Evaluating the Reliability of Bluetooth Low Energy under Wi-Fi Interference

Maja Ovčarik
Graz University of Technology
maja.ovcarik@student.tugraz.at

ABSTRACT

Due to the fact that ISM Band is permanently free for unlicensed use, ISM band has become crowded with devices which use different variation of IEEE 802.11 standard as well as IEEE 802.15.4 standard together with Bluetooth, Bluetooth Low Energy (BLE) and many others. Since BLE was introduced in 2010, it has been one of the key technologyes for Internet of Things. BLE claims to provide very reliable communication even under heavy interference. The objective of this research is to determine reliability of a BLE connection under different Wi-Fi interference scenarios such as varying transmission power, different channels and different positions of the interference source. We measure the BLE connection between a Raspberry Pi 3 Model B, used as IPv6-over-BLE border route and BLE master, and a TI CC2650 SensorTag as BLE slave running BLEach. BLEach is an IPv6-over-BLE communication stack for constrained embedded devices. Raspberry Pi 3 is often used as a border router, and there is no detailed knowledge about its performance under different interference scenarios. The study certainly adds to overall understanding of the behaviour and implementation of adaptive frequency hopping mechanism in the Raspberry Pi 3.

KEYWORDS

Bluetooth Low Energy, Wi-Fi interference, SensorTag, Channels

1 INTRODUCTION

Since it was first used in 1999 by British entrepreneur Kevin Ashton, term Internet of Things (IoT) has been attracting a lot of interest. According to Ashton, IoT is a term that describes a system where the Internet is connected to the physical world via ubiquitous sensors [7]. Rapid development of electronics and digitalization combined with continuously decreasing prices of related technologies and the growth of Internet access led Ericsson's former CEO to predict that by 2020 the world will have 50 billion connected devices [18]. Nowadays, there are around 20 billion devices connected to the Internet and most of them are connected wirelessly using the unlicensed 2.4 GHz ISM band. Devices utilizing this band employ different variation of IEEE 802.11 standard as well as IEEE 802.15.4 standard together with ZigBee, Bluetooth, Bluetooth Low Energy (BLE) and many others.

ISM band has been a popular option for an increasing number of wireless sensors and devices due to the fact that it is permanently free for unlicensed use, but consequently ISM band has become crowded. When Bluetooth Specification v4.0[3] was published in 2010, classical Bluetooth has been redesigned into a new short range, ultra-low power consumption and simple hardware implementation called BLE or Bluetooth Smart[9]. Since then, BLE has been the key enabler technology for The Internet of Things[20]. BLE claims to provide very reliable communication even under heavy interference. To achieve this reliability, BLE uses a mechanism called Adaptive Frequency Hopping.

Although the frequency hopping mechanism was proposed several years ago, there haven't been many researches that evaluate it's performance in the real world. In addition, one research has surveyed how BLE behaves in realistic Wireless environments[6]. According to this research, authors surveyed environments which exhibited high spectrum occupancy due to a large number of WiFi users. Results of this research illustrate BLE's high ability to initiate a connection and successfully exchange data in high interference environments like University food court, Hospital Intensive care unit and Sport Facility. On the contrary, the objectives in this research are to determine reliability of a BLE connection in controlled environment [1] with Wi-Fi interference scenarios such as varying transmission power, different channels and different positions of the interference source. BLE connection is realised between Raspberry Pi 3 Model B running the Raspbian OS[1] and a TI CC2650 SensorTag[11] running BLEach[15]. BLEach is an IPv6-over-BLE communication stack for constrained embedded devices. The Wi-Fi interference in this experiments is generated using the built-in Wi-Fi radio of another Raspberry Pi 3 Model B.

The overall structure of the research takes the form of six chapters. Chapter two begins by summarizing the background information of this research and discussing the adaptive frequency hopping algorithm of BLE. The third chapter is concerned with the methodology used for this study, graphical presentation of measurements and measurement setup description together with data collection during several measurements with different setup scenarios. Chapter four presents the findings of the research, focusing on concrete numbers regarding successful and unsuccessful connections as well as when and which channels are enabled or disabled for communication. To conclude this document, chapter five has a quick overview addressing results and outlook on future work.

2 BACKGROUND

With more than 20 billion devices connected to the Internet, most of them using ISM band, coexistence of different standardized communication technologies in the same frequency band is crucial. At the same time coexistence of different wireless communication technologies in the ISM band means more radio interference. Radio interference decreases reliability and energy-efficiency due to retransmission and lead to packet loss and high latencies while

[1] Environment is not fully controlled because there is possible appearance of interference outside of room in which was everything measured, because room is not specifically isolated for this purpose.

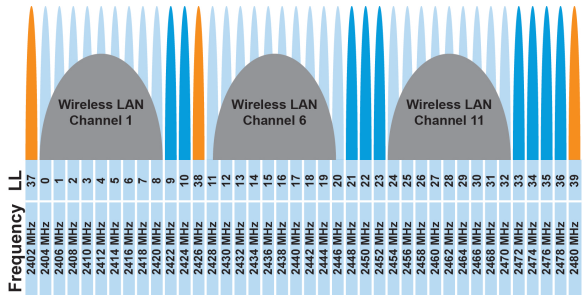

Figure 1: ***Wi-Fi and BLE channels[10]***

decreasing the quality of communication in general. Communication in the ISM frequency band can be affected by electromagnetic interference. Electromagnetic interference (EMI), also called radio-frequency interference (RFI) when in the radio frequency spectrum, is a disturbance generated by an external source that affects an electrical circuit by electromagnetic induction, electrostatic coupling, or conduction[2]. The disturbance may degrade the performance of the circuit or even stop it from functioning. In the case of a data path, these effects can range from an increase in error rate to a total data [16]. In addition, BLE signals are likely to be completely lost due to lower transmission power if there is a large amount of electromagnetic interference in the environment. Electromagnetic interference can be caused by microwave ovens or baby monitors.

2.1 *Bluetooth Low Energy (BLE)*

BLE system uses 40 RF channels. These RF channels have their central frequencies at $2402 + 2kMHz$ where $k \in [0..39]$. Channels 37, 38 and 39 are used for advertising and channels from 0 to 36 are used for bidirectional data exchange using BLE connections. Other than classic Bluetooth where devices continuously scan each channel for incomming connections, BLE uses only the three advertisement channels for connection setup. By using these dedicated channels, a connection between two Bluetooth LE devices can be set up in less than three milliseconds[14]. BLE supports connection-less and connection-based communication mode. Most IoT applications which use BLE usually communicate in connection-less mode [8]. Connection-less communication means that a message can be sent from one end point to another without prior connection establishment. On the other hand, in connection-based communication, devices which exchange messages are in the connected link layer state and they exchange data during periodic connection events. IPv6 over BLE used in this paper, requires devices to communicate bidirectionally using connection-based communication[15].

In order to establish a connection between two devices, the slave needs to advertise that he supports connection, after which the master takes on the role of the initiator and transmits a connection request to create a connection between the initiator and the advertiser. During the connection setup, the initiator provides the advertiser with information for the adaptive frequency hopping and timing parameters for the established connection. The advertiser also receives a channel map which contains information regarding used and unused data channels. Channel map is a 40 bit sequence, that assigns a bit value 1 if channel is marked as used, consequently bit value 0 is assigned if channel is marked as unused[4].

After a connection has been successfully established, data is transmitted using Adaptive Frequency Hopping (*AFH*). AFH is used to improve the performance of the BLE connection in the presence of radio interference as well as reducing the interference caused by physical links on other devices in the ISM band[19]. The basic principle is that BLE channels are classified into two categories, used and unused, where used channels are part of the hopping sequence and unused channels are replaced in the hopping sequence by used channels in a pseudo-random way[4]. During a connection event, the master and slave alternately send and receive packets until there is no more data to be transmitted and the connection event is closed. Beginning of a connection event is also called anchor point[2], during which the adaptive frequency hopping algorithm determines which data channel will be used during the event. Calculation of the unmapped channel index is accomplished by mapping this index to a data channel index from the set of used channels in channel map. Unmapped channel is calculated using the following algorithm :

$$unmappedCh = (lastUnmappedCh + hopIncrement) \bmod 37 \quad (1)$$

The *unmappedCh* and *lastUnmappedCh* are the unmapped channel indices of two consecutive connection events, *hopIncrement* is specified by the master during connection establishement and has random value in the range of 5 to 16. After calculation, if unmapped channel is an unused channel according to the channel map, the unmapped channel is remapped to one of the used channels in channel map with the following algorithm:

$$remappingIndex = unmappedCh \bmod numUsedCh^3 \quad (2)$$

On the other hand, if unmapped channel is a used channel according channel map, the unmapped channel is used for data transmission without remapping[4].

2.2 *Wi-Fi and BLE interference*

Wi-Fi has fourteen channels defined for use in the 2.4 GHz ISM band. However, the FCC allows only 11 channels in what is often called the North American domain, while ETSI [4] defines 13 channels which are used in Europe. The channels used for Wi-Fi are separated by 5 MHz in most cases but have a bandwidth of 22 MHz[12]. As a result, neighbouring channels overlap and it is possible to find a maximum of three non-overlapping channels. Therefore if there are adjacent WLAN devices that need to work on non-interfering channels, there can only be a maximum of three devices in the same environment. Wi-Fi routers are often configured to channel 6 as the default channel which determines that channels 1, 6 and 11 are non-overlapping and therefore most widely used. BLE advertising channels center frequencies are selected to minimize interference with this popular channels, as shown in Figure 1.

In case of Wi-Fi interference during BLE connection between devices, the master determines which channels show lower transmission qualities in order to avoid using them. Such channels are marked as unused. After refreshing the channel map, master will transmit the updated channel. After at least 6 connection events have been carried through, channel map will start being in use. If

[2] Anchor points are timed so that there aren't any overlaps.
[3] *numUsedChannels* is the number of used channels in the channel.

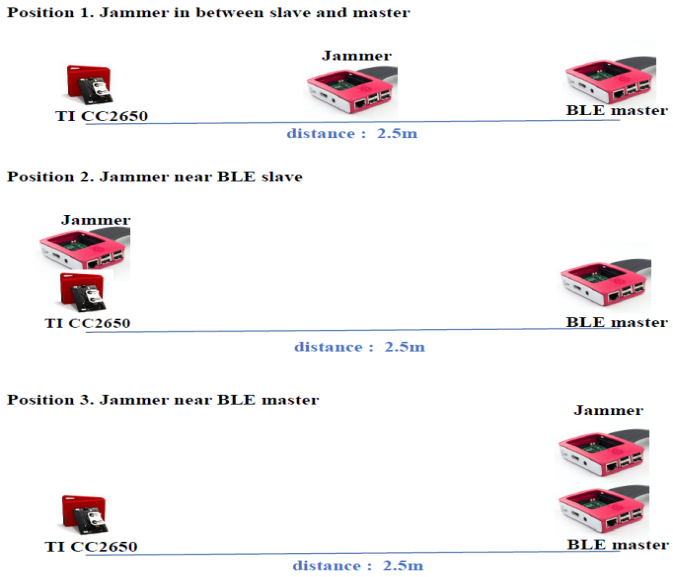

Figure 2: ***Position of devices used for measurements.***

a channel cannot be used for transmission due to interferenc, the packet will be retransmitted during the next connection event on a different data channel. Channels with interference are dynamically disabled and skipped next time, this process is called blacklisting. Blacklisting consists in identifying the channels which exhibit the lowest reliability to avoid using them for the transmissions[17]. Whitelisting is the process of re-enabling channels so that they will be used again.

Risk of communication obstruction is proportional to the level of interference in the channel, which often can result in different errors. Some of them are:

Connection error. In order to avoid such errors, each BLE node will hop over a set of available data channels determined by the master node in channel map. However, there is a possibility that a channel which has been selected for a connection event may be interferred. In such a case, the connection event will fail due to a connection error.

Connection loss. After several subsequent failed connection events, master and slave can become asynchronous. Hence the connection can be lost, meaning that the link between master and slave is completely interrupted.

Failure to start a connection. During connection initialization, it is possible that a BLE device finds fewer available data channels than those needed to establish connection. Standard specifications mandate at least two available data channels for communication, and if that minimum is not met, the connection is blocked.[6]

3 METHODOLOGY

All measurements were performed using the same setup described in following section.

3.1 *Setup*

Raspberry Pi 3 model B containing BCM43438 chip for wireless LAN and Bluetooth Low Energy (BLE) on board[5] and running *Raspbian OS* (Version from before 2017-08-17)[1] is used as BLE Master and IPv6-over-BLE border router. Border router must support Bluetooth low energy and 6LoWPAN over BLE. In order to configure Raspberry Pi 3 model B as a IPv6-over-BLE as a border router, an *nRF5 IoT SDK* installation guide was followed. This installation guide shows how to use BLE links to connect IoT devices to a BLE-enabled router which is connected to the internet via IPv6[13], while using UDP as the default and only transport protocol. To ensure that BLE master and slave are communicating, the script `udp-echo-server.py` is implemented and running on the master device. Moreover, TI CC2650[11] running *Contiki OS* with integrated *BLEach* is used as the slave device. *BLEach* is the first full-fledged IPv6-over-BLE stack that exposes tuning knobs for controlling the energy usage and timeliness of BLE transmissions and allows enforcing a variety of quality-of-service (QoS) metrics. BLEach is lightweight, interoperable with other standard-compliant devices and reduces energy costs by up to 50% while giving QoS guarantees by quickly adapting to changes in interference, traffic priority and traffic load[15]. Another Raspberry PI model B is used as the source of Wi-Fi interference with same specifications as the one previously mentioned. This device implements a jammer for generating interference with controlled parameters.

3.2 *Data collection*

Data acquired from measurements is graphically presented using a Python script which displays the distribution of successful and unsuccessful transmissions on each channel, as well as the distribution of transmission errors during the entire experiment and the usage of channels before, during and after Wi-Fi interference. Connection events were converted into the time domain by using information regarding the interval between two connection events, provided by the BLE master at the starting point of connection establishment process. Interval value was 54 and according to the [4], this value needs to be multiplied with 1.25ms to alter it in time domain. The time accuracy of Wi-Fi interference was achieved by using a bash script which precisely determines the beginning and the end of Wi-Fi dummy data transmission. During the measurements, the distance between BLE master and BLE slave was set to 2,5 m.

External Wi-Fi interference severely affects BLE data channels. In case of high interference within a channel, it will be blacklisted. Channel interference depends on several factors: physical position of source which is generating interference, position of interference in the spectrum, interference power, packet lengths and many others. Consequently, following challenges are addressed and surveyed in this paper:

Blacklisting. First set of measurements examined the impact of Wi-Fi interference with different transmission power and channels on blacklisting. Goal was to determine whether blacklisted BLE channels are the expected ones according Figure 1. For this measurements position 1 shown in Figure 2 was used, together with data length set to maximum of 1500 bytes and transmitting packet every 15 ms. Furthermore, measurements begin with 45 sec without any interference, subsequently 45 sec with Wi-Fi interference and

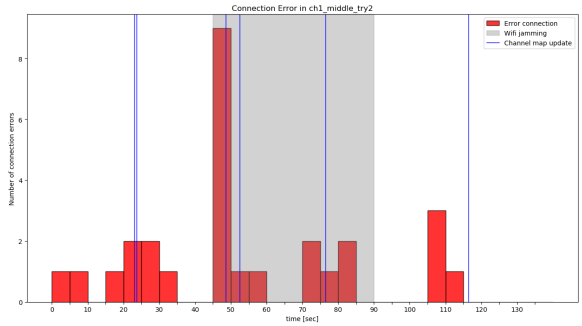

Figure 3: ***Ch1 - pw[100 mW]- duration 135s***

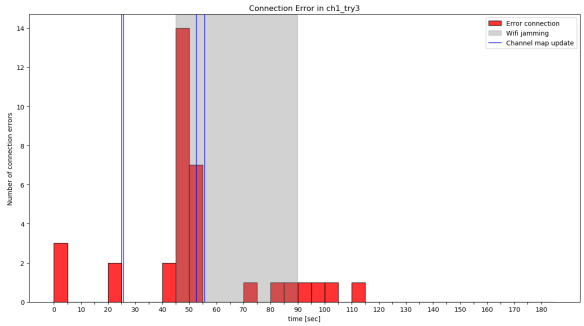

Figure 4: ***Ch1 - pw[100 mW] - duration 180s***

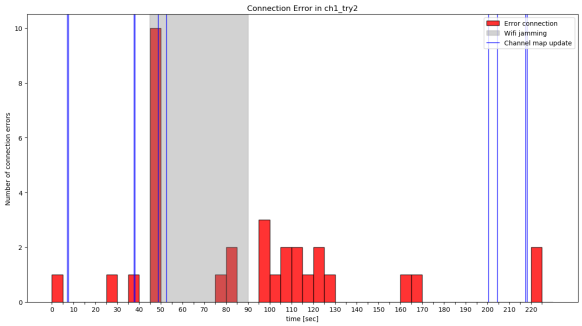

Figure 5: ***Ch1 - pw[100 mW] - duration 225s***

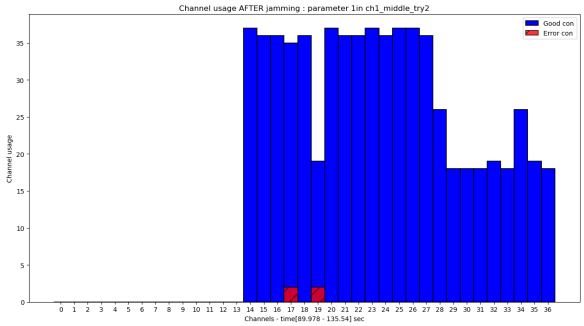

Figure 6: ***Ch1 - pw[100 mW] - duration 135s - channel usage after jamming***

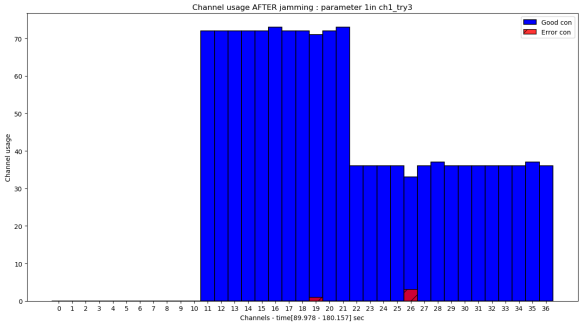

Figure 7: ***Ch1 - pw[100 mW] - duration 180s - channel usage after jamming***

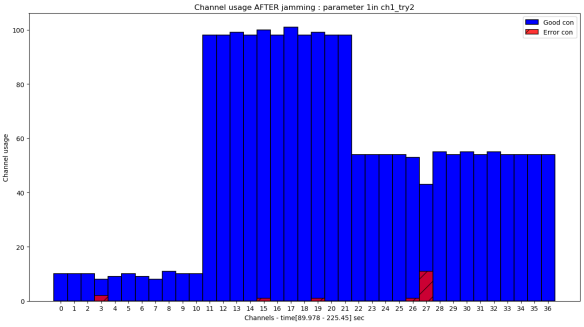

Figure 8: ***Ch1 - pw[100 mW] - duration 225s - channel usage after jamming***

finally last 45 sec without interference. This form was chosen, in order to see how many channels are blacklisted and which ones are used during Wi-Fi interference compared to channel usage before and after interference. This test was also used to determine is the difference in connection errors between various channels.

To achieve higher measurement reliability, each measurement was repeated at least three times. Data collected at the beginning of the research has shown that repeating measurements with rebooting BLE slave in between measurements, has a decreasing number of connection errors (Table 2). In the interest of investigating the relationship between number of connection errors (in repeated measurements) two cases were carried out:

1. Always rebooting BLE master and slave
2. Only rebooting the BLE slave

With constant transmission power set to 100 mW and position 1 shown in Figure 2, time management has been the same as described above. Results showed that BLE master persistently keeps

previously blacklisted channels. Due to the results, all test were made with rebooting both BLE devices.

Whitelisting. Proper understanding of BLE functionalities under severe Wi-Fi interference includes determining when channels are removed from the blacklist by measuring the time before BLE resumes transmitting data on previously blacklisted channels. Besides measuring how long it takes to whitelist all channels after jamming, this paper also analyses whitelisting channels in case of decreasing power of the source of interference. An experimental demonstration of whitelisting was carried out by having slightly different sets of measurements compared to measurements used for blacklisting. Time after jamming was prolonged for the sake of discovering the point when all channels were whitelisted after previously being under Wi-Fi interference. Initially, time was extended for 45 sec, but the results were not satisfactory, so the next iteration of measurements was was extended for another 45 sec which makes total measurement duration exactly 225 sec.

Position of the source of Wi-Fi interference. Does the number of successful connections and ratio of good connection events to error connection events depend on physical positioning of the source of Wi-Fi interference? Initial measurements to answer this questions were conducted with the interference source placed next to the master or slave device, interfering with transmission power 100. In those cases the interference was strong and caused immediate connection loss. Therefore all subsequent measurements, with transmission power 100, were conducted with the interference source placed in the middle between the master and slave, where that transmission power didn't cause connection loss. In further measurements transmission power has been changed from 0 to 100 with a step of 5, in order to confirm highest possible transmission power (Table 1) for each Wi-Fi channel with the setup set as shown in Figure 2 on position 2 and 3. Measurements were also made with rebooting both BLE devices to avoid keeping previously blacklisted channels in master's memory.

4 RESULT

Various interference patterns were observed in order to determine how the Raspberry Pi is blacklisting BLE data channels. When interference was generated on Wi-Fi channel 1 with transmission power 100 mW, first channel map update was received within 3 seconds after starting interference (Figure 14), and after the second channel map update, the drop in the number of connection errors was observable. Wi-Fi channel 1 include BLE channels from 0 to 8, during Wi-Fi interference BLE channels from 0 to 13 were blacklisted, as shown in Figure 9. Therefore, the number of blacklisted BLE channels was larger than number of BLE channels which Wi-Fi channel 1 includes. Furthermore, when interference was generated with transmission power 10 mW on the same channel, blacklisted channels were from 2 to 6 (Figure 10). Hence, one could infer that with decreasing transmission power, the number of blacklisted BLE channels drastically reduces. Therefore, the same measurements were conduced for each Wi-Fi channel. Interference generated on Wi-Fi channel 7 with transmission power 100 mW, had as a result a histogram shown on Figure 11, blacklisted BLE channels were from 11 to 28, whereas Wi-Fi channel 7 include channels from 12 to 23. When transmission power was set on 50 mW on the same Wi-Fi channel, blacklisted BLE channels were from 11 to 25, which means that only 3 channels less have been blacklisted and power was decreased for 50, as shown on Figure 12. Finally, with transmission power 10 mW, no channels were whitelisted compared to previous measurements (Figure 13). Interference generated on other Wi-Fi channels shown similar behaviour to channel 7. Consequentially, it can be concluded that with drastically decreased interference power, number of blacklisted channels is usually smaller for 3.

Data collected during the research has shown, that there is a significant difference between the following two conditions: repeating measurements with re-booting both BLE devices and with re-booting only BLE slave. Re-booting only BLE slave has as result decreasing number of connection errors as shown in Table 2. Every *try* in this table was sampled with 2000 connection events, which is exactly 135 sec, interference duration was 45 sec. With every *try*, connection error was decreasing, whereas connection success was increasing. In the second part of this table, one can find the outline of connection errors, divided in three sections. Connection errors before jamming, during and after jamming for every *try*. Table 3 has same information as Table 2, but this time both BLE devices were re-booted. These results indicate that connection errors and number of repeated measurements are not in any correlation. Therefore, it is concluded that BLE master keep previously blacklisted channels in RAM memory. It also can be concluded that there is no interrelationship between total number of connection errors and interference on specific channel.

The results of measurements made for revealing how long Raspberry Pi 3 model B needs to resume using again all BLE channels after Wi-Fi interference. Figure 3 shows measurements on channel 1 with a duration of 135 sec, last 45 sec shown were without interference and as we can see on Figure 3 there was only one channel map update after interference was terminated. Hence, on Figure 6, it's shown that even when 45 sec passed after jamming, previously blacklisted BLE channels are still not in use for communication. This measurement was repeated with interference on every channel and all results were similar. Further duration of measurements was increased to 180 sec, on Figure 4 can be seen that after the second channel map update during Wi-Fi interference, the number of connection errors has decreased. In the next 90 sec after interference there hasn't been any channel map updates, therefore channels from 0 to 10 were not used for communication (Figure 7). Last measurement showed, that after 110 sec passed from Wi-Fi interference (Figure 5), channel map was updated and from that moment messages were transmitted using every BLE channel (Figure 8). Same conclusions were made for every Wi-Fi channel. Evidence collected in this study indicate that border transmission power with connection loss between BLE devices is: 10 mW for every channel when the jammer is near BLE master, with the jammer positioned near BLE slave transmission power was different for every channel as shown in Table 1. Finally, when jammer is between BLE slave and master, transmission power can be up to 100 mW without connection loss.

5 CONCLUSION

In order to have successful communication using different standards at the same time, knowledge about interference in between

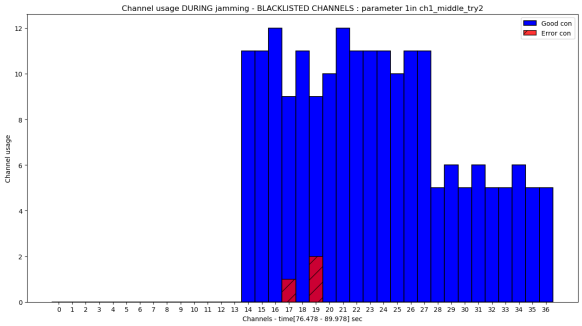

Figure 9: ***Ch1 - pw[100 mW] -blacklisted channels***

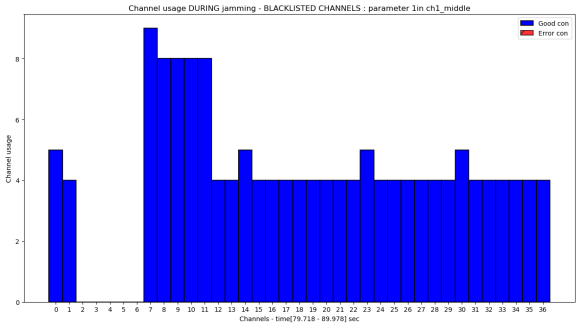

Figure 10: ***Ch1 - pw[10 mW] - blacklisted channels***

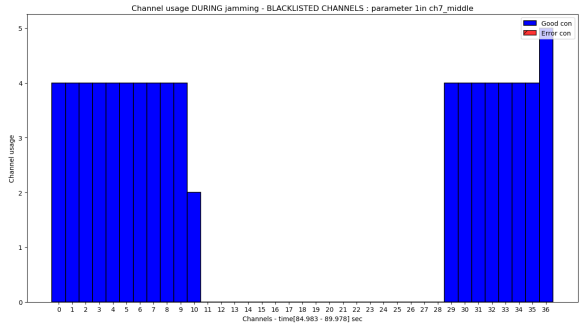

Figure 11: ***Ch7 - pw[100 mW] - blacklisted channels***

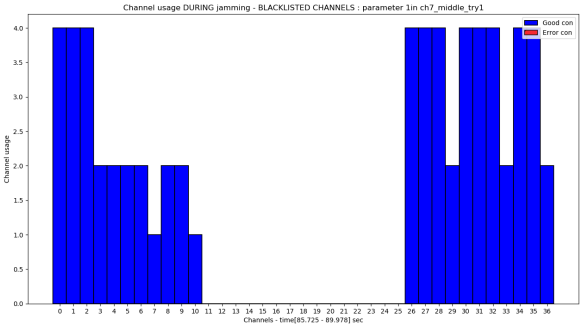

Figure 12: ***Ch 7 - pw[50 mW] - blacklisted channels***

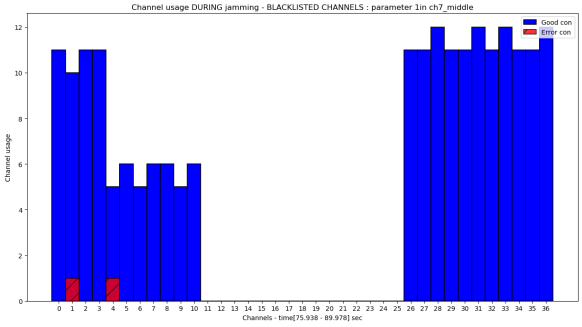

Figure 13: ***Ch 7 - pw[10 mW] - blacklisted channels***

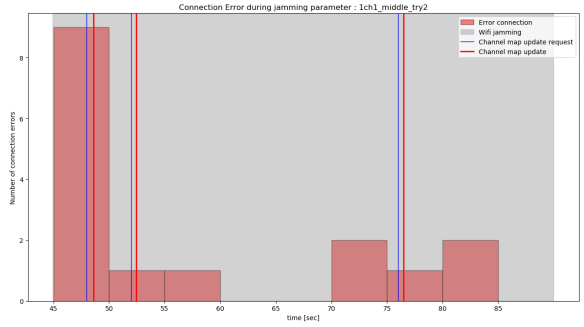

Figure 14: ***Ch1 - pw[100 mW] - channel map update - during jamming***

Wi-Fi channel	1	2	3	4	5	6	7	8	9	10	11	12	13	14
PW[mW]	100	70	70	45	30	25	35	25	35	20	30	40	75	–

Table 1: Border transmission power for connection loss - position 2 in Figure 2

them is crucial. Therefore, this paper has explored how BLE connections work under different Wi-Fi interference patterns, regarding different parameters such as transmission power, position of interference source and different Wi-Fi channels but with the distance between BLE slave and master fixed to 2.5m.

Channel	Ch 6			Ch 7			Ch 12		
Try	try 1	try 2	try 3	try 1	try 2	try 3	try 1	try 2	try 3
connection good	1951	1981	1987	1974	1983	1987	1978	1983	1984
connection error	49	19	13	26	17	13	22	17	16
channel map update	10	9	9	11	14	13	14	12	13
error/try	before	during	after	before	during	after	before	during	after
try 1	10	30	9	3	21	2	3	16	3
try 2	4	12	3	6	9	2	4	10	3
try 3	1	11	1	9	3	1	7	9	0

Table 2: Connection error when rebooting BLE Slave - position 1 in Figure 2

Channel	Ch 6			Ch 7			Ch 12		
Try	try 1	try 2	try 3	try 1	try 2	try 3	try 1	try 2	try 3
connection good	1985	1971	1982	1981	1992	1986	1981	1972	1973
connection error	15	29	18	19	8	14	19	28	27
channel map update	9	14	14	15	10	14	13	11	12
error/try	before	during	after	before	during	after	before	during	after
try 1	5	8	2	9	7	3	5	11	3
try 2	6	9	14	1	7	0	8	16	4
try 3	3	8	7	2	9	3	4	13	0

Table 3: Connection error when rebooting both BLE devices - position 1 in Figure 2

This effort provides insight about performance of the Raspberry Pi 3 model B, a very popular border router, under different Wi-Fi interference scenarios. The results of measurements have shown that border transmission power with connection loss between BLE devices depends on the position of source of Wi-Fi interference. When source of Wi-Fi interference is near master or slave communication will most likely to be lost, whereas when source of Wi-Fi interference is between master and slave, there will appear few connection errors but communication will be sustained. Secondly, the evidence from this study suggests that BLE master keeps previously blacklisted channels in RAM memory. Moreover, it has been shown that during high interference in order to maintain connection, the master device has been blacklisting around 15 to 20 BLE channels, even though interference was generated only on one specific Wi-Fi channel that has frequency which corresponds to around 10 BLE channels. Lastly, no evidence was found that number of errors depends on which Wi-Fi channel the interference is generated.

Limitations. One source of weakness in this study which could have affected the measurements is that the room, where everything was measured, hasn't been fully isolated. Next limitation is the fact that timing the start of interference exactly at same connection event isn't easily repeatable. In spite of its limitations, the study certainly adds to understanding of the AFH mechanism implemented on the Rasperry Pi 3 model B and discovers some of it's limitations.

Considerably more work will needed to determine why the behaviour is different depending on the position of the source of interference. Further studies could assess if there is any difference between connections error and maximum transmission power before losing the connection, if the distance between BLE devices is volatile. Several questions still remain to be answered.

REFERENCES

[1] [n. d.]. *Instaling Operating System Images.* https://www.raspberrypi.org/documentation/installation.

[2] 2005. Based on the "interference" entry of The Concise Oxford English Dictionary (11th edition). (2005).

[3] 2010. *Specification of the Bluetooth system, version 4.0.*

[4] 2014. *Specification of the Bluetooth system, version 4.2.*

[5] 2016. *Raspberry Pi 3 Model B - Specification.*

[6] Mohamad Omar Al Kalaa, Walid Balid, Naim Bitar, and Hazem Refai. 2016. Evaluating Bluetooth Low Energy in Realistic Wireless Environments. (04 2016).

[7] Kevin Ashton. 2009. That 'Internet of Things' Thing. *RFID Journal* (June 2009).

[8] R. Faragher and R. Harle. 2015. Location Fingerprinting With Bluetooth Low Energy Beacons. *IEEE Journal on Selected Areas in Communications* 33, 11 (Nov 2015), 2418–2428. https://doi.org/10.1109/JSAC.2015.2430281

[9] N. Gupta. 2013. *Inside Bluetooth Low Energy.* Artech House. https://books.google.at/books?id=-LMq0NhoEQgC

[10] Mark Hughes. 2017. LABeL: Troubleshooting Tools for Your Next Bluetooth LE Project: Ubertooth and the Nordic nRF Sniffer. (7 2017).

[11] Texas Instruments. 2015. *CC2650 SimpleLink Multistandard Wireless MCU.*

[12] Jim Lansford and Adrian Stephens. 2004. Wi-Fi (802. 11 b) and bluetooth: enabling coexistence - IEEE Network.

[13] Nordic Semiconductor. [n. d.]. *nRF5 IoT SDK, version 0.9.0.*

[14] Michael Spörk. 2016. *IPv6 over Bluetooth Low Energy using Contiki (Master's thesis).* Master's thesis. Graz University of Technology, Graz, Austria.

[15] Michael Spörk, Carlo Alberto Boano, Marco Zimmerling, and Kay Römer. 2017. BLEach: Exploiting the Full Potential of IPv6 over BLE in Constrained Embedded IoT Devices. (11 2017).

[16] M. K. Sue. 1981. *Radio frequency interference at the geostationary orbit.* Technical Report.

[17] Lei Tang, Yanjun Sun, Omer Gurewitz, and David B. Johnson. 2011. EM-MAC: A Dynamic Multichannel Energy-efficient MAC Protocol for Wireless Sensor Networks. In *Proceedings of the Twelfth ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc '11).* ACM, New York, NY, USA, Article 23, 11 pages. https://doi.org/10.1145/2107502.2107533

[18] Hans Vestberg. 2010. CEO to shareholders: 50 billion connections 2020. (April 2010).

[19] Thomas Watteyne, Ankur Mehta, and Kris Pister. 2009. Reliability Through Frequency Diversity: Why Channel Hopping Makes Sense. In *Proceedings of the 6th ACM Symposium on Performance Evaluation of Wireless Ad Hoc, Sensor, and Ubiquitous Networks (PE-WASUN '09).* ACM, New York, NY, USA, 116–123. https://doi.org/10.1145/1641876.1641898

[20] Martin Woolley. 2017. Exploring Bluetooth 5 - Going the Distance. (02 2017).