Research Article

Tao Wang, Li Wang, Pengfei Yan, Renu Popli, Poonam Rani, and Rajeev Kumar*

Application of embedded Linux in the design of Internet of Things gateway

https://doi.org/10.1515/jisys-2021-0208
received October 13, 2021; accepted June 30, 2022

Abstract: To study the development trend of embedded platforms in home appliances Internet of Things (IoT), an application of embedded Linux in the design of IoT gateway is designed. ARM microprocessor S3C2410, EM310 radio frequency (RF) module, and Ethernet DM9000A module are used to realize the home network information-processing platform. The platform utilizes existing network facilities, RF modules, and network interface modules to realize the interconnection of the home perception network through wireless or wired links. For experimental analysis, Ping the central server (gateway) from the IoT node B is set to twice: The first time ttl = 255 time = 1.9 ms, the second time: ttl = 255 time = 2.4 ms. After application verification, the system has the characteristics of low cost, small size, low power consumption, and vital ease of use. It can not only integrate the home Internet and the IoT gateway but also directly run application-layer communication protocols such as HTTP/MQTT between IoT nodes and Internet terminals. The design supports multi-hop communication and significantly increases the deployment range of IoT nodes without signal relay equipment, hence reducing the cost of IoT networking and the complexity of the network architecture.

Keywords: internet of gateway, embedded linux, integrated gateway

1 Introduction

In the last decade, Internet of Things (IoT) technology officially promoted at a national strategic level and becomes one of the key information industries to be researched and implemented. The agricultural IoT also took the opportunity to open the road of development [1,2]. There are a large number of various sensors in the agricultural IoT, and they are of different types. There are multiple heterogeneous networks, the collected data need to be uploaded to the data center through the agricultural IoT gateway (hereinafter referred to as the gateway), at the same time, the device control command is forwarded to the control device in the agricultural IoT through the gateway [3]. The agricultural IoT gateway plays an important role in the

*** Corresponding author: Rajeev Kumar,** Chitkara University Institute of Engineering and Technology, Chitkara University, Punjab, India, e-mail: rajeev.kumar@chitkara.edu.in, rajeev_chauhan364@yahoo.co.in
Tao Wang: Department of Mechanical and Electrical Engineering, Henan University of Technology, Luohe Institute of Technology, Luohe, Henan 462000, China, e-mail: wangtao932@163.com
Li Wang: Department of Information Engineering, Luohe Food Vocational College, Luohe, Henan 462000, China, e-mail: wangli8582@126.com
Pengfei Yan: Department of Information Engineering, Qinhuangdao Vocational and Technical College, Qinhuangdao, Hebei 066000, China, e-mail: yanpengfei35@163.com
Renu Popli: Chitkara University Institute of Engineering and Technology, Chitkara University, Punjab, India, e-mail: renu.popli@chitkarauniversity.edu.in
Poonam Rani: University Institute of Engineering and Technology, Kurukshetra University, Kurukshetra, India, e-mail: prani2015@kuk.ac.in

agricultural IoT system, the quality of the gateway design will affect the stability and accuracy of data collection and device control [4]. IoT technology has always been regarded as an application extension of the Internet. It uses radio frequency identification, infrared sensors, global positioning system, information sensing devices such as laser scanners are connected to the Internet for information exchange, realize intelligent identification, positioning, tracking, monitoring, and management. The concept of the IoT was proposed in 1999, the IoT gateway is a new term, and it will play a very important role in the future IoT era, it will become the link between the perception network and the traditional communication network [5].

With the integration of home IoT, edge computing, artificial intelligence, and other new technologies, the IoT lead to a new era of large-scale development and cross-industry integration. It is even more necessary to lower the barriers for interaction between IoT data and the Internet. It also supports running various application layer protocols such as hypertext terminal protocol/Message queuing telemetry transport (HTTP/MQTT) on the IoT [6]. To solve this kind of problem, a design of IoT integrated gateway based on 6LoWPAN and B.A.T.M.A.N. adv protocol is proposed. The process begins with the home gateway server, it can avoid the cumbersome process of data protocol conversion through the IoT gateway. It significantly improves the deployment range of IoT nodes without signal relay equipment, thus, reducing the networking cost of the home IoT and the complexity of the network architecture. Yi and Choi [7] improved the cycle time performance of the EtherCAT network by developing a Linux Ethernet driver. The authors have removed the participation of the Linux network stack and the new application program interface of the standard Ethernet driver is developed. The Ethernet driver is used to establish a direct interface between the main module of the embedded system and the Ethernet controller. Therefore, the time-consuming memory copy operation is reduced and the EtherCAT frame process is accelerated. To demonstrate the effect of developing the Ethernet driver, an EtherCAT network composed of an embedded Linux-based master device and a commercial off-the-shelf slave was set up. The experimental results confirm that the cycle time performance is significantly improved.

The main contributions of this article are listed as follows:

i. To propose the important role of the IoT gateway in the informatization and intelligence of the home system.
ii. To utilize several existing facilities such as radio frequency (RF) modules and network interfaces to analyze the performance of home perception network through wireless or wired links.
iii. To realize the design of the IoT gateway and its application in the home network using the embedded system, adopt ARM9 microprocessor S3C2410, EM310 RF module, DM9000A network module, and embedded Linux system software and hardware platform.

In the remainder of this article, the most recent work done in the field is discussed in Section 2. Section 3 summarizes the system framework and functional modules. Section 4 summarizes the simulation and test-bed experiment results. Finally, Section 5 concludes this article.

2 Related work

There is as of now little academic and research work on the design and standardization of the IoT Gateway framework [8]. Different operators of international and domestic telecom sectors have launched the connected business in applications joining wireless sensor network (WSN) and telecommunication networks [9]. These organizations are conducting various active explorations as per the requirement of industrial users. Some of such standard organizations like 3GPP and ETSI M2M TC have established relevant standards [10]. The principal objective of ETSI M2M TC is to do exploration of machine-to-machine (M2M) standardization. The work has already advanced after the achievements of ETSI and 3GPP. Currently, the focus of ETSI M2M TC is on the definition of M2M and its application examples [11]. On this basis, the business requirements and standardization is targeted, however, tended to no particular technology yet. A research group is set up by 3GPP, and the basic idea is to analyze the requests, their achievability, and their applications [12]. With the advancement in sensor technology and the design of various security mechanisms, IoT is widely implemented in the majority of applications. Zheng et al. [13] proposed a system for water level detection by using an optical fiber sensor and fuzzy logic control. In one another study Zeng et al. [14] have proposed a secure smart water management system by designing a blockchain-based model for agricultural applications. The experimental results show the effectiveness of the model as the design achieves high accuracy and better security. In one study, Singh and Sharma [15] proposed wireless sensor network and unmanned aerial vehicle-based IoT framework for agriculture application. Their proposed model achieves better accuracy in the agricultural domain when compared with other existing approaches.

In the short term, the household endeavors also accomplished equipment purpose work as per the regulations [16]. The Telecom industry of China reported the MDMP agreement for WSN station management and ensured demonstrating applications in agricultural and home applications for practical understanding [17]. Concerning plan and execution, a research institute of telecom proposed a household plan of IoT, which showed that passage was the central component of data assembling and control [18]. Chen et al. [19] summed up that the vital elements of the entry framework were settlement change, state control, data gathering, terminal tending to, and its authentication. In the IoT work proposed in ref. [20], through the intelligent entrance, the genuine articles can be changed into soothing assets to be coordinated into the current frameworks, to be straightforwardly gotten to by outside HTTP. The intense passage cooperates with sensor hubs using Bluetooth, which designates the URL for sensor hubs and advances the gathered discernment information to the Web server through HTTP parcels including the JSON information portion. Along these lines, sensor organization and conventional media communications networks are connected [21]. The current framework related to the Internet of things is essentially an element that performs information transmission and transmission [22]. In any case, the administration and control issues are of less attention.

3 Research methods

3.1 Overview of the system framework and functional modules

At present, the IoT has a wide range of uses and various access methods. The IoT gateway needs to integrate a variety of access methods, not only to meet the short-distance communication requirements in local areas, but also to realize the connection with the public network, and to complete the processing, forwarding, and control of data packets. Therefore, the IoT gateway as the core device of the IoT should have the following functions. The embedded home IoT gateway system can be divided into an application management layer, network protocol layer, and perception access layer, to realize the functional requirements that the IoT is applied to the home network system that can be fully perceived, reliably transmitted, and intelligently processed. The system framework is shown in Table 1. The functions of each layer are as follows:

Table 1: System frame structure

System shaft	Web browser terminal device (PC)
Layer-by-layer network	Full-featured network servers such as DHCP and Web
	A complete network protocol clusters such as IPv4 and IPv6
Aware access layer	Safe and reliable network interface module and RF module

i. The application management layer realizes the monitoring and management of the entire system through the most familiar Web browser.

ii. A complete network protocol cluster such as IPv4 and IPv6 is embedded in the network protocol layer, and based on it, built a dynamic host configuration protocol (DHCP) server, Web server, and other full-featured servers.
iii. The perception access layer uses the network interface module and RF module in the system to facilitate the access of the wide-area network (WAN) and the interconnection and intercommunication of the perception network within the home through wired or wireless means.

The functional modules of the embedded home IoT gateway system can be divided into wireless communication modules, wireless communication module interface, network interface module, information processing module, and interface management module. The basic working process of the system is that when the embedded home IoT gateway system is started, the information-processing module is automatically started. It controls the process and starts the wireless communication module (RF module) through the wireless communication module interface to access the WAN network and home wireless communication equipment. Then it waits for the interface management module to set the system configuration parameters or use the system default configuration parameters to bridge the interconnection of the home perception network devices or route to the WAN [23].

In the embedded home IoT gateway system, the wireless communication module is connected to the WAN network wirelessly. At the same time, it accepts the family's internal perception network device data forwarded by the information-processing module. In addition, the module is also a connection interface for wireless devices in the home perception network. The information-processing module is the brain of the embedded home IoT gateway system. It starts automatically after power-on and controls the safe start of surrounding modules. The information-processing module is the core of the entire system. At the same time, it accepts data from external network information and internal network equipment. It completes the most core functions such as routing and forwarding data from the internal network to the external network and from the external network to the internal network. The interface management module allows users to configure certain device configurations (network interface module IP, gateway, subnet mask, the built server parameters, wireless communication module connection status, etc.) that are modified and managed to adapt to the user's habits.

3.2 System design and implementation

The embedded home IoT gateway system is a hardware platform with an S3C2410 microprocessor, DM9000A Ethernet module, and EM310 RF module as the core hardware platforms. Among them, S3C2410 is a memory management unit and ARM9 microprocessor supports real-time control. It is a CPU designed for the entire system, responsible for the control of the surrounding circuit modules and it carries the entire embedded minimal system. DM9000A can connect to all home telephone line network equipment or other transceivers that provide support for media-independent interface functions. Its automatic coordination function will automatically complete the configuration to best suit other line bandwidths. It also supports IEEE802.3 full-duplex flow control and simple implementation of features such as plug and play. It is also able to process and encapsulate Ethernet frames, transmit them through network interfaces, and twisted pair cables in real-time for the effective realization of wired connection of the home perception network. The EM310 RF module can be easily connected to the GPRS network to realize the wide area of network access of the gateway system. Additionally, it has functions such as voice messages to facilitate access to the mobile phone network and realize the wireless networking of the home perception network. The system hardware structure diagram is shown in Figure 1.

The microprocessor S3C2410 is the core of the entire hardware system, it controls the startup of surrounding circuits in real-time when it is powered on, shut down, reset, etc., with real-time monitoring of the working status of surrounding circuit modules. The whole hardware system is controlled by the microprocessor S3C2410 to connect the EM310 RF module to the WAN and mobile cellular network. This

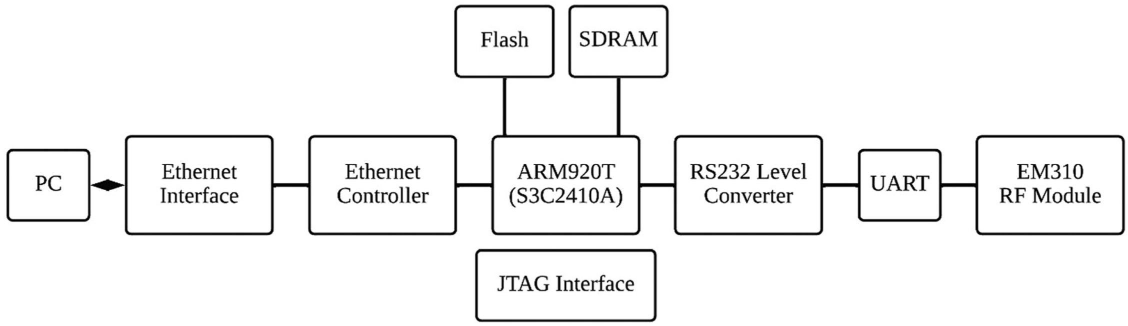

Figure 1: System hardware structure diagram.

network is designed to realize the access of the gateway system to the external network. The wireless networking of the home perception network is controlled by the Ethernet controller DM9000A to read, write, and reset with real-time wired access to the home perception network [24]. In addition, the entire hardware system has also designed a RESET module that is a hardware watchdog module by which the stability of the system can be monitored in real-time scenarios to safeguard that the whole system is reset and restarted when the software reset system fails to work. The 64M SDRAM and 64M NANDFlash fully meet the storage needs of embedded systems and the needs of program operation.

3.3 Software design

The software platform reference model of the embedded home IoT gateway system is shown in Figure 2. The whole system is based on the hardware platform with an embedded Linux operating system as the core which realizes the writing of the network interface module driver and wireless communication module interface driver. The network realizes IPv4/IPv6 protocol cluster, routing, and forwarding whereas firewall transplantation to access the writing and transplantation of upper-level applications (PPP dialer, DHCP server, Web server, and Web page).

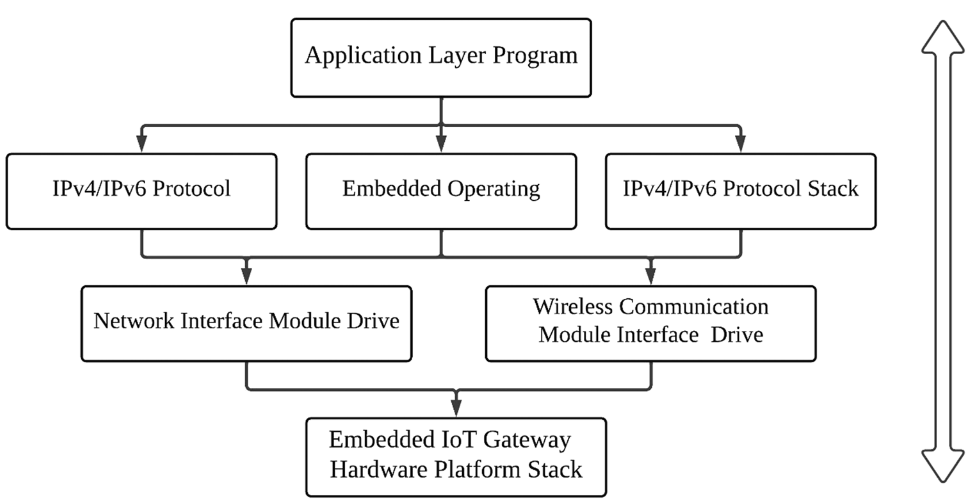

Figure 2: Software platform reference model.

The underlying hardware initialization program, embedded operating system and file system, application layer software, and user layer software, design each module from bottom to top. These modules do not exist in isolation and are based on lower-level software layer programs.

3.3.1 BootLoader porting

The BootLoader of the embedded home IoT gateway system adopts a lightweight vivi specially developed by the mini-company for an arm. Two working modes of startup loading and downloading can be realized and the setting information in the /Vivi/Makefile file can be modified, delete the "*.o" and "*.o.flag" files before compilation to ensure that the compilation is valid. The Vivi executable file generated by the final compilation can be programmed to NANDFlash to realize the booting of the system.

3.3.2 Embedded Linux kernel porting

Linux2.4.18 kernel can run on ARM920T processor at high speed. Its file system supports file formats or functions such as cramfs, yaffs, ext2, and NFS. The kernel transplantation needs to be carried out by kernel tailoring and kernel compilation. It generates the image file by compiling the kernel file, which is the kernel image file. The BootLoader can start the embedded Linux operating system after loading the board.

3.3.3 Root file system migration

Embedded Linux systems support a variety of file systems, and most of them include ext2, NFS, and other file systems when the kernel is made. Here the yaffs file system is uploaded on the embedded Linux system that is a readable and writable file system and convenient for making subsequent applications. Use the file system authoring tool of busybox1.0 version which is known as the Swiss Army Knife of Embedded File System and is convenient and very suitable for the production of the embedded file system. The steps of making a file system are: Configure Makefile, tailor the file system, build the root file system directory, and compile. The root_china.yaffs image file can be generated by compiling and then loaded into the board, start the hardware system and you can see the shell interactive interface on the terminal, which is convenient for subsequent application development.

3.3.4 Iptables transplantation (Network address translation (NAT) implementation)

Iptables is an IP packet filtering system integrated with the latest 2.4.x version of the Linux kernel. If the Linux system is connected to the Internet or LAN, a server or a proxy server that connects the LAN and the Internet, the system is conducive to better control the IP packet filtering and firewall configuration on the Linux system. Netfilter/Iptables provides a series of tables in the Linux 2.4 kernel. Each table is composed of several chains, and each chain can be composed of one or several rules [25]. The kernel module can register a new rule Table and requires data packets to flow through the specified rule Table, used to implement data packet filtering (Filter Table), network address translation (NAT table), and data packet processing (Mangle Table). The NAT Table contains PREROUTING chain, POSTROUTING chain, and OUTPUT chain. Netfilter monitors hook function NF_IP_PRE_ROUTING, NF_IP_POST_ROUTING, and NF_IP_LOCAL_OUT, and according to the rules in the NAT Table performs address translation processing on the data packet. NAT only queries the NAT table for the first packet of a new connection, then the data packets of the same connection will undergo the same conversion process according to the result of the first data packet. Iptables consist of kernel modules and user interface applications. The Iptables kernel module can filter and manage the input and output IP packets, it is an integral part of the Netfilter framework in the

Linux 2.4 kernel; the Iptables user interface program can add, insert, or delete rules in the kernel Table, use the Iptables tool, and use the option “-that” to create and modify the NAT Table.

3.3.5 DHCP service software

DHCP is mainly used to complete the dynamic allocation of the IP address of the terminal equipment in the local area network, wherein at the same time the client subnet mask is set by directly accessing the network without manually setting the subnet mask and IP address. DHCP service software not only realizes the dynamic allocation of IP addresses but also the protocol helps in managing several users in the network such as stopping the allocation of IP addresses for users whose leases have expired.

3.3.6 Point-to-point protocol dial-up service

PPP is a link-layer protocol that carries network-layer data packets on point-to-point links. The PPP dial-up service is used to control the EM310 RF module to access the wireless network, and the physical layer uses standard RS232 to communicate with the Micro-Control Unit [26]. The Network Control Protocol is responsible for sending the data to be processed to the network layer, Link Control Protocol (LCP) is used to exchange configuration information packets in the link establishment phase when the configuration is completed, the LCP is turned on and the authentication phase is entered [27]. Authentication is optional, users who pass the authentication will get an IP address dynamically allocated by PPPD through the IP Control Protocol, and finally realize access to the Internet [28].

3.3.7 Web service software

A user-level interactive web service software installs the Web server in the embedded Linux system. It is a small, simple, and comprehensive Boa embedded Web server that writes the corresponding Common Gateway Interface (CGI) program to realize the interaction between the user Web interface data and the Web server, to realize the user’s control of the system. The realization of the Web page adopts the C language variant based on the build engine-C Language Service Page (CSP), similar to other CGIs (ASP, JSP). The CSP inserts the C language into the HTML template. The “*.cgi” file is finally generated through compilation, and the user can intuitively realize the monitoring and management of the system with the help of the IE browser when loaded into the board [29].

4 Result analysis

4.1 Data exchange patency test between nodes

The data exchange patency is checked by sending Ping packets between devices. The specific Ping scheme is as follows and is as shown in Figure 3. To implement this process, you can directly use the Ping command on the server-side and an IoT node.

The hub server (gateway) (94:A7:8E:32:2D:7B) sends Ping packets to the other two IoT nodes, IoT node A (C4:8E:3B:23:82:7D) sends a Ping packet to the IoT node B (E3:4D:A2:83:D4:42), and the delay in return data is shown in Figure 4 [30].

Based on this data, it is determined that the data interaction of the wireless Mesh network under the topology structure is normal, and the hardware part of the module works normally.

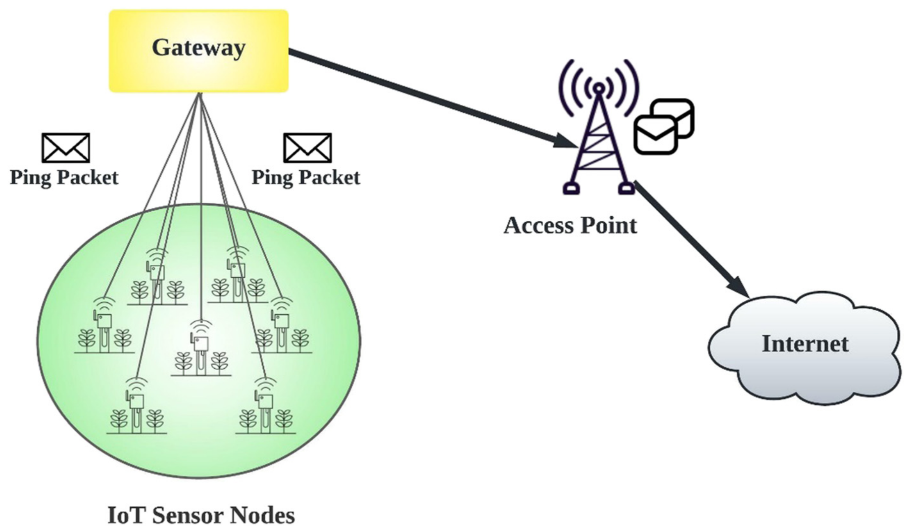

Figure 3: Schematic diagram of Ping scheme for data exchange patency test.

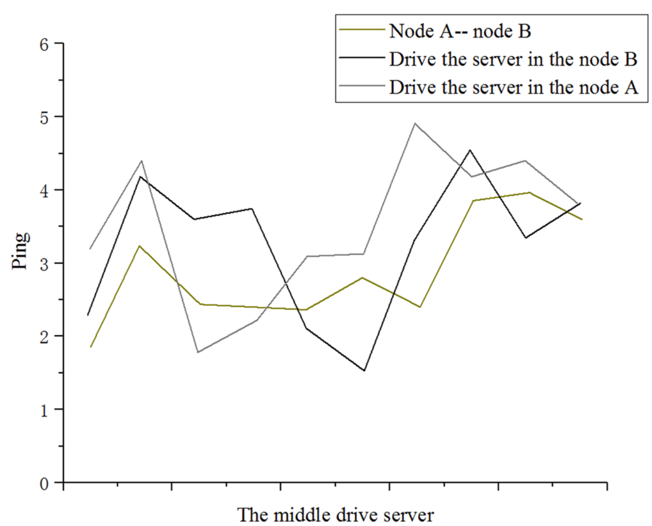

Figure 4: Ping delay test data statistics table.

4.2 Multi-hop communication of B.A.T.M.A.N. adv routing protocol

Multi-hop communication means that IoT nodes use one or more fixed or mobile other IoT nodes to transmit data to the central server gateway or destination IoT node. Therefore, the key to this test is to block the direct connection of a certain IoT node (A or B) to the central server, and test whether it can achieve multi-hop communication through another IoT node. There are two ways to block direct communication between an IoT node and the central server. One is to place the blocked IoT node in a place where it cannot receive the 802.11 protocol signal transmitted by the central server. The second is that the hub server uses the netfilterliptables IP packet filtering system to shield all data packets communicated with the IoT node under the MAC address. Because the former has signal attenuation that causes packet loss or blocked nodes, they can capture the 802.11 signal of the central server; therefore, the relevant variables cannot be controlled well, so the second method is used for testing [31].

Set the IoT node and the hub server to set the IoT node B as the node that is blocked from directly communicating with the hub server: Turn off the IEEE802.11 module of IoT node A first, and then use ipotables to block data packet communication. Figures 5 and 6 show the commands for Hub server-side and IoT node B.

```
ip6tables -AINPUT-p ALL-m mac--mac-source E3:4D:A2:83:D4:42-j DROP
#Discard the incoming data packet from Internet of Things node B
```

Figure 5: Hub server-side command.

```
ip6tables -A INPUT -p ALL -m mac--mac-source 94:A7:8E:32:2D:7B -j DROP
#Discard the incoming data packet from the hub server
```

Figure 6: IoT node B commands.

After completing the blocking work, ensure that the IoT node B and the hub server cannot achieve single-hop communication. Next, open the IEEE 802.11 module of IoT node A and connect to the wireless Mesh network to view the routing table of IoT node A, the discovery node has detected the remaining two devices.

Ping the central server (gateway) from the IoT node B, set it to twice: The first time ttl = 255 time = 1.9 ms, the second time: ttl = 255 time = 2.4 ms. It proves that node B of the IoT can interact with the central server (gateway), and the multi-hop communication verification of the B.A.T.M.A.N. adv protocol is passed. A test environment is generated to test the various functions of the IoT gateway. The results show that the developed IoT gateway system based on embedded Linux can meet the design requirements.

5 Conclusion

This article introduces the research background and research status of the IoT gateway and analyzes the shortcomings and improvement measures of the current IoT gateway. This work completes the perfect transplantation and matching of the Iptables program at the core of the embedded Linux operating system. The proposed uses rich rule settings to realize the firewall mechanism and NAT mechanism of the entire system, which makes the system data transmission more reliable. It enables the data packets of the external network and the internal perception network devices of the home to be forwarded to the corresponding interface smoothly by the rules. The user-level uses the CSP language that inherits the advantages of the C language to refine the CGI program, and a better man–machine interface is realized. The system has the characteristics of good real-time performance and high security and is suitable for use in the home IoT. In the future more developing trends of embedded platforms in home appliances IoT, an application of embedded Linux in the design of IoT gateway can be designed.

Conflict of interest: Authors state no conflict of interest.

References

[1] Qin H, Cao B, He J, Xiao X, Chen W, Peng Y. Cross-interface scheduling toward energy-efficient device-to-gateway communications in IoT. IEEE Internet Things J. 2019;7(3):2247–62.

[2] Yang J, Sharma A, Kumar R. IoT-based framework for smart agriculture. Int J Agric Environ Inf Syst (IJAEIS). 2021;12(2):1–14.

[3] Hayashikoshi M, Tanizaki H, Murai Y, Tsuji T, Kawabata K, Nii K, et al. A cost-effective 1T-4MTJ embedded MRAM architecture with voltage offset self-reference sensing scheme for IoT applications. IEICE Trans Electron. 2019;102(4):287–95.
[4] Dick RP, Shang L, Wolf M, Yang SW. Embedded intelligence in the internet-of-things. IEEE Des Test. 2019;37(1):7–27.
[5] Atzori L, Bellido JL, Bolla R, Genovese G, Iera A, Jara A, et al. SDN&NFV contribution to IoT objects virtualization. Comput Netw. 2019;149:200–12.
[6] Chang CC, Lee WK, Liu Y, Goi BM, Phan RCW. Signature gateway: offloading signature generation to IoT gateway accelerated by GPU. IEEE Internet Things J. 2018;6(3):4448–61.
[7] Yi HC, Choi JY. Cycle time improvement of EtherCAT networks with embedded linux-based master. IEICE TRANS Inf Syst. 2019;102(1):195–7.
[8] Kang B, Choo H. An experimental study of a reliable IoT gateway. ICT Express. 2018;4(3):130–3.
[9] Salman AD, Khalaf OI, Abdulsahib GM. An adaptive intelligent alarm system for wireless sensor network. Indonesian J Electr Eng Comput Sci. 2019;15(1):142–7.
[10] Chang K, Soong A, Tseng M, Xiang Z. Global wireless machine-to-machine standardization. IEEE Internet Comput. 2011;15(2):64–9.
[11] Song J, Kunz A, Schmidt M, Szczytowski P. Connecting and managing m2m devices in the future internet. Mob Netw Appl. 2014;19(1):4–17.
[12] Swetina J, Lu G, Jacobs P, Ennesser F, Song J. Toward a standardized common M2M service layer platform: introduction to oneM2M. IEEE Wirel Commun. 2014;21(3):20–6.
[13] Zheng Y, Dhiman G, Sharma A, Sharma A, Shah MA. An IoT-based water level detection system enabling fuzzy logic control and optical fiber sensor. Security Commun Netw. 2021;2021:4229013.
[14] Zeng H, Dhiman G, Sharma A, Sharma A, Tselykh A. An IoT and Blockchain-based approach for the smart water management system in agriculture. Expert Syst. 2021;2021:e12892.
[15] Singh PK, Sharma A. An intelligent WSN-UAV-based IoT framework for precision agriculture application. Comput Electr Eng. 2022;100:107912.
[16] Nunes BAA, Mendonca M, Nguyen XN, Obraczka K, Turletti T. A survey of software-defined networking: past, present, and future of programmable networks. IEEE Commun Surv Tutor. 2014;16(3):1617–34.
[17] Al-Hamadi H, Chen R. Adaptive network defense management for countering smart attack and selective capture in wireless sensor networks. IEEE Trans Netw Serv Manag. 2015;12(3):451–66.
[18] Kshetri N. The evolution of the internet of things industry and market in China: an interplay of institutions, demands and supply. Telecommun Policy. 2017;41(1):49–67.
[19] Chen S, Xu H, Liu D, Hu B, Wang H. A vision of IoT: applications, challenges, and opportunities with china perspective. IEEE Internet Things J. 2014;1(4):349–59.
[20] Huang H, Hu L, Xiao F, Du A, Ye N, He F. An EEG-based identity authentication system with audiovisual paradigm in IoT. Sensors. 2019;19(7):1664.
[21] Mason SJ, Cleveland SB, Llovet P, Izurieta C, Poole GC. A centralized tool for managing, archiving, and serving point-in-time data in ecological research laboratories. Environ Model Softw. 2014;51:59–69.
[22] Mach P, Becvar Z. Mobile edge computing: a survey on architecture and computation offloading. IEEE Commun Surv & Tutor. 2017;19(3):1628–56.
[23] Dick RP, Shang L, Wolf M, Yang SW. Guest editors' introduction: embedded intelligence in the internet-of-things. IEEE Des Test. 2020;37(1):5–6.
[24] Mahbub M, Hossain MM, Gazi MSA. Cloud-enabled IoT-based embedded system and software for intelligent indoor lighting, ventilation, early stage fire detection and prevention. Comput Netw. 2021;184:107673.
[25] Johns EM, Oestreich S. On the edge: how to provide course-and program-integrated library support without being embedded. J Library Inf Serv Distance Learn. 2019;13(1–2):1–20.
[26] Xiang Y, Chen Y, Ye J, Wen B, Hu H. Design of multi-parameter monitoring system based on embedded Linux + Qt. Zhongguoyi Liao qi xie za zhi = Chin J Med Instrum. 2020;44(2):127–31.
[27] Xue L, Huang Q, Zhang S, Huang H, Wang W. A lightweight three-factor authentication and key agreement scheme for multigateway WSNs in IoT. Security Commun Netw. 2021;2021:1–15.
[28] Agrawal H, Dhall R, Iyer KSS, Chetlapalli V. An improved energy efficient system for IoT enabled precision agriculture. J Ambient Intell Humanized Comput. 2020;11(6):2337–48.
[29] Gaur AS, Budakoti J, Lung CH. Vertical handover decision for mobile IoT edge gateway using multi-criteria and fuzzy logic techniques. Adv Internet Things. 2020;10(04):57.
[30] Yao J, Ansari N. Caching in dynamic IoT networks by deep reinforcement learning. IEEE Internet Things J. 2020;8(5):3268–75.
[31] Rodoplu V, Nakıp M, Eliiyi DT, Güzeliş C. A multiscale algorithm for joint forecasting–scheduling to solve the massive access problem of IoT. IEEE Internet Things J. 2020;7(9):8572–89.