The Network Model of Internet Access Intranet Based on Embedded Platform

Xian Zhang[1,2,3,4], Pengju Tang[1*], Dong Yin[1,3,4], Taiguo Qu[1], Yiwen Liu[1,3,4]

[1] School of Computer Science and Engineering, Huaihua University, China
[2] College of Computer Science and Electronic Engineering, Hunan University, China
[3] Key Laboratory of Intelligent Control Technology for Wuling-Mountain Ecological Agriculture in Hunan Province, China
[4] Key Laboratory of Wuling-Mountain Health Big Data Intelligent Processing and Application in Hunan Province Universities, China
zxian@ hnu.edu.cn, 412482450@qq.com, 443660950@qq.com, 1849310620@qq.com, 87134537@qq.com

Abstract

To better manage and control the embedded devices located in the remote Intranet, the functional requirements of the Internet accessing the Intranet need to be effectively solved. This paper studies and implements a network model of external network accessing intranet network and realizes the network model's function by programming. At the same time, an experimental platform is built to test the performance metrics of the network model, such as the maximum load and CPU utilization, device connection success rate, user access response time, and other performance parameters. The test results show that the network model can meet the network scale of at least 3000 intranet devices and has high practical value and application prospects.

Keywords: Intranet, Network model, Maximum load, Connection success rate, Response time

1 Introduction

With the continuous development of computer technology, network technology, microelectronics technology, and Internet of Things (IoT) technology, the application of IoT and the embedded system has been more deeply into every field of our daily lives [1]. Such as all kinds of communication base station equipment, outdoor monitoring equipment, outdoor acquisition equipment, etc [2-3]. However, it is an urgent problem to connect the IoT devices to the Internet to realize more networked and intelligent management and monitoring. Especially for those embedded devices scattered in the remote end, it becomes more important to realize the access to the remote embedded devices. Realizing the access of remote embedded devices, we can get the running information of the remote devices in real-time and send corresponding control information for monitoring and management [4].

At present, the network communication interface with embedded devices mainly includes Ethernet network interface, GPRS/3G/4G/5G mobile network interface, WIFI network communication interface, etc [5]. It is impossible to assign a public network IP to every remote embedded device that contains an Ethernet network interface, GPRS/3G/4G/5G mobile network interface, or a WIFI network communication interface because the public network IP is limited and the remote embedded device is thousands and the locations are scattered [6-7]. Therefore, to effectively manage and control the remote embedded devices, it is equivalent to realizing the external network access to the internal network.

In view of the problems described above, it is of great value and practical significance to study how to realize the technology of the Internet accessing Intranet effectively. Therefore, this paper explores a network model for external network access to the internal network and implements the network model with C++ language programming. Meanwhile, an experimental platform is built to test the maximum load connection number, external network connection success rate, server CPU utilization, response time, and other model parameters.

2 Requirements Scenario Analysis

There are two main methods to access the internal network from the external network. One is to allocate the public network IP; the other is to convert the network address. Allocation of public network IP: with the rapid development of the Internet, there is a crisis that the originally huge number of IPV4 addresses are exhausted. Since February 3, 2011, the Regional Internet Registry can no longer allocate any available IPV4 addresses, which indicates that the IPV4 address pool has been basically exhausted [8-9]. To alleviate the exhaustion of IPV4 addresses, network researchers proposed the IPV6 solution. Still, with the concept of the Internet of things proposed and implemented, things are connected to things, IPV6 is also difficult to solve [10]. Network Address Translator (NAT) technology: it maps IP Address from private network to public network, thus converting private network address into public network address and realizing the function of Intranet host to access public network [11-12]. External network access to the internal network through network address translator technology, commonly adopted methods are: Application Level Gateway technology, Middle-box Communication technology, Universal plug-in and Play technology, Simple Traversal of User Datagram Protocol(UDP) Through Network Address Translators technology, Traversal Using Relay

*Corresponding Author: Pengju Tang; E-mail: 412482450@qq.com
DOI: 10.53106/160792642022032302001

Network Address consultants technology, Interactive Connectivity Establishment technology, Tunneling technology, Hole Punching technology and so on [13-15].

In the scenario studied in this paper, the external network access to the internal network is mainly aimed at embedded devices scattered in remote corners. Therefore, the network interface of the remote embedded device is based on the GPRS/3G/4G/5G mobile network. At the same time, it is almost impossible to use an Ethernet network interface or WIFI network interface [16-17]. Implementing network access with an Ethernet or WIFI network interface is too expensive for dispersed embedded devices [18].

The network model structure of the external network accessing the internal network is shown in Figure 1.

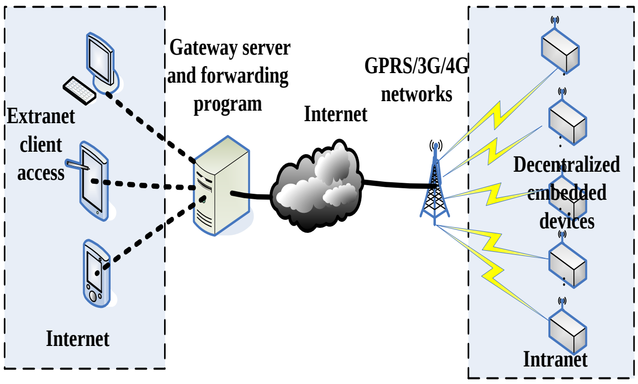

Figure 1. Network access structure diagram

In the figure above, the Intranet is a decentralized embedded device, and the embedded device contains GPRS/3G/4G/5G network module interface; the outside network is the access user. Users of the external network need to access the embedded devices of the internal network to grasp the operation status and information of the embedded devices to realize real-time management of the embedded devices. Normal access is not possible to manage user connections to embedded devices on the Intranet [19]. Therefore, it is necessary for us to research and realize the function that the management user can connect to the embedded device and the external network user can access the internal network device.

3 Model Design and Implementation

In view of the above network structure, external network users need to access the remote decentralized internal network embedded devices. A network model is studied to realize the function of the external network accessing the internal network.

3.1 Network Model Design

This network model realizes the external network to access the internal network mainly through the Transmission Control Protocol (TCP) network programming to realize the server-side forwarding program and the target side forwarding program to complete [20]. Therefore, the intermediate forwarding program specifically includes the server-side intermediate forwarding program and the target server-side intermediate forwarding program. The block diagram that the network structure of the forwarding program of this network model is shown in Figure 2.

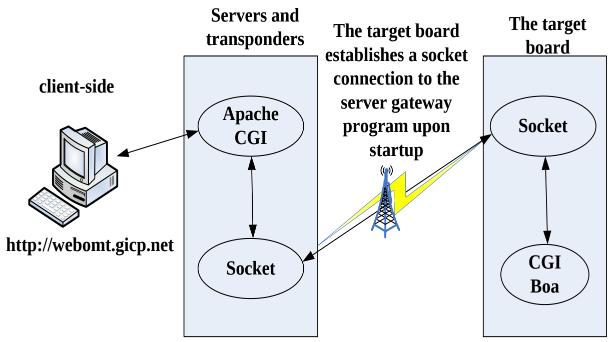

Figure 2. Forwarding model structure diagram

A monitor program for forwarding information is deployed on the server gateway forwarder, and a forwarding program is used on the target machine. The client PC uses Uniform Resource Locator (URL) plus the target device number to access the gateway server. The server receives the request according to the target device number and serial number to determine which connected socket communication with the target board; the target board socket replies to the corresponding data based on the received request. Then the target board forwarding program forwards the replied data to the remote PC. This will display the remote target's content information to the visiting user's browser.

The interfaces of forwarding data communication and interaction in this network model mainly include the information interaction between the server-side forwarding program and XML file, the server-side forwarding program and the browser, the server-side forwarding program and the target board end-to-end forwarding program, and the target board end-to-end forwarding program and the BOA server. The communication interface interaction of this network model is shown in Figure 3.

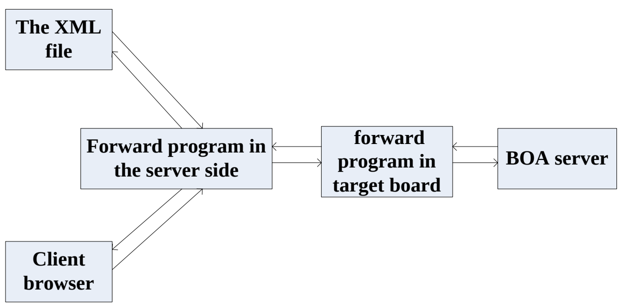

Figure 3. Communication interface diagram

The server-side forwarding program interacts with the target board side forwarding program to obtain the relevant information of the target board (site number, device type, longitude, latitude, socketFd, etc.), and the server-side forwarding program stores the relevant information of the acquired device in the XML file in real-time. Input the corresponding URL + socketFd in the client browser; the server-side forwarding program listens for and receives requests from the browser. The server-side will receive the request from the browser and interact with the target board side forwarding program and forward the request to the target board forwarding program; the target board forwarder forwards the request to the BOA server, the BOA server replies request data to the target board forwarding program, the target board forwarding program sends the reply data to the server-side forwarding program, the server-side forwarder sends the response data to the browser for display.

Intermediate forwarding program includes server-side intermediate forwarding program and target - side intermediate forwarding program. The server-side forwarding program has two socket threads. One thread is used to receive the connection request of the target machine specially, and maintain the socket connection with the target machine constantly open, at the same time, the target machine information is written in the XML file to save; The other thread receives the request from the client browser, parses and determines which target device the request is sent to, based on the socket number corresponding to the target, and then sends the request data to the corresponding target machine through the previously established socket, and waits to receive the reply data from the target machine. Finally, the data returned by the target machine is sent to the client browser for display. The flow of the server-side forwarding program is shown in Figure 4.

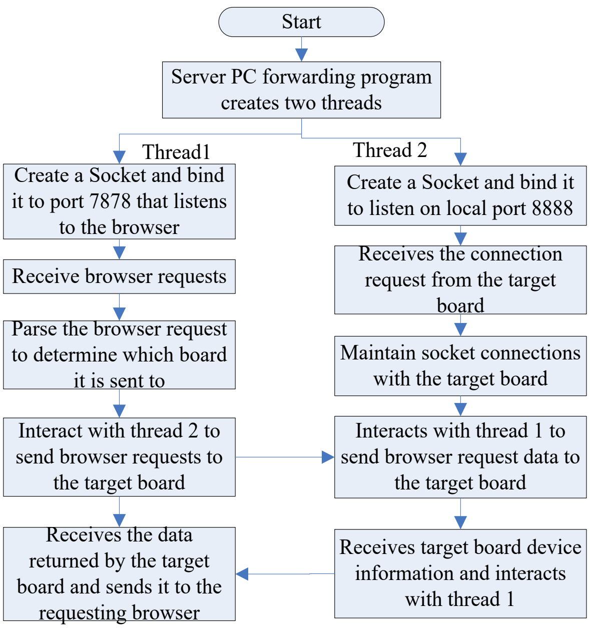

Figure 4. Flow chart of the server-side forwarding program

The target board side forwarding program also uses two socket threads. One of which is specially used to maintain continuous connection with the socket of the server-side forwarding program and send the target machine information to the server-side forwarding program regularly. The other thread is responsible for receiving the request data from the server and forwarding the received request data to the BOA server of the target device, then waiting for the BOA server to parse the request and receive the BOA data. Finally, the BOA replied data is forwarded to the server-side forwarding program. The process of the target board side forwarding program is shown in Figure 5.

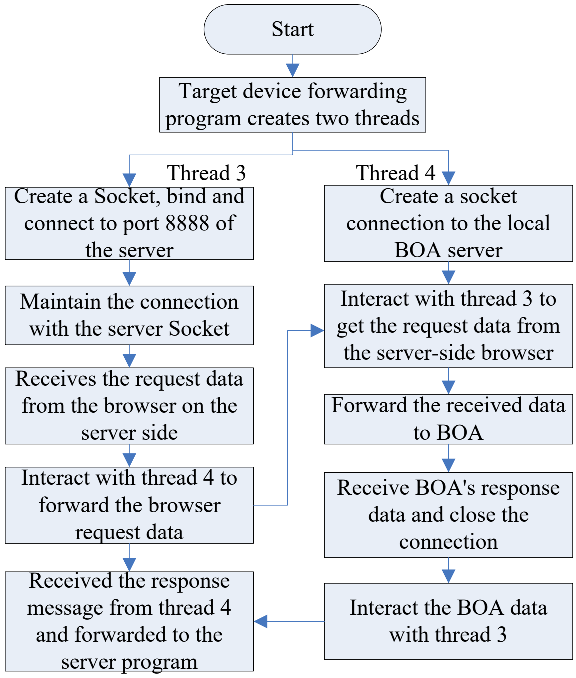

Figure 5. Flow chart of the target board side forwarding program

Through the server-side forwarding program and the target-side forwarding program, the internal network data can be accessed by the external network. The communication block diagram of the overall forwarding program is shown in Figure 6.

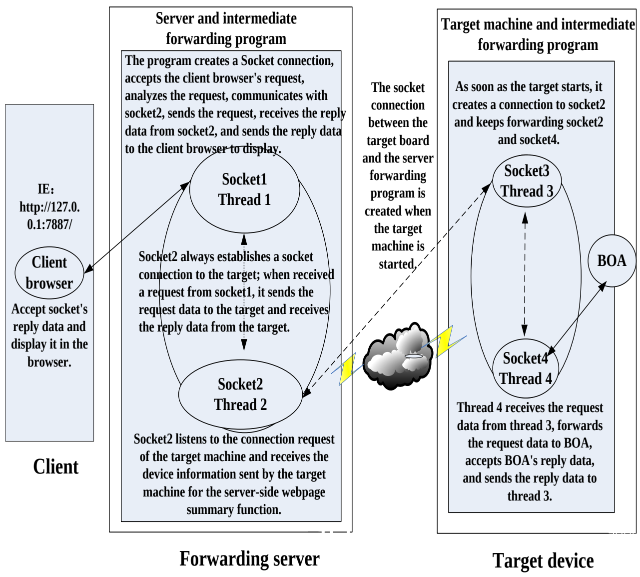

Figure 6. Communication block diagram of the overall forwarding program

The specific process of corresponding to the above forwarding program is as follows. First, the target board forwarding program starts as soon as it is created, takes the initiative to establish a socket connection with the server-side forwarding program, and keeps the link uninterrupted. That is, the Socket2 and Socket3 in the figure above are always connected. Then the server-side forwarding program listens to the request from the client browser and forwards the request to the forwarder on the target board forwarding program;

Socket1 is always listening to the request by the browser and forwards the request to Socket2. The forwarding procedure of the target board Socket4 forwards the request to the BOA server, which replies to the corresponding data after receiving the request to the target board forwarding procedure. The target board forwarding program forwards the response data to the server-side forwarding program, and the server-side forwarding program sends the response data to the browser, which displays the response data.

3.2 Network Model Analysis

The key technologies of this network model mainly include four points: one is to use TCP/IP socket network programming to realize the remote client can access the server PC, while the target board device can connect to the server PC; Second, data forwarding is realized through network forwarding program to achieve the information interaction between the target board and the client. The third is to meet the requirements of accessing multiple embedded devices (more than 2000); Fourthly, the communication link of the whole forwarding procedure is maintained continuously.

Analyze the response time of this network model: according to the process of information interaction between interfaces in Figure 3, analyze the specific time needed for each step to realizing. If the browser request is 570 bytes and the response data is 4KB, it takes about 30ms to go from the browser to the server-side PC forwarding program and about 700ms for a request to go from the server-side PC forwarding program to the target board forwarding program. The response time from target board to BOA server is about 50ms, from target board to server PC is about 2S, and from server PC to browser is about 40ms. Regardless of the BOA server interaction time with the monitor, the time between the browser sending the request data and receiving the reply response data is about 3 seconds. Therefore, the response time of the network model can meet the user's perception time.

The advantages of this network model: first, the network model does not need an additional hardware circuit. It only needs to use the intermediate forwarding program (server-side PC intermediate forwarding program and target computer intermediate forwarding program). At the same time, there is no need to set up a WEB server on the server PC, and no additional software needs to be installed on both the server PC and the target device. Second, the intermediate forwarding program of this network model has good portability, high stability and reliability, and low CPU resource consumption. This network model has high application value in the Internet of things applications such as communication base station equipment, outdoor monitoring equipment, and outdoor collection equipment.

3.3 Network Model Implementation

The server-side forwarding program is implemented in VC++ language. The server-side forwarding program first establishes and maintains socket communication with the target board, then listens to the client browser's request and forward the request to the target board upon receiving it. Wait for the reply from the target board, and then send the reply information back to the browser for display. The target board side forwarding program is developed with C language. As soon as the target board forwarding program starts, it will initiate a socket connection with the server-side forwarding program and maintain the contact. Then it receives the request from the server-side forwarding program, sends the request to the BOA server, receives the BOA server's data reply, and forwards the request-reply data to the server-side so that the server-side can forward it to the browser display. The flow chart of the realization of the specific forwarding program is shown in Figure 7.

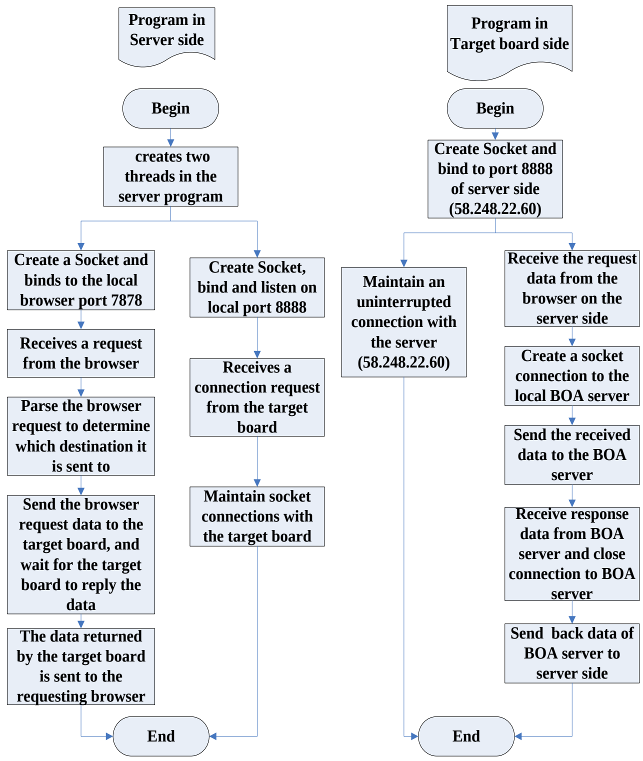

Figure 7. Flow chart of the forwarding program

In the communication process of remote data forwarding, we use two threads in the server-side forwarding program. One thread is used to receive the connection request of the target board, keep the socket connection with the target board, and write the information of the target board to the XML file. Another thread receives the browser's request, parses and determines which target board the request was sent to, then sends the request data through the socket that has been established previously to the corresponding target board, and waits to receive the data replied by the target board. It will send the data replied by the target board to the browser for display.

The target board side forwarding program also uses two threads. One thread is specially used to maintain an uninterrupted connection with the socket of the server-side forwarding program and send the device information to the server-side forwarding program regularly. The other thread is responsible for receiving the request data from the server-side and forwarding the requested data to the BOA server, waiting for the BOA server to parse the request and receive the BOA's reply data, and delivering the BOA's reply data to the server-side forwarding program.

3.4 Exception Analysis and Solution

The anomaly analysis and processing solution of the forwarding program in this network model mainly include the following three aspects

1. Power off or restart in the target board. The target board program regularly sends data packets to the server-side to maintain continuous socket connection. If all the data packets of the maintenance link fail three times in a row, the socket will be re-created, and data packets will be sent regularly to maintain the socket link.

2. Power off or restart the server. The server-side program listens to the remote target machine's connection request again, clears the previous connection data, records the existing target board connection record again, and writes the current data into the XML file while removing the original data in the XML file.

3. The browser failed to reply to the complete data in one request. The designer needs to check the integrity of a data request. If the data transmission is not complete, then re-transmit a data request to the target board through such measures to ensure that the model achieves external network access network reliability and stability.

4 Experimental Design and Verification

To test and verify the performance of the network forwarding model and the forwarding program. This paper designed and tested the maximum load connection number of the network model, the success rate of the external network connection, the server CPU utilization rate, the connection response time, and other parameters [21].

4.1 Experimental Environment

According to the network model to solve the external network access the Intranet scene to build the experimental test platform. The experimental testing platform mainly includes a host server, embedded devices containing GPRS/3G/4G/5G communication modules, laptops, etc. The specific experimental test platform environment is shown in Figure 8.

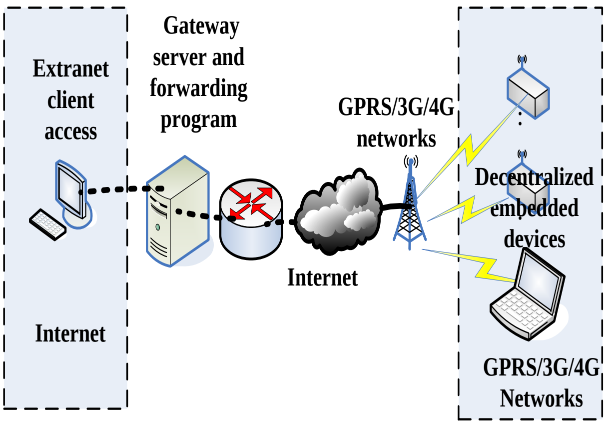

Figure 8. Experimental testing platform

The host server uses an ordinary PC with a 4-core Intel Xeon processor, Intel E7520 memory controller, 8GB DDRII memory, and gigabit Ethernet card. Here the host server needs to have a public network IP address. The basic configuration information of the embedded device is as follows: ARM processor, main frequency 180 MHz, memory 512MB, GPRS/3G/4G/5G modem module. During the test, to simulate a large number of embedded devices, the software is used in the laptop to simulate the generation of a large number of embedded device connections. The detailed parameters of the experimental environment are shown in Table 1.

Table 1. Experimental environmental parameters

Devices	Main Parameters
Host server	CPU: Intel Xeon Quad -Core CPU 3.00GHz Memory:8GB Operating system: Microsoft Windows XP Professional SP3 Includes a public IP address
Embedded device	CPU:ARM9 frequency: 180 MHz, Memory:512MB Booterstrap:AT91Bootstrap1.2 Uboot: u-boot-1.3.4 Kernel: linux-2.6.30 Fs:Yaffs2 Including MODEM interface
MODEM module	GPRS/3G/4G/5G MODEM GSM:900/1800Mhz TD-SCDMA:B34/B39 WCDMA:B1/B8 LTE-FDD:B1/B3/B5/B8 LTE-TDD:B38/B39/B40/B41

4.2 Maximum Load and CPU Utilization Testing

To apply this network model to practical production, it is necessary to test and verify the maximum number of load connections that the network model can undertake. Because the specific industrial application needs to consider the cost, this section designs experiments to test the maximum number of load connections on the main server. The particular configuration of the main server in this experiment is as described above. It is difficult to test the number of connections of embedded devices with actual embedded devices, So here we use software to simulate the connection that generates the remote embedded device. The maximum load connection number and CPU usage of the primary server are tested. The experimental results are shown in Figure 9.

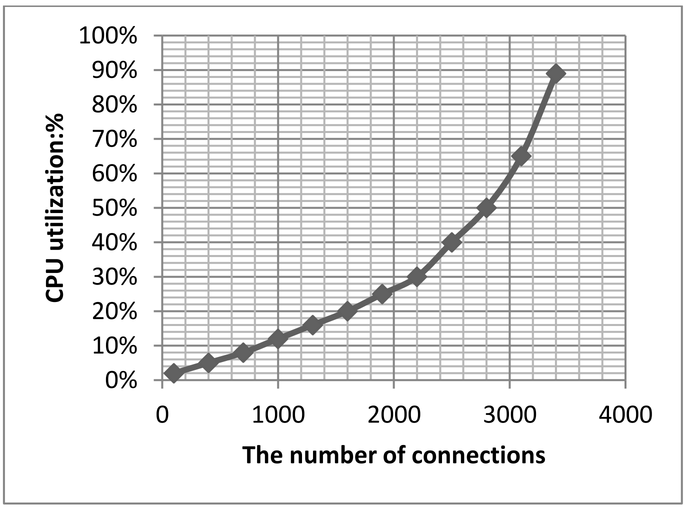

Figure 9. Load connections and CPU utilization

In the figure above, the abscissa represents the number of embedded device connections, and the ordinate represents the CPU utilization of the primary server. It can be seen from the experimental test results that when the number of embedded device connections is below 2000, the CPU utilization of the main server is low. When the number of embedded device connections exceeds 3000, the CPU utilization increases rapidly. Here, according to the CPU utilization rate of less than 60%, the load of this test platform can be up to 3000 embedded devices. This load is sufficient for general area device management. In addition, the configuration of the host server in this test platform is an ordinary PC. If the dedicated server is used as the main server, it can improve the maximum load.

4.3 Connection Success Rate Testing

Users should also focus on the embedded device connection success rate. Therefore, an experiment was designed to test the connection success rate of the device in the network model. In the experiment, 5, 10, 15, 20, 25, 30, 35, 40, 45, 50 embedded devices were used to connect to the host server simultaneously and tested the connection success rate of the host server. The device connection success rate of this network model is shown in Figure 10.

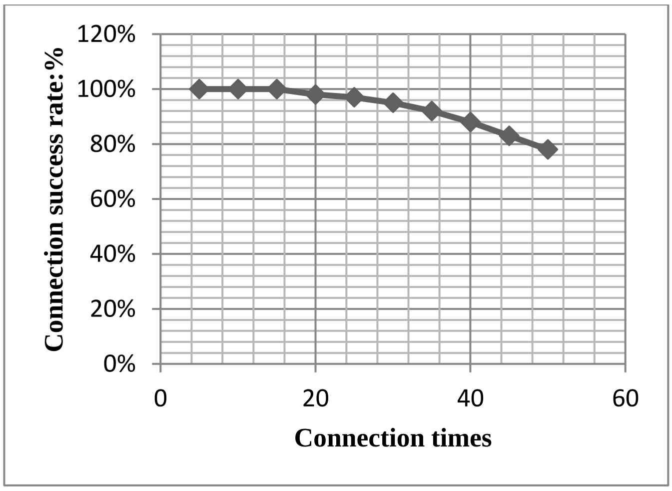

Figure 10. Device connection success rate

In the figure above, the horizontal coordinate is the number of equipment connected at the same time, and the ordinate represents the connection success rate of the device. It can be seen from the experimental test results that when the number of concurrent device connections is around 20, the success rate of the device connection is basically 100, but when the number of simultaneous connections increases, the success rate of connection will decrease to a certain extent.

4.4 Response Time Testing

Another important factor in measuring the implementation of Intranet access is the sense of user experience. The response time of user access is one of the most direct requirements of user experience. Tested the average response time of devices accessed by different concurrent users in the experiment. In the experiment, the average response time was tested when concurrent users were 50, 100, 150, 200, 250, 300, 350 and 400, respectively. The test results for concurrent users and response time are shown in Figure 11.

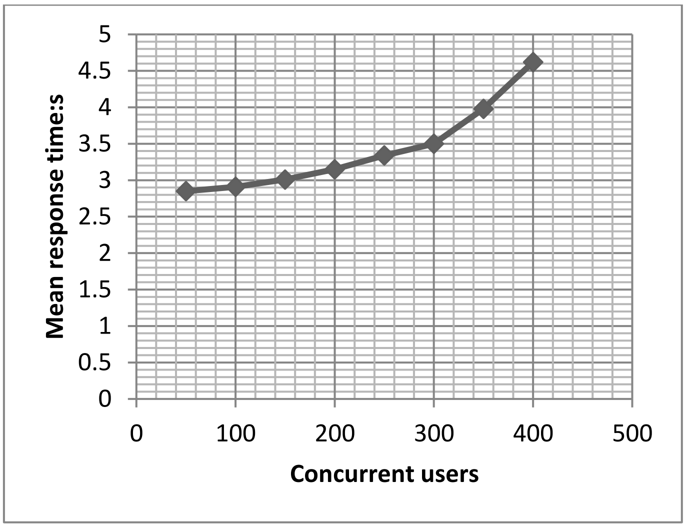

Figure 11. Response time testing

The abscissa represents the number of concurrent user visits, and the ordinate represents the average response time in the figure above. It can be seen from the experimental test results that when the number of concurrent users is around 200, the average response time is around 3S, which can meet the requirements of users.

4.5 Algorithm Complexity

The algorithm complexity of the network model of the external network accessing the internal network is mainly concentrated in the intermediate forwarding program. The intermediate forwarding program establishes a thread with each embedded device in the intranet and maintains the information interaction with the remote embedded device in the intranet through this thread. Therefore, the algorithm time complexity of the network model is mainly related to the network bandwidth n of the embedded device and the distance s of the embedded device. The time complexity can be expressed as O (n * s). The spatial complexity is mainly related to the connection number m of remote embedded devices so that the spatial complexity can be expressed as O (m). At the same time, through the previous test of connection success rate and response time, the results show that the network model has good robustness and applicability.

5 Conclusions

For the IoT, more and more embedded device is deeply into our daily lives to better manage and monitor these devices located in the remote internal network and solve the problem of IP deficiency in the public network. In this paper, a network model of Internet accessing Intranet is studied and implemented to realize Internet access function. In this paper, an experimental test platform for external network access to the internal network is built to verify and test the reliability and stability of the network model and the forwarding program. The performance parameters of the network model are tested, such as the maximum load, CPU utilization rate, device connection success rate and user access response time. It can see from the test results that this network model can meet the network scale of at least 3000 devices of the internal network, so that it can meet the application scenarios of most external networks accessing the internal network. Therefore, this network model can effectively solve the problem of external networks accessing internal networks. It has high practical value and application prospects in communication base station equipment, outdoor monitoring equipment and outdoor acquisition equipment.

Acknowledgement

This work was supported by the Province Natural Science Foundation of Hunan (2019JJ50477); This work was supported by the Project of Social Science Achievements Evaluation Committee of Hunan Province(XSP20YBC199); This work was supported by Huaihua Science and technology Innovation plan project (2021R3130); This work was supported by the National Natural Science Foundation of China (62172182); This work was supported in part by the Huaihua University Double First-Class initiative Applied Characteristic Discipline of Control Science and Engineering.

References

[1] S. Vimal, R. H. Harold, S. C. Kadry, H. V. Long, Y. Nam, IoT based smart health monitoring with CNN using edge computing, *Journal of Internet Technology*, Vol. 22, No. 1, pp. 173-185, January, 2021.

[2] P. P. Zhou, *Research on High-Robustness NAT Traversal Technology Based on PPTP*, Master's thesis, Nanjing University of Information Science & Technology, Nanjing, China, 2015.

[3] H. J. Qing, C. X. Liang, L. P. Chen, A New Method of VPN Based on LSP Technology, *Proceedings of the 2017 2nd Joint International Information Technology*, Mechanical and Electronic Engineering Conference, Chongqing, China, 2017, pp. 216-221.

[4] S. L. Dirk, Network Models for the Industrial Intranet: A Survey of Requirements, Challenges, and the State of the Art, *2016 IEEE 21ST International Conference on Emerging Technologies and Factory Automation*, Berlin, Germany, 2016, pp. 110-119.

[5] Y. Mai, J. Y. Chen, M. Lin, Design of load-based power saving and scheduling scheme integrating real-time and non-real-time services in WiMAX networks, *Journal of Internet Technology*, Vol. 21, No. 5, pp. 1263-1275, September, 2020.

[6] M. Lall, H. Jeske, O. Kogeda, A Real-Time Plagiarism Detection Tool for Computer-based Assessments, *Journal of Information Technology Education: Innovations in Practice*, Vol. 17, No. 2, pp. 23-35, February, 2018.

[7] D. S. Purnomo, Sucherly, Y. S. Suryana, D. Sari, The Effect of Business Partnership and Innovation Management to Business Performance of Business Units of Multiplay Provider in Indonesia, *Academy of Strategic Management Journal*, Vol. 17, No. 4, pp. 71-82, April, 2018.

[8] F. Y. Fan, T. H. Fan, Y. M. Mao, Design and Implementation of NAT Traversal based on SCDMA Access Gateway, *Journal of High Speed Networks*, Vol. 23, No. 4, pp. 341-359, April, 2017.

[9] L. L. Leonardo, R. Dario, P. L. Salvatore, B. Dave, M. J. Damjan, P. F. Pierre, High-speed Data Plane and Network Functions Virtualization by Vectorizing Packet Processing, *Computer Networks*, Vol. 149, pp. 187-199, February, 2019.

[10] K. Treseangrat, *Performance Analysis of Defense Mechanisms Against UDP Flood Attacks*, Master's thesis, Unitec Institute of Technology, Auckland, New Zealand, 2014.

[11] T. Kozik, P. Kuna, S. Marek, P. Arras, Remote Experiments, from Internet to Intranet, *Edukacja-Technika-Informatyka*, Vol. 13, No. 3, pp. 328-332, June, 2015.

[12] C. F. Lai, H. Y. Weng, H. Y. Chou, Y. M. Huang, A novel NAT-based approach for resource load balancing in fog computing architecture, *Journal of Internet Technology*, Vol. 22, No. 3, pp. 513-520, March, 2021.

[13] M. Yoon, *Securing Computer Networks: Access Control Management and Attack Source Identification*, Ph. D. Thesis, University of Florida, Florida, America, 2008.

[14] L. Brian, V. Roman, R. Franklin, J. Paul, Situational Awareness based Risk-adapatable Access Control in Enterprise Networks, *Cryptography and Security*, arXiv:1710.09696, October, 2017.

[15] A. Romanova, M. Kolbanev, N. Verzun, A Two-phase Model of Information Interaction in a Heterogeneous Internet of Things Network at the Last Mile, *Proceedings of the 12th Majorov International Conference on Software Engineering and Computer Systems, Saint Petersburg*, Russia, 2020, pp. 78-89.

[16] N. Maroua, A. Bekrar, A. Giret, O. Cardin, D. Trentesaux, A Multi-agent Model for the Multi-plant Multi-product Physical Internet Supply Chain Network, *Service Oriented, Holonic and Multi-Agent Manufacturing Systems for Industry of the Future*, Paris, France, 2020, pp. 435-448.

[17] F. C. Andrea, V. Fabrizio, Robustness and stability of enterprise intranet social networks: The impact of moderators, *Information Processing and Management*, Vol. 53, No. 6, pp. 1287-1298, June, 2021.

[18] M. Takada, M. Namiki, End-to-End Delay Model for Remote Surveillance over Internet and Mobile Networks, *2018 International Symposium on Intelligent Signal Processing and Communication Systems (ISPACS)*, Ishigaki, Japan, 2018, pp. 226-230.

[19] Š. Mandik, T. Pevny, *Mapping the Internet: Modelling Entity Interactions in Complex Heterogeneous Networks*, Master's thesis, Faculty of Electrical Engineering, Czech Technical University in Prague, Prague, Czech, 2020.

[20] M. K. Elmezughi, T. J. Afullo, An Efficient Approach of Improving Path Loss Models for Future Mobile Networks in Enclosed Indoor Environments, *IEEE Access*, Vol. 9, pp. 332-345, August, 2021.

[21] A. Dafeur, B. Cousin, R. Ziani, Multicast routing and allocation of wavelengths in a WDM network with splitters and converters, *International Journal of Internet Protocol Technology (IJIPT)*, Vol. 14, No. 2, pp. 120-130, July, 2021.

Biographies

Xian Zhang received a Master's degree in software engineering from the College of Computer Science and Electronic Engineering, Hunan University, China, in 2011. From 2013 to 2021, he has been a faculty with the School of Computer Science and Engineering at Huaihua University, China. He is currently an associate professor in computers. His research interests include embedded systems, network management and network measurement, data processing and analysis, data mining, knowledge graph, Graph Processing Accelerators.

Pengju Tang graduated from Guizhou University, for the degree of Master. His research interests include network management, network measurement.

Dong Yin graduated from Guizhou University, for the degree of Master. His research interests include Data processing, auxiliary engineering, network application, image processing, information security, etc.

Taiguo Qu graduated from the College of Computer Science and Electronic Engineering, Central South University, for the degree of Ph. D. His research interests include machine learning and bioinformatics.

Yiwen Liu graduated from Xiamen University, for the degree of Master. His research interests include program language analysis, software test, digital image processing, etc.