

BLE and IEEE 802.15.4 in the IoT: Evaluation and Interoperability Considerations

PrithviRaj Narendra¹, Simon Duquennoy¹ and Thiemo Voigt^{1,2}

¹ SICS Swedish ICT, Sweden
prithvi@sics.se, simonduq@sics.se, thiemo@sics.se
² Uppsala University, Sweden

Abstract. As the Internet of Things is gaining momentum, low-power communication technologies proliferate. In this paper, we focus on Bluetooth Low Energy (BLE) and IEEE 802.15.4 (CSMA, Low-power listening, and TSCH), and advocate low-power IPv6 for interoperability between the two. We perform a thorough experimental comparison of their link-layer performance, both in idle radio environment and when facing heavy (controlled) external interference. Our results suggest that both technologies can achieve interesting and complementary latency-energy trade-offs. Based on our results, we discuss possible interoperability between BLE and IEEE 802.15.4 and present related open issues.

Key words: interoperability, iot, link-layer, ble, ieee802154, ipv6

1 Introduction

Over the past few years, Internet of Things (IoT) applications have flourished, where everyday objects are made ‘smart’ by including sensors, processing units and low-power wireless communication means. Current applications, ranging from e-health, entertainment to smart buildings and energy, however, work in isolation. To enable them to interoperate and reach the full potential of the IoT, a practical approach is to run a low-power IPv6 stack in these devices. With this approach, the devices can access the Internet and benefit from network-layer compatibility. This enables interaction between devices running various applications and relying on heterogeneous technologies.

In this paper, we focus on two of the currently dominating technologies: Bluetooth Low Energy (BLE) and IEEE 802.15.4¹. We review both technologies, evaluate their performance, and discuss their interoperability. An example application of BLE – 802.15.4 interoperability is a smart health/fitness tracking band (BLE) that can communicate via IPv6 to the home thermostat (802.15.4) to regulate the temperature/humidity based on multiple users’ pulse and body temperature. Because the application builds upon standards, it can be extended easily, for example, with a cloud server used to store/process sensor data and make more complex decisions based on user preferences etc. The devices can also be accessed directly by the user via a smartphone or other means.

¹ In this paper we use ‘IEEE 802.15.4’ and ‘802.15.4’ interchangeably.

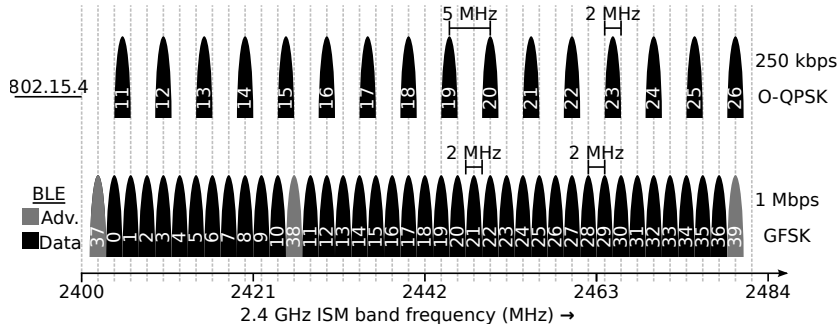


Fig. 1. The BLE and 802.15.4 channel maps share the 2.4 GHz ISM spectrum.

We port Contiki, a leading operating system for the IoT, to the nrf51822 System on Chip (SoC), a BLE platform. As our prototype does not support IPv6 yet, we focus our evaluation on the link-layer. We conduct an extensive experimental comparison of the BLE and IEEE 802.15.4 link layers (CSMA, Low-power listening and TSCH), and we challenge both technologies by exposing them to controlled external WiFi interference. We look at both request-response and bulk transmission schemes. Our focus is on reliability, energy consumption, latency and data rate. Overall, we find that both technologies are able to make different latency-energy trade-offs, which makes them suitable for a variety of applications. We believe there is room for both technologies in the IoT, which makes interoperability an even more important concern. We discuss practical issues and open research questions in making this interoperability happen.

The paper is organized as follows: §2 gives necessary background on BLE and 802.15.4. §3 describes the hardware platforms used for experimentation. §4 presents an experimental comparison of the BLE and 802.15.4 link layers. §5 discusses current trends and open issues in achieving full BLE-802.15.4 interoperability. We finally review related work in §6 and conclude in §7.

2 Background: BLE and 802.15.4

This section reviews BLE [5] and IEEE 802.15.4 [3] with a focus on the link layer, as this is the layer we evaluate experimentally. As illustrated in Figure 1, both technologies share the 2.4 GHz ISM spectrum, but have their own modulation scheme, bit rate, channel map and channel spacing, and upper layers.

2.1 Bluetooth Low Energy (BLE)

BLE employs frequency hopping over 37 channels for (bidirectional) communication and 3 for (unidirectional) advertising, with a bitrate of 1 Mbps. In Bluetooth 4.0, the link-layer MTU is 27 bytes², protected with a 24-bit CRC.

² This has been increased to 251 bytes in Bluetooth 4.2 [6].

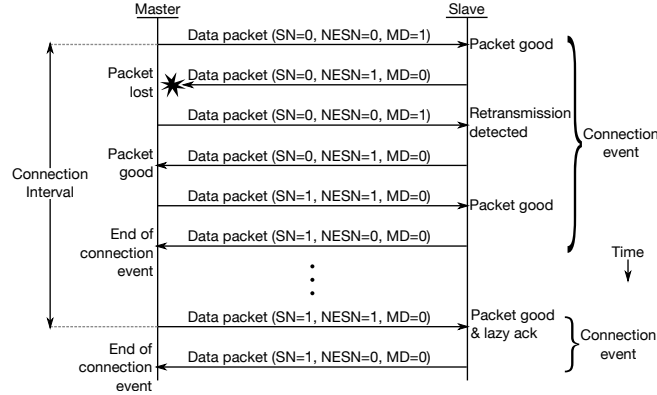


Fig. 2. Illustration of a connection event and connection interval (inspired from [14]). SN: Sequence Number. NESN: Next Expected Sequence Number. MD: More Data bit.

At the link layer, BLE forms a star network, with a master orchestrating bidirectional communication with one or several slaves. Nodes have different link-layer states including advertising, scanning and connection. An advertising device (typically a low-power node) periodically broadcasts packets over channels 37, 38, 39 which are spread over the 2.4 GHz ISM spectrum such that they do not overlap the most common WiFi channels 1, 6 and 11. A scanning device (e.g. a smartphone) listens on these advertising channels, waiting for advertisement packets. Upon receiving an advertisement, the scanning device may initiate a connection with a ‘connection request’ packet. The advertising device becomes the slave and the scanning device the master.

After the connection is established, communication takes place in ‘connection events’, which occur at a period called ‘connection interval’. At every connection event, the master transmits a packet to the slave, which may or may not respond. Several master-slave transmissions can occur in the same connection event, as driven by the ‘More Data’ bit (as illustrated by Figure 2). Sequence numbers and acknowledgments ensure reliable, in-order delivery. We review a number of relevant connection parameters (set by the master during establishment).

Connection Interval. Interval between connection events. Can range from 7.5 ms to 4 s, and can be changed even after a connection is established.

Slave Latency. Slave devices can save further energy by skipping connection events. The slave latency tells how many such events can be skipped in a row.

Channel Map. Specifies the channels used for channel hopping. Can be all 37 channels of BLE, or any subset.

Hop Increment. Every connection event BLE employs a frequency hopping mechanism to hop to a different frequency in the 2.4 GHz band. The hopping of the master is synchronized with the slave with the following equation: $f_{n+1} = (f_n + hop) \bmod 37$. Where f_{n+1} is the next channel, f_n the current channel used and hop is the hop increment connection parameter. The process is iterated until reaching a channel that is part of the connection’s channel map.

Above the link layer the BLE stack consists of the Logical Link Control and Adaptation Protocol (L2CAP), the Attribute Protocol (ATT), the Generic Attribute Profile (GATT), the Security Manager Protocol (SMP) and the Generic Access Profile (GAP) [12]. On top of them there are standardized profiles for various use cases that the application layer can use. Unlike 802.15.4, which is restricted to the physical and MAC layers, BLE is a full protocol stack. The adoption of 6LoWPAN, IPv6 and higher layers on top of BLE’s L2CAP layers has been proposed in a currently active IETF Internet-Draft [19]. Hence, the lower layers are important indicator of the performance of a protocol.

2.2 IEEE 802.15.4

802.15.4 features a total of 27 non-overlapping channels, including 16 in the 2.4 GHz and 11 in the sub-GHz bands. We focus on the 2.4 GHz band, which has a bitrate of 250 kbps. The maximum transmission unit is 127 bytes, and frames are protected with a 16-bit CRC.

Unlike BLE, IEEE 802.15.4 can form various topologies, such as star network, cluster tree or mesh. 802.15.4 is widely used as a research platform and features many different MAC layers. Some of the latter are part of the standard, others research prototypes. In this paper, we focus on the three MAC protocols we selected for evaluation: CSMA, ContikiMAC and TSCH.

802.15.4 + CSMA. CSMA (non-persistent CSMA-CA) is one of the MAC layers defined in 802.15.4-2011 [3]. We use the Contiki implementation, which departs slightly from the standard configuration (non-standard back-off procedure) but follows the same overall operation.

With CSMA, nodes keep their radio always on, operate on a single channel, and access the medium through contention. Link-layer acknowledgments are used to confirm reception and enable retransmissions at the sender. Using the ‘frame pending’ bit of the 802.15.4 header, nodes can tell the receiver that they have more packets in their send queue. With this feature, nodes can send many consecutive packets, which is useful in high bandwidth applications.

802.15.4 + ContikiMAC. ContikiMAC is Contiki’s default power-saving MAC layer [10]. It is an asynchronous low-power listening MAC, similar to BoX-MAC [18]. In ContikiMAC, nodes sleep most of the time, and wake up periodically (*e.g.*, every 125 ms) to sense the medium. Whenever detecting a signal, a node keeps the radio on for a few milliseconds, attempting a reception. At the sender side, the node transmits the data packet repeatedly until it receives an ACK from the receiver. ContikiMAC optimizes unicast using a so-called ‘phase-lock’ mechanism, where nodes stay loosely synchronized to minimize strobing time. ContikiMAC also supports the ‘frame pending’ bit for burst transmissions, but at a rate lower than with CSMA.

802.15.4 + TSCH. Time Slotted Channel Hopping (TSCH) is a MAC layer defined in the IEEE 802.15.4e [4] amendment. TSCH forms a mesh network where all nodes are globally synchronized. Communication is organized in slots, with a typical duration of 10 or 15 ms (we use 15 ms in this paper). Every slot

is long enough to accommodate the transmission of a frame and its acknowledgment. Slots are grouped in one or several slotframes, which repeat over time and form a schedule. At every slot, nodes know exactly whether they are supposed to sleep, transmit or receive. TSCH combats interference and link dynamics through channel hopping: at every slot, a channel is selected deterministically from a pseudo-random hopping sequence.

802.15.4: Upper Layers. There are many possible upper layers for 802.15.4, including ZigBee, 6LoWPAN, or custom network stacks. In interoperable low-power IP networks, the network stack is IPv6 along with the 6LoWPAN [17] adaptation layer. For mesh networking, a routing protocol such as RPL can be used, allowing low-power IPv6 nodes to be addressed over multiple hops. Upper layers typically employ UDP and CoAP. The IETF Working Group 6TiSCH [22] is currently proposing an architecture for 6LoWPAN/TSCH networks.

3 Platforms and Implementations

For this study, we work with two platforms: the nrf51822-based PCA10000 for BLE and the Tmote-Sky for 802.15.4. We ported the Contiki OS [1] to nrf51822 in order to use the same OS in both cases. Note that although Contiki supports a full-fledged low-power IPv6 stack, we run our experiments without it and focus on the performance at the link-layer, in part because we do not have yet all the mechanisms required to run IPv6 over BLE.

3.1 BLE: Contiki over nrf51822

As a BLE platform, we use the PCA1000 board, containing the nrf51822 SoC by Nordic Semiconductor. The nrf51822 SoC has a ARM Cortex M0 processor, BLE compliant radio and various on-chip peripherals. The BLE stack used is a precompiled and linked binary file called SoftDevice, S110 v6.0.0 for the slave role and S120 v1.0.0 for the master role. This stack provides an implementation of Bluetooth 4.0 specification. We perform all communication at the GATT layer because the binary from Nordic Semiconductor used in this project does not provide an API to access the lower layers in both master and slave devices. We enable the radio state change notification in the SoftDevice to enable logging the radio on-time, which we use for energy estimation.

3.2 IEEE 802.15.4: Contiki over Tmote-Sky

For 802.15.4, we use the Tmote-Sky platform, a widely used research platform that supports all Contiki features. The Tmote-Sky features an MSP430-F1611 MCU and CC2420 2.4 GHz transceiver. We use the Contiki energest module to log the radio on-time for energy estimation. Note that the CSMA experiments use the Contiki CSMA+NullRDC layers (*i.e.*, CSMA with no additional radio duty cycling protocol). As our experiments are without IPv6, we run Contiki with the Rime stack (a simple ad-hoc network layer).

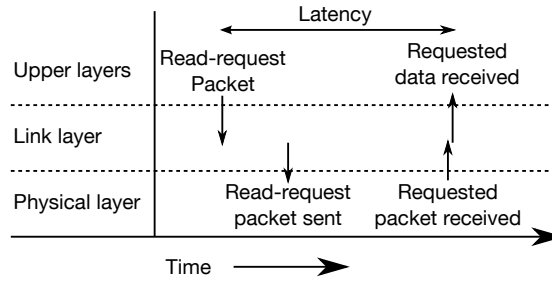


Fig. 3. Timeline of one request-response cycle. With both BLE and 802.15.4, we measure latency at the application layer.

4 Performance Comparison

This section compares the performance of BLE vs. 802.15.4 in terms of latency, data rate, reliability, and energy consumption. The goal is to gain a better understanding of the properties and trade-offs of each technology.

4.1 Setup

In order to assess both technologies in different scenarios, we have run two different series of experiments: Request-Response (RR) and Bulk-Transmission (BT). Both experiments involve communication between two nodes: a master and a slave for BLE, a coordinator and simple node for 802.15.4. In all cases we use a transmission power of 0 dBm (the default of both the nrf51822 and Tmote Sky’s cc2420 radios). We focus on the following metrics:

Latency. The latency is the time it takes to poll data in a request-response process. It is measured at the application layer as illustrated in Figure 3.

Data Rate. The data rate is measured at the link-layer, i.e. the amount of link-layer payload (excluding link-layer header) per unit of time.

Energy. As a metric of energy, we measure the portion of time spent with the radio turned on, referred to as Radio Duty Cycle (RDC). The RDC is a commonly used metric which does not allow to derive exact consumption numbers, but has the advantage of being completely platform-independent. RDC numbers can be fairly compared across protocols and platforms. We measure the RDC by logging, from the firmware, the total time spent with the radio turned on.

Reliability. We measure reliability at the link-layer as Packet Reception Ratio (PRR), i.e. the portion of packets received over sent. This does not account for retransmissions, i.e. the reception ratio at higher layers is typically higher than the PRR thanks to repeated attempts.

4.2 Request-Response (RR)

The RR experiment aims at determining the latency for reading data from another node. In the RR experiments we keep the nodes close to each other (10 cm)

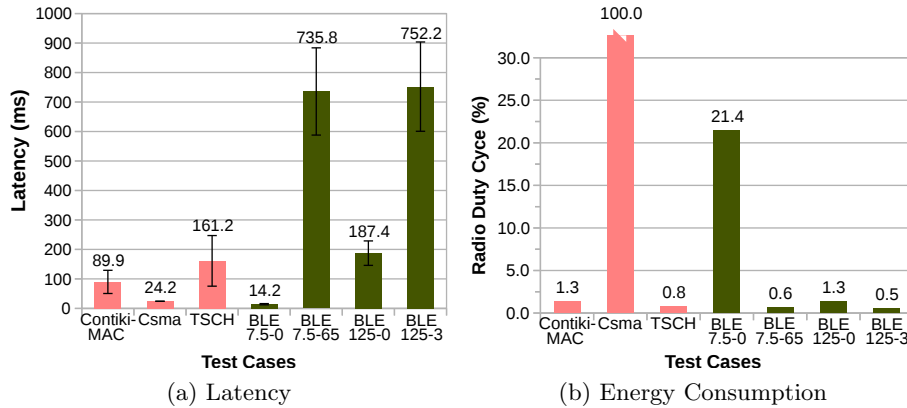


Fig. 4. Request-Response (RR) experiment. The various configurations of BLE and 802.15.4 offer complementary energy-latency trade-offs. ‘BLE-x-y’ denotes a connection interval ‘x’, and a slave latency ‘y’.

to reflect an almost error-free environment. We repeat the request-response cycle 1000 times for test case. The goal is to find the relation between energy consumption and latency based on different link layer configurations. Thus, the metrics evaluated in this experiment are **latency** and **energy**.

We run 802.15.4 with ContikiMAC, CSMA and TSCH. For TSCH, we use a slotframe of length 17 (repeats every $17 \times 15ms$), with two dedicated unicast slots: one for the request and one for the response. We run BLE with a connection interval of 7.5 ms (denoted BLE 7.5) or 125 ms (BLE 125). We also evaluate the BLE slave latency mechanism, which allows the slave to skip a given number of connection intervals. With a 7.5 ms interval we use a slave latency of 65 (denoted BLE 7.5-65), while with a 125 ms we set the slave latency to 3 (BLE 125-3). This results in a maximum sleep time of 495 ms resp. 500 ms.

Figure 4 shows our results in latency and energy. The lowest latency is achieved with BLE with a short connection interval (BLE 7.5), but this leads to a high duty cycle of above 20%. Next comes CSMA, with a duty cycle that is even higher (100%). TSCH, ContikiMAC and BLE 125 strike an interesting latency-energy balance, with a duty cycle between 0.6 and 1.3%, and a latency below 200 ms. Finally, BLE, using slave latency, achieves the lowest duty cycle, at the price of much a higher latency, in the 750 ms range. Overall, both 802.15.4 and BLE are able to achieve interesting latency-energy trade-offs, allowing to prioritize responsiveness and lifetime depending on application requirements.

4.3 Bulk Transmission (BT)

The BT experiment aims at measuring the maximum data rate of BLE and 802.15.4 CSMA and TSCH at the link layer with and without WiFi interference. In all cases, the link layer contains the maximum payload allowed, namely

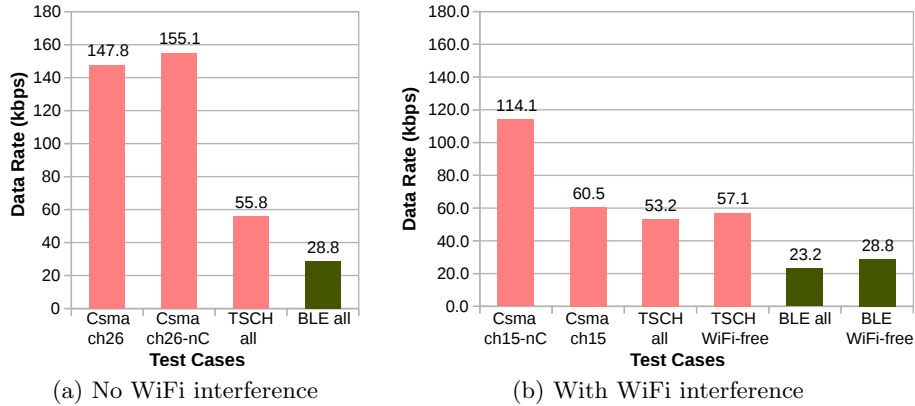


Fig. 5. Bulk Transmission (BT) experiment. 802.15.4 CSMA reaches higher throughput than BLE. BLE is less affected by interference thanks to channel hopping.

27 bytes for BLE and 110 bytes for 802.15.4. For 802.15.4 TSCH, we run a simple schedule with a single unicast slot from source to destination, repeating every 15 ms. The nodes are kept 1 m apart and in case of environment with interference, a WiFi router is placed 2.5 m away from the two nodes. Each run of the experiment lasts one minute in order to measure a stable mean data rate. The metrics measured in this experiment are **data rate**, **energy**, and **reliability**.

To introduce external WiFi interference, we use the tool `iperf` running on a ThinkPad T420 to transmit UDP traffic at 3 MB/s to a WiFi router (Netgear N300). The WiFi devices use the 802.11b/g/n standard over channel 5. We run TSCH and BLE either over all channels, or the subset of channels that are WiFi-free. For CSMA we select channel 15 (overlapping WiFi channel 5) or 26 (not affected by WiFi). With CSMA we rely on the burst transmission mode and disable link-layer acknowledgments to reach the highest throughput. We evaluate CSMA both with and without CCA (test cases without CCA are denoted ‘nC’).

Figure 5a, Figure 6a and Figure 7a show the results without interference. 802.15.4 + CSMA achieves the highest data rate with 155 kbps, but with a higher energy consumption as it keeps the radio turned on nearly all the time. BLE is the most reliable, with 99.9% PRR, which we attribute to channel hopping and short packet air time (high data rate, short frames).

We show the results with external WiFi interference in Figure 5b, Figure 6b and Figure 7b. Overall, we see the same trends as without WiFi, i.e. 802.15.4 has a higher data rate and consumes more energy. Disabling CCA helps CSMA to achieve a higher data rate, at the price of a PRR reduction from 80 to 70%. TSCH and BLE, through channel hopping, are less affected by interference than the single-channel CSMA. With the help of link-layer retransmissions, both BLE and TSCH achieved 100% reliability.

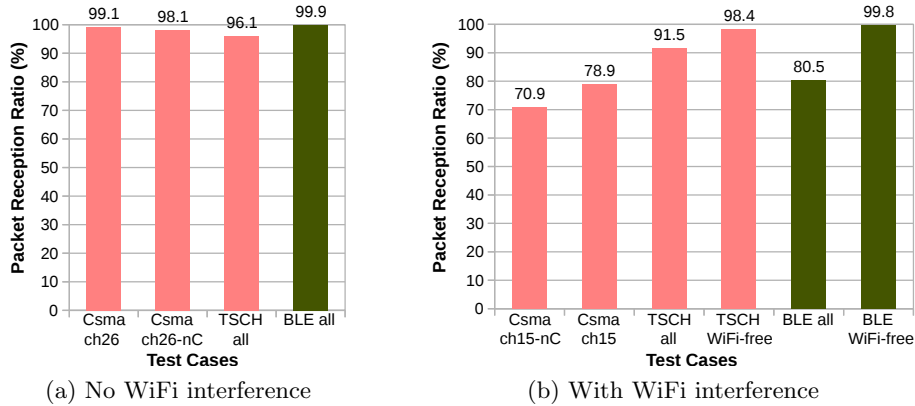


Fig. 6. BT experiment. Both protocols achieve similar link-layer reception ratios.

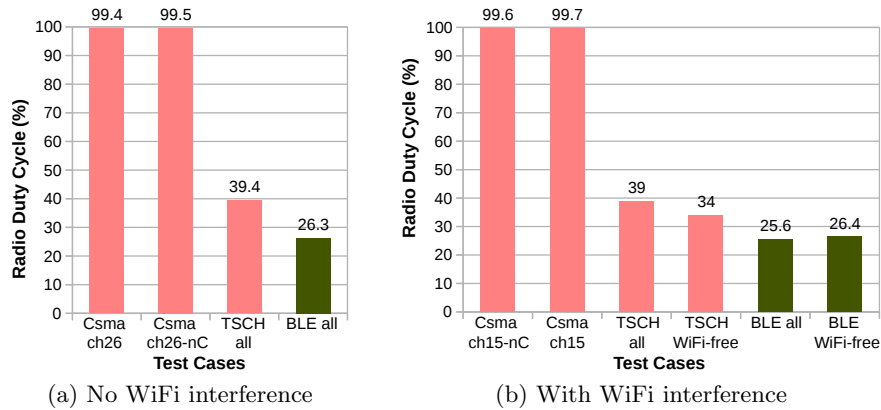


Fig. 7. BT experiment. BLE keeps saving energy during bulk transmission.

5 Discussion

5.1 IPv6 over BLE

The Internet Protocol Support Profile (IPSP) [7] defines how BLE devices running IPv6 discover each other and communicate at the L2CAP layer. With a focus on the layers above L2CAP, the IETF Working Group ‘6lo’ is currently defining how to transmit IPv6 datagrams over BLE (Bluetooth version 4.1 or greater) using adapted 6LoWPAN [19]. A number of mechanisms already defined in 6LoWPAN are applied to BLE, among which: address auto-configuration, neighbor discovery (star network case) and header compression. 6LoWPAN fragmentation is not used, as the L2CAP layer can take care of segmentation and reassembly in case the payload exceeds the link layer MTU.

In a 6LoWPAN over BLE network, the master is a 6LoWPAN Border Routers (6LBR), while the slaves are 6LoWPAN nodes (6LN). This provides Internet connectivity to the nodes, and enables 6LN-to-6LN communication via the 6LBR. Multicast is implemented as individual unicast by the 6LBR to target 6LNs. Even though the specification is still in a draft state, Nordic Semiconductor has already released an IoT SDK based on it [2].

5.2 Dual-protocol Devices

In the past years quite a few platforms have been released that can support both BLE and 802.15.4. As a module, Redpine Signal’s RS9113 also supports 2.4GHz/5GHz 802.11b/g/n WiFi in addition to a dual mode Bluetooth 4.0 and 802.15.4-based ZigBee. CC2650, a SoC released by TI in 2015 has a low power 2.4 GHz radio compatible with both 802.15.4 and BLE. This SoC consists of an ARM Cortex-M3 for main application, ARM Cortex-M0 for radio operation and a low power 16 bit processor for simple periodic routines. Similarly KW40Z, a new SoC based on ARM Cortex M0+ also has a low power radio compatible with these two protocols. With the radio capable of changing its operation on the fly between the two protocols and availability of low level radio commands these SoCs can be good candidates in both commercial and research scenarios.

Since we foresee that both the protocols will have their role to play in the IoT, these platforms would be ideal as a bridge between the two technologies at a hardware level, similar to IPv6 at the network level. Since both these protocols need border routers to connect to the Internet, these platforms can provide a solution for both, while also enabling direct inter-technology communication.

5.3 BLE – 802.15.4 Co-existence

In joint BLE – 802.15.4 networks, spectrum sharing might become challenging. How to best handle co-existence between cross-technology (*e.g.*, BLE and TSCH) channel hopping networks is an open issue. Assuming coordination is possible through *e.g.* a home gateway, one could easily allocate disjoint parts of the spectrum to each network. In uncoordinated cases, possible directions include detecting external interference and restricting BLE to the subset of channels that does not overlap 802.15.4, or devising mechanisms for adaptive blacklisting in low-power channel hopping networks.

6 Related Work

We review previous work on comparing BLE vs. 802.15.4, and on IPv6 for both. **BLE – 802.15.4 Comparison.** Several studies [11, 21, 9, 15] compared BLE with 802.15.4 in simulation or experimentally, and under various application assumptions and protocol configurations. Siekkinen *et al.* [21] focused on the relation between throughput and energy, and found that BLE had a constant

energy utility, while 802.15.4 got more energy-efficient as throughput increases. The paper also characterizes the overhead of both protocols at different payload sizes and runs experiment with external WiFi interference, but with BLE running in non-connected mode only. Other works [11, 9, 15] explored the energy-performance trade-off of both stacks for different configurations, and the found results aligned with ours in particular for BLE with 7.5 ms connection interval, and for 802.15.4 CSMA. To the best of our knowledge our paper is the first to explore 802.15.4 with low-power listening (ContikiMAC) in comparison with BLE at different connection intervals and under controlled interference.

Low-power IPv6 for BLE and 802.15.4. IPv6 in low-power wireless is gaining more and more academic and industrial attention. While 6LoWPAN is being adapted to BLE by the IETF [19], there are already several implementations of IPv6 or 6LoWPAN stacks for BLE [13, 20, 23]. For instance, the Contiki low-power IPv6 stack was ported to CC2541 for BLE communication with a Linux system as a master [13]. Extending BLE to multi-hop, which could be useful in IPv6 IoT networks, is also currently under consideration [16, 8]. As for 802.15.4, 6LoWPAN [17] was defined in 2007 and soon found its way to IoT OSes such as Contiki, TinyOS, OpenWSN, or RIOT. Several recent standardization efforts such as ZigBee IP, the Thread Group, and the IETF Working Groups Roll, 6lo, 6tisch (among others) demonstrate the rising interest in IPv6-based 802.15.4 and IPv6 in the IoT at large.

7 Conclusion

BLE and IEEE 802.15.4 have their own set of properties and performance, which makes them suitable to different applications. As a result, both technologies will likely continue to co-exist, making interoperability a central concern. This paper sheds the light on open interoperability issues, and presents experimental results on how each technology performs in a challenging environment.

References

1. Contiki: The Open Source Operating System for the Internet of Things.
2. NordicSemiconductor IPv6 over Bluetooth Smart Protocol Stack for nRF51 SoCs Enables Small, Low Cost, Ultra-low Power IoT Applications, 2014.
3. 802.15.4 Task Group. 802.15.4-2011: IEEE Standard for Local and metropolitan area networks—Part 15.4: Low-Rate Wireless Personal Area Networks (LR-WPANs), 6 Sep 2011.
4. 802.15.4e Task Group. 802.15.4e-2012: IEEE Standard for Local and metropolitan area networks—Part 15.4: Low-Rate Wireless Personal Area Networks (LR-WPANs) Amendment 1: MAC sublayer, 16 April 2012.
5. Bluetooth SIG. Bluetooth Core Specification Version 4.0. *Specification of the Bluetooth System*, 2010.
6. Bluetooth SIG. Bluetooth Core Specification Version 4.2. *Specification of the Bluetooth System*, 2014.

7. Bluetooth SIG. Bluetooth Internet Protocol Support Profile Specification Version 1.0.0, 2014.
8. J. Decuir. Bluetooth Smart Support for 6LoBTLE: Applications and connection questions. *IEEE Consumer Electronics Magazine*, 4(2):67–70, Apr. 2015.
9. A. Dementyev, S. Hodges, S. Taylor, and J. Smith. Power consumption analysis of Bluetooth Low Energy, ZigBee and ANT sensor nodes in a cyclic sleep scenario. In *IEEE International Wireless Symposium (IWS)*, pages 1–4. IEEE, Apr. 2013.
10. A. Dunkels. The ContikiMAC Radio Duty Cycling Protocol. Technical Report T2011:13, Swedish Institute of Computer Science, 2011.
11. C. Gomez, I. Demirkol, and J. Paradells. Modeling the Maximum Throughput of Bluetooth Low Energy in an Error-Prone Link. *IEEE Communications Letters*, 15(11):1187–1189, Nov. 2011.
12. C. Gomez, J. Oller, and J. Paradells. Overview and evaluation of bluetooth low energy: an emerging low-power wireless technology. *Sensors (Basel, Switzerland)*, 12(9):11734–53, Jan. 2012.
13. Haolin Wang, Minjun Xi, Jia Liu, and Canfeng Chen. Transmitting IPv6 packets over Bluetooth low energy based on BlueZ. In *Advanced Communication Technology (ICACT)*, pages 72–77, 2013.
14. R. Heydon. *Bluetooth Low Energy: The Developer’s Handbook*. Prentice Hall, 1 edition, 2012.
15. K. Mikhaylov, N. Plevritakis, and J. Tervonen. Performance Analysis and Comparison of Bluetooth Low Energy with IEEE 802.15.4 and SimpliCI. *Journal of Sensor and Actuator Networks*, 2(3):589–613, Aug. 2013.
16. K. Mikhaylov and J. Tervonen. Multihop data transfer service for Bluetooth Low Energy. In *2013 13th International Conference on ITS Telecommunications (ITST)*, pages 319–324. IEEE, Nov. 2013.
17. G. Montenegro et al. Transmission of IPv6 Packets over IEEE 802.15.4 Networks, Sept. 2007. RFC 4944.
18. D. Moss and P. Levis. BoX-MACs: Exploiting Physical and Link Layer Boundaries in Low-Power Networking. Technical Report SING-08-00, Stanford, 2008.
19. J. Nieminen, B. Patil, T. Savolainen, M. Isomaki, Z. Shelby, and C. Gomez. Transmission of IPv6 Packets over Bluetooth Low Energy [Working Draft], 2015.
20. G. M. Shrestha, J. Imtiaz, and J. Jasperneite. An optimized OPC UA transport profile to bringing Bluetooth Low Energy Device into IP networks. In *2013 IEEE 18th Conference on Emerging Technologies & Factory Automation (ETFA)*, pages 1–5. IEEE, Sept. 2013.
21. M. Siekkinen, M. Hienkari, J. K. Nurminen, and J. Nieminen. How low energy is bluetooth low energy? Comparative measurements with ZigBee/802.15.4. In *2012 IEEE Wireless Communications and Networking Conference Workshops (WCNCW)*, pages 232–237. IEEE, Apr. 2012.
22. X. Thubert (Ed.), T. Watteyne, R. Struik, and M. Richardson. An Architecture for IPv6 over the TSCH mode of IEEE 802.15.4e - draft-ietf-6tisch-architecture-06, Mar. 2015. IETF Draft.
23. D. Yoon, Wondeuk and Ha, Minkeun and Kwon, Kiwoong and Kim. 6Lo Bluetooth Low Energy for Patient-Centric Healthcare Service on the Internet of Things. In *Proceedings of the international conference on the Internet of Things*, 2014.