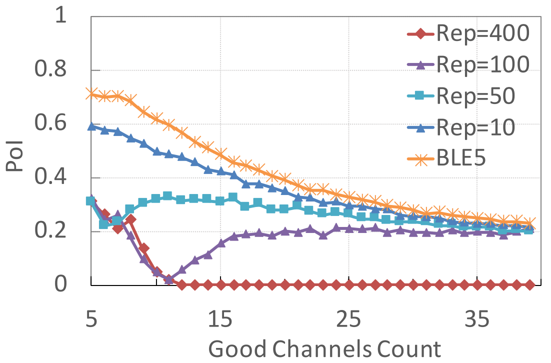
2. The quantity of neighbouring piconets significantly contributes to the escalation of interference. For instance, when there are merely ten piconets present, the probability of interference is about 20%. This value rises to 60% with



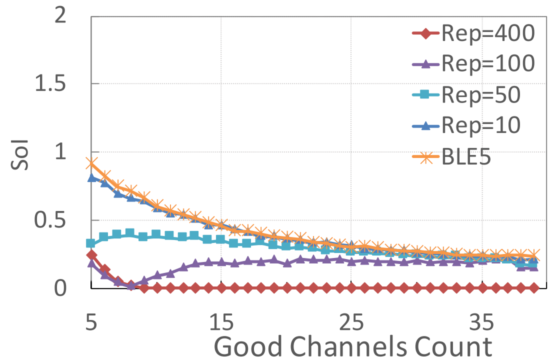
a) PoI for channel maps with bit-diff of 0



b) SoI for channel maps with bit-diff of 0



c) PoI for channel maps with bit-diff of 1



d) SoI for channel maps with bit-diff of 1

Fig. 4. Performance of ICR in Probability of Interference (PoI) and Severity of Interference (SoI) for a network of 10 piconets and channel maps generated with 0 or 1-bit-diff.

40 networks and surpasses 90% with 100 networks, which is consistent with the findings of other researchers.

Figure 3b repeats the simulation with identical channel maps for all piconets. As the number of good channels decreases, indicating increased interference severity, the probability of system interference rises. With 40 piconets active in a confined area, halving the number of good channels boosts interference probability by 30%. The graph shows that with only ten good channels, a maximum of 20 neighbouring piconets can coexist with an 86% interference probability. Doubling the good channels allows 40 piconets to coexist.

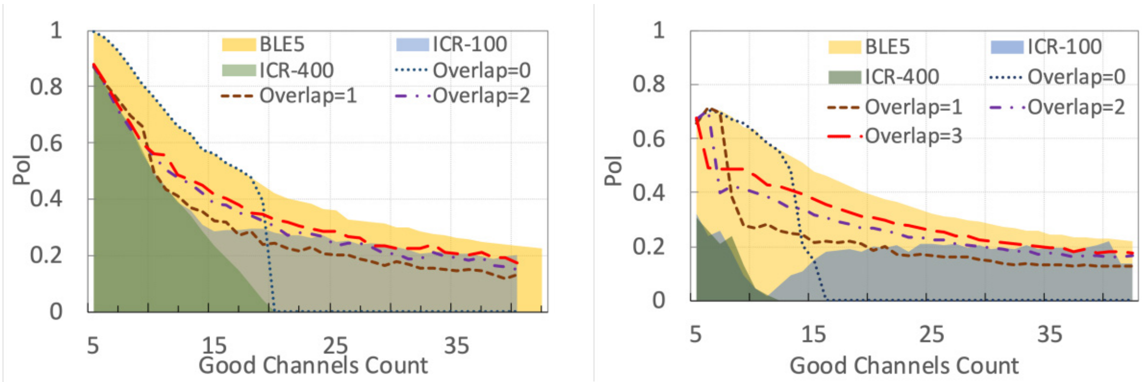
In the subsequent simulation, the results depicted in Figure 3c, we generated a random channel map with a specific number of good channels. Then, for each piconet, we randomly substituted one good bit with a bad bit. It can be observed that the probability of interference, especially with a lower number of good channels, exhibits a noticeable decrease. For instance, with ten good channels and 20 Piconets, the probability of interference has decreased from 86% to 76%. However, this reduction has a lesser impact in scenarios where good channels exceed 10. Increasing the size of bit differences in generating channel maps does not have a notable impact on Ð, evident in Figure 3d for a network of 10 piconets.

These results clearly indicate that the inter-piconet interference rises considerably as the number of neighbouring piconets grows. The probability of such interference becomes particularly concerning when the pattern of channel maps becomes similar.

6- 2- Evaluating ICR

In the second simulation set, we focus on examining the performance of ICR. We conducted ICR tests once in a network consisting of 10 piconets and then in a network with 40 piconets, based on the number of rounds and channel map bit differences of 0 or 1. The results are depicted in Figures 4 and 5.

In the experiment with ten piconets, it is expected that when the number of good channels exceeds 20, Ð would



a) PoI for channel maps with bit-diff of 0

b) PoI for channel maps with bit-diff of 1

Fig. 5. Comparing the performance of ICA and ICR in Probability of Interference (PoI) for a network of 10 piconets and channel maps generated with 0 or 1-bit-diff.

be around 0, as each piconet has a minimum of two distinct channels. Figure 4a illustrates the probability of interference when the channel map of piconets is identical. According to this figure, ICR could only achieve desirable results in many rounds, while it had less than a 10% impact on scenarios with fewer than 100 rounds. Repeating the experiment from 5 to 50 rounds did not significantly reduce interference.

Notably, as the number of good channels increases, there needs to be a substantial increase in the number of rounds to achieve minimum desirable outcomes. This could be attributed to the fact that in these conditions, piconets are active in a larger number of channels, and removing channels one by one in each round does not have a remarkable effect on Đ . Although a direct relationship between the reduction in S and the decrease in Đ is observable, for each unit decrease in S, Đ decreases by less than half a unit.

In Figures 4c and 4d, the results of simulations are presented for the scenario in which the channel maps differ by one bit. This time, due to the presence of at least one distinct channel for each piconet, ICR successfully reduced Đ by more than 50% in fewer rounds, such as 50 rounds. However, with increased good channels, as previously mentioned, the number of rounds for executing ICR must increase to achieve the desired outcome.

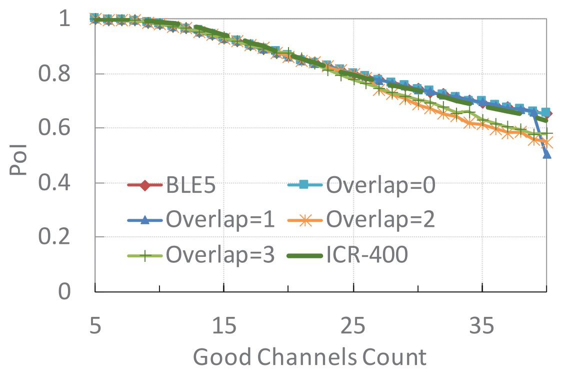
6- 3- Evaluating ICA

The final set of simulations compares the performance of ICA and ICR. Remember that ICR operates based on piconets' precise frequency hopping patterns, while ICA makes decisions solely based on channel map similarity. The ICA Channel map assignment is based on equations (12) and (16). For comparison, Đ and S of both methods are calculated using CSA#2 and the frequency hopping patterns of piconets.

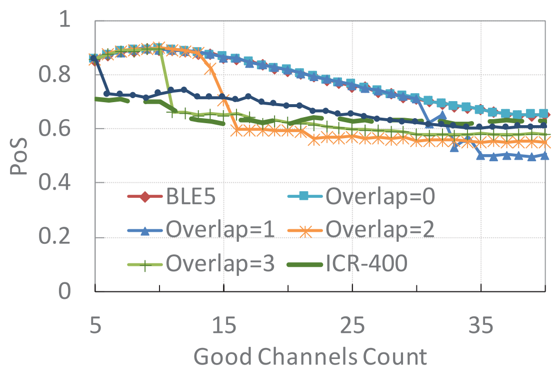
In Figure 5, the performance of ICR with two 100 and 400 rounds and ICA with overlap values ranging from 0 to 3 are depicted. With 0 overlap, the ICA does not change the channel maps until it can allocate two channels for each piconet. Therefore, as shown in Figure 5a, ICA results are similar to standard BLE until the number of good channels reaches 20. However, after that point, it performs similarly to ICR-400.

Including overlap in ICA allows it to start reducing interference from fewer good channels. The higher the overlap, the lower the number of good channels from which the reduction of interference starts, but the extent of interference reduction decreases. Figures 5a and 5b show that ICA performs better with an overlap of 1 to 2 than ICR with 100 rounds. In summary, ICR performs better in environments with high interference and fewer good channels. The formula $c_s = n/(v+2)$ can be used to estimate the amount of perfect amount of overlap in ICA, in which c_s is the minimum good channel count, n is the number of piconets, and v is the overlap amount.

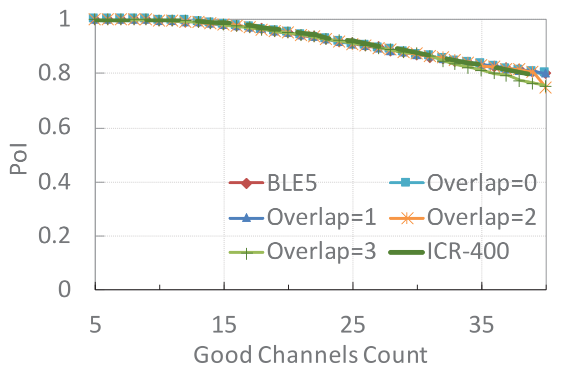
Figure 6 showcases the performance comparison of three methods in larger networks of 40, and 60 piconets. None of the methods significantly reduced interference in a network with 40 piconets and identical channel maps. Only when more than 30 good channels were available ICA and ICR were able to reduce interference by less than 10%. This trend continues in Figure 6c when there are 60 adjacent piconets, and the level of reduction drops below 5%. The possibility of reducing interference arises when piconets have at least one separate channel. However, the extent of interference reduction decreases with an increase in the number of piconets. In the 40 piconet scenario, it is less than 20%,



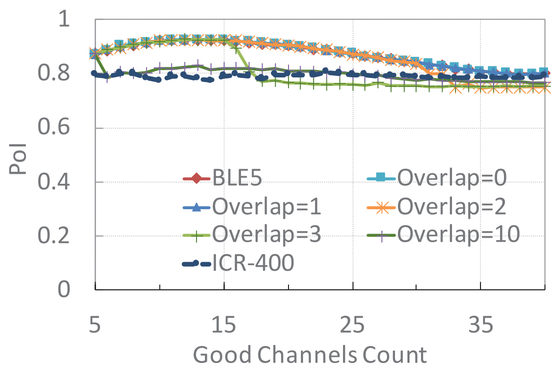
a) PoI of 40 piconets with channel maps of 0 bit-diff



b) PoI of 40 piconets with channel maps of 1 bit-diff



c) PoI of 60 piconets with channel maps of 0 bit-diff



d) PoI of 60 piconets with channel maps of 1 bit-diff

Fig. 6. Comparing performance of ICA and ICR in Probability of Interference (PoI) for networks consisting of 40, and 60 piconets and channel maps generated with 0 or 1-bit-diff.

around 10% in the 60 piconet scenario, and in the scenario of 100 piconets, it is below 8%.

In general, ICR has performed better in most cases with 400 rounds, and the difference in performance between ICR and ICA is clear in scenarios with fewer channels. This performance is achieved at the cost of a significantly longer execution time. Figure 7 illustrates the execution time ratio of ICR-400 and ICR-100 to ICA. The execution time of ICR-400 is more than 13 times greater for ten piconets, and ICR-100 is ten times greater than the execution time of ICA for the same scenario. This difference increases to 20 and 100 times for a network with 40 piconets, respectively, which poses a significant burden for a gateway primarily designed with embedded processors. Furthermore, as previously mentioned, ICR cannot be implemented in BLE5 without serious vulnerability risks.

7- Conclusion and Future Work

This paper addresses inter-piconet interference issues in BLE5, revealing that the BLE5's new channel selection method struggles as the number of neighbouring piconets increases. We've shown that similar channel map patterns among neighbouring piconets escalate interference, significantly impacting even small networks with up to 10 piconets.

To tackle this, we proposed the BlueMIC framework, wherein a gateway periodically collects and suggests new channel maps to reduce interference. We also introduced the ICA algorithm, a lightweight and BLE5-compatible solution for channel allocation. Simulations demonstrate that BlueMIC and ICA significantly enhance execution speed without security risks, performing well even under severe interference and limited "good" channels. Due to their lightweight nature,

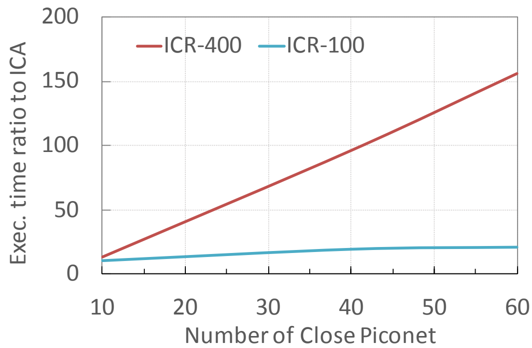


Fig. 7. Execution time ratio of ICR with 400 and 100 rounds against ICA

BlueMIC and ICA can be easily integrated into application layers without needing knowledge of the piconets' frequency hopping sequences.

Future research should focus on practical experiments to further explore channel map patterns and their similarity in limited environments. Additionally, examining packet length variation and transmission power as interference reduction mechanisms is necessary. We are also interested in developing a distributed approach for configuring channel maps without a central node.

References

[1] R. Natarajan, P. Zand, M. Nabi, Analysis of coexistence between ieee 802.15.4, BLE and ieee 802.11 in the 2.4GHz ism band, in: IECON 2016- 42nd Annual Conference of the IEEE Industrial Electronics Society, IEEE, 2016, pp. 6025–6032.

[2] Collotta M, Pau G, Talty T, Tonguz OK. Bluetooth 5: A concrete step forward toward the IoT. IEEE Communications Magazine. 2018 Jul 25;56(7):125-31.

[3] Bluetooth Special Interest Group, Bluetooth core specification v5.2, in: Bluetooth core specification v5.2, 2019. URL: https://www.bluetooth.org/docman/handlers/downloaddoc.ashx? doc id=478726, retrieved: 2022-09-14.

[4] Tosi J, Taffoni F, Santacatterina M, Sannino R, Formica D. Performance evaluation of bluetooth low energy: A systematic review. Sensors. 2017 Dec 13;17(12):2898.

[5] F. Mazzenga, D. Cassioli, P. Loreti, F. Vatalaro, Evaluation of packet loss probability in Bluetooth networks, in: 2002 IEEE International Conference on Communications. Conference Proceedings, volume 1, ieee, 2002, pp. 313–317.

[6] Mazzenga F, Cassioli D, Detti A, Habib I, Loreti P, Vatalaro F. Performance evaluation in Bluetooth dense piconet areas. IEEE Transactions on Wireless Communications. 2004 Nov;3(6):2362-73.

[7] M. O. Al Kalaa, H. H. Refai, Selection probability of data channels in Bluetooth low energy, in: 2015 International Wireless Communications and Mobile Computing Conference (IWCMC), IEEE, 2015, pp. 148–152.

[8] S. B ̈ocker, C. Arendt, C. Wietfeld, On the suitability of Bluetooth 5 for the internet of things: Performance and scalability analysis, in: 2017 IEEE 28th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC), 2017, pp. 1–7.

[9] A. Ancans, J. Ormanis, R. Cacurs, M. Greitans, E. Saoutieff, A. Faucorr, S. Boisseau, Bluetooth low energy throughput in densely deployed radio environment, in: 2019 23rd International Conference Electronics, IEEE, 2019, pp. 1–5.

[10] M. S. Bahbahani, E. Alsusa, Crowdconnect: A quality of experience enhancement solution for dense stadium networks, in: 2018 IEEE Global Communications Conference (GLOBECOM), volume 1, 2018, pp. 1–7.

[11] La QD, Nguyen-Nam D, Ngo MV, Hoang HT, Quek TQ. Dense deployment of BLE-based body area networks: A coexistence study. IEEE Transactions on Green Communications and Networking. 2018 Jul 24;2(4):972-81.

[12] K. Mekki, E. Bajic, F. Chaxel, F. Meyer, Overview of cellular lpwan technologies for IoT deployment: Sigfox, LoRaWAN, and NB-IoT, in: 2018 ieee international conference on pervasive computing and communications workshops (percom workshops), IEEE, 2018, pp. 197–202.

[13] J. Lansford, A. Stephens, R. Nevo, Wi-fi (802.11 b) and Bluetooth: enabling coexistence, IEEE network 15 (2001) 20–27.

[14] C.-F. Chiasserini, R. R. Rao, Coexistence mechanisms for interference mitigation between ieee 802.11 WLANs and Bluetooth, in: Proceedings. Twenty- First Annual Joint Conference of the IEEE Computer and Communications Societies, volume 2, IEEE, 2002, pp. 590–598.

[15] M. C.-H. Chek, Y.-K. Kwok, On adaptive frequency hopping to combat coexistence interference between Bluetooth and ieee 802.11 b with practical resource constraints, in: 7th International Symposium on Parallel Architectures, Algorithms and Networks, 2004. Proceedings., IEEE, 2004, pp. 391–396.

[16] Shao C, Roh H, Lee W. BuSAR: Bluetooth slot availability randomization for better coexistence with dense Wi-Fi networks. IEEE Transactions on Mobile Computing. 2019 Nov 22;20(3):846-60.

[17] Chen D, Zhuang Y, Huai J, Sun X, Yang X, Javed MA, Brown J, Sheng Z, Thompson J. Coexistence and interference mitigation for WPANs and WLANs from traditional approaches to deep learning: A review. IEEE Sensors Journal. 2021 Oct 4;21(22):25561-89.

[18] Lee SH, Kim HS, Lee YH. Mitigation of co-channel interference in Bluetooth piconets. IEEE transactions on wireless communications. 2012 Feb 16;11(4):1249-54.

[19] Zhang X, Shin KG. Cooperative carrier signaling: Harmonizing coexisting WPAN and WLAN devices. IEEE/ACM Transactions On Networking. 2012 Jun 5;21(2):426-39.

[20] W. Sun, J. Koo, S. Byeon, W. Park, S. Lim, D. Ban, S. Choi, Bluecode: Bluetooth coordination in dense environment for better coexistence, in: 2017 IEEE 25th International Conference on Network Protocols (ICNP), IEEE, 2017, pp. 1–10.

[21] Z. Jiang, V. C. Leung, V. W. Wong, Reducing collisions between Bluetooth piconets by orthogonal hop set partitioning, in: Radio and Wireless Conference, 2003. RAWCON'03. Proceedings, IEEE, 2003, pp. 229–232.

[22] Y. Hu, G. Wang, L. Shan, Z. Yuan, Y. Ouyang, Inter-piconet interference mitigation schemes for converged BTLE and cellular network, in: 2011 International Conference on Wireless Communications and Signal Processing (WCSP), IEEE, 2011, pp. 1–5.

[23] B. Pang, T. Claeys, D. Pissoort, H. Hallez, J. Boydens, A study on the impact of the number of devices on communication interference in Bluetooth low energy, in: 2020 XXIX International Scientific Conference Electronics (ET), IEEE, 2020, pp. 1–4.

[24] Pang B, T'Jonck K, Claeys T, Pissoort D, Hallez H, Boydens J. Bluetooth low energy interference awareness scheme and improved channel selection algorithm for connection robustness. Sensors. 2021 Mar 24;21(7):2257.

[25] Baddeley M, Raza U, Stanoev A, Oikonomou G, Nejabati R, Sooriyabandara M, Simeonidou D. Atomic-SDN: Is synchronous flooding the solution to software-defined networking in IoT?. IEEE Access. 2019 May 30;7:96019-34.

[26] Murillo Y, Chiumento A, Reynders B, Pollin S. An all-wireless SDN framework for BLE mesh. ACM Transactions on Internet of Things. 2020 Aug 4;1(4):1-30.

[27] B. Pang, J. Vankeirsbilck, H. Hallez and J. Boydens, "Experimental Validation of Common Assumptions in Bluetooth Low Energy Interference Studies," 2023 IEEE 22nd International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom), Exeter, United Kingdom, 2023, pp. 2495-2502, doi: 10.1109/TrustCom60117.2023.00350.

[28] Lun YZ, Rinaldi C, D'Innocenzo A, Santucci F. Co-Designing Wireless Networked Control Systems on IEEE 802.15. 4-Based Links Under Wi-Fi Interference. IEEE Access. 2024 May 16.

HOW TO CITE THIS ARTICLE

H. Khanmirza, P. Teymoori, BlueMIC: Coordination for Mitigating Inter-Piconet Interference in BLE5 Networks , AUT J. Model. Simul., 56(1) (2024) 19-32.

DOI: 10.22060/miscj.2024.23295.5363