2021 IEEE 27th Real-Time and Embedded Technology and Applications Symposium (RTAS) | 978-1-6654-0386-3/20/$31.00 ©2021 IEEE | DOI: 10.1109/RTAS52030.2021.00067

Demo Abstract: A Full-Blown 6TiSCH Network with Partition-based Resource Management for Large-Scale Real-Time Wireless Applications

Jiachen Wang*, Tianyu Zhang†, Song Han*, Xiaobo Sharon Hu‡
*University of Connecticut, Email: {jc.wang, song.han}@uconn.edu
†The Hong Kong Polytechnic University, Email: tianyu1.zhang@polyu.edu.hk
‡University of Notre Dame, Email: shu@nd.edu

I. Introduction

Industrial Internet of Things (IIoT) systems aim to interconnect a large number of heterogeneous industrial sensing and actuation devices through both wired and wireless communication technologies and further connect them to the Internet to achieve ubiquitous sensing, computing and control services [1]. As a representative IIoT technology, 6TiSCH [2] targets at gluing together the 802.15.4e data link layer (offering industrial performance in terms of timing, reliability and power consumption) and an IP-enabled upper layer stack to achieve both deterministic network performance and seamless integration with Internet services. In recent years, 6TiSCH has been receiving increasing attentions from both industry and academia. We have witnessed its wide deployment in many industrial domains, including advanced manufacturing, industrial process control, smart grids, and healthcare.

Although 6TiSCH has a promising future to become the de facto standard for the real-time wireless edge networks in the IIoT world, the current design and development efforts on 6TiSCH, especially on dynamic and scalable network resource management, are still preliminary. In this work, we present a comprehensive 6TiSCH implementation that addresses both the network scalability and adaptability issues through a novel partition-based network management framework, called A-PaS [3]. A-PaS employs the concept of resource partitioning and can guarantee the end-to-end packet transmission latency in multi-hop 6TiSCH networks, even in the presence of frequent network topology changes. In the following, we first present the overall 6TiSCH system architecture, and then lay out our demonstration plan to show how our proposed technologies contribute towards meeting the stringent performance requirements in 6TiSCH networks on system robustness, communication latency and reliability, and user-friendly network monitoring and diagnosis.

II. 6TiSCH System Architecture

Fig. 1 presents the overall architecture of our full-blown 6TiSCH network. It consists of the following key components:

6TiSCH end devices: A 6TiSCH end device is an embedded device equipped with sensors and/or actuators and running a 6TiSCH stack. Its layered structure is presented on the right side of Fig. 1. The stack runs on the TSCH (time slot channel hopping) mode of IEEE 802.15.4e to provide deterministic channel access with low-latency and reliable packet delivery. The 6LoWPAN layer provides protocol adaptation between standard IPv6 datagrams and 802.15.4e frames. In the application layer, a HTTP-like CoAP protocol [4] provides access to individual end devices from the Internet.

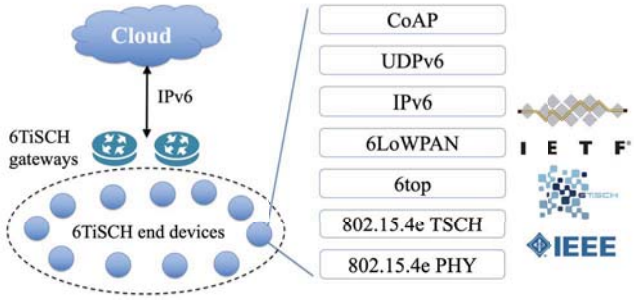

Fig. 1. Overview of the 6TiSCH system architecture.

6TiSCH gateway: The 6TiSCH gateway (an embedded Linux OS with a 6TiSCH Access Point) serves as the coordinator and the boarder router of a 6TiSCH network. To maintain a large-scale 6TiSCH network, we have developed a rich set of topology and schedule management functions on the gateway to achieve real-time, robust and self-adaptive network resource management, including the A-PaS framework.

Network management and visualization system: To provide real-time monitoring, analysis and visualization of 6TiSCH networks, we developed a comprehensive RESTful web service. Benefiting from this web service, end users can monitor the network performance and the current communication schedules in real time. It also allows the network engineers and researchers to export the network statistic data and perform various network analysis tasks.

III. Testbed Setup and Demonstration

In this section, we demonstrate the key features of the full-blown 6TiSCH network using our 122-node testbed.

Testbed setup: Our 6TiSCH testbed (see Fig. 2(a)) consists of 121 CC2650-based 6TiSCH end devices built on Ti-RTOS [5], and a Raspberry Pi 4B running 32-bit ARM Linux attached with a CC2652 board being configured as the gateway.

Demo 1: Network formation and operations. In this demo, we will show the joining process of 6TiSCH end devices to form the 122-node network and how the sensing/performance

978-1-6654-0386-3/21/$31.00 ©2021 IEEE
DOI 10.1109/RTAS52030.2021.00067

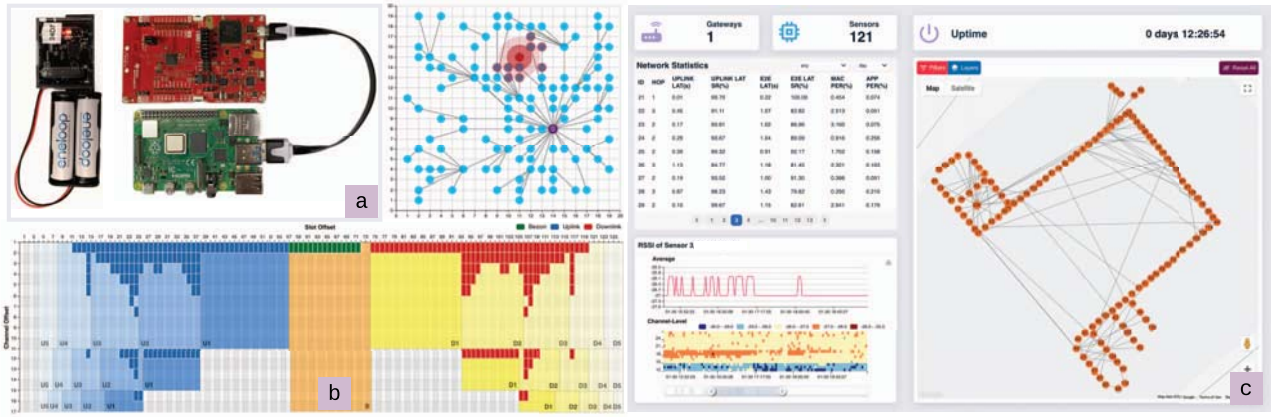

Fig. 2. (a) Hardware platforms for the 6TiSCH testbed, (b) online 6TiSCH network simulation tools, (c) cloud-based management system.

data are exchanged between the 6TiSCH gateway and end devices during the network operation. The gateway will be first powered up and broadcast Enhance Beacon (EB) messages. Each 6TiSCH end device will then be powered up and listen for the EB messages. Once the EB messages are received, they will associate with a designated or randomly picked parent node to join into the network. After the multi-hop 6TiSCH network is formed, the gateway will send CoAP requests to subscribe the sensing data from each device. The network manager module running on the gateway will also be configured to send topology control command and perform schedule update during the network operation.

Demo 2: End-to-end MAC layer latency measurement. In this demo, we will show how the end-to-end (e2e) MAC layer latency of a packet traversing through its routing path will be measured without modifying the 6TiSCH packet header. We will also present the key design principles of the A-PaS network resource management framework and show that benefiting from A-PaS, the e2e MAC layer latency of all the packets in the network (regardless of their number of hops towards the gateway) are bounded within 1 slotframe length.

Demo 3: Network performance with dynamic topology. In this demo, we will compare the performance of three link-based scheduling methods in 6TiSCH (A-PaS, LLSF [6] and Randomized Scheduler) in the presence of dynamic network topologies. We will randomly select to restart a set of end devices in the network, and force them and their child devices as well to rejoin the network by choosing new parents and thus change the network topology. We will illustrate the communication schedules before and after the network topology change and show how the network performance is affected. For fair comparison, we will make sure that the selected devices to be restarted and the resulted new network topologies are the same under all the three methods.

Demo 4: Cloud-based network management system. In this demo, we will show our cloud-based network management system for 6TiSCH network monitoring, analysis and visualization. As shown in Fig. 2(c), the key modules of the system include: 1) visualization of the 6TiSCH communication schedule and device/network performance data (*e.g.,* e2e latency, packet error rate, RSSI); 2) Google Map based network topology visualization, with the power usage information of individual devices and the constructed noise level heatmap in the deployed area; 3) display of network events with a timeline chart (*e.g.,* topology changes and device restart); and 4) data exporting function to support offline network analysis.

Demo 5: Online 6TiSCH network simulation tool. In this demo, we will show our 6TiSCH network simulator developed based on HTML5 and Vue.js. It is part of the cloud-based network management system, providing the users a flexible way to simulate a 6TiSCH network with arbitrary network topology. As shown in Fig. 2(b), this simulation tool supports: 1) multi-hop network topology generation following the RPL routing protocol [7] in 6TiSCH; 2) online communication schedule construction based on the given network topology and the specified scheduling framework; and 3) simulation of runtime events in the network, including device poweroff/restart and network topology changes.

IV. Acknowledgement

This research is partially supported by National Science Foundation under awards CCF-2028875 and CCF-2028879.

References

[1] E. Sisinni, A. Saifullah, S. Han, U. Jennehag, and M. Gidlund, "Industrial internet of things: Challenges, opportunities, and directions," *IEEE Trans. on Industrial Informatics*, vol. 14, no. 11, pp. 4724–4734, 2018.

[2] D. Dujovne, T. Watteyne, X. Vilajosana, and P. Thubert, "6tisch: deterministic ip-enabled industrial internet (of things)," *IEEE Communications Magazine*, vol. 52, no. 12, pp. 36–41, 2014.

[3] J. Wang, T. Zhang, D. Shen, X. S. Hu, and S. Han, "Apas: An adaptive partition-based scheduling framework for 6tisch networks," in *RTAS 2021*.

[4] C. Bormann, A. P. Castellani, and Z. Shelby, "Coap: An application protocol for billions of tiny internet nodes," *IEEE Internet Computing*, vol. 16, no. 2, pp. 62–67, 2012.

[5] Texas Instruments, *TI-RTOS 2.20 User's Guide*, 2016. [Online]. Available: http://www.ti.com/lit/ug/spruhd4m/spruhd4m.pdf

[6] T. Chang, T. Watteyne, Q. Wang, and X. Vilajosana, "LLSF: Low latency scheduling function for 6TiSCH networks," in *2016 International Conference on Distributed Computing in Sensor Systems (DCOSS)*. IEEE, 2016, pp. 93–95.

[7] T. Winter, P. Thubert, A. Brandt, J. W. Hui, R. Kelsey, P. Levis, K. Pister, R. Struik, J.-P. Vasseur, R. K. Alexander *et al.*, "Rpl: Ipv6 routing protocol for low-power and lossy networks." *rfc*, vol. 6550, pp. 1–157, 2012.