How Low Energy is Bluetooth Low Energy? Comparative Measurements with ZigBee/802.15.4

Matti Siekkinen, Markus Hiienkari, Jukka K. Nurminen
Dept. of Computer Science and Engineering
Aalto University, School of Science, Finland
{matti.siekkinen,markus.hiienkari,jukka.k.nurminen}@aalto.fi

Johanna Nieminen
Nokia Research Center, Finland
johanna.1.nieminen@nokia.com

***Abstract*—Ultra low power communication mechanisms are essential for future Internet of Things deployments. Bluetooth Low Energy (BLE) is one promising candidate for such deployments. We study the energy consumption of BLE by measuring real devices with a power monitor and derive models of the basic energy consumption behavior observed from the measurement results. We investigate also the overhead of IPv6-based communication over BLE, which is relevant for future IoT scenarios. We contrast our results by performing similar measurements with ZigBee/802.15.4 devices. Our results show that when compared to ZigBee, BLE is indeed very energy efficient in terms of number of bytes transferred per Joule spent. In addition, IPv6 communication energy overhead remains reasonable. We also point out a few specific limitations with current stack implementations and explain that removing those limitations could improve energy utility significantly.**

I. Introduction

Internet of Things is about extending the scope of the Internet to cover also physical objects which are not the kind of traditional networked devices. To this end, a large variety of communication solutions have been introduced which enable, for instance, sensors to feed small amounts of data to the Internet energy efficienctly for long periods of time. Bluetooth Low Energy (BLE) [1] is one of the latest developments in this arena. It is a very low power, relatively short range (50m) technology that promises sensors to be able to communicate using a coin cell battery even up to two years [2].

IETF is active in promoting the adoption of IPv6 for these types of networks, which would enable, e.g., sensors to communicate directly to the Internet. The 6LoWPAN [3] working group has worked on specifications especially for IPv6 functionality over ZigBee networks [4]. More recently, work towards IPv6 over BLE links have been initiated [5].

We investigated the energy consumption of BLE by performing measurements with real BLE devices and report the results in this paper. To put our results into perspective, we measured also the energy consumption of ZigBee/802.15.4 in a comparative manner. Our results confirm that BLE indeed consumes very little energy. Based on the results, we also provide simple models which we also use to study the energy consumption in the case of IPv6 over BLE. We also discover some limitations with the BLE stack that we experimented with. Especially, the adaptive frequency hopping was not implemented because of which interference poses a problem.

Classic Bluetooth and ZigBee has been extensively studied in the literature. See, e.g., [6]–[10] for some example studies. To the best of our knowledge, this is the first work to measure and model BLE energy consumption and to contrast the results with a “competing” technology. Our contributions are the following:

- We perform comparative energy measurements with real BLE and ZigBee devices and characterize energy consumed in typical setup.
- Based on the measurements, we build simple models of the energy consumption. We also include scenarios to compute the overhead of future 6LoWPAN deployments.
- We study the impact of interference on both BLE and ZigBee.

II. Bluetooth Low Energy and ZigBee

BLE is a low energy version of Bluetooth specified in the version 4.0 [1]. Two of the lowest layers of BLE stack are Physical (PHY) and the Link Layer (LL). PHY takes care of transmitting and receiving bits. The Link Layer provides medium access, connection establishment, error control, and flow control. The upper layers are Logical Link Control and Adaptation Protocol (L2CAP), Generic Attribute protocol (GATT), and Generic Access Profile (GAP). L2CAP is able to multiplex the data channels from the above layers and provides fragmentation and reassembly for large data packets. Similar to classic Bluetooth (BT), BLE uses adaptive frequency hopping spread spectrum to access the shared channel. However, the number of hops is 43 and the channel width is 2MHz as opposed to 79 hops and 1MHz channel width in classic BT.

BLE device can operate either in master or slave role. A master can manage multiple simultaneous connections with a number of slave devices, but a slave can only be connected to a single master. Therefore, a BLE network topology is a star. Differently from classic BT, discovery is done so that slave advertises on one or several of the three designated advertisement channels. Master scans these channels in order to discover slaves. After discovery, data transmission happens in the form of connection events in which the master and the slave wake up in synchrony to exchange frames. Both devices sleep the rest of time.

ZigBee protocol stack consist of PHY and MAC layers specified in the 802.15.4 standard and a set of layers above them

978-1-4673-0682-9/12/$31.00 ©2012 IEEE

which are specified by the ZigBee alliance. Our comparative measurements focus on the two lowest layers which is why we do not detail the higher layers further. The channel access in 802.15.4 is CSMA/CA as opposed to the frequency hopping of BLE. The over the air data rate is only 250kbit/s compared to 1Mbit/s of BLE.

In ZigBee terminology, a network can consist of end devices, routers, and coordinators. End devices are typically most resource constrained and coordinators have the most capacity. End devices can connect to routers or coordinators which can connect to each other. The resulting topologies can be star or peer-to-peer. In the non-beacon mode, an end device can transmit whenever it desires and the router/coordinator should listen to the medium for any incoming transmissions.

III. Measurement Setup

The measurements were performed using a Monsoon Power Monitor [11] which provides an adjustable voltage output terminal to which the device to be measured can be directly connected. To measure the energy consumption of BLE, we plugged a BlueGiga BLE112 module (based on TI's CC2540 System-On-Chip) to the power monitor and configured it to be a slave device. The slave was communicating with a BLE112 USB dongle which was configured as a master device. Besides what is necessary to communicate over BLE, the slave device does not contain any other hardware which could consume energy and bias the results. It is run with TI's BLE stack (v 1.0) in single chip mode and represents a realistic example of a sensor where the amount of hardware and power consumption is tried to be kept at minimum. The slave is set to generic discoverability and undirected connectability mode.

For the ZigBee measurements, we used TI's sensor boards consisting of MSP430F2274 application microcontroller and CC2530 ZigBee network processor. The boards are meant to be used in dual-chip configuration (MSP430 running the application and CC2530 running network processor firmware). However, we wanted to study the energy consumption of just the CC2530. Therefore, CC2530 was flashed with firmware containing the full Z-Stack (v.2.4.0-1.4.0) and MSP430 is programmed to go to sleep on startup, making this configuration effectively single-chip. It is worth pointing out that except for the radio module, the CC2530 SoC is identical to the BLE's equivalent (CC2540), which makes it possible to do a fair comparison of energy consumption between the two. One sensor was then plugged to the power monitor and that sensor was talking to another one connected to a PC and acting as PAN coordinator. The MAC layer was set to not use the superframe structure (beacons used only for network discovery), which removes the need for end-devices to listen and wait for the beacon before sending data.

Several parameters of the two link layer protocols can be modified. We list the most relevant of them in Table I. In BLE scanning cycle, the scanning interval, i.e. the time between subsequent scanning events, can be determined in addition to the duty interval which determines the fraction of time from the scanning interval that the device spends actively scanning. Z-Stack's poll-rate is the interval in between events where the end-device queries the coordinator whether it has pending data. In our measurements, this did not play a role since the end device was just transmitting data. ACK request is a MAC-level acknowledgment mechanism in 802.15.4.

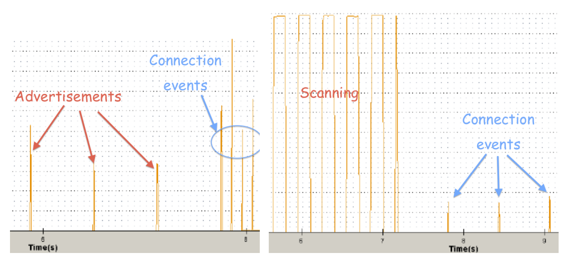
BLE stack	Z-stack
Device role (master/slave) Discoverability and connectability modes Enabled advertisement channels (0-3) Advertisement interval and data Scanning duty cycle Connection interval, timeout, slave latency Output power	Device role (coordinator/end-device) PAN channel and ID Poll rate (end-device) ACK request on/off Output power

Table I
Some of the tunable parameters in BLE and ZigBee stacks.

IV. BLE Energy Consumption Measurements and Modeling

A. Non-Connected States

We first study the energy consumption before the slave and master are connected. Figure 1 shows a power trace of the slave and master while advertizing and scanning, respectively. The slave was set to send advertisements with half a second intervals and the master was configured to scan using a 50% duty cycle. The two traces are not from the same discovery process and did not have exactly the same configurations which can be noticed from the differing connection event intervals.



(a) Slave advertizing (b) Master scanning

Figure 1. Power trace of advertisement/scanning and the connection events that follow.

With no specific advertising data embedded, the advertisement is 8B long. Each advertisement consumed roughly 0.1mJ, which gives us power consumption of about 0.2mW for the slave with this configuration. Advertisements can be sent also to several different channels. Sending to three channels instead of one during a single advertising event consumes about 0.08 mJ more almost doubling the energy consumption. However, using all three channels increases chances of faster discovery. The master was consuming about 33mW during the scanning phase when using 50% duty cycle.

The total energy consumption during this phase depends on the chosen parameter values. For the slave, the parameters of interest are the advertisement interval and the number of channels to send them. The scanning duty cycle of the master is an equally important parameter. The specification does not mandate how the master should scan the three different channels. The TI stack implementation scanned one channel during one cycle of duty and the next channel during the next cycle etc. The optimal parameter configuration depends on the usage scenario. For instance, imagine an IoT scenario where a number of sensors are deployed in surrounding environments and a smart phone is used to collect information from those sensors over BLE. Given that the phone is mobile, it might be more important to complete the discovery phase quickly in order to be able to harvest the surrounding information than to save energy by using a low scanning duty cycle. However, the best strategy also depends on mobility patterns, density of sensors, etc.

The duty cycle (d) corresponds to the probability that the master finds the slave during one of its advertisement interval (I_a). We can model the discovery time using binomial distribution in which case the expectation for the discovery time is I_a/d. Thus, the estimate for energy consumed by master is $E_m^{disc} = dP_{scan}I_a/d = P_{scan}I_a$, where P_{scan} is the power consumption while scanning. The energy consumption for the slave is $E_s^{disc} = E_{adv}/d$, where E_{adv} is the energy consumed by a single advertisement. Fastest discovery is therefore obtained when master scans all the time. Although the duty cycle does not have an influence on the overall energy consumed by the master when discovering slaves that are in vicinity, the energy consumed by a high duty cycle starts to count when the master is scanning even if no slaves are present. Since the master does not know and cannot influence the slave's advertisement interval and slave cannot change master's duty cycle, neither slave nor master can optimize their energy consumption by tweaking their own parameters.

However, considering the IoT scenario depicted above, a cooperative protocol could be developed. For example, several sensors within range of each other could exchange information about discovery process and when the master discovers one of them it would be informed of how to discover the others. Another approach could be to have the master learn and remember where sensors with particular configurations are located. An adaptive discovery protocol for classic Bluetooth is described in [12]. These kind of optimization mechanisms would be very useful because unfortunate choice of discovery parameter values can lead to long discovery times and a lot of energy spent in the worst case where the advertisement and scan cycles converge very slowly to a mutual point in time. Though, BLE specification defines that a pseudorandom part is to be included in the advertising interval to avoid situations where the advertising and scanning cycles are synchronized and never meet.

B. Connection State

In connected mode, the connection events happen periodically. Figure 2 shows a power trace of an example event captured at the slave. The figure shows the different phases during a connection event: wakeup and pre-processing, transmit, receive, and post-processing. In between subsequent transmission and reception a short delay called inter-frame space (IFS) is added according to the specification. The slight differences in the heights of peaks in the figure are due to the limited sampling rate of the power meter.

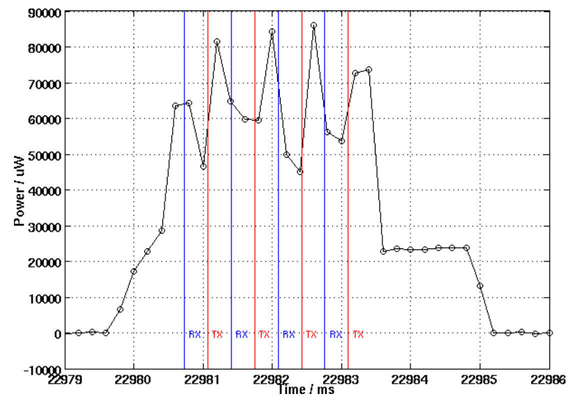


Figure 2. Power consumption of a single connection event at slave during which it sends 4 packets.

Phase	Power draw (V_{DD} = 3V)	Duration
1. wakeup & pre-processing	P_{wu} = 15mW	D_{wu} = 1ms
2. RX	P_{rx} = 66mW	D_{rx} = 8μs/B
3. IFS	P_{ifs} = 45mW	D_{ifs} = 150μs
4. TX	P_{tx} = 84mW	D_{tx} = 8μs/B
5. post-processing	P_{mcu} = 24mW	D_{mcu} = 1.4ms

Table II
CURRENT DRAW IN DIFFERENT PHASES OF A CONNECTION EVENT.

Table II lists the power draw and duration of each phase. The absolute values are naturally hardware dependent and the post-processing time depends on what kind of processing needs to be done which depends on the application. Note that radio is already off during that phase. The energy consumption is similar for slave and master except that the RX and TX durations are interchanged. From this information, we infer that the energy consumption during a connection event for slave (E_s^{ce}) and master (E_m^{ce}) consist of energy to wakeup (E_{wu}), transmit (E_{tx}), receive (E_{rx}), wait for required number of IFS (E_{ifs}), and do post-processing (E_{mcu}), as follows:

$$
\begin{aligned}
E_s^{ce} &= E_{wu} + E_{tx} + E_{rx} + E_{ifs} + E_{mcu} \\
&= P_{wu}D_{wu} + P_{tx}D_{tx}((n-1)(l_{hdr} + l_{pl}) + l_{last} + l_{hdr}) + \\
&\quad P_{rx}D_{rx}nl_{hdr} + P_{ifs}D_{ifs}(2n-1) + P_{mcu}D_{mcu} \\
E_m^{ce} &= P_{wu}D_{wu} + P_{rx}D_{rx}((n-1)(l_{hdr} + l_{pl}) + l_{last} + l_{hdr}) + \\
&\quad P_{tx}D_{tx}nl_{hdr} + P_{ifs}D_{ifs}(2n-1) + P_{mcu}D_{mcu}
\end{aligned}
$$

In the above equations, l_{hdr}, l_{pl}, and l_{last} denote length of BLE link layer frame headers and trailers, maximum payload that fits into BLE link layer frame, and the amount of data in the last frame payload, respectively, in bytes. Furthermore, n, which is the number of frames sent, can be computed as $n = \lceil (4+s)/l_{pl} \rceil$, where s is the total length of the application

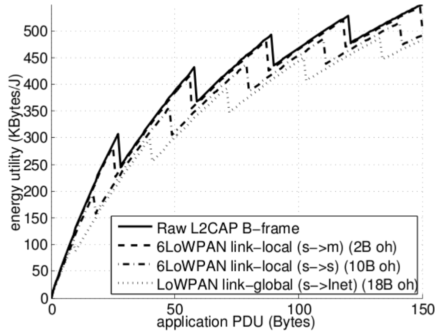


Figure 3. Energy utility of a single connection event at slave.

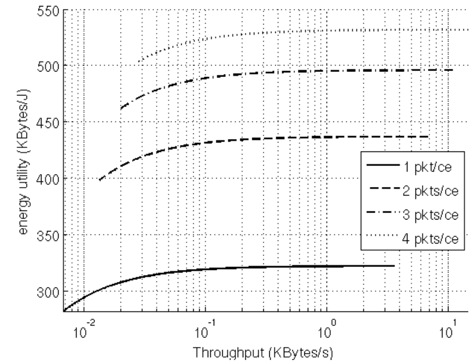


Figure 4. Energy utility vs. throughput when varying connection event size and connection interval (note the logarithmic scale).

message to send in bytes and 4 is the header length of L2CAP messages in bytes. The rest of the terms are listed in Table II.

Using the above described models, we compute the slave's energy utility, bytes sent per Joule of energy consumed, for different amounts of data sent during a single connection event (Figure 3). It has a step-wise nature because a maximum of 27B of payload data can be transmitted at the Link Layer which corresponds to 23B of application data when taking L2CAP header into account. We observe that the energy utility rapidly improves when incresing the amount of application data per connection event. The results for master look exactly the same except that the energy consumed is slightly less and, consequently, the energy utility is higher since reception draws less power than transmission.

In addition to the basic case of slave transmitting raw L2CAP frames, the figure shows several different curves which correspond to different scenarios. In BLE+6LoWPAN, the target is to examine the overhead of 6LoWPAN headers (DISPATCH+IPHC) in different scopes [5]. In the calculations, stateless compression is used and fields are elided if possible. This would result in optimal case in short-length scenarios where stateful compression is unsuitable. In link-local communications between master and slave (Slave <-> Master (link-local)), the IPv6 source and destination addresses can be fully elided. Traffic class and flow are also elided, which reduces the total overhead to only 2B. A connection between 2 slaves (Slave <- (Master) -> Slave (link-local)) must be routed via master. This requires sending the IID part (64bits) of destination IPv6 address inline from which master associates the BLE access address of the receiving slave. The master can now remove the destination IID and add the IID of the sender (also from association table), which makes the total overhead to 10B. Sending packets to global IPv6 addresses (Slave <- (Master) -> Internet) requires the full address carried inline, unless context is used, which increases the overhead to 18B. If slave can have multiple global IPv6 addresses, the IID part must also be carried inline (to/from master), making the total overhead 26B.

BLE specification allows to set the connection interval to values ranging from 7.25ms to 4s. In addition, the slave may skip some of the connection events so that maximum effective connection interval is 32s. Transmit window size, i.e. the time window during which connection event must happen, has a maximum duration of 10ms. According to our measurements, there should be enough time to transmit up to 12 full size frames (27B) during a 10ms connection event. However, the TI stack limits the maximum number of frames to four. These parameter limits together yield a maximum throughput of roughly 10KB/s. A given throughput can be achieved in several ways since there are two parameters to tune: the interval between connection events and the number of frames sent during a single event. We computed the energy utility for different throughput values and the results are plotted in Figure 4. It is easy to see that the energy utility is rather insensitive to changes in length of the connection interval, which stems from the fact that the BLE transciever draws only about 1μA of current when sleeping. However, the number of frames sent during a connection event has a big impact on the energy utility. Therefore, to maximize the energy utility for a given data rate, the slave should transmit as many frames within a single connection event as possible and use a longer connection interval. Such strategy is very useful when the data to be transmitted is not time critical.

In order to put the energy utility of BLE to perspective, we contrast our results with those reported in [13] for Wi-Fi. Wi-Fi energy utility is highly dependent on throughput even when operated with Power Saving Mode enabled. The authors show that when the throughput varies from 16 to 256 KBps, the energy utility ranges from roughly 20 to 240 KB/J which is clearly lower than that of BLE.

V. Comparative Measurements with 802.15.4

We contrast the energy efficiency of BLE by performing a similar measurement study with ZigBee. The energy consumption of a ZigBee end-device can be divided into 2 phases: network discovery and data transfers. In the discovery phase, the end-devices scan for network beacons sent by PAN coordinator and when found, joins the network. After this, the end-device can transmit data according to the 802.15.4 MAC protocol. The energy consumption during discovery phase is similar (slightly higher) to that of BLE except for one important difference: the scanning is done by the end device.

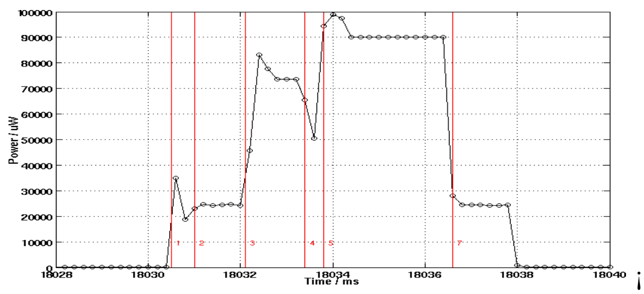


Figure 5. 802.15.4 end device power consumption when transmitting 70B.

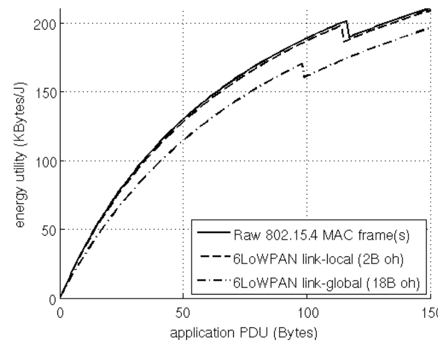


Figure 6. Energy utility of end device transmitting different amounts of data.

Figure 5 shows a power trace of an end device which transmits 70B. The different phases and their power draw and duration are listed in Table III. A notable difference is the channel access method which is contention based CSMA/CA in 802.15.4. We report a range for the duration of that phase in the table because it varies depending on the amount of contention and the random backoff values chosen in case the channel is found busy. No other competing devices were in the vicinity during our experiments. The pre-processing phase is around 1ms when sending raw 802.15.4 frames but increases to at least 3.5ms when sending ZigBee frames.

Phase	Power draw ($V_{DD}=3$V)	Duration
1. wakeup	$P_{wu}=20$mW	$D_{wu}=0.5$ms
2. pre-processing	$P_{mcu1}=24$mW	$D_{mcu1}=1/3.5$ms
3. CSMA/CA	$P_{ca}=72$mW	$D_{ca}=0.6-2$ms
4. RX-TX switch	$P_{rxtx}=54$mW	$D_{rxtx}=0.4$ms
5. TX	$P_{rx}=90$mW	$D_{rx}=32\mu$s/B
6. RX	$P_{tx}=72$mW	$D_{tx}=32\mu$s/B
7. post-processing	$P_{mcu2}=24$mA	$D_{mcu2}=1.4$ms

Table III
CURRENT DRAW IN DIFFERENT PHASES OF A TRANSMISSION EVENT.

Based on the measured values in Table III, we can derive an equation to compute the energy consumption in the same way we did for BLE in Section IV-B (omitted due to space constraints). Figure 6 shows the resulting energy utility. Comparing these curves to those of BLE in Figure 3, we see that the shapes are similar but the BLE energy utility is 2.5 times better. The reason is the four times higher over-the-air data rate of BLE and the CSMA/CA which somewhat prolongs the time that the 802.15.4 radio is powered on. As 802.15.4 supports mesh networking and IEEE addresses are carried within link-layer packets, the DISPATCH+IPHC overhead in all link-local communications is only 2 B. In link-global communications, the overhead is 18/26 B as in BLE.

Figure 7 visualizes the relationship between throughput and energy utility. CSMA/CA was assumed not to be done in between transmitting subsequent frames. We can observe that 300KB/J is a limit for the energy utility in this scenario. Comparison to the equivalent results on BLE plotted in Figure 4 tells us that the 802.15.4 energy utility corresponds roughly to that of BLE when transmitting a single full packet per connection event but is far inferior to that of BLE with multiple packets per connection event.

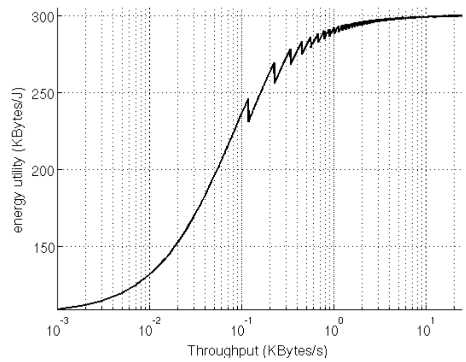


Figure 7. Energy utility vs. throughput for 802.15.4.

We restricted our attention to the energy consumption of an end-device and excluded the behavior of coordinator which in ZigBee usage scenarios is often a more powerful device and might even be directly attach to a power outlet. Therefore, the energy consumption is also rather asymmetric compared to BLE requiring the coordinator to listen whether end devices decide to transmit data, which may consume significantly more in order to let end devices sleep more.

Concerning our use case where the mobile device would continuously gather information from sensors in its vicinity, the mobile device would in fact need to act as a coordinator and, therefore, would spend more energy than the analysis of an end device shows above. Hence, BLE seems to be more suitable for that scenario.

VI. IMPACT OF INTERFERENCE

We studied also how interference influences the two technologies. We set up in close range a maximum rate UDP transmission over a given Wi-Fi (54Mbps) channel as a source of interference. BLE should use adaptive frequency hopping (AFH) similarly to classic Bluetooth. However, the TI stack used did not implement the adaptation, which means that it did not detect channels under interference and exclude them from the hopping sequence. Instead, all channels were used regardless of the interference. 802.15.4 uses CSMA/CA and, thus, senses the channel and backs off in case of interference.

From energy consumption perspective, the impact of interference is visible in different ways. BLE experiences packet loss because of the lack of AFH, while 802.15.4 transmitter needs to back off due to channel contention which increases the transmit time and causes extra energy to be spent for multiple clear channel analysis. The exact amount of extra energy spent depends on the number of retransmissions or clear channel analysis per packet.

Roughly 60% of packets were succesfully transmitted over BLE when the interferer was very close to the master which corresponds roughly to the fraction of BLE channels that are not covered by a single Wi-Fi channel. We did not see much sense in experimenting further with the frequency hopping since AFH was not implemented. Therefore, for the following experiments, we configured the slave to advertise on a single channel and set up the interferer on the same or close-by channel in the frequency spectrum. In addition, we varied the distance of the slave to the interferer. The results are shown in Figure 8. For a given distance between the slave and interferer, there are two stacked bars which correspond to two different Wi-Fi channels of the interferer: one completely overlapping the slave's advertising channel and another somewhat overlapping one. The results show that just 1.5 meters of distance is enough to avoid almost completely the negative impact of Wi-Fi interferer. Another interesting observation is that fraction of received packets actually grows when bringing the slave very close to the source of interference. The only logical explanation for this phenomenon is that in this case the Wi-Fi sender, which uses CSMA/CA, is occasionally able to sense that the channel is occupied and backs off as a result.

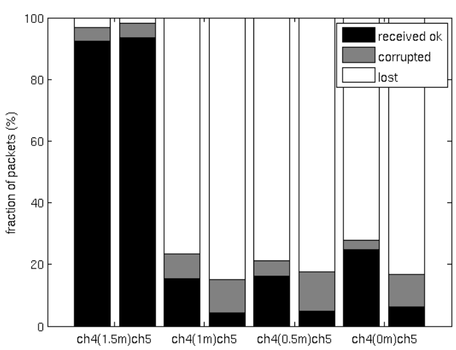


Figure 8. Impact of interference for BLE.

Similar setup was done for ZigBee end device. It was first allowed to join the network with interferer turned off. Afterwards, the end device started to broadcast packets with 70B MAC payload at 4 second interval and the interferer was started. In this experiment, the end device managed to succesfully transmit roughly 35% of packets on same the channel as the interferer. Varying the distance between 0m and 1.5m had little impact on the results. However, growing the distance to 1m and beyond increased the ratio of succesfully transmitted packets from 38% to almost 100% on the channel adjacent to the one with the interferer.

VII. Conclusion

Future IoT deployments are likely to use ultra-low power, short range communication technologies. In this paper, we measure and model the energy consumption of BLE and compare it to that of 802.15.4. Our results show that BLE indeed consumes extremely little energy and has a very attractive ratio of energy per bit transmitted. However, the stack that we experimented with had certain limitations. Indeed, the energy efficiency could be further improved by allowing more packets to be sent within a connection event and by implementing AFH to combat interference. In addition, the discovery energy could be reduced through design of cooperative mechanisms.

Acknowledgments

This work was supported by TEKES as part of the Future Internet program of TIVIT (Finnish Strategic Centre for Science, Technology and Innovation in the field of ICT) and by the Academy of Finland (grant number 253860).

References

[1] B. SIG, "Bluetooth specification version 4.0 [vol 0]," Available at http://www.bluetooth.org, June 2010.

[2] C. Sorrel, "Casio bluetooth low energy watch has two year battery life: http://www.wired.com/gadgetlab/2011/03/casio-bluetooth-low-energy-watch-has-two-year-battery-life/," Wired magazine, Mar. 2011.

[3] N. Kushalnagar, G. Montenegro, and C. Schumacher, "IPv6 over Low-Power Wireless Personal Area Networks (6LoWPANs): Overview, Assumptions, Problem Statement, and Goals," RFC 4919 (Informational), Internet Engineering Task Force, Aug. 2007. [Online]. Available: http://www.ietf.org/rfc/rfc4919.txt

[4] G. Montenegro, N. Kushalnagar, J. Hui, and D. Culler, "Transmission of IPv6 Packets over IEEE 802.15.4 Networks," RFC 4944 (Proposed Standard), Internet Engineering Task Force, Sept. 2007. [Online]. Available: http://www.ietf.org/rfc/rfc4944.txt

[5] J. Nieminen, B. Patil, T. Savolainen, M. Isomaki, Z. Shelby, and C. Gomez, "Transmission of ipv6 packets over bluetooth low energy," draft-ietf-6lowpan-btle-05, work in progress, Dec. 2011.

[6] M. Blom, M. Ekstrom, J. Castano, and M. Linden, "Bluetooth energy characteristics in wireless sensor networks," in *Wireless Pervasive Computing, 2008. ISWPC 2008. 3rd International Symposium on*, may 2008, pp. 198 –202.

[7] R. de Francisco, L. Huang, G. Dolmans, and H. de Groot, "Coexistence of zigbee wireless sensor networks and bluetooth inside a vehicle," in *Personal, Indoor and Mobile Radio Communications, 2009 IEEE 20th International Symposium on*, sept. 2009, pp. 2700 –2704.

[8] J.-S. Lee, Y.-W. Su, and C.-C. Shen, "A comparative study of wireless protocols: Bluetooth, uwb, zigbee, and wi-fi," in *Industrial Electronics Society, 2007. IECON 2007. 33rd Annual Conference of the IEEE*, nov. 2007, pp. 46 –51.

[9] S. Gollakota, F. Adib, D. Katabi, and S. Seshan, "Clearing the rf smog: making 802.11n robust to cross-technology interference," in *Proceedings of the ACM SIGCOMM 2011 conference on SIGCOMM*, ser. SIGCOMM '11. New York, NY, USA: ACM, 2011, pp. 170–181.

[10] R. Friedman, A. Kogan, and Y. Krivolapov, "On power and throughput tradeoffs of wifi and bluetooth in smartphones," in *INFOCOM, 2011 Proceedings IEEE*, april 2011, pp. 900 –908.

[11] "Monsoon: www.msoon.com."

[12] C. Drula, C. Amza, F. Rousseau, and A. Duda, "Adaptive energy conserving algorithms for neighbor discovery in opportunistic bluetooth networks," *Selected Areas in Communications, IEEE Journal on*, vol. 25, no. 1, pp. 96 –107, jan. 2007.

[13] Y. Xiao, P. Savolainen, A. Karppanen, M. Siekkinen, and A. Ylä-Jääski, "Practical power modeling of data transmission over 802.11g for wireless applications," in *e-Energy '10: Proceedings of the 1st International Conference on Energy-Efficient Computing and Networking*. New York, NY, USA: ACM, 2010, pp. 75–84.