

The Construction and Implementation of Wireless Network Online Learning System Based on Edge Computing

<https://doi.org/10.3991/ijet.v17i21.33029>

Jingmin Yang^{1,2}(✉), Trong-Yen Lee¹, Wen-Ta Lee¹, Li Xu²
¹Department of Electronic Engineering, National Taipei University of Technology, Taipei, Taiwan
²Key Laboratory of Network Security and Cryptology, Fujian Normal University, Fuzhou, China
yjm4201@163.com

Abstract—In the critical period of COVID-19 epidemic prevention, universities are required to stop all in-person classes. These issues have a huge impact on wireless network education. In addition, the traditional wireless network laboratory construction also has the problem of high hardware investment. To address these issues, we propose a novel wireless network online learning system (WNOLS) based on edge computing (EC). Firstly, we realize the simultaneous operation of multiple virtual network devices (VNDs) on the same physical device through EC and virtualization technology. Secondly, we design a unified experimental resource description method based on XML format, which is parsed by the parser and executed by the topology switcher to realize the rapid deployment of the experimental environment. An isolated design of in-band and out-of-band networks to avoid mutual interference between administrative and experimental data is proposed. Furthermore, we present a blended learning mode integrating online learning and project-based learning in the field of wireless network teaching. The practice results show that the WNOLS effectively alleviates the impact of the epidemic on the offline classes and improves the learning efficiency of learners. Besides, compared with the traditional approach, it reduces the total fixed investment cost by 30.25%.

Keywords—edge computing, network device virtualization, project-based learning, self-directed online learning, wireless network online learning system

1 Introduction

Wireless local area network (WLAN) refers to a wireless distribution method for two or more devices that transmit and receive data by using high-frequency radio waves. Compared with the wired network, WLAN has several obvious advantages such as flexible usage, easy expansion, low investment, and simple installation [1]. Therefore, it is widely used in smart cities, smart campuses, and other scenarios [2]. Following this trend, many colleges and universities take the WLAN technology and application

course as a basic compulsory course for computer science-related majors. This course is a multidisciplinary engineering discipline, which is characterized by the close combination of computer technology and communication technology, software and hardware, theory and practice, and requires a lot of experimental practice.

Engineering practice teaching plays an important role in higher engineering education. It is the main way to cultivate students' engineering ability, and also an important way to realize the combination of theory and practice [3]. In this regard, education experts have done a lot of practice and achieved notable results [4–7]. In practice, we find that how to overcome simple knowledge-based learning, promote the reorganization of students' cognitive structure, and guide students to apply knowledge and principles to actual situations is a key issue.

The Corona Virus Disease 2019 (COVID-19), which broke out at the end of 2019 and spread rapidly around the world, has made a huge impact on education, especially on practice teaching [8]. Many outstanding scholars have studied how to carry out high-quality teaching in the context of the epidemic, and have produced many excellent results. For example, Unung et al. discuss online learning to deal with the COVID-19 outbreak [9]. Wahidaturrahmi et al. point out that online learning will not only be an option for the pandemic situation but will also become one of the main delivery methods in the era of technology development [10]. Muhammad and Srinivasan present an effective approach of transition from in-class to online lectures in a short time [11]. K. Y. et al. recommend that under the background of the COVID-19, face-to-screen lectures together with the recorded lesson are the most appropriate ways of online learning [12]. Berga et al. conduct a comparative study of blended learning and face-to-face learning and point out that blended learning has the potential to foster innovative and flexible learning opportunities [13]. Ginting et al. believe that blended learning could create a more pleasant atmosphere for students and teachers [14]. Srinivasan et al. indicate that compared with the entirely virtual form of online learning, students prefer the blended learning with a large number of supplementary learning materials. They present four key pillars for a flexible blended learning format [15]. However, some studies also indicate that online learning and blended learning also have some inevitable challenges, such as lack of infrastructure, financial and technical resources, unstable networks, and student saturation [14][15].

In recent years, most of the wireless network teaching is based on simulation platforms, such as network simulator (NS-2), optimized network engineering tool (OPNET), quality network engineering tool (QualNet), labview, and packet tracer [16][17][18]. There are two main disadvantages of using software simulation platform. One is that some special properties of wireless networks, such as WLAN roaming, are difficult to be simulated by software. The other is that compared with the actual equipment operation, students gain less engineering experience through simulators. Therefore, more and more laboratories have purchased a large number of physical equipment for experiments. In practice, this method has obvious problems: high hardware investment [19][20]. In the field of Internet of Things (IoT) laboratory construction, in order to address these issues, Mostafa M. Soliman and Ishwar Singh propose a low cost and easy to use IoT laboratory by exploiting widely available software and cloud platforms [21]. Nevertheless, in the field of wireless network construction, it is rarely studied in previous literature.

Inspired by the above remarkable research results, we propose a novel WNOLS based on EC to solve the above issues. The main contributions of our work are as follows.

New blended learning mode. WNOLS realizes the combination of online learning and project-based learning (PBL) in the field of wireless network teaching. Specifically, the project-based case teaching is based on the actual operation of industrial-grade network equipment. The selected engineering cases come from the improved actual cases of the enterprise, which are realistic and pertinent. Furthermore, WNOLS provides reference teaching cases and supplementary learning materials to help learners carry out self-directed online learning. This blended learning mode well compiles with the government-issued social distancing norms, as well as improves the learners' engineering practice skills.

Cost-effective. The edge computing (EC) brings the computing closer to the network, while network device virtualization (NDV) virtualizes a single physical device into multiple logically independent devices. Adopting these two technologies, WNOLS allows multiple users to operate their own virtual devices simultaneously without mutual interference, which improves the equipment utilization rate and reduces the equipment procurement costs.

Time-effective. WNOLS proposes a new pattern based on extensible markup language (XML) to uniformly deploy the laboratory resources required in one experimental teaching case. This pattern can quickly deploy the experimental environment, shorten the experimental deployment time, and thereby improve the experimental efficiency.

The rest of this paper is organized as follows: Section 2 introduces the system architecture design of the WNOLS. Section 3 presents the design of key modules of WNOLS in detail. Section 4 illustrates the operation of the system through specific experimental case, and analyzes the key evaluation indicators. Finally, conclusions and suggestions for future works are discussed in Section 5.

2 System architecture design

This section mainly includes three parts: system design principles, key evaluation indicators and system architecture design. The principles of system design define the user requirements that need to be met and the boundary of the system. The key evaluation indicators are used to evaluate the performance of the system. The system architecture design mainly presents the key components in the physical layer, the edge computing layer and the application layer and the relationship among these components.

2.1 Technical background

EC refers to enabling technologies that allow computation to be performed at the edge of the network, with downstream data on behalf of cloud services and upstream data on behalf of IoT services [22]. EC conducts the computing at the proximity of data sources. Compared with the traditional cloud-based computing paradigm, EC improves the capacities of computing and storage at the network edge, decreases the

energy consumption, reduces network operation and service delivery delay, and further enhances the Quality of Experience (QoE) of users. In recent years, EC has become a research and application hotspot that attracts widespread attention, and has produced many meaningful research results. For example, Fang J et al. propose an efficient computation offloading strategy with mobile edge computing for IoT [23]. Pham K.V et al. design an asymmetrical training scheme of binary-memristor-crossbar-based neural networks for energy-efficient edge-computing nanoscale systems [24].

Current computing platforms encourage the integration of processing cores and their interconnections into a single chip. More and more network devices, such as routers and switches, use Multi-Core System-on-Chips (SoCs) to exploit computing power and parallelism to meet the dynamic workload requirements [25]. In addition, Network Device Virtualization (NDV) is the act of creating multiple Virtual Network Devices (VNDs). These VNDs have the same functions as physical devices and concurrently run on the same physical network device. Applications executing on different VNDs are independent of each other, although these virtual machines share the same physical resources. Objectively, the NDV technology has obvious key benefits, including higher efficiency, lower cost, faster deployment, and higher availability. It is a challenging and meaningful work to develop VND functions on network devices with Multi-Core SoCs.

XML is a markup language that defines a set of rules for encoding, storing, and transmitting data in human-readable and machine-readable formats. In the experimental case design phase, the experimental resources required for each experiment are defined in XML. In the experimental environment initialization phase, the user logs into WNOLS and selects the experimental case. WNOLS extracts the information about the experimental resources contained in the XML and deploys physical devices accordingly. Through this automatic deployment method, the rapid deployment of experimental resources is realized.

2.2 Design principles

On the basis network engineering experience and teaching practice accumulated for many years, this research puts forward the following design principles. The first basic design principle is to combine the experimental contents with the wireless network engineering practice and the key principles of wireless technology. To realize this design principle, the modular design method is a good choice. The experimental content based on modular design is convenient for dynamic adjustment according to experimental needs. Besides, the following principles should be followed and implemented in the system design.

Cost-effective. The hardware resources of system should have the ability to be reused to meet the needs of multiple people conducting experiments at the same time. The main purpose of this design principle is to improve the utilization of hardware resources, including switches, routers, wireless controllers, etc., while reduce the Equipment Procurement Costs (EPC). Since EPC varies with the price of equipment, it is more objective to use the Number of Concurrent Experiments supported per Unit Hardware Costs (NCEUHC) indicator to measure cost-effectiveness. To facilitate the measurement and the comparison effects, the NCEUHC of traditional equipment scheme is quantified as 1, and used as the comparison benchmark with the WNOLS scheme.

Time-effective. The experimental environment should be deployed rapidly to save time. According to the historical statistics we collected, the standard wireless network experiment course is 45 minutes, and the Average Deployment Time of Experimental Environment (ADTEE) with traditional equipment scheme is 18 minutes, accounting for 40%. The time is mainly used for equipment movement, equipment topology connection, equipment initialization configuration, etc. As a result, it is usually impossible for students to complete the experiment on time, and the Timely Completion Rate of Experiments (TCRE) is about 60%. If the time occupied by the above activities can be shortened, the TCRE indicator and the quality of experimental teaching can be greatly improved.

Scalability. The new system should be able to meet the needs of subsequent expansion. These expansion requirements mainly include adding physical devices and expanding system function modules. At present, there are 30 students in one wireless network experiment class in the school of computer science, and a class has a maximum of 50 students. Based on an estimate of 30% expansion in system capacity, the newly designed system needs to support 65 students doing experiments simultaneously.

Independent personal learning space. Students should have an individual learning space to support individualized teaching and autonomous learning. Individual learning space stores student’s learning materials and records the student’s learning process data, such as the experiment time, wrong experiment steps, experimental results, and student’s feedback. These data can be used for teaching analysis to improve the teaching quality.

2.3 Key evaluation indicators

A reasonable system architecture design is the key to realize user needs. There are many aspects to be considered during the system design phase, and the importance of each consideration should reflect the goals of the system and the expectations of users. To design a reasonable system, first of all, it is necessary to formulate the principles and key evaluation indicators to direct the system design. Based on the historical data collected and laboratory management standards, the following three key indicators are formulated: the NCEUHC, the ADTEE, and the TCRE. Table 1 shows the detailed description of these three indicators. Secondly, under the guidance of design principles, the overall framework of the WNOLS is divided into three layers based on the hierarchical design approach. There are physical layer, edge computing layer, and application layer. Each layer contains corresponding software and hardware components. Finally, the key technologies involved in each layer and the logical relationship among these components of each layer are defined. The following sections discuss the above content in detail.

Table 1. Key evaluation indicators of the WNOLS

Index	Indicator	Current Value	Target Value	Description
1	NCEUHC	1	≥ 2	$\frac{\text{number of concurrent experiments}}{\text{hardware cost}}$
2	ADTEE	18	≤ 5	$\frac{\text{deployment time for each experiment}}{\text{total number of experiments}}$
3	TCRE	60%	$\geq 90\%$	$\frac{\text{number of experiments completed}}{\text{total number of experiments}}$

2.4 Architecture design

Based on the above design principles, this study proposes the newly designed system WNOLS that integrates hardware components and software components. In terms of the system architecture, it consists of three layers, namely, the physical layer, the edge computing layer, and the application layer. Figure 1 illustrates the hierarchical model of WNOLS.

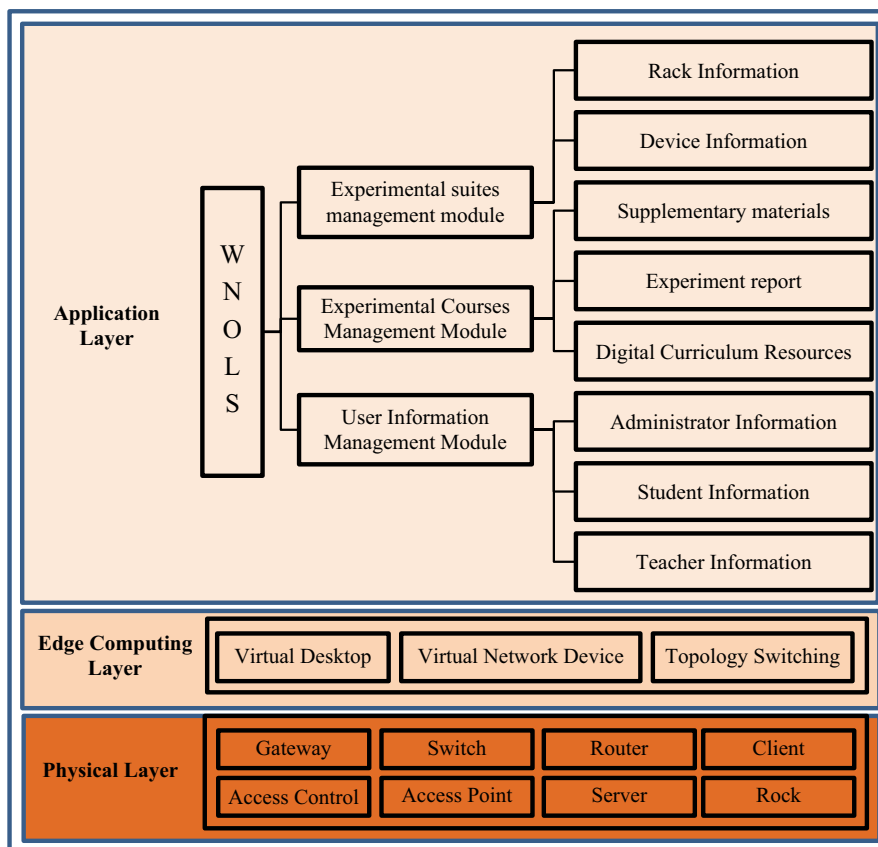


Fig. 1. System architecture of WNOLS

Physical layer. The physical layer consists of physical experimental instruments and network devices, including communication gateway, topology controller, switches, routers, wireless controllers, wireless access points, application server, and client computers. Devices are uniformly installed on racks. These racks serve as experimental management units. The experimental suites management module at the application layer uses the XML to describe the experimental configuration, and then sends XML instructions to the communication gateway. Each experimental case has an independent

initialization environment configuration based on XML. XML instructions contain the device parameters required for the experiment, such as the type and number of network devices needed in one experiment case, topology connections between devices, and selected routing protocols. The communication gateway receives and parses the instructions, and then initializes the experimental environment according to the parsed content. When the initial deployment of the experimental environment is completed, users can conduct experiments.

Edge computing layer. The edge computing layer mainly creates computational logic components for experiments. On the network side, NDV technology creates VNDs used for experimentation. On the user side, virtual machine technology creates client virtual machines (CVMs) that run the application software. They set up the experimental environment together. Notably, due to the technology used in this research, the type of physical device determines the type of the virtual device. For example, if the physical device is a switch, only virtual switches of the same model can be created. This study explains this technical feature in detail in Section 3. Furthermore, VMware creates different virtual operating systems on the same X86 architecture server. These operating systems include the most commonly used systems, such as windows 7, windows 10, Linux, etc. Users can run software on these operating systems to implement wireless network experiments.

In addition, the topology controller on the network side, called Topology Switching Device (TSD) in this study, is also an important component of the edge computing layer. According to the network topology parameters transmitted from the communication gateway, TSD interconnects the ports of the experimental devices by controlling the switches of the connection ports to constitute the experimental topology. It should be noted that the technology used by TSD is Ethernet switching technology rather than virtualization technology. Section 3 introduces the technical implementation of TSD in detail.

Application layer. The application layer is the closest to the end-user and consists of the Experimental Suites Management Module (ESMM), the Experimental Courses Management Module (ECMM), and the User Information Management Module (UIMM). The main functions of ESMM typically include registering and deleting racks, adding, and removing the rack-specific experimental equipment, and setting parameters of racks and equipment. ECMM provides different functions for teachers and students. The functions provided for teachers mainly include setting curriculum resources and arrangements, managing experimental classes, and reviewing students' experimental reports. While the functions provided for students mainly include selecting experimental courses, conducting experiments, and recording experimental reports. The main functions of the UIMM include the permission settings for administrators, teachers and students, and the maintenance of the user information.

3 Key modules design

This section describes the technical solutions adopted by the key modules of the WNOLS.

3.1 Network device virtualization

Virtualization principles. Each VND created by NDV possesses its application software process, kernel process, hardware switching chip, and physical ports for data forwarding. From the perspective of the user’s operating experience, VND is no different from an independent physical device. To achieve this effect, specifically, the operating system divides its software space into two parts: process space and kernel space. The software process cluster of VND runs in the process space. Different VND software process clusters run in different user spaces and are independent of each other. A significant advantage of this method is that when one VND fails, it can be restarted via software reset without affecting other VNDs. Each VND is distinguished by an integer number, starting from 0. When VND-0 is created, VND-M is also created in the process space of VND-0 at the same time. VND-M is the server process that VND0 uses to communicate with other VNDs. When another VND is created, a communicate client process, named VND-A, is created in this VND’s process space to communicate with VND-M. Two important management operations for VND include creation and deletion. VND-0 is called root-VND, and other VNDs created by VND-0 are called client VND. Figure 2 shows the operating principles of VND.

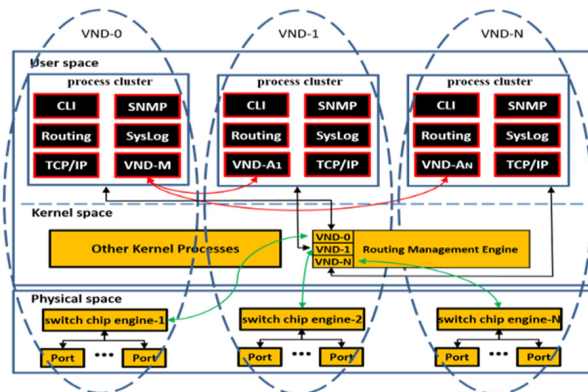


Fig. 2. The operating principle of VND

VND management plane. The application software process of VND is a cluster of running programs. Each application creates an instance in each VND, and its running space is also in the process space of this VND. In other words, instances of the same application software in different VNDs are unrelated and isolated at the process level. When a VND is shutdown, application instances running in this VND are also revoked. Typically, these application instances are divided into three main categories: communication management programs, routing protocols, and TCP/IP stacks. The communication management programs mainly include Command-Line Interface (CLI), Telnet, Syslog and Simple Network Management Protocol (SNMP); whereas routing protocols mainly refer to OSPF, RIP, etc., as well as the TCP/IP stacks hosting these applications.

Application processes running in the VND process space use Inter Process Communication (IPC) for communication. The IPC needs to know the address where processes communicate with each other. For example, UNIX domain sockets communicate in a way that requires knowledge of the file path to which the peer process is bound. When WNOLS

creates a VND, it creates a separate file system space and working directory for each VND simultaneously as the communication address of the process in the VND.

VND control plane. The network routing process of control plane, which runs in the kernel space of VND, is the main component of the routing architecture. This process involves computing the network topology information in the routing table that defines the packet processing strategy. The routing information calculated by different routing protocols in the management plane is passed to the routing process in the control plane simultaneously. The network routing process performs the final route selection according to the configured policy, and then forms a routing and forwarding table, which is transmitted to the data plane and installed on the forwarding chip. The forwarding chip processes incoming packets at high speed according to this routing and forwarding table. The normal operation of the routing mechanism requires multiple modules in the control plane and the forwarding plane to work cooperatively. Besides, the kernel process of operating system controls other core resources such as Central Processing Unit (CPU) and memory. Therefore, to reduce the implementation difficulty and failure rate, a memory-block-independent mechanism is adopted in the implementation of the control plane. Specifically, the routing process in the control plane allocates an independent memory block identified by an integer to each VND for storing routing and forwarding parameters of different VNDs.

VND data plane. In the routing mechanism, the data plane defines the part of the router architecture that implements the packets processing strategy. The main physical components of the data plane include switching chip engines and physical ports controlled by these chips. By default, the entire physical device is logically set to VND-0. When creating a new VND, the system allocates the required resources from the hardware resource pool, including CPU, memory, file system, switching chip engines, and physical ports. To facilitate the allocation of hardware resources and solve the problem of data interference between different VNDs, an allocation scheme with a switching engine as one allocation unit is adopted. This scheme allocates a set of switching chip engines and corresponding physical ports controlled by these chip engines to one VND. In other words, switching chip engines and the corresponding driven physical ports are not shared and cannot be used by other VNDs after allocation. Therefore, the number of VNDs that can create on a physical device depends on the number of switching chip engines supported by this device. The device model of the VND that can be created is also determined by the hardware model of the physical device.

Data separation. An inevitable problem in remote experiment is that the changes of routing protocol or the changes of topology often occur during the experiment, resulting in the interruption of the routing protocol. This results a disruption of the management communication between the user and the experimental system. To solve this problem, this study proposes an architecture design that completely isolates the management data from the experimental data. The communication gateway constructs two independent transmission networks, one is the management command transmission network, and the other is the experimental data transmission network. In the management command transmission network, the high-density asynchronous network modules on the communication gateway are interconnected with the console ports of each controlled network device on the rack through the asynchronous transmission cable. This communication is performed by the out-band communication module of the network device. While in the experimental data transmission network, all the in-band transmission ports of

network devices are interconnected with the topology switching device. In the experimental environment deployment phase, the system sends the device interconnection instruction to the topology switching device for parsing. After parsing this instruction, the topology switching device assigns ports of the network devices that need to be interconnected to the same logical link communication group to form an experimental network. The topology switching device is a switch with high-density ports. When the physical ports of network devices need to be interconnected, the switching device maps these ports to the same Virtual Local Area Network (VLAN). That is, the connectivity between devices can be controlled by controlling VLANs. In this way, the management data is transmitted over out-of-band management links, while experimental data is transmitted over in-band links. Both are independent of each other. Figure 3 shows the overall design framework for data separation, and Figure 4 defines the detailed topology design between devices.

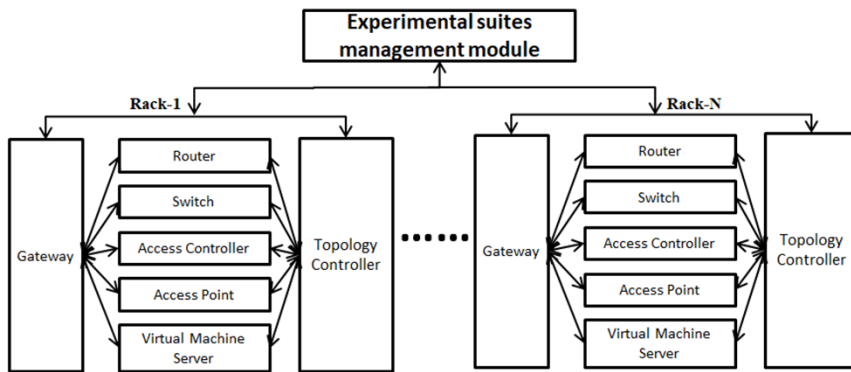


Fig. 3. The overall design framework for data separation

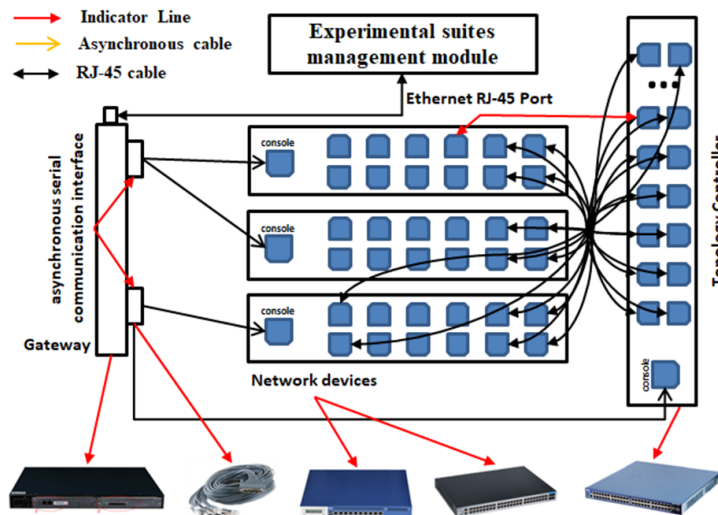


Fig. 4. The detailed design for data separation

Formal description. Manually deploying the network environment is a cumbersome, time-consuming, and error-prone process. To address this issue, this study designs a format method to describe the experimental network uniformly, which can be automatically parsed and executed by the parser. This section focuses on this XML-based approach to formatting network deployment, which allows users to define their own hierarchies of labels and information, with the advantages of simplicity, generality, and usability.

The XML-based experimental deployment tree description file is divided into 4 layers, as shown in Figure 5. The first layer is the root element, indicating that it is an XML file for wireless experiment deployment. The second layer defines the identifier, title, the target of the experiment, and the storage address link of the experimental documents. The third layer defines the types of network device required for the experiment. Current network devices include routers, switches, ACs, APs, and client virtual machines. The fourth layer defines the interface type, interface number, interface description, interface address, and routing protocol parameters of the specific device. The specific definition of each layer is as follows.

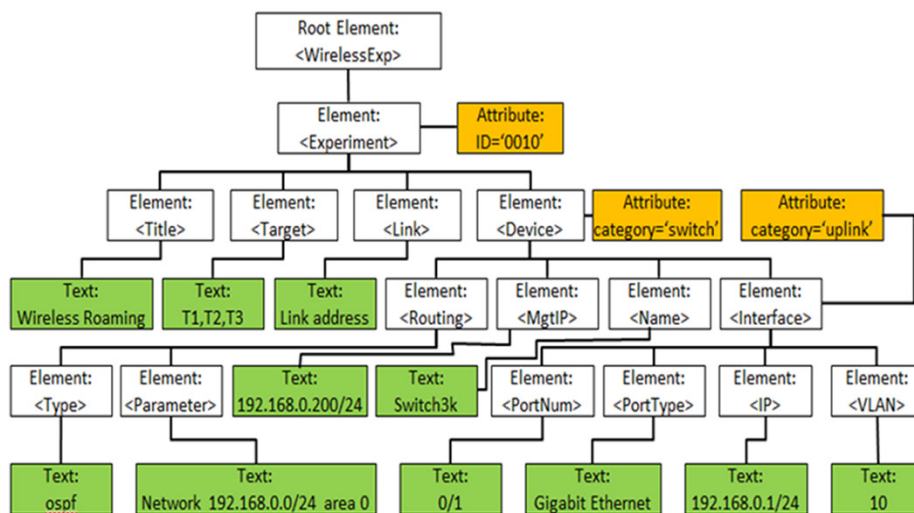


Fig. 5. XML-based environment deployment tree structure

The first layer is the root of XML-based environment deployment tree. The XML format is defined as follows:

```
<WirelessExp>...</WirelessExp> <!--root element-->
```

The second layer defines the XML elements and attributes of experiment, as shown in Table 2.

Table 2. Definition of element experiment’s markup

Markup	Character	Meaning
ID	Attribute	Experiment’s ID
Title	Child Element	Experiment’s description
Target	Child Element	Experiment’s target
Link	Child Element	Experimental data link
Device	Child Element	Experimental Device

The markup ‘<Device>...</Device>’ in Table 2 is in the third layer of XML-based environment deployment tree, which defines the parameters of experimental devices. The Table 3 describes the definition of element Device’s Markup.

Table 3. Definition of element device’s markup

Markup	Character	Meaning
category	Attribute	Device’s type
<Name> </Name>	Child Element	Device’s description
<MgtIP> </MgtIP>	Child Element	Device’s IP address for Management
<Routing> </Routing>	Child Element	Device’s routing information for data transmission
<Interface> </Interface>	Child Element	Device’s Interfaces

The markup ‘<Routing>...</Routing>’ and ‘<Interface>...</Interface>’ in Table 3 are in the fourth layer of XML-based environment deployment tree, which defines the parameters of routing and interfaces of experimental devices. The markup definitions for elements ‘<Routing>...</Routing>’ and ‘<Interface>...</Interface>’ are shown in Tables 4 and 5, respectively.

Table 4. Definition of element routing’s markup

Markup	Character	Meaning
<Type> </Type>	Child Element	Routing protocol’s description
<Parameter> </Parameter>	Child Element	Routing protocol’s parameter

Table 5. Definition of element interface’s markup

Markup	Character	Meaning
catagory	Attribute	Device’s type
<PortName> </PortName>	Child Element	Interface’s description
<PortType> </PortType>	Child Element	Type of interface, such as Gigabit Optical Fiber Port
<IP></IP>	Child Element	Interface’s IP address
<Vlan></Vlan>	Child Element	The vlan to which the port belongs

4 Experiment case

This section takes the “Case_007: Wireless Network Roaming within AC” experiment as an example to introduce the main process of WNOLS. In addition, the key evaluation indicators of WNOLS listed in Table 1 are analyzed based on the data collected from the project implementation and teaching process.

In the context of IEEE 802.11 wireless network, WLAN roaming refers to the process by which wireless clients move from one access point to another in a Wi-Fi wireless network without losing connectivity. Roaming is an extremely important feature of Wi-Fi networks because it guarantees the seamless mobility. However, due to its protocol complexity and device dependence, it is more difficult to simulate with a simulator.

The target of this experiment is to enable students to understand the basic working principles and application scenarios of Wi-Fi roaming, and to master the basic configuration of the device. Figure 6 illustrates the experimental topology, and Table 6 shows the experimental devices needed.

4.1 Experimental procedure

1. Visit the WNOLS website, enter the student’s username and password to login the student’s learning space, and then view the assigned experimental courses. Here, select the “Case_007: Wireless Network Roaming in AC” experiment.
2. Select the rack preassigned by the teacher, as shown in Figure 7(a).
3. In the initial experimental environment, the experimental e-book is on the left side of the page and experimental devices are on the right side. First, click the device figure and then click the “Select Device” button to enter the device’s command line configuration interface, as shown in Figure 7(b).
4. During the experiment, if students want to change the topology, first click one port with the mouse, then click another port that needs to be connected. At this point, a line is created between these two ports, and the topology switching device connects the two ports physically. When students want to unlink these two ports, just only double click this line to delete it.

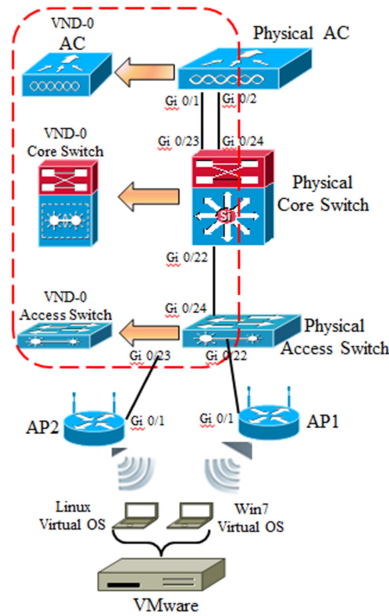


Fig. 6. Topology of wireless network roaming

Table 6. Experimental devices

Device Type	Number	Description
AC	1	Controls the APs and wireless roaming, can create VND
Core switch	1	Responsible for network routing, can create VND
Access Switch	1	Interconnect the core switch and APs
AP	2	Responsible for wireless client connection
Client operating system	2	Operating system running wireless application software, created by VMware
Wireless interface card	2	Plug in the wireless user's virtual machine for data sending and receiving

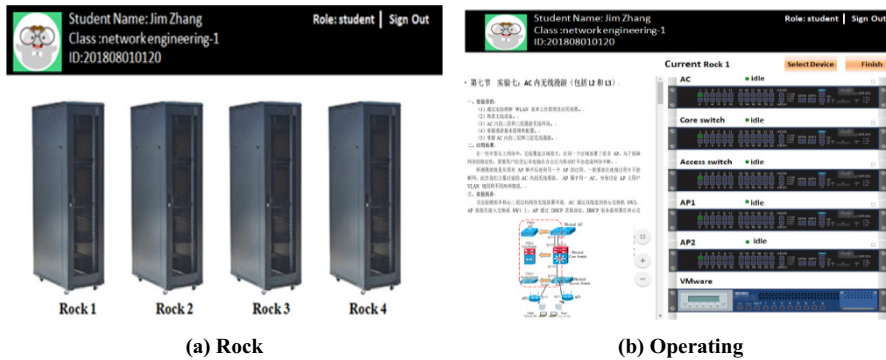


Fig. 7. User interface of WNOLS. (a) Racks; (b) Experimental operation interface

4.2 Laboratory environment

The Figure 8 shows the entire teaching environment, including (a) laboratory environment, (b) physical racks, (c) physical devices, (d) servers, (e) teaching process, and (f) group discussion.



Fig. 8. Holistic teaching environment. (a) Laboratory; (b) Physical racks; (c) Physical devices; (d) Servers; (e) Teaching process; (f) Group discussion

4.3 Key codes

This section shows the key codes of this experiment, including the configuration on the core switch, AC, and access switch, as shown in Table 7.

Table 7. Key codes of the “Case_007: wireless network roaming within AC” experiment

Step1: Core Switch Configuration	
Configuration	Description
SW2(config)#interface vlan 10	Create Switch Virtual Interface (SVI), which acts as the gateway’s IP address of AP
SW2(config-if-VLAN10)#ip address 172.16.10.254 255.255.255.0	
SW2(config)#interface vlan 20	Create SVI, which acts as the gateway’s IP address of users
SW2(config-if-VLAN 20)#ip address 172.16.20.254 255.255.255.0	
SW2(config)#interface vlan 100	Create SVI, which acts as the gateway’s IP address of access switch (option)
SW2(config-if-VLAN 100)#ip address 172.16.100.254 255.255.255.0	
SW2(config)#interface gigabitEthernet 0/1	Configure the interconnect interface with the access switch as trunk mode
SW2(config-if-GigabitEthernet0/1)#switchport mode trunk	
SW2(config)#interface gigabitEthernet 0/23	Configure the interconnect interface with the AC as routed-port mode
SW2(config-if-GigabitEthernet 0/23)#no switchport	
SW2(config-if-GigabitEthernet 0/23)#ip address 172.16.200.1 255.255.255.252	
SW2(config)#service dhcp	Configure the DHCP service, including two address pools for APs and wireless users.
SW2(config)#ip dhcp pool ap	
SW2(dhcp-config)#network 172.16.10.0 255.255.255.0	
SW2(dhcp-config)#default-router 172.16.10.254	
SW2(dhcp-config)#option 138 ip 1.1.1.1	
SW2(config)#ip dhcp pool users	
SW2(dhcp-config)#network 172.16.20.0 255.255.255.0	
SW2(dhcp-config)#default-router 172.16.20.254	
SW2(config)#ip route 1.1.1.1 255.255.255.255 172.16.200.2	Configure routing information
Step 2: Access Switch Configuration	
SW1(config)#vlan 10	Create an AP VLAN and the working mode of the interface.
SW1(config-vlan)#vlan 100	
SW1(config)#interface gigabitEthernet 0/1	
SW1(config-if-GigabitEthernet 0/1)#switchport access vlan 10	
SW1(config)#interface gigabitEthernet 0/2	
SW1(config-if-GigabitEthernet 0/2)#switchport access vlan 10	
SW1(config)#interface gigabitEthernet 0/24	
SW1(config-if-GigabitEthernet 0/24)#switchport mode trunk	

(Continued)

Table 7. Key codes of the “Case_007: wireless network roaming within AC” experiment (Continued)

Step 3: AC Configuration	
Configuration	Description
AC(config)#interface loopback 0	Configure the IP address of AC’s interface loopback 0. This IP address is the same as the address of the option 138 field in the DHCP address pool. After the AP obtains its IP address by DHCP, it actively establishes a tunnel with the IP address of AC’s interface loopback 0.
AC(config-if-Loopback 0)#ip address 1.1.1.1 255.255.255.255	
AC(config)#interface gigabitEthernet 0/1	Configure two types of ports on the AC, one interconnects with the core switch and the other interconnects with the wireless user.
AC(config-if-GigabitEthernet 0/1)#no switchport	
AC(config-if-GigabitEthernet 0/1)#ip address 172.16.200.2 255.255.255.252	
AC(config)#interface gigabitEthernet 0/2	
AC(config-if-GigabitEthernet 0/2)#switchport mode trunk	
AC(config)#ip route 0.0.0.0 0.0.0.0 172.16.200.1	
AC(config)#wlan-config 1 test wifi-test	Configure WLAN
AC(config-wlan)#enable-broad-ssid	Create user VLAN
AC(config)#vlan 20	
AC(config)#interface vlan 20	Create an AP group, associate WLAN and VLAN
AC(config)#ap-group ap-group	
AC(config-ap-group)#interface-mapping 1 20	
AC(config)#ap-config ap	
AC(config-ap)#ap-name ap1	
AC(config)#ap-config 1414.4b6d.1069	
AC(config-ap)#ap-name ap2	
AC(config)#ap-config ap1	
AC(config-ap)#ap-group ap-group	
AC(config)#ap-config ap2	
AC(config-ap)#ap-group ap-group	

4.4 Analysis of system key indicators

Due to the limited teaching funding, fixed investment cost is the key indicator, which should be considered carefully. After adopting WNOLS, although the hardware cost of a single experimental device increases by 80%, the number of concurrent users supported by the device increases from one to four. Under the condition of supporting 30 users concurrently, NCEUHC increases from 1 to 2.08, exceeding the design goal. In fact, the fixed investment of the project also includes laboratory construction cost, rack procurement cost and experimental equipment procurement cost, accounting for 35%,

10% and 55%, respectively. In this project, the saved experimental equipment purchase cost reduces the total fixed investment cost of the project by 30.25%.

The ADTEE of the eight experiments with and without the WNOLS system are calculated, respectively. The results are shown in Figure 9(a). A total of 30 students participates in the experiment. Without the automatic deployment of WNOLS, the ADTEE is gradually decreases with the improvement of students' wireless network technology proficiency, with a final average of 18.3 minutes. After automated deployment using WNOLS, the ADTEE is 2 for relatively simple experiments, and 3 for complicated experiments, with a final average of 2.75 minutes. The improvement effect of the system on ADTEE is obvious.

At the same time, the analysis of TCRE for each experiment is shown in Figure 9(b). Without the automatic deployment of WNOLS, although the students' proficiency in wireless network technology improves, the experimental difficulty is also increasing. The TCRE gradually decreases, from 70% to 50%, with the final average of 59%. With WNOLS, although the experimental time increases, the average TCRE only increases from 59% to 70%. The experimental data shows that the automatic deployment can improve TCRE and the quality of teaching.

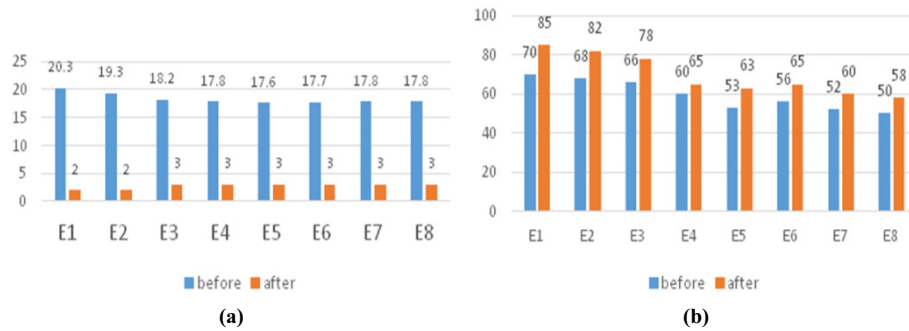


Fig. 9. Statistics of system key indicators. (a) ADTEE; (b) TCRE

In the process of equipment debugging, system capacity testing is performed by increasing the number of racks and experimental equipment. When the number of concurrent users reaches 65, the system runs normally. Affected by laboratory space and funding, the concurrency capacity of the final purchased system is 30.

5 Conclusion

This paper details the construction and implementation of WNOLS. By utilizing EC and NDV technologies, WNOLS allows multiple users to operate their own virtual devices simultaneously without mutual interference. WNOLS also integrates teaching materials resources, physical network equipment resources and experimental teaching environment, realizes the combination of online learning and PBL in the field of wireless network teaching. Especially in the critical period of COVID-19 epidemic prevention, WNOLS allows learners to carry out self-directed online learning. Compared to existing approaches in the literature, WNOLS expands the blended learning mode,

reduces the equipment procurement costs and improves the experimental efficiency. However, the analysis of TCRE also reveals the substantial improvement of teaching quality requires further innovation in teaching methods. Future works may further consider the design of the online assessment strategies and the online instruction to ensure good quality education. In addition, the use of artificial intelligence and big data analysis technology to analyze the learners' learning behaviors and optimize the blended learning quality also needs further consideration.

6 Acknowledge

This work is supported by the Fujian Province Nature Science Foundation under Grant No.2020J01813, the Zhangzhou Municipal Natural Science Foundation under Grant ZZ2021J23, the Research project on education and teaching reform of undergraduate colleges and universities in Fujian Province under grant FBJG20210070, and the Opening Foundation of Fujian Provincial Key Laboratory of Network Security and Cryptology Research Fund, Fujian Normal University under grant NSCL-KF2021-07, and the 2022 Annual Project of the Fourteenth Five-Year Plan for Fujian Educational Science under Grant FJKBK22-173.

7 References

- [1] Jingmin Yang, Wangfei Liu, Hailiang Huang. (2018). *Wireless Network Technology: Principles, Experiments and Network Design*. Xiamen: Xiamen University Press, 35–40.
- [2] Jingmin Yang, Zhixiang Chen. (2017). Research on Architecture of Wisdom Campus Based on Internet Plus. *Journal of Zhangzhou Teachers College (Natural Science Edition)*, 30(1):32–37.
- [3] Jingmin Yang, Trong-Yen Lee, Baoxing Chen, Wenjie Zhang. (2019). A Comprehensive Teaching Reform Model for a Computer Networks Course Based on Integrated Information Systems. *International Journal of Emerging Technologies in Learning (IJET)*, 14(18):76–90. <https://doi.org/10.3991/ijet.v14i18.11040>
- [4] Arvind Singh, R. Sean, et al. (2019). Improving Student Engagement in Teaching Electric Machines Through Blended Learning. *IEEE Trans. Educ.*, 62(4):297–304. <https://doi.org/10.1109/TE.2019.2918097>
- [5] Jianxin Liao, Jingmin Yang, Wenjie Zhang. (2021). The Student-centred STEM Learning Model Based on Artificial Intelligence Project: A Case Study on Intelligent Car. *International Journal of Emerging Technologies in Learning*, 16(21):100–120. <https://doi.org/10.3991/ijet.v16i21.25105>
- [6] J. Chen, J. Zhao. (2018). An Educational Data Mining Model for Supervision of Network Learning Process [J]. *International Journal of Emerging Technologies in Learning (IJET)*, 13(11):67–77. <https://doi.org/10.3991/ijet.v13i11.9599>
- [7] Guolei Zhang, Jia Li and Li Hao. (2015). Research on Cloud Computing and Its Application in Big Data Processing of Distance Higher Education. *International Journal of Emerging Technologies in Learning (IJET)*, 10(08):55–58. <https://doi.org/10.3991/ijet.v10i8.5280>
- [8] Nasim Muhammad, Seshasai Srinivasan. (2021). Online Education During a Pandemic – Adaptation and Impact on Student Learning. *International Journal of Engineering Pedagogy*, 11(3):71–83. <https://doi.org/10.3991/ijep.v11i3.20449>

- [9] Verawardina, Unung; Asnur, Lise, et al. (2020). Reviewing Online Learning Facing the Covid-19 Outbreak. *Talent Development & Excellence*, 12(3s):385–392.
- [10] W. Wahidaturrahmi, M. P. Sari, L. Hayati, E. Kurniawan and F. P. Astria. (2021). Analysis of Student Perceptions about Online Lectures In Terms of Student Learning Styles: Study of the Covid-19 Pandemic. *Advances in Social Science, Education and Humanities Research*: 80–83. <https://doi.org/10.2991/assehr.k.210525.050>
- [11] Nasim Muhammad, Seshasai Srinivasan (2021). Transition from In-Class to Online Lectures During A Pandemic: In “Visions and Concepts for Education 4.0”, Eds M. E. Auer and D. Centea, ICBL 2020, AISC 1314, pp 1–8, Published by Springer Nature, Switzerland.
- [12] K. Y., G. W. (2021). Students’ Perceptions of Online Lecture Delivery Modes: Higher Education During Covid-19 Pandemic and Beyond. *International Journal of Emerging Technologies in Learning (iJET)*, 16(21):58–73. <https://doi.org/10.3991/ijet.v16i21.25305>
- [13] K.-A. Berga, E. Vadnais, J. Nelson, S. Johnston, K. Buro, R. Hu and B. Olaiya. (2021). Blended Learning Versus Face-to-Face Learning in an Undergraduate Nursing Health Assessment Course: Aquasi-Experimental Study. *Nurse Education Today*, 96:1–6. <https://doi.org/10.1016/j.nedt.2020.104622>
- [14] S. Ginting, M. Tjandra and E. Wianto. (2021). Blended Learning: Post Pandemic Solutions. *AKSARA*, 7(2):425–438. <https://doi.org/10.37905/aksara.7.2.425-438.2021>
- [15] Seshasai Srinivasan, Juan Antonio Lopez Ramos, Nasim Muhammad. (2021). Flexible Future Education Model—Strategies Drawn from Teaching During the COVID-19 Pandemic. *Education Sciences*, 11:557–575. <https://doi.org/10.3390/educsci11090557>
- [16] Fu Xiuwen, Yang Yongsheng. (2019). Simulation Platform for Invulnerability of Industrial Wireless Sensor Networks. *Journal of System Simulation*, 31(7):1342–1350.
- [17] CHEN Weihong, LI Renfa. (2014). Simulation Technologies Analysis of Wireless Sensor Networks. *Control Engineering of China*, (2):149–155.
- [18] Haider TH. Alrikabi, Abdul Hadi M. Alaidi, et al. (2019). Student’s Performance Assessment and Learning Skill towards Wireless Network Simulation Tool – Cisco Packet Tracer. *International Journal of Emerging Technologies in Learning (iJET)*, 14(07):196–208. <https://doi.org/10.3991/ijet.v14i07.10351>
- [19] Yu Jihong. (2021). Communication and Network Laboratory of Beijing University of Technology. *Journal of Data Acquisition and Processing*, 36(2):391.
- [20] Chen Qianbin. (2021). Key Laboratory of Mobile Communication Technology, Journal of Chongqing University of Posts and Telecommunications (Natural Science Edition), 33(4):510.
- [21] Mostafa M. Soliman, Ishwar Singh. (2021). Low Cost and User Friendly IoT Laboratory: Design and Implementation. *Visions and Concepts for Education 4.0*, 1314:15–23. https://doi.org/10.1007/978-3-030-67209-6_2
- [22] P. Porambage, J. Okwuibe, M. Liyanage, et al. (2018). Survey on Multi-Access Edge Computing for Internet of Things Realization. *IEEE Communications Surveys & Tutorials*, (99):1–12. <https://doi.org/10.1109/COMST.2018.2849509>
- [23] J. Fang, J. Shi, S. Lu, M. Zhang, Z. Ye, (2021). An Efficient Computation Offloading Strategy with Mobile Edge Computing for IoT. *Micromachines*, 12(2):204–215. <https://doi.org/10.3390/mi12020204>
- [24] K. V. Pham, S. B. Tran, T. V. Nguyen, K.-S. Min. (2019). Asymmetrical Training Scheme of Binary-Memristor-Crossbar-Based Neural Networks for Energy-Efficient Edge-Computing Nanoscale Systems. *Micromachines*, 10(2):141–159. <https://doi.org/10.3390/mi10020141>
- [25] J. R. Gomez-Rodriguez, R. Sandoval-Arechiga, et al. (2021). A Survey of Software-Defined Networks-on-Chip: Motivations, Challenges and Opportunities. *Micromachines*, 12(2): 183–195. <https://doi.org/10.3390/mi12020183>

8 Authors

Jingmin Yang is an associate professor at Minnan Normal University of Computer Science, 36 Xianqianzhi Road, Xiangcheng District, Zhangzhou, 363000, China. He is also a visiting scholar at the Fujian Provincial Key Laboratory of Network Security and Cryptography, Fujian Normal University, Fuzhou, China. His research interests include hardware-software co-design of embedded systems, machine learning, wireless communications and networks. Currently he is studying toward the Ph.D. degree in the Department of Electronic Engineering, National Taipei University of Technology, Taipei, Taiwan.

Trong-Yen Lee received his Ph.D. degree in Electrical Engineering from National Taiwan University, Taipei, China, in 2001. Since 2002, he has been a member of the faculty in the Department of Electronic Engineering, National Taipei University of Technology, where he is currently a professor. His research interests include hardware-software co-design of embedded systems, FPGA systems design, and VLSI design.

Wen-Ta Lee received his Ph.D. degree in Electrical Engineering from National Taiwan University, Taipei, Taiwan, in 1995. Since 1997, he has been a member of the faculty in the Department of Electronic Engineering, National Taipei University of Technology, where he is currently an associate professor. His research interests include IC design, and VLSI circuit design.

Li Xu received the Ph.D. degree in applied mathematics from the Nanjing University of Posts and Telecommunications (NJUPT), Nanjing, China, in 2004. Since 1992, he has been a member of the faculty in the College of Computer and Cyber Security, Fujian Normal University, Fuzhou, China, where he is currently a professor and the president of the college. He is also the director of the Fujian Provincial Key Laboratory of Network Security and Cryptology, Fuzhou, China. His research interests include network and information security, Internet of things and wireless network communication.

Article submitted 2022-07-11. Resubmitted 2022-09-14. Final acceptance 2022-09-15. Final version published as submitted by the authors.