6LoWPAN Overview and Implementations

Zengxu Yang
Tufts Wireless Lab
Department of Electric and Computer Engineering
Tufts University
zengxu.yang@tufts.edu

C. Hwa Chang
Tufts Wireless Lab
Department of Electric and Computer Engineering
Tufts University
chorng.chang@tufts.edu

Abstract

This paper provides an overview of the 6LoWPAN standard. 6LoWPAN is an open standard developed by IETF. It is an IPv6 adaptation layer running on top of IEEE 802.15.4 standard. This paper reviews the history that led to the concept of Internet of Things (IoT) and how IoT evolved from simple, non-IP networks to 6LoWPAN based IP networks. In this paper, we argue why 6LoWPAN is an important building block for the future of IoT. We also describe the challenges and most important features of 6LoWPAN, including two application protocol standards CoAP and MQTT-SN that are used in 6LoWPAN. We also list some of the popular implementations of 6LoWPAN, including some Thread products based on 6LoWPAN.

1 Introduction

The Internet evolved from ARPANET in the 1970s and has been the most popular, globally interconnected network based on the TCP/IP suite. It has replaced several proprietary and incompatible network protocols such as Novell Internetwork Packet Exchange (IPX) and IBM Systems Network Architecture (SNA). Currently the IPv4 based Internet is the most important network and has revolutionized people's lives. Internet of Things (IoT) is a new technology that has wide potential applications and has the potential of changing our lives even more profoundly[9].

Constrained networks formed by connected constrained nodes are the backbone of IoT. Constrained networks and constrained nodes have unique challenges and the normal network protocols usually cannot be directly applied. Constrained nodes falls into different classes, as show in Table 1.

Most of the popular network protocols, like TCP/IP, need more resources than the constrained nodes can afford, especially for class 0 and class 1 devices. Either new protocols or adaptation of popular protocols are used. Class 0 devices are too limited even for specially designed IoT network protocols, while class 1 devices are capable of communicating with the external Internet using network protocols specially designed for IoT devices and some gateway/router devices. Even if class 2 devices are capable of running some normal network protocols like TCP and HTTP, they can still benefit from specially designed IoT network protocols. A certain type of constrained networks called Low Power Wireless Personal Area Networks (LoWPANs) has been widely used in a variety of applications in the field of Internet of Things, including wearable or implantable devices, urban monitoring, control of large buildings, and industrial control applications. The LoWPANs are not limited to personal usage and the "Personal" is just a vestige[13]. The most popular LoWPAN implementations are IEEE 802.15.4 defined by the IEEE 802.15 working group[26] and Bluetooth Low Energy (BLE)[1] developed by the Bluetooth Special Interest Group[24]. Several network protocols based on IEEE 802.15.4 like ZigBee[19] and 6LoWPAN[32] have been developed and standardized.

Table 1. Classes of constrained devices

Name	data size (e.g., RAM)	code size (e.g., Flash)
Class 0, C0	$\ll$10 KB	$\ll$100 KB
Class 1, C1	~ 10 KB	~ 100 KB
Class 2, C2	~ 50 KB	~ 250 KB

6LoWPAN brings IPv6 to the constrained IoT networks, reusing the time-proven TCP/IP protocols on IoT, making it a natural and future proof choice for IoT network protocols.

2 6LoWPAN

2.1 Why 6LoWPAN

There are two different categories of constrained IoT networks, non-IP or IP. For a non-IP IoT network, an application layer gateway is used to translate the communications between the IoT network and the Internet. One benefit of an IP IoT network is that only an IP layer gateway is needed, which is much more lightweight and more flexible. And because of the success of the IP, an IP IoT network can reuse applications by using existing, proven protocol implementations, thus greatly shortening development time. The disadvantage of IP IoT network is that TCP/IP is a fairly complex

[1] Marketed as Bluetooth Smart 2011-2016.

International Conference on Embedded Wireless Systems and Networks (EWSN) 2019
25–27 February, Beijing, China

ISBN: 978-0-9949886-3-8

protocol suite compared to simple IoT network protocols like ZigBee, so it is more difficult to implement on constrained nodes. Because of hardware advancement, the processing power has become more affordable even for constrained devices and networks now, making an IP IoT network more attractive[33]. 6LoWPAN is an IP IoT network and runs IPv6 over IEEE 802.15.4 networks. IPv6 was chosen to be the fabric of 6LoWPAN because the currently widely use IPv4 address space, which uses 32 bit addresses and has no more than $2^{32} = 4.3 \times 10^9$ addresses, is facing address exhaustion. IPv6 uses 128 bit addresses, which provides up to $2^{128} = 3.4 \times 10^{38}$ addresses. IPv6 can support the vast number of IoT devices and ubiquitous networks in the foreseeable future.

Non-IP IoT networks used to be more popular than IP IoT networks because of its simplicity, with some popular protocols such as ZigBee and BLE, but IP IoT networks are gaining momentum as hardware prices go down. IP IoT is so attractive that some previous non-IP IoT networks started to support IPv6 recently. For example, Bluetooth used to be a point-to-point communication protocol, but since Bluetooth 4.2, BLE started to add IPv6 support in its Internet Protocol Support Profile (IPSP)[41]. RFC 7668 added IPv6 over BLE, but it only supported star network topology[38]. A new IETF draft plans to add IPv6 mesh network support to BLE, so it is expected that BLE may fully support 6LoWPAN in the future[23]. ZigBee IP also incorporates IP into its ZigBee networks but it is not widely used[22].

2.2 Challenges of 6LoWPAN

Because 6LoWPAN runs IPv6 on constrained nodes, with constrained resources like energy, memory, and processing power, there are several challenges present in 6LoWPAN design and implementations. Some of the most important challenges are:

Header overhead The link layer of 6LoWPAN is IEEE 802.15.4, which has an MTU of only 127 bytes, while the IPv6 MTU is at least 1280 bytes with a header length of 40 bytes. This means that directly transmitting standard IPv6 packets over IEEE 802.15.4 is inefficient because of high header/payload ratio and frequent fragmentations and defragmentations[36].

Neighbor discovery IPv6 uses Neighbor Discovery Protocol (NDP) to configure itself statelessly by combining the network prefix information from Router Advertisement messages and host ID from its link layer address, forming a 128 bit address[37]. In 6LoWPAN, NDP should also be used because it greatly simplifies the task of assigning IP addresses to a large number of devices. Yet it must be revised to accommodate the constrained networks.

Lossy networks The wireless links in constrained networks are usually not reliable because of mobility, interference, etc. We call such networks lossy networks. Lossy networks present a challenge to routing protocol designs because most of the routing protocols require a relatively stable, slowly changing network topology.

Security Security has always been an important issue in computer networks. The standard TCP/IP networks emphasize end-to-end security, like the widely used Internet Protocol Security (IPSec) on the network layer or Transport Layer Security (TLS) on the application layer[43, 45, 16]. Those standard security measures usually require more resources than can be provided by the constrained IoT devices.

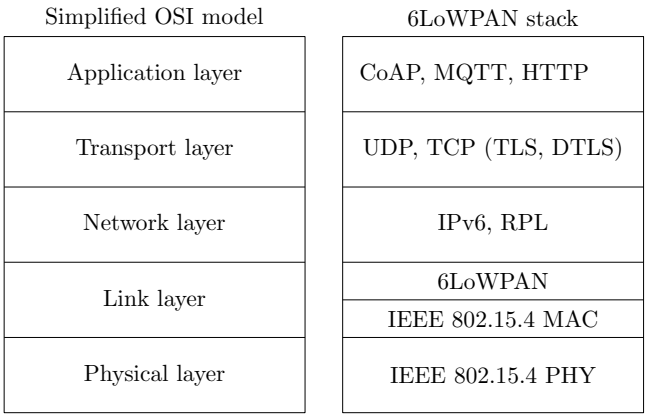

Figure 1. 6LoWPAN stack overview

Applications One of the most mature and popular application layer protocols is Hypertext Transfer Protocol (HTTP), which is based on TCP and provides Web services[20]. The Representational State Transfer (REST) architecture based on HTTP has been very popular in web services because it minimizes latency and network communication while at the same time maximizing the independence and scalability of component implementations[21]. The vast amount of RESTful web services based on HTTP makes it a very attractive protocol for 6LoWPAN applications. Nonetheless the regular HTTP and TCP are not suitable for constrained networks that 6LoWPAN runs on because of their complexity and the amount of resources required.

2.3 Main Features of 6LoWPAN

6LoWPAN, an IoT IPv6 adaptation layer, runs on top of IEEE 802.15.4, as shown in Figure 1.

IEEE 802.15.4 is a low power, low speed wireless personal area network (WPAN) standard that uses CSMA/CA and has a typical configuraion with a range of 10 m - 100 m and raw data rate of 2 - 250 kbit/s on the 2.4 GHz ISM band[26]. IEEE 802.15.4 can also operate on the 900 MHz (sub-G) band, with a lower data rate and a longer range up to a few kilometers. The range and data rate depend on the frequency band, environment, and hardware used[4, 5, 6, 7].

Header compression To efficiently transmit IPv6 packets over IEEE 802.15.4 links, we need to add an adaptation layer using header compression, which compresses the IPv6 header size from 40 bytes down to 7 bytes. IPv6 Extension headers are compressed as well. Higher layer headers have also been compressed, like UDP headers[29].

Routing Most of the routing protocols used on the regular IPv6 networks are not suitable for the constrained net-

works that 6LoWPAN runs on, so a new routing protocol called IPv6 Routing Protocol for Low Power and Lossy Networks (RPL) has been developed to be used by 6LoWPAN. Because a lossy network does not have a predefined topology, RPL organizes a topology as a Directed Acyclic Graph (DAG) partitioned into one or more Destination Oriented DAGs (DODAGs)[47]. RPL is supposed to be used in networks with up to thousands of nodes, where the majority of the nodes have very constrained resources, where the network to a large degree is "managed" by a (single or few) central "supernodes" (for example, the 6LoWPAN border routers)[15].

Security As IEEE 802.15.4 is a wireless physical and link layer protocol, like IEEE 802.11, a strong link layer security protocol is desirable. IEEE 802.15.4 uses Advanced Encryption Standard (AES) in the Counter with CBC-MAC mode[46]. The problem with link layer security is that it only covers the link segment that uses it, but not end-to-end security. To ensure end-to-end security, just like the Internet, we need to add security measures on higher layers, mainly on the network layer and the application layer. On the network layer, an adapted IPSec protocol has been proposed to support 6LoWPAN, which promises to support end-to-end security[39, 40]. On the application layer, Datagram Transport Layer Security (DTLS) is a main candidate. Unlike TLS that works on TCP, DTLS works on UDP, which is more suitable for constrained networks like 6LoWPAN[42].

Application protocols There are several application protocols available to 6LoWPAN, with the most popular ones being Constrained Application Protocol (CoAP) and Message Queuing Telemetry Transport-Sensor Network (MQTT-SN)[2] .

Due to the huge popularity of web services, CoAP was developed as a lightweight HTTP on 6LoWPAN. To work on constrained devices, CoAP uses UDP with its own message based retransmission mechanism instead of TCP. UDP is much simpler than TCP, has a smaller header size, and requires much less resources. CoAP keeps REST architecture yet requires much less resources, which enables Web developers to easily write programs for IoT. By using a proxy device, it is easy to translate between CoAP and HTTP and no significant web application redesign is needed. CoAP also uses options like block transfer, observations, and Web discover to further optimize for constrained networks. CoAP also works with security measures like DTLS, just like HTTP works with TLS. A secure CoAP with compressed DTLS has also been proposed[12, 42].

Another popular application protocol for 6LoWPAN is Message Queuing Telemetry Transport (MQTT)[11]. A special version of MQTT called MQTT-SN has been developed to be used on constrained IoT networks like 6LoWPAN. MQTT-SN can use UDP instead of TCP[30, 44, 25]. Unlike CoAP, MQTT is a lightweight publisher-subscriber protocol. A publisher-subscriber system decouples producers and consumers, but a broker is needed.

[2]MQTT-SN used to be called MQTT-S.

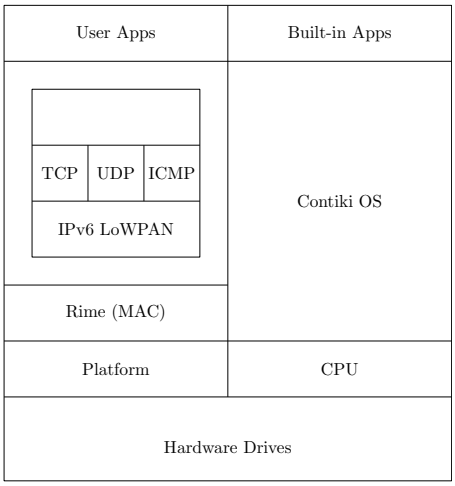

Figure 2. Contiki architecture

3 6LoWPAN Implementations

There are numerous open source and commercial implementations of 6LoWPAN. The most popular ones are: Contiki OS, TinyOS, Linux, and Openthread. They have been ported to a wide range of hardware platforms and architectures. Here we list some of the recent 6LoWPAN implementations.

3.1 Contiki

Contiki is a BSD licensed open source, lightweight operating system that implements 6LoWPAN using uIP on tiny, severely resource constrained nodes with IEEE 802.15.4 wireless communication capabilities, with memory as low as around 20 KB RAM and 100 KB ROM. It is implemented in the C language and has been ported to a wide range of devices on a variety of architectures, like chips based on the Atmel AVR, MSP430, and ARM Cortex-M3 architectures. Contiki supports dynamic loading and replacement of individual programs and services. It is built around an event-driven kernel but provides optional preemptive multithreading that can be applied to individual processes[18]. The architecture of Contiki is shown in Figure 2. By using a duty cycling protocol called ContikiMAC, Contiki achieves ultra low power consumption by turning off the wireless transceivers 99% of the time[17]. Contiki includes a network simulator called Cooja.

3.2 TinyOS

TinyOS is a BSD licensed open source operating system. It runs on low power, constrained devices that have around 16 KB of memory. The core TinyOS code size is just about 400 bytes[35]. TinyOS applications are written using a language called nesC, which is a dialect of C with features to reduce RAM and code size, enable significant optimizations, and help prevent low-level bugs like race conditions[34]. TinyOS has a component based programming model. Each component is an independent computational entity that exposes one or more interfaces. Com-

ponents have three computational abstractions: *commands*, *events*, and *tasks*. TinyOS 2.X supports 6LoWPAN by the Berkeley Low-power IP stack (BLIP) and TinyRPL module[28, 31].

3.3 RIOT

RIOT is a relatively new, LGPLv2 licensed open source operating system that is specifically designed for IoT. RIOT implements a microkernel architecture and allows for standard C and C++ programming, provides multi-threading as well as real-time capabilities, and needs only a minimum of 1.5 KB of RAM[10]. RIOT is a very versatile OS. When the hardware is capable, it competes with Linux; when running on constrained devices, it competes with Contiki, TinyOS, FreeRTOS, etc. Currently, RIOT supports basic network protocols including 6LoWPAN, RPL, IPv6, TCP, UDP, CoAP, and provides CCN-lite to experiment with content-centric networking[27].

3.4 OpenWSN

OpenWSN is an open source IoT operating system created by UC Berkeley. The goal of the OpenWSN project is to provide open-source implementations of a complete protocol stack based on Internet of Things standards on a variety of software and hardware platforms. OpenWSN supports 6LoWPAN and CoAP. It runs on a variety of motes, including OpenMote[14].

3.5 Zephyr

Zephyr is a new IoT operating system originally developed by WindRiver System and later became a project of the Linux Foundation. Zephyr is open source with an Apache 2.0 license. The Zephyr kernel supports multiple architectures, including ARM Cortex-M, Intel x86, ARC, NIOS II, Tensilica Xtensa and RISC-V 32. Zephyr supports multi-threading and includes POSIX pthreads compatible API support. Zephyr supports 6LoWPAN with its IP networking stack[3].

3.6 Linux

Although Linux is not for the severely constrained devices, adding native 6LoWPAN support in the Linux kernel makes inter-operating between Linux devices and constrained IoT devices easy. Sometimes a more powerful nodes that can directly talk in 6LoWPAN are desirable, for example border routers, brokers, and other computation-intensive nodes. Currently the native support of 6LoWPAN in the Linux kernel is still a work-in-progress. More information can be obtained at `http://wpan.cakelab.org/`.

3.7 Thread

Thread is a royalty free IoT specification based on 6LoWPAN developed by Google, Samsung, and some other companies. According to the Thread Specification, the Thread stack is an open standard for reliable, cost-effective, low power, wireless device-to-device communication. It is designed specifically for Connected Home applications where IP-based networking is desired[8]. Thread's primary features include:

Simplicity — Simple installation, start up, and operation.

Security — All devices in a Thread network are authenticated and all communications are encrypted.

Reliability — Self-healing mesh networking, with no single point of failure, and spread-spectrum techniques to provide immunity to interference.

Efficiency — Low-power Thread devices can sleep and operate on battery power for years Scalability — Thread networks can scale up to hundreds of devices.

Openthread released by Nest is an open source implementation of the Thread and implements all features defined in the Thread 1.1.1 Specification[2].

3.8 Mbed OS

Mbed is a platform and operating system for internet-connected devices based on 32-bit ARM Cortex-M microcontrollers developed by ARM and its partners. Mbed OS supports 6LoWPAN natively[1]. It is now a Thread Certified Component, which means it supports the Thread specification.

4 Conclusion

Internet of Things is the future and 6LoWPAN, an open and IPv6 based IoT network standard, is gaining popularity among IoT developers and manufacturers. Traditional non-IP IoT technologies like BLE and ZigBee are transitioning to be IPv6 based. Google, Samsung, and other companies are developing their Thread IoT products, which is based on 6LoWPAN. 6LoWPAN is being actively researched and newer IoT operating systems use their support of 6LoWPAN as a selling point. Nonetheless, challenges still exist for 6LoWPAN. End-to-end security is still under development, alternative link layers other than IEEE 802.15.4 is still in its infancy. We believe that for IoT researchers, studying and working on improving 6LoWPAN is a worthwhile investment. We believe that 6LoWPAN will be an important engine for the Internet of Things, just like IP is for the Internet.

5 References

[1] Mbed OS | Mbed. www.mbed.com/en.

[2] OpenThread. https://openthread.io/.

[3] The Zephyr™ Project. https://www.zephyrproject.org/.

[4] IEEE Standard for Information Technology - Telecommunications and Information Exchange Between Systems - Local and Metropolitan Area Networks Specific Requirements Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (LR-WPANs). *IEEE Std 802.15.4-2003*, pages 1–670, 2003.

[5] IEEE Standard for Information technology– Local and metropolitan area networks– Specific requirements– Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low Rate Wireless Personal Area Networks (WPANs). *IEEE Std 802.15.4-2006 (Revision of IEEE Std 802.15.4-2003)*, pages 1–320, Sept. 2006.

[6] IEEE Standard for Local and metropolitan area networks–Part 15.4: Low-Rate Wireless Personal Area Networks (LR-WPANs). *IEEE Std 802.15.4-2011 (Revision of IEEE Std 802.15.4-2006)*, pages 1–314, Sept. 2011.

[7] IEEE Standard for Low-Rate Wireless Networks. *IEEE Std 802.15.4-2015 (Revision of IEEE Std 802.15.4-2011)*, pages 1–709, Apr. 2016.

[8] Thread 1.1.1 Specification. Technical report, Thread Group, Feb. 2017.

[9] L. Atzori, A. Iera, and G. Morabito. The Internet of Things: A survey. *Computer Networks*, 54(15):2787–2805, Oct. 2010.

[10] E. Baccelli, O. Hahm, M. Gunes, M. Wahlisch, and T. Schmidt. RIOT OS: Towards an OS for the Internet of Things. In *2013 IEEE Con-*

ference on Computer Communications Workshops (INFOCOM WKSHPS), pages 79–80, Turin, Apr. 2013. IEEE.

[11] A. Banks and R. Gupta. MQTT Version 3.1. 1. *OASIS standard*, 29, 2014.

[12] C. Bormann, A. P. Castellani, and Z. Shelby. CoAP: An Application Protocol for Billions of Tiny Internet Nodes. *IEEE Internet Computing*, 16(2):62–67, Mar. 2012.

[13] C. Bormann, M. Ersue, and A. Keranen. Terminology for Constrained-Node Networks. Technical Report RFC7228, RFC Editor, May 2014.

[14] T. Chang, P. Tuset-Peiro, X. Vilajosana, and T. Watteyne. OpenWSN & OpenMote: Demo'ing a Complete Ecosystem for the Industrial Internet of Things. In *2016 13th Annual IEEE International Conference on Sensing, Communication, and Networking (SECON)*, pages 1–3, London, United Kingdom, June 2016. IEEE.

[15] T. Clausen, U. Herberg, and M. Philipp. A critical evaluation of the IPv6 Routing Protocol for Low Power and Lossy Networks (RPL). In *2011 IEEE 7th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*, pages 365–372, Shanghai, China, Oct. 2011. IEEE.

[16] T. Dierks and E. Rescorla. The Transport Layer Security (TLS) Protocol Version 1.2. 2008.

[17] A. Dunkels. The ContikiMAC Radio Duty Cycling Protocol. page 11.

[18] A. Dunkels, B. Gronvall, and T. Voigt. Contiki - a lightweight and flexible operating system for tiny networked sensors. In *29th Annual IEEE International Conference on Local Computer Networks*, pages 455–462, Tampa, FL, USA, 2004. IEEE (Comput. Soc.).

[19] S. C. Ergen. ZigBee/IEEE 802.15.4 Summary. page 37.

[20] R. Fielding, J. Gettys, J. Mogul, H. Frystyk, L. Masinter, P. Leach, and T. Berners-Lee. Hypertext Transfer Protocol – HTTP/1.1. 1999.

[21] R. T. Fielding. Architectural Styles and the Design of Network-based Software Architectures. page 180, 2000.

[22] M. Franceschinis, C. Pastrone, M. A. Spirito, and C. Borean. On the performance of ZigBee Pro and ZigBee IP in IEEE 802.15.4 networks. In *2013 IEEE 9th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*, pages 83–88, Lyon, France, Oct. 2013. IEEE.

[23] C. Gomez, S. M. Darroudi, T. Savolainen, and M. Spoerk. IPv6 Mesh over BLUETOOTH(R) Low Energy using IPSP. Internet-Draft draft-ietf-6lo-blemesh-03, Internet Engineering Task Force, July 2018.

[24] C. Gomez, J. Oller, and J. Paradells. Overview and Evaluation of Bluetooth Low Energy: An Emerging Low-Power Wireless Technology. *Sensors*, 12(9):11734–11753, Sept. 2012.

[25] K. Govindan and A. P. Azad. End-to-end service assurance in IoT MQTT-SN. In *2015 12th Annual IEEE Consumer Communications and Networking Conference (CCNC)*, pages 290–296, Las Vegas, NV, USA, Jan. 2015. IEEE.

[26] J. A. Gutierrez, M. Naeve, E. Callaway, M. Bourgeois, V. Mitter, and B. Heile. IEEE 802.15. 4: a developing standard for low-power low-cost wireless personal area networks. *IEEE network*, 15(5):12–19, 2001.

[27] O. Hahm, E. Baccelli, H. Petersen, M. Wahlisch, and T. C. Schmidt. Demonstration abstract: Simply RIOT — Teaching and experimental research in the Internet of Things. In *IPSN-14 Proceedings of the 13th International Symposium on Information Processing in Sensor Networks*, pages 329–330, Berlin, Apr. 2014. IEEE.

[28] M. Harvan. Connecting Wireless Sensor Networks to the Internet – a 6lowpan Implementation for TinyOS 2.0. page 69.

[29] J. Hui and P. Thubert. Compression Format for IPv6 Datagrams over IEEE 802.15.4-Based Networks. Technical Report RFC6282, RFC Editor, Sept. 2011.

[30] U. Hunkeler, H. L. Truong, and A. Stanford-Clark. MQTT-S – A publish/subscribe protocol for Wireless Sensor Networks. In *2008 3rd International Conference on Communication Systems Software and Middleware and Workshops (COMSWARE '08)*, pages 791–798, Bangalore, India, Jan. 2008. IEEE.

[31] J. Ko and O. Gnawali. Evaluating the Performance of RPL and 6lowpan in TinyOS. page 6.

[32] N. Kushalnagar, G. Montenegro, and C. Schumacher. IPv6 over Low-Power Wireless Personal Area Networks (6lowpans): Overview, Assumptions, Problem Statement, and Goals. Technical Report RFC4919, RFC Editor, Aug. 2007.

[33] N. Lethaby. Wireless connectivity for the Internet of Things: One size does not fit all. page 16, 2017.

[34] P. Levis and D. Gay. *TinyOS Programming*. Cambridge University Press, Cambridge, 2009.

[35] P. Levis, S. Madden, J. Polastre, R. Szewczyk, K. Whitehouse, A. Woo, D. Gay, J. Hill, M. Welsh, E. Brewer, and D. Culler. TinyOS: An Operating System for Sensor Networks. In W. Weber, J. M. Rabaey, and E. Aarts, editors, *Ambient Intelligence*, pages 115–148. Springer-Verlag, Berlin/Heidelberg, 2005.

[36] G. Montenegro, N. Kushalnagar, J. Hui, and D. Culler. Transmission of IPv6 Packets over IEEE 802.15.4 Networks. Technical Report RFC4944, RFC Editor, 2007.

[37] T. Narten, E. Nordmark, W. Simpson, and H. Soliman. Neighbor Discovery for IP version 6 (IPv6). 2007.

[38] J. Nieminen, T. Savolainen, M. Isomaki, B. Patil, Z. Shelby, and C. Gomez. IPv6 over BLUETOOTH(R) Low Energy. Technical Report RFC7668, RFC Editor, Oct. 2015.

[39] S. Raza, T. Chung, S. Duquennoy, D. Yazar, T. Voigt, and U. Roedig. Securing internet of things with lightweight ipsec. Mar. 2011.

[40] S. Raza, S. Duquennoy, J. Höglund, U. Roedig, and T. Voigt. Secure communication for the Internet of Things-a comparison of link-layer security and IPsec for 6lowpan: Secure communication for the Internet of Things. *Security and Communication Networks*, 7(12):2654–2668, Dec. 2014.

[41] S. Raza, P. Misra, Z. He, and T. Voigt. Bluetooth smart: An enabling technology for the Internet of Things. In *2015 IEEE 11th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*, pages 155–162, Abu Dhabi, United Arab Emirates, Oct. 2015. IEEE.

[42] S. Raza, H. Shafagh, K. Hewage, R. Hummen, and T. Voigt. Lithe: Lightweight Secure CoAP for the Internet of Things. *IEEE Sensors Journal*, 13, Oct. 2013.

[43] J. H. Saltzer, D. P. Reed, and D. D. Clark. End-to-end Arguments in System Design. *ACM Trans. Comput. Syst.*, 2(4):277–288, Nov. 1984.

[44] A. Stanford-Clark and H. L. Truong. MQTT For Sensor Networks (MQTT-SN) Protocol Specification. page 28, 1999.

[45] R. Thayer, N. Doraswamy, and R. Glenn. IP Security Document Roadmap. Technical Report RFC2411, RFC Editor, Nov. 1998.

[46] D. Whiting, R. Housley, and N. Ferguson. Counter with CBC-MAC (CCM). Technical Report RFC3610, RFC Editor, Sept. 2003.

[47] T. Winter, P. Thubert, A. Brandt, J. Hui, R. Kelsey, P. Levis, K. Pister, R. Struik, JP. Vasseur, and R. Alexander. RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks. Technical Report RFC6550, RFC Editor, Mar. 2012.