Competition: OpenWSN, a Development Environment for 6TiSCH

Tengfei Chang, Thomas Watteyne, Xavier Vilajosana

Competition: OpenWSN, a Development Environment for 6TiSCH

Tengfei Chang
EVA team, Inria-Paris
tengfei.chang@inria.fr

Thomas Watteyne
EVA team, Inria-Paris
thomas.watteyne@inria.fr

Xavi Vilajosana
Universitat Oberta de Catalunya, Barcelona, Spain
xvilajosana@uoc.edu

Abstract

6TiSCH is a standardization effort at the IETF that combines industrial performance with seamless Internet connectivity. It brings together a number of existing standards, and adds scheduling capabilities. One of these standards is IEEE802.15.4 Time Slotted Channel Hopping (TSCH), a medium access control layer technique that provides high reliability and ultra low power consumption. Another is RPL, a routing protocol that turns the topology into a multi-hop mesh network. OpenWSN is an open-source implementation of 6TiSCH. It provides a set of developing tools and supports multiple hardware platforms. OpenWSN is the basis of our solution for the EWSN 2019 dependability competition.

1 6TiSCH, the Industrial IoT Protocol Stack

6TiSCH [1] defines a protocol stack for Industrial IoT, depicted in Fig. 1. This protocol stack is rooted in the IEEE802.15.4 physical layer. Time Slotted channel hopping (TSCH), as Medium Access Control (MAC) layer protocol, provides ultra-high reliability and ultra-low energy consumption. The 6TiSCH protocol stack uses the 6top protocol [4], one or more 6TiSCH scheduling functions, and the 6LoWPAN adaption layer to transport large IPv6 packets into short IEEE802.15.4 frames [3]. RPL [7] is a routing protocol for low-power lossy networks. It allows communication both from devices inside the mesh network to its gateway ("upstream") and from the gateway to devices ("downstream"). 6TiSCH uses CoAP [2] at its application layer, including for carrying the secure joining messages.

According to the competition categories, there are two traffic patterns, multipoint-to-point (MP2P) and point-to-multipoint (P2MP). RPL supports both traffic pattern. Moreover, the competition evaluation metrics are end-to-end reliability, energy consumption and latency. 6TiSCH is a natural fit for these types of traffic. Adapting the scheduling function further allows us to trade-off lower latency for higher energy consumption.

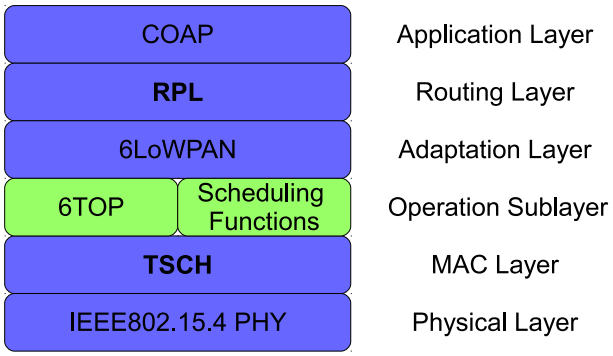

Figure 1. In the 6TiSCH protocol stack, TSCH and RPL are the main two protocols that influence network performance.

The following two sections describes TSCH and RPL, the main two protocols that influence network performance.

2 Time Slotted Channel Hopping (TSCH)

TSCH achieves high reliability through channel hopping, and ultra low-power consumption through time synchronization.

In a TSCH network, time is split into time slots. A pair of communicating neighbor nodes are synchronized when their time slot boundaries are aligned. Link-layer frames are exchanged within slots. Since nodes are synchronized, the receiver knows when in the slot the transmitter starts sending a frame. This time offset is `TxOffset` in Fig. 2. This allows a receiving node to only turns on its radio when it is about to receive a frame (withing a guard time of `TxOffset` to account for clock drift). This mechanism significantly the radio-on time, hence the average energy consumption of the device. All nodes share a common sense of time, as each time slot is identified by the Absolute Slot Number (ASN), an ever-increasing number. This shared notion of time allows neighbor nodes to turn on their radio only when the are scheduled to exchange a frame.

In TSCH, the 83 MHz frequency band at 2.4 GHz is split into 16 equally-spaced channels. All communication is orchestrated by a schedule. That schedule can be represented as a slot $\times$ channel matrix, as shown in Fig. 2. When neighbor nodes are about to exchange a frame, they

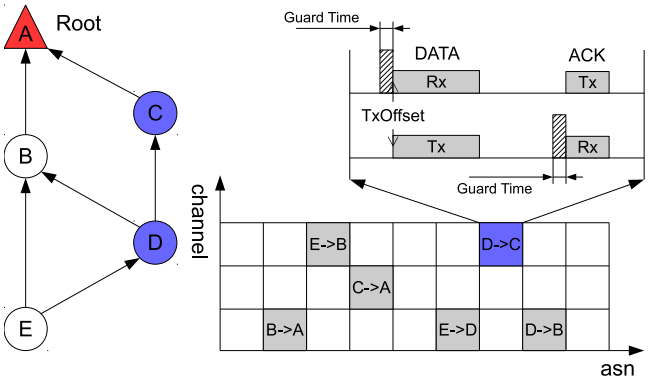

Figure 2. The TSCH schedule indicates to each now what to do in each time slot. The cell shaded blue is use by node D to send a frame to node C, and get a link-layer acknowledgement.

use Eq. 1 to compute the frequency they will tune their radio to. $chOffset$ is the logic channel offset used by a slot. $numChannels$ indicates the number of channels used by the network. Fig. 2 shows a 5 node network using 3 channels to communicate with one another. Channel hopping is the mechanism by which successive frames exchanged between two neighbors are done so on a different frequency. Channel hopping is effective at combating external interference and multiple-path fading [5].

$$channel = (ASN + chOffset)\%numChannels \quad (1)$$

3 Routing Protocol for Low-Power and Lossy Networks (RPL)

The RPL routing protocol enables both upstream and downstream communication. It organizes the network into a Directed Acyclic Graph (DAG) by assigning a rank to each node in the network. The rank of a node indicates the cost to send a packet to the root node. A node's rank increases as it is multiple hops away from the root, and because the it has to its routing parent is weak. Upstream communication a achieved when each node forward a message to its parent. Fig. 2 shows a DODAG with 5 nodes.

A node can have multiple neighbors in its parent set. Fig. 2 shows that node D has two neighbors which are closer than itself to the root: nodes B and C. In case of node B is powered off, node D can still forward its packets to node C, so no data is lost. Having multiple nodes in the parent set increases the reliability and stability of the network.

4 OpenWSN

OpenWSN [6] is the reference open-source implementation of the 6TiSCH protocol stack. It mainly contains two parts: the firmware running on the embedded devices, and the software running on the PC. The firmware can be run on 11 different hardware platforms, including the TelosB board used in the competition. The software – called OpenVisualizer – contains a set of supporting scripts, including visualizing the status of the node in the network, or configuring one of the nodes as the root of the network.

The following sections introduces two features: HDLC-based serial debugging, and the build system.

4.1 HDLC-based Serial Debugging

In OpenWSN firmware provides a component called **openserial**. The component is responsible for forwarding the status of nodes, the packet to the root or Internet, and receiving commands from OpenVisualizer. We use High-Level Data Link Control (HDLC) as the framing mechanism on top of the raw serial communication. The status of the node includes the node's identifiers, the node's rank, the current ASN, whether the node is synchronized, as well as details about the node's schedule, neighbor and transmit queue. The OpenVisualizer gathers this status information printed by the nodes, and displays it conveniently on a dynamic web page.

4.2 A Complete Build System

OpenWSN uses **SCons** to build the 6TiSCH protocol stack for different target hardware platforms. Scons is a build system, similar to "Make", but written in Python. With the `gcc` toolchain for msp430, the source code can be compiled for the TelosB. OpenWSN also provides several bootloader scripts, allowing a user to program her devices after compiling the code and generating the image.

5 Conclusion

We propose to use OpenWSN for the EWSN 2019 dependability competition. This extended abstract introduces the 6TiSCH protocol, as well as its two main protocols: TSCH and RPL. We describe OpenWSN, and detail its HDLC-based serial debugging capability, and the build system used.

6 References

[1] D. Dujovne, T. Watteyne, X. Vilajosana, and P. Thubert. 6TiSCH: Deterministic IP-enabled Industrial Internet (of Things). *IEEE Communications Magazine*, 52(12):36–41, December 2014.

[2] Z. Shelby, K. Hartke, and C. Bormann. The Constrained Application Protocol (CoAP). RFC 7252, June 2014.

[3] P. Thubert and J. Hui. Compression Format for IPv6 Datagrams over IEEE 802.15.4-Based Networks. RFC 6282, Sept. 2011.

[4] Q. Wang, X. Vilajosana, and T. Watteyne. 6TiSCH Operation Sublayer (6top) Protocol (6P). RFC 8480, Nov. 2018.

[5] T. Watteyne, A. Mehta, and K. S. Pister. Reliability Through Frequency Diversity: Why Channel Hopping Makes Sense. In *ACM Symposium on Performance Evaluation of Wireless Ad Hoc, Sensor, and Ubiquitous Networks (PE-WASUN)*, pages 116–123. ACM, 2009.

[6] T. Watteyne, X. Vilajosana, B. Kerkez, F. Chraim, K. Weekly, Q. Wang, S. Glaser, and K. S. Pister. OpenWSN: a standards-based low-power wireless development environment. *Transactions on Emerging Telecommunications Technologies (ETT)*, 23(5):480–493, 2012.

[7] T. Winter, P. Thubert, A. Brandt, J. Hui, R. Kelsey, P. Levis, K. S. Pister, R. Struik, J. Vasseur, and R. Alexander. RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks, March 2012.