Design and Implementation of a Low-Cost Software Defined Wireless Network Testbed for Smart Home

Watipatsa W. Nsunza, Samuel Rutunda, and Xiaojun Hei(✉)

School of Electronic Information and Communications,
Huazhong University of Science and Technology, Wuhan 430074, China
{nsunza,rutunda,heixj}@hust.edu.cn

Abstract. The evolvable nature of software defined wireless networking offers great opportunities toward the design and implementation of a low-cost network testbed for smart home. Programmability is an essential component on a network gateway to enable efficient management of energy and other network resources for secure, scalable, and cost-effective solutions. In this paper, we proposed a software defined edge-cloud network architecture for smart home. We studied the programmable features of several popular SoC and FPGA platforms and design a software defined wireless network testbed for smart home by integrating several open-source projects including OpenWrt, Lede, and OpenFlow, which may be extended for other application scenarios such as smart grid and Internet-of-Things. We implemented WiFi, BLE, and ZigBee networking features on our low-cost FPGA and SoC platforms and evaluated the TCP and UDP throughput on our testbed. We conducted a series of experiments on our testbed and examined optimization issues based on recent developments in SDN. Our testbed may provide experiment supports for advancing smart home research and development.

Keywords: Smart home · Network testbed · Home area networks
Internet-of-Things · Software defined networking
Wireless networking · OpenFlow

1 Introduction

Smart home applications have been penetrating into our daily life in recent years. Gartner has predicted that over 25 billion IoT devices will be connected by 2020. A smart home is a communications network linking key electrical appliances and services accessible and monitored remotely. Smart homes can be classified into 2 categories: autonomous houses based on sensor-driven activation, or intelligent houses which can learn without human intervention. Smart homes utilize recent development in different domains such as smart grid, wearable IoT and a variety of sensors utilizing different protocols. Smart homes can also be centralized or decentralized, however, most systems have adapted a centralized approach

G. Wang et al. (Eds.): SpaCCS 2017 Workshops, LNCS 10658, pp. 379–388, 2017.
https://doi.org/10.1007/978-3-319-72395-2_35

where all devices are connected to a single gateway [1,2]. There are a few challenges in standardizing smart home protocols such as energy efficiency, security, and efficient management. In recent years, a huge number of WiFi networks have been deployed at homes as dominant broadband network access in an unplanned manner and compete for unlicensed bandwidth in same areas, which may lead to significant degradation of network performance [3,4]. The emerging software defined networking (SDN) has been proposed to support smart home research and development. As a new paradigm shift in networking, SDN separates the control plane from the data plane, offering network flexibility, introducing programmability and ease of management [5]. SDN is often combined with network function virtualization (NFV) to enable manageable and controllable networks. To meet the needs of smart home applications, SDN can unite different communication technologies, provide better security against external attacks, and offer better energy saving schemes [6].

In this paper, we design and implement a low-cost software defined wireless network testbed for smart homes based on multiple SoC development boards. The testbed platforms include the Digilent Zybo™ FPGA with a dual-core ARM®Cortex®-A9 processor, Intel®Galileo Gen 2 with an Intel®Quark™ SoC X1000 application processor, Raspberry Pi 1 Model B+ with the ARM11 CPU, and Raspberry Pi 3 Model B with a quad-core ARM®Cortex®-A53 CPU (see Table 1). These platforms have much higher processing speeds than conventional routers. We investigated the programmable features of these hardware platforms as a network testbed for smart home research to provide connectivity solutions for software defined IoT.

Table 1. Hardware specification

Platform[a]	Processor			Memory
	CPU	*Cores/Threads*	*Freq* (MHz)	*DRAM* (MB)
Galileo 2	Quark™ SoC X1000	1/1	400	256
Zybo	Cortex®-A9	2/2	650	512
RPi B+	ARM11	1/1	700	512
RPi 3B	Cortex®-A53	4/4	1200	1024

[a] The FPGA/SoC platforms augmented are priced roughly between 20 to 200 USD.

The remaining of this paper is organized as follows. First, we present a software-defined edge-cloud network architecture for smart home in Sect. 2. Next, we describe the design and implementation of our testbed to support the research of the proposed architecture in Sect. 3. In Sect. 4, we report the evaluation results of our testbed followed by discussing optimization issues. Then, in Sect. 5, we review some related work. Finally, we conclude this paper and outline some future work in Sect. 6.

2 A Software Defined Edge-Cloud Network Architecture

The essential components in building a software defined edge-cloud network architecture for smart home are depicted in Fig. 1. There are four key components in a smart home environment, including the network gateway, an SDN controller, and high bandwidth and low bandwidth communication channels to manage appliances with efficiency and security. The data paths to the communication interfaces between the gateway device and smart home network appliances or external networks are illustrated by grey arrows and wireless links to the the appliances and home devices are depicted by the bold dashed lines. The SDN control plane can receive status information from home appliances and send control messages to the appliances through the SmartWLAN and SmartWPAN access networks. The controller can also implement security schemes by extending the OpenFlow protocol to secure the data transmission.

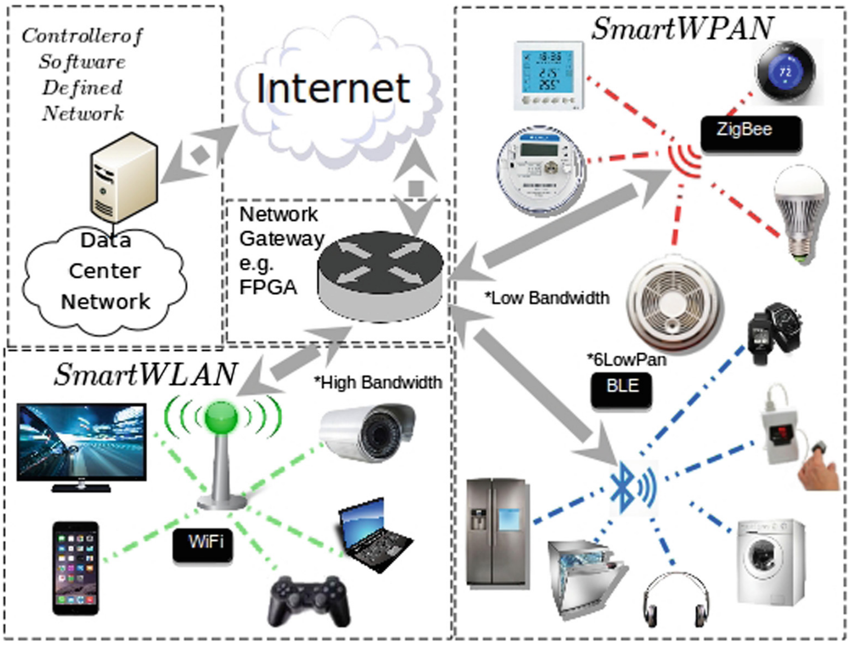


Fig. 1. A proposed software-defined edge-cloud network architecture.

2.1 Smart Home Gateway

We studied the programmable features of low-cost SoC platforms with both FPGA and non-FPGA architectures as network gateways for a smart home. The gateway device can be implemented on programmable SoC architectures with the latest Linux kernels. The proposed design does not incorporate wired links on the data path to home appliances though wired links are used for linking the gateway to the Internet. Wired communication links are much secure and provide high throughput; however, we consider the modern structure of smart homes and cabling costs of wired systems and integrated the latest wireless communication standards in this design. Wireless systems also have a low complexity during setup and configuration when compared with wired links which is preferred in Home Area Networking (HAN) scenarios. The proposed smart home gateway accommodates wireless communication networks much more easily by adapting wireless interfaces such as Bluetooth (IEEE 802.15), WiFi (IEEE 802.11) and BLE and ZigBee (IEEE 802.15.4).

2.2 SmartWLAN

WiFi has become the most dominant networking technology for implementing wireless local area networks "WLAN". WiFi technology is built on top of the IEEE 802.11 standard set of media access control (MAC) and physical (PHY) specifications that support applications in computers, smart phones, and other bandwidth sensitive networking devices. WiFi standards include IEEE 802.11a, 802.11b/g/n, and 802.11ac wireless communication standards which operate at the 900 MHz and 2.4, 3.6, 5, and 60 GHz communication frequency bands. "802.11ac" is the latest WiFi standard with dual band support. It supports multiple connections at once and operates at the 2.4 and 5 GHz WiFi frequency bands. 802.11ac is also backward compatible with 802.11b/g/n wireless devices and supports data bandwidth rates up to 1300 Mbps when operating at 5 GHz and 450 Mbps at a 2.4 GHz frequency.

2.3 SmartWPAN

BLE and ZigBee are designed for low data rate applications to efficiently conserve energy. This enables devices and sensors to operate for a number of years depending on the amount of activity and stability of the energy source. These low complexity wireless standards have specifications on both Layer 1 (PHY) and Layer 2 (MAC) and are highly adopted and anticipated solutions for connecting smart grid devices in IoT. The short range and restricted topology in BLE and ZigBee devices requires mesh and start networking protocols with multi-hop support to overcome the limitations. WiFi radios don't efficiently manage power. Using WiFi to manage appliances in the smart home network would exhaust battery powered home appliances much frequently. This has encouraged the development of power efficient home appliances with other wireless technologies. BLE and ZigBee provide sufficient throughput for smart home network communication in low bandwidth devices, which efficiently utilize energy to improve connectivity for smart home.

2.4 SDN Controller

As a new method for managing a smart home network, we propose a software defined edge-cloud architecture to efficiently manage network traffic from an extensible number of connected IoT devices. OpenFlow extends generic features of TCAM switches and routers and provides an open protocol for configuring different switches and flow tables on routers. While using OpenFlow to manage the network, researchers can distinguish between experimental streams and workflows to control their own experimental streams by selecting packets, routing lines, and handling received packets. This enables experimenting of routing protocols, security models, address scheduling, and select IP. The data path of the OpenFlow switch contains a flow table and an action corresponding to each flow entry. These operations are extensible to a subset of switches for a limited and useful set of operations. The OpenFlow switch matches each flow entry in the table to a corresponding operation with instructions on handling an incoming flow and provides a secure channel to link the switch to a remote controller and transmits control commands using the OpenFlow protocol.

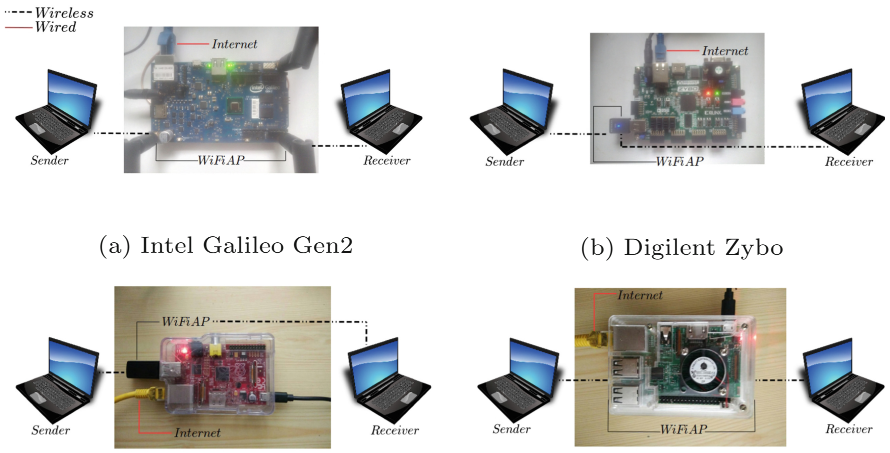
3 Testbed

3.1 Overview

In this section, we describe the design and implementation of a network testbed to support the research of the proposed edge-cloud architecture. In our testbed, the gateway is instrumented using different platforms and deploys the OpenFlow protocol to manage communication channels via a few interfaces. All the platforms followed an overall similar development structure, with different levels of difficulty and procedures for building our system firmware. We developed an "OpenWrt" and "Lede" Linux system with supporting drivers for the on-board input and output interfaces (i.e. Ethernet, USB, PCI expansion, etc.) specific to the hardware requirements on each platform. We implemented a WiFi access point based on each platforms hardware and software requirements. Our "Zybo" and "RPi 3" platforms have been implemented with the latest supporting Linux kernel, and are therefore capable of supporting 802.11ac WiFi, while other platforms only support 802.11n. We installed packages to support the OpenFlow protocol as well as QoS routing schemes which are currently under test for all our systems.

3.2 Firmware

The Intel®Galileo Gen 2 (Fig. 2a) system firmware is built based on the "Linux kernel 3.8.13" OpenWrt "Trunk" source code. The wireless access point (WAP) on this platform is built on an Atheros AR9380 PCI Card. Linux has supports 6LowPan for Bluetooth Low Energy "BLE" on kernel version 3.17 and above. We're still investigating alternative solutions for implementing ZigBee and BLE support on this platform. Digilent Zybo (Fig. 2b) supports the latest OpenWrt trunk "Linux kernel 4.4.14". This firmware supports OpenFlow and QoS, BLE, ZigBee, and WiFi devices via USB. The challenges with this platform are due to



(a) Intel Galileo Gen2

(b) Digilent Zybo

(c) Raspberry PI 1

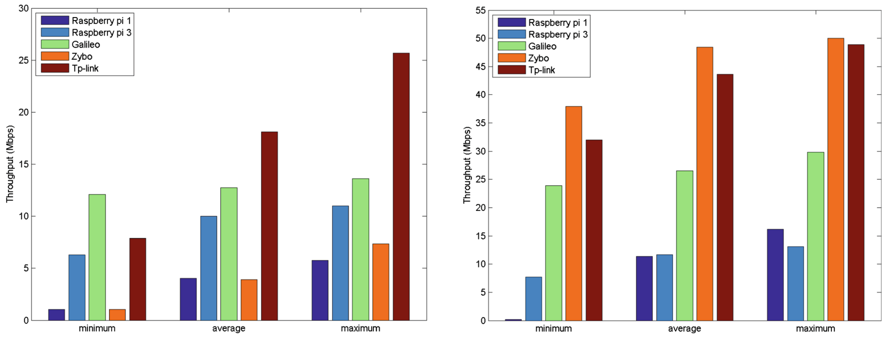
(d) Raspberry PI 3

Fig. 2. Testbed platforms.

the limited number of USB interfaces which limits the number of active connections to one communication technology at time based on our current implementation. To resolve this issue we have been looking into interfacing through other peripheral module inputs on the device. The Raspberry Pi 3 platform (Fig. 2d) offers an in-built 802.11n WiFi and Bluetooth 4.1 chipset. The Raspberry Pi 1 (Fig. 2c) however contains no built-in communication interfaces and requires a USB based dongle to support both Bluetooth and WiFi features. The firmware on the Raspberry Pi testbed has been developed on the OpenWrt Chaos Calmer source codes and the Raspbberry Pi 3 firmware is based on the Lede 17.01.0 source codes. All these firmwares also support the latest developments of OpenFlow, QoS, BLE, ZigBee, and WiFi.

4 Evaluation

We evaluated the testbeds under both TCP and UDP against the performance of a traditional TP-link router running on the vendor firmware using iPerf. With the datagram size set at 1470 bytes, a TCP window size of 416 Kbytes and the UDP buffer size at 208 Kbytes, we conducted 100+ experiments on each platform to summarize our measurement results. We observed that the Intel®Galileo Gen 2 achieved the highest TCP throughput when compared with other platforms, though this performance was still less than the average when compared to the TP-link router as shown in (Fig. 3a). The maximum TCP throughput for Galileo ranged at 13.6 Mbps while the TP-link router achieved a 25.7 Mbps TCP throughput. Our Digilent Zybo platform however achieved the highest throughput on UDP traffic even compared with the TP-link router, achieving a maximum throughput of 50 Mbps; a performance equivalent to the line rate (see Fig. 3b) and (Table 2). Zybo offers a decreased latency and also encountered the lowest UDP jitter ratio averaged at 0.224 ms on a minimum UDP buffer size, and caused by a single packet loss.



(a) TCP

(b) UDP

Fig. 3. Throughput performance.

Table 2. Performance summary

Platform	TCP max (Mbps)	UDP max (Mbps)	Latency avg. (ms)	Jitter avg. (ms)
Galileo 2	13.6	15.7	4.12	1.57
Zybo	7.34	50.0	0.04	0.22
RPi B+	5.77	16.2	6.82	1.63
RPi 3B	11.0	13.1	7.42	1272.2

Our measurement results show a very low performance of the software based packet switching module on our testbed. We conjecture that by utilizing the latest developments in SDN, we may optimize the TCP and UDP throughput. The performance tests for our testbed are still being analyzed as we continue the developments. The above tests demonstrate the most basic analysis. Due to the current progress, other parts of our testbed have yet to be evaluated. As an on-going research project, we have been conducting further experiments and analysis on our all the required parameters of our network testbed.

5 Related Work

Prior research on the smart home testbed has centered primarily on creating simulation environments for conducting experiments on energy saving. In [7], Perumal et al. reviewed machine-to-machine distributed home networks and describe the performance trade-offs which covers quality of service, energy efficiency, and security issues. In [8], Suh and Ko designed and implemented a multi-purpose smart house simulation system for designing and simulating all aspects of a smart house environment. This simulator provides the ability to design the house plan and different virtual sensors and appliances in a two dimensional model of a virtual house environment. This simulator can connect to any external smart house remote controlling system, extending the evaluation capabilities to the system. In [9], Louis proposed to improve the energy efficiency of smart buildings as an essential part of the smart grid system. The theoretical aspect of the paper introduced ideas for promoting energy efficiency in smart home while underlining that data safety and privacy are major concerns in the system. The practical aspect based on Matlab and Simulink modeled key aspects of a smart building including 10-year climate data, a lighting system, twenty-one appliances with different power rates, and variables including the number of inhabitants & bed-rooms and small-scale energy production systems including wind, solar, and fuel cell. In [10], Fensel et al. introduced a smart home system for energy efficiency. Their study emphasized that energy efficiency is an important element in smart home development due to the rising energy cost, which have created a growing need for energy saving systems and increased demands for energy saving solutions world-wide. Companies including Apple, Cisco, and Google have also introduced semantic energy saving solutions for homes on the market.

The energy-saving approach in [10] similar to one we're developing on our testbed demonstrated the efficiency of using non-semantic interface specific solutions such as a ZigBee communication interface approach for home controlling. Zigbee-like Bluetooth low energy devices are designed to be energy efficient. Integrating support interfaces for these technologies on our testbed promotes energy efficiency in our Smart Home. In [11], Gill et al. also presented a low-cost stand-alone ZigBee Smart Home automation system. The system adapted a low-complexity architecture to lower financial costs by eliminating complex and expensive hardware components. This architecture consisted of a home gateway for inter-operability between heterogeneous networks including ZigBee and WiFi for connecting to smart home devices from the Internet-enabled devices through serial and parallel interfaces on a Jennic JN5139 Micro-controller. The architecture was managed by a virtual environment responsible for administering security to the home automation system. This architecture however adapted traditional advanced encryption standard (AES) public/private key exchange methods in the virtual environment for securing the network devices. This may present intrusion threats to an IoT network including smart homes from reverse engineering with a microcontroller unit (MCU) debugger to obtain keys stored on the devices, which will affect all attached TCP/IP networks [12]. Once the session keys are compromised, security will be ineffective. Other testbeds in [8,13,14] also employed new routing schemes to optimize performance while maintaining traditional methodologies of managing the network devices and therefore face similar potential threats previously described. In [15] Tang et al. also proposed a simulation testbed for the Cyber-security experimentation on a smart home network based on traditional wireless network protocols.

In [1,2], Zahid et al. conducted a measurement-based empirical study on the design space of different OpenFlow switches in multiple scenarios of a smart home network. In that study, they focused on the throughput performance for different software-based OpenFlow switches based on ONetSwitch20, ONetSwitch45, and the NetGear WNDR3700v4. Their results demonstrated significant higher throughput with hardware-based switching than software-based switching. The design space was discussed for high-bandwidth WiFi devices while we are interested to investigate efficient schemes to manage energy and other network resources to increase the number of connected devices in smart home. Our study also covered several popular low-cost SoC architectures by extending our previous work [16].

6 Conclusion

In this paper, we proposed a software defined edge-cloud network architecture for smart home. We designed and implemented a network testbed based on popular SoC and FPGA platforms. We conducted comprehensive measurement experiments to evaluate the TCP and UDP throughput on our testbed. Our results demonstrate a very low performance of the software based packet switching module on our testbed. This has motivated us to conduct further experiments to track the performance bottleneck of our testbed. We plan to continue analyzing the performance of our switch implementation of the "WiFi", "BLE", and ZigBee modules. Our experiments will include energy consumption, network performance, bandwidth utilization and CPU load. We have been examining the prototype results with the emulation results and the theoretical results [17]. Such a comparison study may provide more insights into the design space of a software defined edge-cloud architecture for the emerging edge computing and networking applications. We also plan to instrument a representative SDN controller in a software/hardwared co-design approach [18] to manage both the data forwarding and the MAC-layer parameters to implement a software defined wireless network testbed for performance evaluation. The emerging software defined edge-cloud architecture is advantageous to achieve measurable, manageable and controllable high-density smart homes [19].

Acknowledgments. This work was supported in part by the National Natural Science Foundation of China (No. 61370231), and in part by the Fundamental Research Funds for the Central Universities (No. HUST:2016YXMS303).

References

1. Zahid, T., Dar, F.Y., Hei, X., Cheng, W.: An empirical study of the design space of smart home routers. In: Chang, C.K., Chiari, L., Cao, Y., Jin, H., Mokhtari, M., Aloulou, H. (eds.) ICOST 2016. LNCS, vol. 9677, pp. 109–120. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-39601-9_10

2. Zahid, T., Hei, X., Cheng, W.: Understanding the design space of a software defined WiFi network testbed. In: International Conference on Frontiers of Information Technology (FIT), pp. 170–175, December 2016
3. Zhang, C., Qiu, D., Mao, S., Hei, X., Cheng, W.: Characterizing interference in a campus WiFi network via mobile crowd sensing. In: Guo, S., Liao, X., Liu, F., Zhu, Y. (eds.) CollaborateCom 2015. LNICST, vol. 163, pp. 173–182. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-28910-6_16
4. Gao, Y., Dai, L., Hei, X.: Throughput optimization of multi-BSS IEEE 802.11 networks with universal frequency reuse. IEEE Trans. Commun. **65**(8), 3399–3414 (2017)
5. McKeown, N., Anderson, T., Balakrishnan, H., Parulkar, G., Peterson, L., Rexford, J., Shenker, S., Turner, J.: OpenFlow: enabling innovation in campus networks. SIGCOMM Comput. Commun. Rev. **8**(2), 69–74 (2008)
6. Shin, S., et al.: Enhancing network security through software defined networking (SDN). In: IEEE ICCCN (2016)
7. Perumal, T., Ramli, A.R., Leong, C.Y.: Interoperability framework for smart home systems. IEEE Trans. Consum. Electron. **57**(4) (2011)
8. Suh, C., Ko, Y.B.: Design and implementation of intelligent home control systems based on active sensor networks. IEEE Trans. Consum. Electron. **54**(3), 1177–1184 (2008)
9. Louis, J.N.: Smart buildings to improve energy efficiency in the residential sector. Ph.D. thesis, University of Oulu (2012)
10. Fensel, A., et al.: Sesame-S: Semantic smart home system for energy efficiency. Informatik-Spektrum **36**(1), 46–57 (2013)
11. Gill, K., Yang, S.H., Yao, F., Lu, X.: A ZigBee-based home automation system. IEEE Trans. Consum. Electron. **55**(2), 422–430 (2009)
12. Sood, K., Yu, S., Xiang, Y.: Software-defined wireless networking opportunities and challenges for Internet-of-Things: a review. IEEE Internet Things J. **3**(4), 453–463 (2016)
13. Han, D.M., Lim, J.H.: Design and implementation of smart home energy management systems based on ZigBee. IEEE Trans. Consum. Electron. **56**(3), 1417–1425 (2010)
14. Osiegbu, C., et al.: Design and implementation of an autonomous wireless sensor-based smart home. In: IEEE ICCCN (2015)
15. Tong, J., Sun, W., Wang, L.: A smart home network simulation testbed for cybersecurity experimentation. In: Leung, V.C.M., Chen, M., Wan, J., Zhang, Y. (eds.) TridentCom 2014. LNICST, vol. 137, pp. 136–145. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-13326-3_14
16. Nsunza, W.W., Hei, X.: Design and implementation of a smart home router based on Intel Galileo Gen 2. In: EAI TRIDENTCOM, December 2017
17. Chen, Z., Fu, D., Gao, Y., Hei, X.: Performance evaluation for WiFi DCF networks from theory to testbed. In: The 16th IEEE International Conference on Ubiquitous Computing and Communications (IUCC), December 2017
18. Kang, J., Hei, X., Song, J.: A comparative study of Zynq-based OpenFlow switches in a software/hardeware co-design. In: International Workshop on Network Optimization and Performance Evaluation (NOPE), December 2017
19. Chen, Z., Manzoor, S., Gao, Y., Hei, X.: Achieving load balancing in high-density software defined WiFi networks. In: International Conference on Frontiers of Information Technology (FIT), December 2017