Teaching Communication Technologies and Standards for the Industrial IoT? Use 6TiSCH!

Thomas Watteyne, Pere Tuset-Peiró, Xavier Vilajosana
Sofie Pollin, Bhaskar Krishnamachari

***Abstract*—The IETF 6TiSCH stack encompasses IEEE802.15.4 TSCH, IETF 6LoWPAN, RPL and CoAP. It is one of the key standards-based technology to enable industrial process monitoring and control, and unleash the Industrial Internet of Things (IIoT). The 6TiSCH stack is also a valuable asset for educational purposes, as it integrates an Internet-enabled IPv6-based upper-stack with state-of-the-art low-power wireless mesh communication technologies. Teaching with 6TiSCH empowers students with a valuable set of competences, including topics related to computer networking (Medium Access Control operation, IPv6 networking), embedded systems (process scheduling, concurrency) and wireless communications (multi-path propagation, interference effects), as well as application requirements for the IIoT. This article discusses how the 6TiSCH stack can be incorporated into existing and new curricula to teach the next generation of Electrical Engineering and Computer Science professionals designing and deploying such networks. It also gives a comprehensive overview of the 6TiSCH stack and of the tools that exist to support a course based on it.**

***Index Terms*—IEEE802.15.4 TSCH, IETF 6TiSCH, 6LoWPAN, RPL, CoAP.**

I. INTRODUCTION

All signs indicate that the "Industrial Internet of Things" (IIoT) revolution is upon us. This revolution relies on ubiquitous low-cost sensor- and actuator-enabled devices with constrained computational capabilities that operate autonomously and are connected to the Internet through low-power wireless communication technologies. The deployment of such devices within the factory floor will bring together OT (Operation Technologies) and IT (Information Technologies) and enable industrial process monitoring and control applications. It is therefore of great importance to train the next generation Electrical Engineering (EE) and Computer Science (CS) professionals on these low-power wireless communication technologies and standards that will be used to connect devices to the Internet in the industrial space.

T. Watteyne is with the EVA team, Inria, Paris, France. E-Mail: thomas.watteyne@inria.fr.

P. Tuset-Peiro is with the Department of Computer Science, Multimedia and Telecommunications (EIMT), Universitat Oberta de Catalunya (UOC), Barcelona, Spain. E-Mail: peretuset@uoc.edu.

X. Vilajosana is with the Internet Interdisciplinary Institute (IN3), Universitat Oberta de Catalunya (UOC), Barcelona, Spain. E-Mail: xvilajosana@uoc.edu.

S. Pollin is with the Department of Electrical Engineering (ESAT), KU Leuven, Leuven, Belgium. E-Mail: sofie.pollin@esat.kuleuven.be.

B. Krishnamachari is with the Department of Electrical Engineering and Computer Science (EECS), University of Southern California (USC), Los Angeles, CA, USA. E-Mail: bkrishna@usc.edu.

Manuscript received January 2, 2017; revised XXX, XXX, 2017.

We have observed a rising trend to use Wi-Fi (IEEE 802.11) and the IPv4 stack as the communication standards to teach IIoT-related courses. We acknowledge that there are many benefits to using Wi-Fi and IPv4 for such purpose, including the fact that Wi-Fi connectivity is ubiquitous, and that IPv4 is still the *de facto* protocol stack to connect to the Internet. However, if the focus is on connecting devices to the Internet for industrial applications, then this approach is flawed for several reasons. *First*, Wi-Fi is not low-power and it does not provide the determinism and robustness levels required by industrial applications, thus limiting its applicability. *Second*, the IPv4 stack has been superseded by the IPv6 stack, which provides several advantages in terms of simplicity, scalability, efficiency, quality of service and security. *Third*, chips implementing Wi-Fi and IPv4 are typically provided as a black box, thereby limiting the opportunities to learn how they operate at the lower levels.

In contrast, standards for low-power wireless communications specifically targeted at the IIoT are reaching technical maturity and being adopted by industries around the world. One particular set of standards is being developed by the IETF 6TiSCH standardization Working Group. 6TiSCH builds upon the IEEE802.15.4 TSCH link-layer without changing it, and combines it with an IPv6-ready "upper stack". The result is the best of both worlds: industrial-grade performance and ease of integration into the Internet. Despite the fact that the standardization process is not finished, there are "pre-6TiSCH" products already available on the market, such as Linear Technology's SmartMesh IP.

Compared to the Wi-Fi/IPv4 approach, there are several elements that make 6TiSCH an ideal match to teach how to connect devices to the Internet. *First*, it is based on truly low-power wireless mesh communication technology that has been validated in the industry for the past decade. *Second*, it is based on next-generation Internet standards currently being developed, which allows to follow the standardization process and learn from it. *Third*, it is a simple yet complete stack, making it easy to understand the basic concepts but, allowing to go deep into protocols and the implementation details.

This article shows how 6TiSCH can serve as an accelerator for students to learn a fully standards-based communication stack, and get familiar with key IIoT protocols.

II. TECHNICAL OVERVIEW OF 6TISCH

This section offers an overview of 6TiSCH, discussing both the protocol stack (Section II-A), its relevance and success stories (Section II-B).

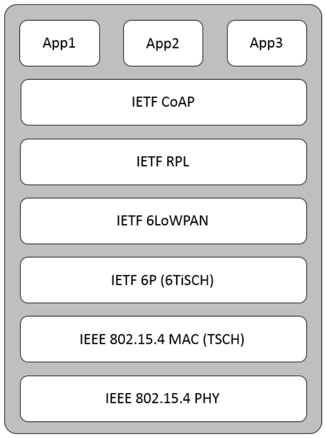

Fig. 1. The IETF 6TiSCH protocol stack, composed of IEEE 802.15.4, IETF 6P, 6LoWPAN, RPL and CoAP.

A. The 6TiSCH Protocol Stack

Fig. 1 shows the 6TiSCH protocol stack. It is composed of standards developed by the IEEE[1] and the IETF[2]. Together, these standards achieve IPv6-based end-to-end connectivity while meeting the determinism, robustness and low-power operation required for battery-operated devices targeted at industrial process monitoring and control applications [1].

This section provides a brief introduction to each layer in the 6TiSCH stack – from the bottom up – highlighting the educational potential of each.

IEEE802.15.4 defines the physical and data-link layers for low-power wireless communication between battery-operated devices. It offers an appropriate trade-off between transmit power (0-10 dBm), data-rate (250 kbps) and maximum payload size (127 bytes). The new IEEE 802.15.4 Time Slotted Channel Hopping (TSCH) mode is targeted at demanding industrial applications. TSCH combines network-wide synchronization and channel hopping to achieve over 99.999% end-to-end reliability and over a decade of battery lifetime.

TSCH is a basic yet complete link-layer protocol to teach students the basics of multiple channel access, frequency diversity, scheduling and coexistence. It trades-off throughput with packet error rate and delay, by scheduling more or less redundant transmissions. Students can derive equations and perform simulations to learn how to do the performance analysis of TSCH-enabled networks of various sizes.

The **IETF 6top Protocol** (6P) [2] is being standardized by the 6TiSCH Working Group. It defines a distributed scheduling protocol whereby neighbor nodes negotiate to add/remove one or multiple cells in the TSCH schedule. Each cell is a "communication opportunity" for the neighbor nodes to exchange a link-layer frame. Adding more cells increases the bandwidth and lowers the end-to-end latency, while at the same time increasing the nodes' power consumption.

6P is a simple protocol which allows students to not only see how the TSCH schedule is managed "down to the wire", but also touch upon notions such as schedule consistency, network stability and network churn.

IETF 6LoWPAN [3] allows long IPv6 packets (up to 1280 bytes) to fit into short IEEE802.15.4 frames (at most 127 bytes). It is composed of two main mechanisms. *First*, it defines rules for compacting the IPv6 header. This is done by (1) removing fields that aren't needed, (2) removing fields which always have the same contents, and (3) compressing the IPv6 addresses by inferring them from link-layer addresses. The result is that the 40-byte IPv6 header gets compacted down to a couple of bytes in the most favorable case. A Low-power Border Router (LBR) sits at the edge of the low-power wireless network and is responsible for doing the transparent IPv6→6LoWPAN and 6LoWPAN→IPv6 translation: a computer outside the low-power wireless network interacts with a low-power wireless directly with its IPv6 address. *Second*, it defines fragmentation rules so multiple IEEE802.15.4 frames can make up one IPv6 packet.

6LoWPAN allows the students to really understand every single bit in the IPv6 header, and feel the power of the Internet. That is, once the devices implement 6LoWPAN correctly and the LBR is running, just by injecting a correctly-formatted packet into the Internet, the tiny low-power wireless device sitting on the student's desk can be interacting with a computer half-way around the globe.

IETF RPL [4] is an intra-domain routing protocol for low-power wireless mesh networks. RPL organizes the network into a Directed Acyclic Graph (DAG) rooted at a gateway device that connects to the Internet. As other distance-vector protocols, devices regularly advertise their distance to the root, allowing neighbors to compute their own.

Despite being "simple" (there are only two required packet formats), by playing with RPL students can understand the complexity of maintaining a coherent multi-hop routing structure in a network made up of "lossy" wireless links.

IETF CoAP [5] is the "HTTP for constrained devices". It turns every low-power wireless device into a web server and a browser, and allows web-like interactions. A constrained node can publish its sensor readings onto a server on the Internet or a smart-phone can operate smart blinds, all by using the communication paradigms of the Internet. Extensions to popular browsers and open-source libraries make adding CoAP support to an computer application easy.

CoAP is not particularly complicated from a protocol point of view; it's header is 4 bytes long and it is "just" an application protocol. But by using this protocol, students can "put everything together" and really get that interacting with a 6TiSCH devices is just as easy as interacting with a web browser. Moreover, thanks to standardization, there is no need to learn exotic protocols, as third party libraries and tools are readily available.

B. Success Stories and Relevance

The 6TiSCH stack does not come out of thin air, it is the result of a rigorous multi-year standardization effort.

6TiSCH is the latest generation of protocols exploiting TSCH technology, and inherits from the lessons learnt from

[1] https://www.ieee.org/
[2] https://www.ietf.org/

the TSMP (2006, proprietary protocol), WirelessHART (2008) and ISA100.11a (2011).

Since its formation in October 2013, 6TiSCH has followed the rigorous IETF standardization process. 398 people – with a mix of technology providers, end-user companies and academics – follow the standardization activity of the group through its mailing list and periodic face-to-face and online meetings.

Several companies have commercialized or are preparing "pre-6TiSCH" products. One example is Linear Technolgy's SmartMesh IP product line. A vibrant community has built around 6TiSCH, with open-source implementations, open hardware platforms, open testbeds, "pre-6TiSCH" commercial off-the-shelf solutions, simulation platforms and other tools.

There is a clear momentum behind the 6TiSCH stack. From an educational perspective this means that it makes sense to teach 6TiSCH to the next generation EE and CS professionals, facilitated by a wide availability of tools and implementations. One challenge is of course that the standardization activity is ongoing, which means the technology might still evolve.

III. IIoT Course Objectives

The topic of "Industrial IoT" is an educational goldmine.

A course of IIoT should be targeted at advanced undergraduate or graduate levels. It can target both Electrical Engineering (EE) and Computer Science (CS) students, and can be taught to mixed groups. Ideally, pre-requisites include a classical computer networking, computer architecture, computer programming and digital electronics. Depending on the type of course (see Section V), some of these requirements can be lifted, at the cost of more "hand-holding" the students.

Probably the most specific of an IIoT course – and a key differentiator with other courses at that level – is its systems nature. That is, the course will allow the student to "put everything together" and build a minimal, but fully functional device, that connects the physical and digital worlds by interfacing physical sensors and actuators to a micro-controller and that interacts with other devices on the Internet. During the process, the student will be able to get his/her hands on the complete chain (hardware, software, standards) and put his/her prior knowledge to test.

One important limitation of 6TiSCH is that standardization is still underway, so the technical specifications the students will be working on might not be stable.

A. Core Educational Focus

There are 3 core educational focus points to an IIoT course.

Core Focus 1: Medium Access Protocols. What makes 6TiSCH different from other low-power wireless technologies is the fact that it uses TSCH technology. Hence, a good first focus point are Medium Access Control (MAC) protocols. In particular, the first goal is to teach the students about the main challenges of building a reliable low-power wireless network: multi-path fading and external interference. The instructor can then touch upon TSCH, the technique used to achieve both channel hopping (to bring reliability) and low-power operation.

Core Focus 2: Standardization process. Based upon Core Focus 1, the course can then focus on the standards which exploit channel hopping and synchronization. This discussion is a great way to introduce the standardization process: What are the main standardization bodies? Who contributes to them and for which reasons? It is a good idea to highlight what makes the IEEE and the IETF different, but stressing the point that they both contribute equally to building the Internet. Somewhat counter-intuitively, standardization is usually well-received by students who appreciate learning about what it takes to turn an idea into inter-working devices. 6TiSCH still being standardized is a great opportunity to learn about the process, as standardization happens.

Core Focus 3: End-to-end systems. The goal of any IoT system, including the IIoT, is to connect low-power wireless devices to the Internet. Probably the most important focus in a course is to take a systems approach and *build* an end-to-end system. Students are usually very engaged when "seeing it all work together". It is therefore important to spend some time discussing the back-end system (including turn-key solutions such as MQTT, Node-Red, Thingworx or IBM Bluemix).

B. Optional Educational Focus

An additional 3 focus points can be added as an option to the course, depending on the students' background and interests.

Optional Focus 1: Low-level programming *(for CS)*. CS students usually enjoy getting to the bottom of a computer system, and the embedded nature of an IIoT system is a great example. The availability of open-source implementations (Section IV-A) are a great opportunity to discover the inner-details of embedded operating systems (concurrency, task switching, memory footprint) and low-level peripherals (timers, communication buses, radio transceivers).

Optional Focus 2: Hardware *(for EE)*. IIoT devices are low-power wireless and battery operated in the vast majority of the cases. They consist of a micro-controller and radio connected to sensors and actuators. Hence, they present a unique opportunity to delve into the design of low-power circuits (including the trade-off between performance and power consumption when choosing a micro-controller and a radio chip) and the integration of sensors and actuators. Advanced groups can even go through the exercise of manufacturing and characterizing a printed circuit board.

Optional Focus 3: Security *(for CS and EE)*. The cyber-physical nature of the IIoT, combined with the critical requirements of industrial applications it is used in, makes the importance of security clear to the students. With IIoT, it is important to stress that security does not come as an optional add-on, but must be built in from the start. As with all systems, security is built into every layer of the protocol stack, including link-layer (CCM*), transport (DTLS) and application (object security). A security focus point should contain an overview of these different levels, and some hands-on exposure to at least one of them.

IV. Tools and Resources around 6TiSCH

This section serves as a listing of the 6TiSCH-related tools and resource available to the person teaching IIoT using

6TiSCH. We only list the tools and resources the authors of this article have either used for teaching, or have directly contributed to.

A. Open-Source Academic Implementations

As detailed in [6], 6TiSCH is now supported by the three major open-source low-power wireless stacks:

OpenWSN[3] [7] is an open-source implementation specifically targeted at 6TiSCH. It was started at UC Berkeley in 2010, and has been considered the reference implementation for the 6TiSCH protocol stack, including in 6TiSCH interoperability events. Its use in research and industrial applications is increasing and OpenWSN is now supported on the most popular platforms. There are several tutorials available. OpenWSN includes the OpenSim emulator, detailed below.

Contiki[4] is an open-source operating system targeting low-power devices and aiming to provide the user with a complete toolbox for the design of IoT solutions. There are plenty of examples on how to use it online, and it is coupled elegantly with the Cooja simulator, which enables simulation of a protocol implementation before it is deployed in a testbed. Currently Contiki supports TSCH.

RIOT[5] is an open-source operating system for the IoT. It is developer friendly, as it allows easy programming in C/C++ and is supported by a wide range of platforms. Currently RIOT supports TSCH through the OpenWSN project.

B. Open Hardware Platforms

In principle, all hardware platforms that include a micro-controller and an IEEE802.15.4 radio transceiver are susceptible of implementing the 6TiSCH stack, since the core requirement to support it is a timer with capture and compare registers. For example, the 6TiSCH stack has been successfully ported on the TelosB platform, which was developed at UC Berkeley back in 2004. Modern platforms based on the Cortex-M architecture are better suited to support the 6TiSCH stack given the fact that they are less constrained in terms of RAM and Flash memory.

A good example is the OpenMote platform[6] (Fig. 2), which was developed at UC Berkeley in 2013 as a replacement for the TelosB. The OpenMote platform is based on the Texas Instruments CC2538 System on Chip (SoC). The main advantage of the SoC architecture compared to the dual-chip approach is that it removes the timing limitations associated to the micro-controller to radio transceiver communications based on a bus. This is particularly important for the 6TiSCH time sensitive implementation.

C. Open Testbeds

Testbeds are an important tool for teaching since they enable large-scale experimentation while removing the need for each student to buy hardware [8].

[3] https://www.openwsn.org/
[4] http://www.contiki-os.org/
[5] https://riot-os.org/
[6] http://www.openmote.com/

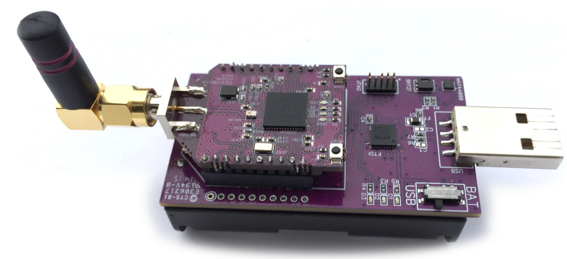

Fig. 2. The OpenMote platform with the OpenUSB carrier board. The OpenUSB contains various sensors (temperature, relative humidity, light, 3-axis acceleration) and allows a user to reprogram it over USB.

The **IoT-Lab**[7] is a 2728-node testbed deployed over 7 locations in France and Germany. Three main hardware platforms are supported in the IoT-LAB to conduct experiments, representative of the variety of IoT hardware. IoT-LAB offers a web-based reservation system, a set of tools that enable application development and real-time debugging, and tools to measure current consumption.

TutorNet[8] is a 100-node low-power wireless embedded IoT testbed that has been operational for nearly 10 years. It has been used for research and teaching of many wireless sensor network protocols, including multi-channel scheduling and routing protocols relevant to 6TiSCH. Originally consisting of TMote Sky nodes, it is currently being upgraded to the OpenMote platform to support new operating systems, such as OpenWSN and RIOT, and new standards such as 6TiSCH. The testbed allows researchers to deploy code on any subset of nodes, run the experiments and collect the data. Currently visual tools to program the nodes are being developed and the whole software package is being documented to allow for replication in other institutions.

D. "Pre-6TiSCH" Commercial Off-the-Shelf Solutions

The 6TiSCH standardization process is not finished yet, so commercial 6TiSCH products do not exist. That being said, several "pre-6TiSCH" commercial solutions exist, which combine the industrial performance of TSCH with an IPv6-enabled stack in a way very similar to the approach taken by 6TiSCH.

One example is Linear Technology's SmartMesh IP product line, which has been commercially available since 2011. The specificity of this solution is that Linear Technology manufactures its own system-on-chip. The result is a best-in-class solution which achieves over 99.999% end-to-end reliability and over a decade of battery lifetime. A SmartMesh IP starter kit[9] (Fig. 3) is available which contains all the pieces necessary to build and benchmark a network. This kit has been used for numerous courses and educational material.

E. Simulation Platforms

Several tools have been created to simulate the performance of a 6TiSCH network.

[7] https://www.iot-lab.info/
[8] http://anrg.usc.edu/www/tutornet/
[9] http://www.linear.com/solutions/3106

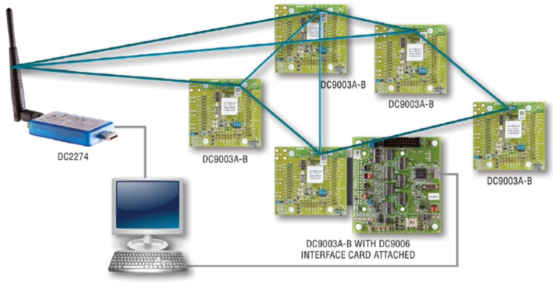

Fig. 3. Linear Technology's SmartMesh IP starter kit allows students to experiment with the market leading low-power wireless IIoT solution.

NS-3 is the most widely adopted tool for network simulation. It provides a large database of protocols, tools and propagation models to emulate almost any kind of network. There are several ongoing efforts to add support for IEEE802.15.4 TSCH and 6TiSCH[10].

The **6TiSCH simulator**[11] is a Python-based high-level simulator maintained by the 6TiSCH working group. It allows students vto benchmark the performance of a network implementing the full 6TiSCH stack.

OpenSim is the emulator which is part of the OpenWSN project. That is, the OpenWSN firmware can be compiled as a Python extension module, and run within OpenVisualizer, the OpenWSN software running on the gateway computer. In this setup, the same firmware runs, allows debugging and performance estimation without requiring real hardware.

Cooja is a network simulator linked to the Contiki project. It is developed in Java and is built around the concept of the COOJA mote. The Contiki native code, written in C, is compiled as a shared library for the MSP430 16-bit architecture using the GCC tool-chain for the MSP430 platform. Since Contiki supports the 6TiSCH stack using a platform-agnostic approach, it is possible to simulate the operation of a node and a whole 6TiSCH network in Cooja.

F. Miscellaneous Tools

The channel-hopping nature of a 6TiSCH networks makes it challenging to capture over-the-air traffic, as typical sniffers can only listen to a single channel at a time. Beamlogic's sniffer[12] listens to all 16 frequencies in the 2.4 GHz band simultaneously, thus allowing to capture all packets on air. The Argus project https://github.com/openwsn-berkeley/argus builds upon it to allow co-located students to share one sniffer through the Internet.

Recent versions of Wireshark[13] come with 6TiSCH dissectors, including IEEE802.15.4-2015 and the 6TiSCH 6top protocol. These dissectors are developed and maintained by the OpenWSN project team[14].

[10] https://github.com/EIT-ICT-RICH/ns-3-dev-TSCH
[11] https://bitbucket.org/6tisch/simulator/
[12] http://www.beamlogic.com/
[13] https://www.wireshark.org/
[14] https://github.com/openwsn-berkeley/dissectors

There are countless CoAP libraries written in virtually all programming languages (including Python[15]), which allow to add CoAP support to computer applications. The Copper extension to the Firefox browser[16] is still the most easy to use.

G. Available Experimental Results

Over the last couple of years dozens of papers have been published presenting results on reliability [9], energy consumption [10], synchronization accuracy [11] of 6TiSCH, complemented with application results exploiting and evaluating the 6TiSCH technology [12], [13]. These results bring fundamental insights into the performance of 6TiSCH.

V. Teaching the IIoT with 6TiSCH

The availability of numerous related resources and tools make it easy to use 6TiSCH as a basis for an IIoT course. The instructor's role is to select the pieces he/she wants the students to focus on and to assemble them in a logical order. With such a diversity of educational focus points and tools to choose from, we believe it would be wrong to recommend a single course outline. Rather, we highlight in Sections V-A and V-B the two main questions an instructor should ask him/herself when preparing a 6TiSCH-based course. For the instructors in a hurry, Section V-C lists several completely packaged and ready-to-teach 6TiSCH material.

A. Add to Existing Course or New Course?

As an instructor, a first question to ask is whether the IIoT material will be added to an existing course, or whether it will be the base for a full new course.

The first approach is adding 6TiSCH to an existing course. Courses suitable for such an addition include introductory courses on computer networks, wireless networks, distributed systems and embedded systems. Materials pertinent to 6TiSCH could be taught as a single module to illustrate concepts ranging from protocol layering in computer networks to interference mitigation in wireless networks. Alternatively, they may be inter-leaved throughout the course, for instance by exposing students sequentially to different layers of the 6TiSCH stack, as they cover different layers of the protocol stack in a bottom up (physical to application layer) or top-down (application to physical layer) manner.

The second approach is creating a full course based on 6TiSCH. The main challenge for such a course is that it should be broad and multidisciplinary, going from hardware all the way up to the application layer. When relying on the 6TiSCH standard, it is possible to cover all required protocols and technologies in one course, as 6TiSCH is supported by a broad range of tools used in networking design and test, from analysis to implementation. Within a full 6TiSCH course, we recommend a four-fold methodology that allows to cover both the theoretical and hands-on knowledge: *(i)* Deployment, *(ii)* Simulation, *(iii)* Testbeds, *(iv)* Analysis. Depending on the

[15] https://github.com/openwsn-berkeley/coap
[16] https://addons.mozilla.org/en-US/firefox/addon/copper-270430/

choice of the instructor, it is possible to first start with hands-on experiments and then gradually move to simulation and analysis for understanding and explaining the experimental results. Alternatively, a course can follow the traditional design flow starting from analysis, to design and simulation and finally deployment.

B. Open-Source or Commercial?

One important question to ask ourselves as instructors is whether to use an open-source implementation (Section IV-A) or an off-the-shelf commercial solution (Section IV-D).

Understanding the distinction is crucial, and it is important for students to also understand it. No matter the enthusiasm of the communities behind them, open-source implementations are never at the level of commercial solutions. The latter tend to "just work", while the former are often riddled with little bugs and undocumented subtleties, even if they might appear similar on paper.

As educators ourselves, our rule-of-thumb has been simple. For a course which focuses on building a real system that just works, we use a commercial solution for the low-power wireless portion. For a course which focuses on the internals of the wireless system, an open-source implementation is acceptable, as long as the students are reasonably comfortable with the tools involved.

Building an IIoT system is complex, as it consists of a chain of elements (sensors, operating system, protocol stack, back-end system, etc.) where each of which can fail. In our experience, using a commercial solution has allowed students to "go further" and build more polished solutions, at the cost of hiding some of the details of the low-power wireless network.

C. Teaching Track Record and Teaching Resources

The authors of this article have been using 6TiSCH as a vehicle for teaching the IIoT from half-day tutorials at conferences to a multi-week advanced undergraduate teaching courses. As a result, a lot of material has been prepared which is readily available for instructors to use.

For a course using the open-source approach, the OpenWSN project has been the foundation for half a dozen short-courses. All the material, including presented slides, virtual machines and pictures are maintained on the project website.

For a course using the commercial approach, the DustCloud community[17] has been maintaining the "Dust Academy", a series of 32 labs student can follow using a SmartMesh IP starter kit. This material is ready to be presented and instructors can get access to instructor-specific material, including answers to the questions.

VI. Conclusion

The 6TiSCH stack is an educational goldmine. It is a combination of protocols for the IIoT, resulting from a rigorous standardization process in the most prominent standardization development organizations. The 6TiSCH stack is widely seen as the key enabler for the IIoT, as it combines the performance of proven industrial low-power wireless communication technologies with the ease of use of an IPv6-enabled stack. A vibrant community has built around it, making a plethora of tools and resources available to the instructors, from open-source implementations to "pre-6TiSCH" commercial products.

This article serves as a guide for the instructor to teaching an IIoT course using 6TiSCH. Rather than recommending a single approach, it describes the educational focus points of an IIoT course, provides an exhaustive list of the tools and resource available, and discusses the key options an instructor has to take when preparing the course.

[17] http://www.dustcloud.org/

Acknowledgments

This work is partially supported by the EU H2020 F-Interop and H2020 ARMOUR projects, Inria through the DIVERSITY associate team, the Spanish Ministry of Economy and the FEDER regional development fund through the SINERGIA project (TEC2015-71303-R), the Center for Cyber-Physical Systems and the Internet of Things at the University of Southern California.

References

[1] Maria Rita Palattella, Nicola Accettura, Xavier Vilajosana, Thomas Watteyne, Luigi Alfredo Grieco, Gennaro Boggia, and Mischa Dohler. Standardized Protocol Stack for the Internet of (Important) Things. *IEEE Communications Surveys Tutorials*, 15(3):1389–1406, March 2013.

[2] Qin Wang and Xavier Vilajosana. 6top Protocol (6P), 31 October 2016. IETF, draft-ietf-6tisch-6top-protocol-03 [work-in-progress].

[3] Jonathan Hui and Pascal Thubert. Compression Format for IPv6 Datagrams over IEEE 802.15.4-Based Networks, September 2011. IETF, RFC6282.

[4] Tim Winter, Pascal Thubert, Anders Brandt, Jonathan Hui, Richard Kelsey, Philip Levis, Kris Pister, Rene Struik, JP Vasseur, and Roger Alexander. RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks, March 2012. IETF, RFC6550.

[5] Zach Shelby, Klaus Hartke, and Carsten Bormann. The Constrained Application Protocol (CoAP), June 2014. IETF, RFC7252.

[6] Thomas Watteyne, Vlado Handziski, Xavier Vilajosana, Simon Duquennoy, Oliver Hahm, Emmanuel Baccelli, and Adam Wolisz. Industrial Wireless IP-Based Cyber-Physical Systems. *Proceedings of the IEEE*, 104(5):1025–1038, May 2016.

[7] Thomas Watteyne, Xavier Vilajosana, Branko Kerkez, Fabien Chraim, Kevin Weekly, Qin Wang, Steven Glaser, and Kris Pister. OpenWSN: a standards-based low-power wireless development environment. *Transactions on Emerging Telecommunications Technologies (ETT)*, 23(5):480–493, 2012.

[8] A. Gluhak, S. Krco, M. Nati, D. Pfisterer, N. Mitton, and T. Razafindralambo. A Survey on Facilities for Experimental Internet of Things Research. *IEEE Communications Magazine*, 49(11):58–67, November 2011.

[9] Thomas Watteyne, Ankur Mehta, and Kris Pister. Reliability Through Frequency Diversity: Why Channel Hopping Makes Sense. In *International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems (PE-WASUN)*, pages 116–123, Tenerife, Canary Islands, Spain, 26-30, October 2009. ACM.

[10] Xavier Vilajosana, Qin Wang, Fabien Chraim, Thomas Watteyne, Tengfei Chang, and Kris Pister. A Realistic Energy Consumption Model for TSCH Networks. *IEEE Sensors Journal*, 14(2):482–489, 2014.

[11] David Stanislowski, Xavier Vilajosana, Qin Wang, Thomas Watteyne, and Kristofer SJ Pister. Adaptive Synchronization in IEEE802.15.4e Networks. *IEEE Transactions on Industrial Informatics*, 10(1):795–802, 2014.

[12] Thomas Watteyne, Ana Laura Diedrichs, Keoma Brun-Laguna, Javier Emilio Chaar, Diego Dujovne, Juan Carlos Taffernaberry, and Gustavo Mercado. PEACH: Predicting Frost Events in Peach Orchards Using IoT Technology. *EAI Endorsed Transactions on Internet of Things*, 16(5), December 2016.

[13] Ziran Zhang, Steven D. Glaser, Thomas Watteyne, and Sami Malek. Long-term Monitoring of the Sierra Nevada Snowpack Using Wireless Sensor Networks. *IEEE Internet of Things Journal*, December 2016.

Thomas Watteyne is a researcher at Inria, in Paris, and a Senior Networking Design Engineer in the Dust Networks product group at Linear Technology, in Silicon Valley. He co-chairs the IETF 6TiSCH working group. He did his postdoctoral work in Prof. Kristofer Pister's lab at UC Berkeley. He founded and co-leads Berkeley's OpenWSN project. He holds a PhD in Computer Science (2008), an MSc in Networking (2005) and an MEng in Telecommunications (2005) from INSA Lyon, France.

Pere Tuset-Peiró is Assistant Professor at the Computer Science, Multimedia and Telecommunications of Universitat Oberta de Catalunya (UOC) in Barcelona, Spain. He received the BSc and MSC in Telecommunications Engineering from Universitat Politcnica de Catalunya (UPC) in 2007 and 2011 respectively, and the PhD in Network and Information Technologies from Universitat Oberta de Catalunya (UOC) in 2015. Currently, Pere has published over 20 journal and conference publications, and holds 7 international patents related to embedded systems, wireless communications and signal processing.

Xavier Vilajosana is Principal Investigator of the Wireless Networks Research Lab at Universitat Oberta de Catalunya (UOC). Xavier is also co-founder of Worldsensing and OpenMote Technologies. Xavier is an active member of the IETF 6TiSCH WG where he authored different standard proposals. Xavier also holds 30 patents. Xavier has been visiting professor at the Prof. Pister UC Berkeley lab. He co-leads Berkeley's OpenWSN project. He has been Senior Researcher at the HP R&D labs (2014-2016) and visiting researcher at the France Telecom R&D Labs Paris (2008). He holds a PhD (2009) from UOC, MSc and MEng (2004) from UPC, Barcelona, Spain.

Sofie Pollin obtained her Ph.D. degree at KU Leuven with honors in 2006. From 2006 to 2008 she continued her research on wireless communication, energy-efficient networks, cross-layer design, coexistence, and cognitive radio at UC Berkeley. In November 2008 she returned to imec to become a principal scientist on the green radio team. Since 2012 she has been a tenured track assistant professor in the Department of Electrical Engineering, KU Leuven, Belgium. Her research centers around networked systems that require networks that are ever more dense, heterogeneous, battery powered, and spectrum constrained. She is a BAEF and Marie Curie fellow, and an IEEE senior member.

Bhaskar Krishnamachari (M02 – SM14) received the B.E. degree in electrical engineering at The Cooper Union, New York, NY, USA, in 1998, and the M.S. and Ph.D. degrees from Cornell University, Ithaca, NY, USA, in 1999 and 2002, respectively. He is currently Professor and Ming Hsieh Faculty Fellow in the Department of Electrical Engineering at the Viterbi School of Engineering, University of Southern California, Los Angeles, CA, USA. He is also the Director of the USC Viterbi School of Engineering Center on Cyber-Physical Systems and the Internet of Things. His primary research interest is in the design and analysis of algorithms, protocols, and applications for next-generation wireless networks.