Reducing Inter-Piconet Interference in BLE 5 Networks

Abstract: Bluetooth Low Energy (BLE) operates in theP: within the congested 2.4GHz ISM band, crowded with manyP: shared by numerous short-range wireless protocols. The Adaptive Frequency Hopping (AFH) mechanism is designed for Bluetooth to minimize interference effects with other protocols. Due to the widespread use of the Bluetooth protocol in various applications and numerous types of equipment, many piconet networks in a limited area are expected today. P: While the Adaptive Frequency Hopping (AFH) mechanism was introduced to mitigate interference effects from other protocols, the proliferation of Bluetooth in diverse applications has led to a dense clustering of piconet networks in confined spaces. (better to use "mitigate", not minimize since a minimum cannot be again minimized!) However, AFH is inadequate for interferences originating from the large number of close inter-piconets. P: Addressing this challenge, this paper studies the problem of the inter-piconet interference problemP: (repeated "problem" word) and proposesP: presents a P: novel BLE5-based solution. The proposedP: presented solution considersP: employs/devises a BLE gateway that detects its surrounding piconet masters and computes a better channel map to reduce interference. The paper also presents two BLE5-compatible channel-map computation algorithms for this gateway.P: Add one or two sentences on these algorithms. Simulation results showP: confirm/validate the effectiveness of the proposed solution and algorithms in reducing interference and preserving the link quality. P: How much is it improved? What was the exact metrics/criteria?

Keywords: inter-piconet interference,BLE5, CSA#2

1 Introduction and Motivation

P: (Motivation can be a section after Introduction, or a subsection in Introduction. Better to remove it from the title.) The Internet of Things (IoT) has emerged in various applications, including medicine, industry, entertainment, shopping, automation, and smart home. Several low-power protocols are proposed to operate on low-capability IoT devices along with such vast application contexts. Bluetooth low energyP: Low Energy (BLE) is among the most popular protocols in this field due to its great features and availability in smart devices globally used by most people for their work or personal affairs (Collotta et al. (2018); Natarajan et al. (2016)). BLE has gained more attraction by introducing its fifth version, known as BLE5, which brings improved speed, a larger connection range, higher security, less power consumption, and many other features to the protocol. P: (I removed all the line breaks between paragraphs to make them more visible.)

BLE has several operation modes. Piconet is a master-slave network formed using the "connected" operation mode P: of BLE. Each piconet supportsP: accommodates a master device with at most seven slave devices communicating with each other by frequency hopping over 40 channels. Channels 0 to 36 are used for data exchange, andP: while the last three channels are usually used for advertisement.

Frequency Hopping is a technique that forces communicating devices to change their channel synchronously over timeP: (rephrasing the sentence:) The cornerstone of Bluetooth's interference mitigation strategy is Frequency Hopping, 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 significantlyP: (probably better to remove "significantly" since **your** method is supposed to "significantly" improve it!) reduced.

Like other IoT protocols such as WiFi and Zigbee, BLE operates in theP: within the crowded 2.4 GHz ISM band. Optimal use of the 2.4 GHz band has become a challenge due to the growing number of IoT devices. The adaptive Frequency Hopping mechanism (AFH) has been embedded in Bluetooth specification since revision 1.2 to counteract the interference of near networks and surrounding WiFi signals. WithP: Under AFH, the master P: device keepsP: maintains a 40-bit channel-map bit vector where the value of each bit represents the condition of the corresponding communication channel. Having a '1' bit means the corresponding channel is in good condition, while having a '0' bit implies interference over that channelP: implies the presence of interference on that channel (Tosi et al. (2017)). The communicating devices may avoid those bad channels in their hopping sequenceP: (suggestion to rephrase the sentence to:) This information is vital for communicating devices, allowing them to intelligently avoid channels with poor conditions during their hopping sequence..

While AFH provides a strongP: robust base for reducing interference in general, it is not effective as expected in crowded environmentsP: (to show the problem is critical, better to change the sentence to:), its effectiveness diminishes considerably in densely populated environments. Specifically, BLE has no particular solution to reduce inter-piconet interference, and AFH is the only wayP: (good to use BIG words!:) the primary means to improve the situationP: address the issue. Al Kalaa and Refai (2015) have shown that with an increasing number of the BLE connection pairs in an environment, the probability of collision in one or more channels grows. As figure 1 shows, with only 25 BLE pairs, the collision probability is as largeP: high as 50%. The problem of inter-piconet

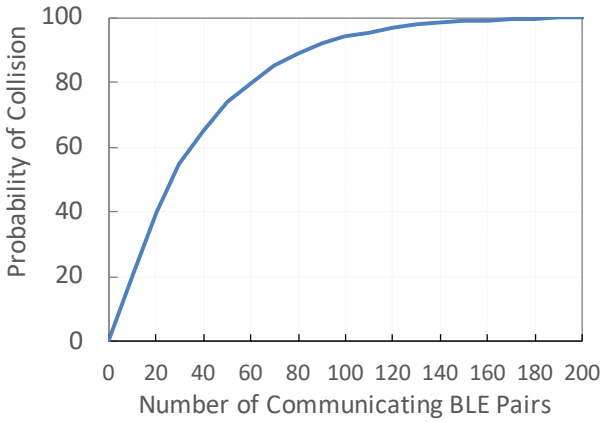


Figure 1 Collision probability vs. number of surrounding BLE pairs (Al Kalaa and Refai (2015))

collision and its consequences is studied in several P: some (if you want to imply the topic is new, change several to some!) researches such as El-Hoiydi (2001); Hu et al. (2011); Naik et al. (2003, 2005); Yufeng et al. (2009).

P: (Are there any statistics on the average number of piconets in an area? If there is, cite it and use as the justification. For example, where are we now in Fig.1? 20? 40?)

By applying AFH, master nodes change their channel map with P: in response to increasing interference in P: on some channels. Consequently, the channel selection algorithm will avoid those channels. Since the channel selection algorithm in BLE chooses the next channel randomly, in a crowded environment, it is highly possible for masters in different piconets to choose similar channels as BLE masters has no way to coordinate. Operating simultaneously in the same channels, AFH will mark those channels as bad, and consequently, piconets return to the previous channels as they are now in good condition. Such a back-and-forth channel selection significantly degrades the utility of channels and the quality of links.

This paper focuses on the inter-piconet channel selection problem in crowded environments and proposes a fast solution for masters to choose low-interference connection channels smartly. In our proposed solution, masters regularly advertise their channel map to a BLE gateway device. Due to a lack of coordination mechanism between piconets, this gateway acts as a coordinator of the surrounding masters. By receiving the channel map of several masters, the BLE gateway calculates and sends globally low-interference channel maps for each of the masters. We also propose and evaluate two algorithms for calculating low-interference channel maps in BLE5 for gateway devices. P: This paragraph should be extended to at least 3 paragraphs: 1) the general idea of your method and differences with previous work, 2) some details of your methods, and 3) a subsection/bullet-points of your novelties.

This paper is organized as follows. Section 2 reviews the related literature, and Section 4.4 describes the new channel selection algorithm in BLE5. In section 3, we explain the architecture of our proposal and introduce our channel map selection algorithms. Section 5.3 presents the evaluation results, and finally, Section 6 concludes the paper.

2 Literature Review

P: Extend this by extending the text and description of each method, or giving it subsections and categorizing the methods, etc. Use a classification chart if possible.

Most of the literature on the interference of Bluetooth networks like Chek and Kwok (2004) and Ancans et al. (2019) has focused on reducing interference with external interference sources, such as WiFi. Recent studies compare the performance of Bluetooth5 with older versions or measure the interference in various conditions like in Böcker et al. (2017); Pang et al. (2021), and Hu et al. (2011). Böcker et al. (2017) study the performance of Bluetooth 5 in IoT applications and investigate the packet error rate (PER) with increasing the number of piconets. Hallez et al. in Pang et al. (2019) compare the performance of channel selection algorithms 1 and 2 (CSA#1 and CSA#2) in different interfering environments, and Pang et al. (2021) propose a new channel selection algorithm for decreasing the interference.

Hu et al. (2011) 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 neighborhood interference. This solution necessitates having master nodes equipped with Bluetooth and LTE interfaces, making them very 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 of piconets.

This paper also considers the problem studied by Hu et al. (2011) and uses a central gateway to reach a less-interfered state among piconets. However, in our proposed solution, all devices need only one BLE-supported interface. It is also designed based on the latest BLE5 stack and CSA#2 channel selection algorithm and fulfills their specific requirements.

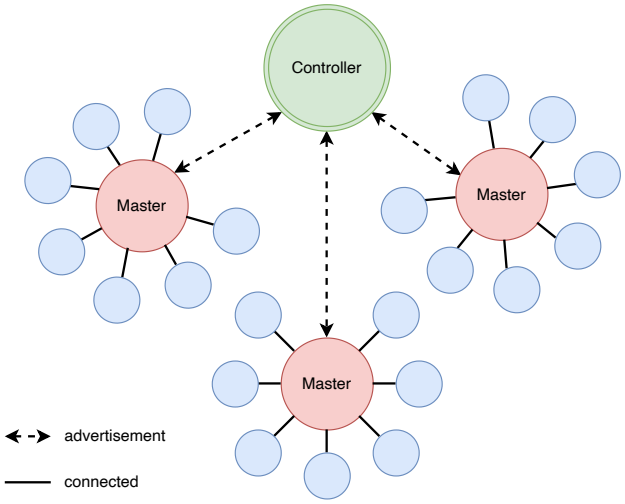


Figure 2 A gateway managing interference level of surrounding piconets

3 Solution Architecture

Figure 2 presents the overall architecture of our solution. A network consisting of a central gateway and several piconets cooperate to reduce interference. The gateway can be any device with a BLE interface and processing capability. Apparently, cooperation takes place if the gateway and masters are within range. Masters exchange data with the gateway by advertising, whereas they communicate with their piconet slaves using BLE connected mode. During the operation of piconets, the masters periodically scan their environment and save the good channels in their channel map. Then, they advertise their channel map to the gateway. The gateway calculates the optimal map for each piconet to reduce the surrounding interference using the advertised channel maps and returns the maps for the masters. We present two algorithms for obtaining channel maps in sections 4.3 and 4.4. By receiving the corresponding channel map from the gateway, masters update their channel map and start the new hopping sequence.

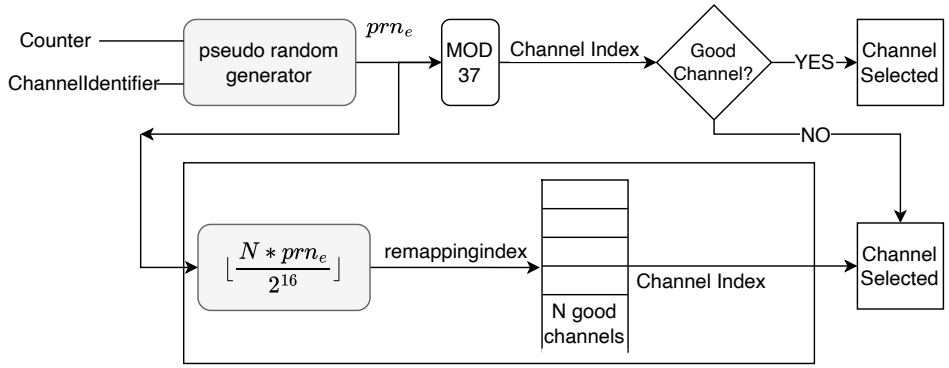


Figure 3 BLE5 Channel Selection Algorithm #2 Block Diagram Bluetooth Special Interest Group (2019)

4 Channel Selection Algorithms

To reduce the interference of the surrounding piconets, the gateway has to compute a set of channels for each piconet. In the following sections, we first explain the channel selection process of the BLE5, known as CSA#2. Then, we introduce our interference model inspired by Hu et al. (2011) and propose two central channel selection algorithms for piconets.

4.1 BLE5 Channel Selection Algorithm

According to Bluetooth Special Interest Group (2019), BLE5 uses Channel Selection Algorithm 2 (CSA#2) by default to select the hopping channels in each connection event. Figure 3 shows the steps of this algorithm. It uses the following three inputs to generate the channel index sequence:

- Channel map
- ChannelIdentifer: it is calculated from Access Address with the following equation

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

  Access address 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.
- Counter: this is a 16-bit integer number and is 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 mod 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 $remappingindex$ is calculated by $\lfloor \frac{N * prn_e}{2^{16}} \rfloor$ expression where N is the number of good channels indicated by the channel map. The channel with the index of $remappingindex$ among the good channels is selected for the next hopping.

4.2 Interference Model

System Interference Probability (I) [1] 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 $N(t)$ is the number of time segments the piconet, t, interferes with other piconets.

$$I = \frac{\sum_t^n N(t)}{n \times L} \tag{1}$$

The Interference Probability of a channel in a piconet is obtained from the following equation.

$$I(c,t) = \frac{M(c,t)}{n \times L}, c \in [0,36] \tag{2}$$

$N(c,t)$ refers to the number of time segments the piconet, t, interferes with other piconets in channel c. The total Interference Probability of a channel denoted by $I(c)$ is given by Eq. 3, which effectively is the sum of the $I(c,t)$ in all piconets.

$$I(c) = \frac{\sum_t^n I(c,t)}{n} \tag{3}$$

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

$$I(t) = \sum_{c \in G(t)} I(c) \tag{4}$$

The set of interference probability parameters, in essence, shows whether a piconet should expect interference on at least one of its channels or not and how much it is probable. However, they do not reflect the severity of the interference. Let $N(c,t,l)$ show the number of channel conflicts for a piconet in its l^{th} hopping, and $N(c,l)$ refers to the number of interference conflicts in l^{th} hopping on channel c. $N(c)$, in the following equation, measures how many piconets are competing for a channel, or the bandwidth of a channel is divided between how many piconets.

$$N(c) = \frac{\sum_l^L \sum_t^n N(c,t,l)}{\sum_l^L N(c,l)} \tag{5}$$

Using $N(c)$, we define a new parameter *throughput indicator* of a piconet, which is computed with the following equation:

$$T(t) = \sum_{c \in G(t)} \frac{1}{N(c)} \tag{6}$$

Throughput indicator is the reverse of the average channel contention. It shows the extent to which the channels used by a piconet are subject to interference with others. The higher the throughput indicator for a piconet, we expect a high-quality connection and better throughput.

Algorithm 1 Pseudo code for Channel Removal Algorithm

```
1: procedure channel_removal
2:     while true do
3:         C ← {} // set of channel maps
4:         A ← {} // set of access addresses
5:         {C, A} ← receive_masters_info()
6:         if |A| < MIN_PICOS then
7:             Continue
8:         end if
9:         CM ← execute_csa2_for_masters(C,A)
10:        I ← compute_interference(C,CM)
11:        while I ≥ MIN_INTERF| no change is done do
12:            c_max ← arg max I(c)
13:            t_max ← arg max I(c,t) / 2^G
14:            if G > MIN_CHANNELS then
15:                remove_channel_for_piconet(t_max,c_max)
16:            end if
17:        end while
18:    end while
19: end procedure
```

4.3 Channel Removal

The channel Removal (CR) algorithm is adopted from Hu et al. (2011) and redesigned to be compatible with CSA#2 and BLE5. Algorithm 1 shows the steps of CR. Based on CR, 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, CR 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 I becomes greater than a threshold (line 11), CR finds the channel with the highest interference and the piconet which has the highest interference in that channel using the following equations.

$$c_{max} = \arg\max I(c) \tag{7}$$

$$\mathbf{G} = |G(t_{max})| \tag{8}$$

$$t_{max} = argmax_t \frac{I(c,t)}{2^{\mathbf{G}}} \tag{9}$$

Then, The channel index of c_{max} is marked as a bad channel in the channel map of the piconet, t_{max} (line 15). This procedure is repeated until I is reduced below a specified threshold. When CR is terminated, the updated channel maps are advertised to the corresponding masters. Once the new channel maps are received and adjusted by the masters, the slaves are quickly notified and adjust their connection according to the new channel map.

Algorithm 2 Pseudo code for Channel Separation Algorithm

```
   procedure channel_separation
2:   while true do
       C ← {} // set of channel maps
4:     C ← receive_masters_info()
       if |C| < MIN_PICOS then
6:       Continue
       end if
8:     CM ← build_collision_map_for_masters(C)
       I ← compute_interference(C)
10:    while changed do
         changed ← FALSE
12:      for C_i ∈ C do
           if fisrt round then
14:          {ch_1, ch_2} ← find_two_good_channels(CM_i, C_i)
             if ch_1 & ch_2 then
16:            enable_channels(C_i,,ch_1, ch_2)
               changed ← TRUE
18:          end if
           else
20:          ch_1 ←find_one_good_channel(CM_i, C_i)
             if ch_1 then
22:            enable_channels(C_i, ch_1)
               changed ← TRUE
24:          end if
           end if
26:      end for
       end while
28:  end while
   end procedure
```

4.4 Channel Separation

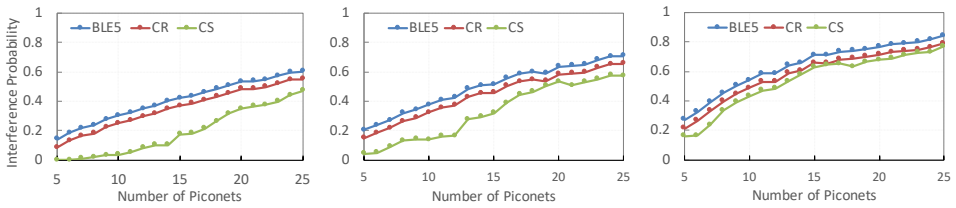
Although the CR algorithm performs well in reducing the interference among piconets, most operating systems or SDKs do not provide API to read the access address of connections for security reasons. With the access address, the hop sequence of a connection can be easily discovered. The Channel Separation algorithm (CS) is more practical, operates independently of the access address, and does not need the emulation of CSA#2. However, since CR does not use the access address, it cannot calculate the exact hop sequence of masters and consequently loses the accuracy of CR in estimating the actual interference. When we have the exact hopping sequence, we can accurately determine the overlapping channels of the interfering piconets. However, in CS, we have channel maps of piconets, and we only know which channels have probability of a collision.

As shown in Algorithm 2, CS tries to assign isolated channels to masters from their good channel set to ensure they do not collide at any time, even by chance. Using CS, masters advertise only their channel map to the gateway. After receiving enough advertisements, the gateway executes the CS algorithm. In the beginning, CS calculates I for all channels

and piconets based on the channel maps.
From then, CS is executed in rounds. As BLE requires at least two good channels to establish a connection (Pang et al. (2021)), CS 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 $N(c)$. The corresponding bits of the selected channels in the channel map of the master are set to 1, and the other bits are set to 0. If CS could not find at least two channels for a master, the channel map remains untouched. CS tries to assign more isolated channels to masters in the subsequent rounds. CS terminates when no channel map can be improved. Updated channel maps are advertised back to the masters one by one.
Sometimes it is impossible to apply the CS algorithm due to the lack of distinct good channels, for example, when the environment is very noisy, or there are more than 40 adjacent piconets. We know that the existence of a common channel between the piconets does not necessarily mean interference because, with the frequency hopping mechanism, this possibility is less. In these cases, the CS algorithm allows the channels to be overlapped to a certain extent. In the simulation section, we will show the effect of overlapping.
Since the gateway running the CS algorithm does not have access to the hopping sequence of masters and must decide only by the channel maps, we have to modify some the formulations. For interference probability expressions, we assume $L = 40$, and the channel map is supposed as 40 hops. The formulation of the channel throughput indicator is corrected as follows.

$$N(c) = \sum_{t}^{n} c \in G(t) \tag{10}$$

With these corrections, the gateway can decide based on the similar formulation of CR and BLE5 without considerable changes.



(a) m=0.3 (b) m=0.5 (c) m=0.7

Figure 4 Interference Probability of network of piconets with different interference severities

5 Simulation Results

This section presents our simulation settings and results. Our simulations compare the total interference (I) and average Throughput Indicator ($T(t)$) of piconets obtained with standard BLE, CR, and CS algorithms.

5.1 Simulation Settings

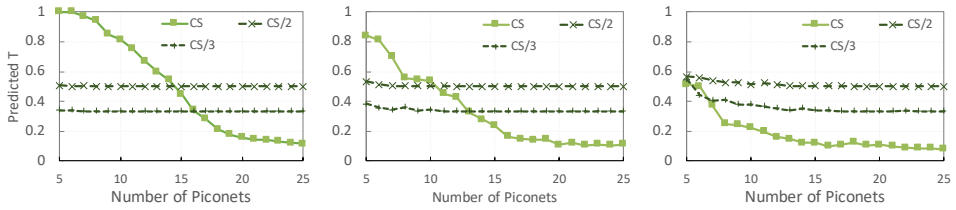
In our tests, the number of piconets varies from 5-25 networks, and the hopping length is set to 100. We generated random channel maps for each piconet based on an environment interference severity parameter m. This parameter controls the percentage of bad channels in the initially generated channel maps. Tests performed with $m = 0.3, 0.5$, and 0.7 values. Since the channel maps are generated randomly, the algorithm is executed 20 times and averaged on piconets to get stable results with a margin of 5%. We also assume the interference threshold to start the interference resolution procedure is 5%.

5.2 Network Interference Probability

Figure 4 shows the I in three environments with different amounts of noise. From the figures, we see that when the number of piconets in an environment grows from 5 to 25 networks, the probability of interference increases three to four times. This result indicates the importance of studying and addressing the inter-piconet interference issue. In addition, the interference probability increases by 1.5 to 2 times when the environment noise (m) rises from 0.3 to 0.7.
The CR algorithm closely follows the BLE5 trace as CR re-maps the channels of piconets based on CSA#2 emulation. CR results in lower I by 2-10%. In contrast, CS halves the interference probability in low to moderate environments. With a large number of piconets within a range and high interference severities, there is no room for channel isolation, and CS like CR converges to the standard BLE.

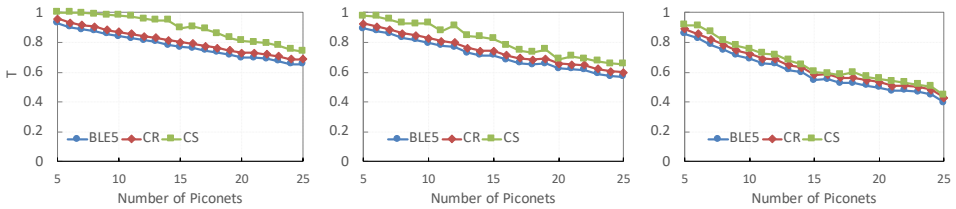
5.3 Throughput Indicator

Figure 5.3 shows the throughput indicator prediction based on the equation 6. We added two CS/2 and Cs/3 lines to the charts that show the value of predicted T when CS allows two and three-channel overlaps. CS with no overlap provides the possibility of very high throughput, but it declines very fast with the increasing number of piconets. CS/2 and CS/3 present a middle performance while allowing a certain and controlled amount of overlap in



(a) m=0.3 (b) m=0.5 (c) m=0.7

Figure 5 Average Throughput Indicator ($T(t)$) of a network of piconets with different interference severities predicted by modified interference formulation



(a) m=0.3 (b) m=0.5 (c) m=0.7

Figure 6 Average Throughput Indicator ($T(t)$) of the network of piconets with different interference severities by emulation of CSA#2

channels. Distributing distinct channels among piconets is possible in low density and noise environments, as is seen in figure 5c, which shows the T reaches under 0.2 with under ten piconets. From another angle, the value of this parameter shows how effective the proposed algorithm can be in improving the interference and link quality.

Figure 6 presents the result of the actual throughput indicator for all three methods. With the actual T, we mean the gateway determines the channel map of the masters using the CS method without knowing their exact hop sequence. Nodes receive the channel maps and hop on channels determined by the normal CSA#2 method. We save the hop sequence of all masters for more than 100 hops and calculate the mean of T for all masters.
In low and medium noise conditions, the CS algorithm results in a 10% better throughput indicator, which means lower interference and channel quality. While in $m = 0.7$ condition this gap is around 1-2%. By the way, CS gives the best channel quality with a lighter algorithm and without a perfect knowledge of the hopping sequence of masters.

6 Conclusion and Future Works

This research studies the BLE5 inter-piconet interference problem and proposes a BLE-based solution to improve the situation. The solution includes a BLE gateway that gathers the channel map of its surrounding masters. The gateway computes a new set of channel maps for all masters to reduce inter-piconet interference. The research also proposes two BLE5-compatible algorithms to compute the channel maps. We showed the effectiveness of the proposed algorithms with the emulation of the BLE5 channel selection #2 algorithm.

References

Al Kalaa, M.O., Refai, H.H., 2015. Selection probability of data channels in bluetooth low energy, in: 2015 International Wireless Communications and Mobile Computing Conference (IWCMC), IEEE. pp. 148–152. URL: http://ieeexplore.ieee.org/document/7289073/, doi:10.1109/IWCMC.2015.7289073.

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

Bluetooth Special Interest Group, 2019. Bluetooth core specification rev. 5.2. URL: https://www.bluetooth.org/docman/handlers/downloaddoc.ashx?doc_id=478726. retrieved: 2022-09-14.

Böcker, S., Arendt, C., Wietfeld, C., 2017. 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), pp. 1–7. doi:10.1109/PIMRC.2017.8292720.

Chek, M.C.H., Kwok, Y.K., 2004. 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. pp. 391–396.

Collotta, M., Pau, G., Talty, T., Tonguz, O.K., 2018. Bluetooth 5: A concrete step forward toward the iot. IEEE Communications Magazine 56, 125–131.

El-Hoiydi, A., 2001. Interference between bluetooth networks-upper bound on the packet error rate. IEEE Communications letters 5, 245–247.

Hu, Y., Wang, G., Shan, L., Yuan, Z., Ouyang, Y., 2011. Inter-piconet interference mitigation schemes for converged BTLE and cellular network, in: 2011 International Conference on Wireless Communications and Signal Processing (WCSP), IEEE. pp. 1–5. URL: http://ieeexplore.ieee.org/document/6096748/, doi:10.1109/WCSP.2011.6096748.

Naik, K., Wei, D.S., Su, Y.T., 2003. Packet interference in a heterogeneous cluster of bluetooth piconets, in: 2003 IEEE 58th Vehicular Technology Conference. VTC 2003-Fall (IEEE Cat. No. 03CH37484), IEEE. pp. 582–586.

Naik, K., Wei, D.S., Su, Y.T., Shiratori, N., 2005. Analysis of packet interference and aggregated throughput in a cluster of bluetooth piconets under different traffic conditions. IEEE Journal on selected areas in communications 23, 1205–1218.

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

Pang, B., T'Jonck, K., Claeys, T., Pissoort, D., Hallez, H., Boydens, J., 2021. Bluetooth low energy interference awareness scheme and improved channel selection algorithm for connection robustness. Sensors 21, 2257.

Pang, B.Z., Claeys, T., Pissoort, D., Hallez, H., Boydens, J., 2019. Comparative study on afh techniques in different interference environments, in: 2019 IEEE XXVIII International Scientific Conference Electronics (ET), IEEE. pp. 1–4.

Tosi, J., Taffoni, F., Santacatterina, M., Sannino, R., Formica, D., 2017. Performance evaluation of bluetooth low energy: A systematic review. Sensors 17, 2898.

Yufeng, N., Yiqi, Z., Naikuo, C., 2009. Analysis of packet interference and aggregated throughput of bluetooth piconets, in: 2009 International Conference on Networks Security, Wireless Communications and Trusted Computing, IEEE. pp. 129–132.