Design and Implementation of the CSNS-II Accelerator Control Network Management System

Authors: Xue Kangjia, Zhang Yuliang, Zhu Peng, Wu Xuan, Wang Lin, Li Mingtao, He Yongcheng, Cheng Sinong, Peng Na

Date: 2025-03-28T00:00:00+00:00

Abstract

As the scale and complexity of large-scale scientific facilities based on particle accelerators continue to increase, their control networks face challenges such as a surge in equipment quantity, difficult security management and control, and low maintenance efficiency. To address these issues, a control network management system for large-scale accelerators has been designed and developed. The system implements three core functions: unified management of control network IP addresses, automatic collection of network dynamic information, and network access control. Through a centralized application and approval mechanism, IP conflicts are avoided; based on switch operation data, real-time monitoring of device online status and precise physical location tracking are achieved; an IP-port binding scheme is employed to ensure control network access security. The system is designed and implemented using a Web architecture, with the frontend developed based on the Vue.js framework, the backend employing a hybrid Node.js and Python technology stack, and MongoDB database selected for data storage. The system has been successfully deployed and is operating stably in the CSNS accelerator control network, effectively resolving security risks and maintenance challenges in network management, and laying the foundation for CSNS-II network management.

Full Text

Preamble

Vol. XX, No. X, XXX 20XX
NUCLEAR TECHNIQUES

Design and Implementation of the CSNS-II Accelerator Control Network Management System

XUE Kangjia1,2, ZHANG Yuliang1,2,3, ZHU Peng1,2,3, WU Xuan1,2,3, WANG Lin1,2,3, LI Mingtao1,2,3, HE Yongcheng1,2,3, CHENG Sinong1,2, PENG Na1,2

1 Institute of High Energy Physics, Chinese Academy of Sciences, Beijing 100049, China
2 Spallation Neutron Source Science Center, Dongguan, Guangdong 523803, China
3 University of Chinese Academy of Sciences, Beijing 100049, China

Abstract

With the continuous expansion in scale and complexity of large scientific facilities based on particle accelerators, their control networks face significant challenges, including a surge in device numbers, difficulties in security management, and low maintenance efficiency. To address these issues, a control network management system tailored for large-scale accelerators was designed and developed. The system implements three core functions: unified IP address management for the control network, automatic collection of dynamic network information, and network access control. IP conflicts are prevented through a centralized application and approval mechanism; real-time monitoring of device online status and precise physical location identification are achieved based on switch operation data; and control network access security is ensured through an IP-port binding scheme. The system adopts a Web architecture for design and implementation, with the frontend developed based on the Vue.js framework, a hybrid Node.js and Python technology stack for the backend, and MongoDB database for data storage. The system has been successfully deployed and is operating stably in the CSNS accelerator control network, effectively resolving security risks and maintenance challenges in network management, laying a solid foundation for CSNS-II network management.

Keywords: Control network; Control system; Accelerator; Network management; CSNS
Classification: TL503.6

[**Background**]: The increasing scale and complexity of large scientific facilities have led to significant challenges in control networks, including a proliferation of connected devices, security management difficulties, and inefficient maintenance processes.
[**Purpose**]: To address these challenges, a comprehensive control network management system was designed and implemented, specifically tailored for large-scale accelerator facilities.
[**Methods**]: The developed system integrates three core functionalities: unified IP address management, dynamic network information acquisition, and network access control. A centralized application and approval mechanism was implemented to eliminate IP conflicts. Real-time monitoring of device status

and precise physical location identification were achieved through continuous collection of switch operation data. Network security was enhanced through an IP-port binding scheme that strictly controls access to the control network. The system architecture employs a web-based approach with a Vue.js frontend framework, a hybrid Node.js and Python backend technology stack, and MongoDB for data persistence.
[Results]: The system has been successfully deployed and is running stably in the CSNS accelerator control network. It effectively resolves security vulnerabilities and maintenance challenges in network management.
[Conclusions]: The implemented management system significantly enhances both the operational security and maintenance efficiency of accelerator control networks. It establishes a robust foundation for CSNS-II network management and offers a promising solution for improving control network administration across various large-scale scientific facilities.

Key words: Control network; Control system; Accelerator; Network management; CSNS

As accelerators grow in scale and complexity, the challenges of control network management and maintenance increase accordingly. Taking the China Spallation Neutron Source (CSNS) accelerator as an example, its control network currently has 650 online devices, and this number is expected to exceed 1,500 network devices after the completion of CSNS-II. Against this backdrop of rapid device growth, traditional network management models can no longer meet the demands for efficient and secure operation, making it imperative to establish a comprehensive network management system to prevent potential risks.

The current CSNS accelerator control network management faces multifaceted challenges. At the security level, the lack of a unified access approval mechanism allows unauthorized personnel to connect devices to the network at will, increasing security vulnerabilities. Additionally, after the control system allocates IP address ranges to each subsystem, the subsystems assign IP addresses to their devices independently, which often leads to IP conflicts and compromises network stability. At the maintenance level, administrators struggle to quickly obtain detailed information about devices corresponding to specific IP addresses, and the lack of data associating IPs with network ports complicates troubleshooting and network maintenance.

These challenges will be further exacerbated with the expansion of the CSNS-II project and the significant increase in network devices. To address these issues, this paper proposes and implements the CSNS-II accelerator control network management system. The system prevents IP conflicts through a centralized application and approval mechanism; achieves real-time monitoring of device online status and precise physical location identification based on switch operation data; and ensures control network access security through an IP-port binding scheme. These functionalities aim to resolve the current challenges in CSNS accelerator control network management and provide strong support for the efficient and secure operation of CSNS-II.

1. Overall System Design

1.1 CSNS-II Accelerator Control Network Overview

The CSNS-II accelerator control system is a complex large-scale distributed control system whose primary task is to enable operators to control system devices through human-machine interface devices, performing real-time control according to the physical requirements of CSNS-II design. This involves precise positioning and control of high-speed proton beams. To achieve this, the control system monitors and controls equipment distributed across multiple areas, including the linear accelerator, beam transport lines, and storage ring [1,2]. The control of these devices is performed by computers connected via Ethernet for real-time monitoring, enabling collaborative work to achieve device control.

[Figure 1: see original paper]

The CSNS-II control network employs a star-shaped two-tier structure, as shown in Figure 1. The core layer is located in the server room of the central control room, where two core switches are deployed and interconnected via high-speed fiber links in a redundant configuration. The Virtual Router Redundancy Protocol (VRRP) is used to implement hot standby, ensuring high system availability. When one core switch fails, the other can seamlessly take over data traffic [3,4]. The access layer is distributed across various equipment halls, responsible for connecting terminal devices. After the completion of CSNS-II, the network is expected to have 50 access switches and over 1,500 network devices. Each access layer switch connects to the core layer switches via dual 10G uplinks, forming fully redundant uplink paths. This dual-uplink design ensures that even if one uplink path or one core switch fails, data can still be transmitted normally through the alternative path, guaranteeing stable operation of the entire control network. The 10G links between core and access layers employ link aggregation technology, providing not only 20G total bandwidth capacity but also intelligent traffic distribution. Given the large number of access devices, the control network is divided into multiple Virtual Local Area Networks (VLANs), with IP segments further subdivided within each VLAN and allocated to various subsystems. This VLAN segmentation scheme effectively isolates broadcast traffic from different subsystems, reduces network congestion risks, and enhances network security [5,6].

1.2 Overall Design of the Network Management System

Although numerous commercial network management software solutions exist, they are primarily designed for conventional campus network management and cannot meet the special requirements of accelerator network management. For instance, compared to campus office computers, most accelerator network devices—such as control, power supply, RF, and vacuum systems—lack the capability to independently request network access. Moreover, due to the requirement for continuous accelerator operation, MAC address binding schemes are unsuitable as they would make device replacement difficult and potentially affect

accelerator operational efficiency. Additionally, while campus networks typically only need to record user and device names, control networks require more comprehensive information about connected devices.

To address these special requirements, the network management system described in this paper aims to enhance control network security and maintenance efficiency without impacting accelerator operations. In terms of security, an IP-based binding access scheme was designed based on the characteristics of accelerator equipment. When devices are replaced, they can access the network simply by configuring the same IP address without requiring re-application for network access, thus ensuring uninterrupted accelerator operation. For maintenance efficiency, a unified management scheme for control network IP addresses was designed, which records detailed device information and combines static device information with dynamic network information to enable rapid device location and online status monitoring.

The network management system adopts a three-layer architecture design, including a data presentation layer, data processing layer, and device interaction layer, as shown in Figure 2 [Figure 2: see original paper]. This layered architecture not only improves system modularity but also enhances scalability and maintainability. System implementation primarily involves Web application development and network management technologies, with the overall technical scheme illustrated in Figure 3 [Figure 3: see original paper].

[Figure 2: see original paper]

The data presentation layer employs a Web architecture as the user interaction interface, integrated with WeChat and email messaging systems for system notifications. Compared to traditional desktop applications, Web applications are easier to deploy and update, and can run directly on different operating systems and devices. Users only need a standard Web browser and network connection to access the system anytime. The system supports querying IP information, filling out and approving IP address usage applications, viewing switch information, port information, and network segment information through the browser. For Web frontend technology selection, the system adopts the Vue.js framework as the primary technology stack. Vue.js is a progressive JavaScript framework that has become one of the mainstream frontend development frameworks due to its lightweight nature, high performance, and component-based development model [7]. To improve development efficiency and user experience, the system integrates the PrimeVue component library as the UI implementation solution. PrimeVue is a UI component library designed specifically for Vue.js, providing feature-rich and customizable components that simplify the implementation of complex UIs while ensuring system interface consistency and professionalism [8].

The data processing layer is designed with a microservices architecture, providing services for authentication, data entry, data retrieval, message push, and calling the device interaction layer. The system employs the Express framework

based on the Node.js runtime environment and the Flask framework based on the Python runtime environment. Node.js, as a high-performance, event-driven JavaScript runtime, can effectively support high-concurrency, low-latency network services [9]. Python has rich libraries and a mature ecosystem for network programming and device interaction, which can significantly simplify the development process of data transmission and interaction with switches, improving development efficiency [10]. The heterogeneous language microservices architecture leverages the advantages of different programming languages, ensuring overall system performance while reserving space for future technology expansion.

The device interaction layer consists of multiple device interaction modules written in Python, primarily implementing network data collection, IP and port binding operations, port enable status modification, and port VLAN ID changes. The network data collection module sets different timed execution cycles based on the type of data collected, while device operation-related modules are executed through passive calls from the data processing layer.

For data storage, the system adopts the MongoDB database, a NoSQL (non-relational) database [11] widely used in modern Web system development due to its high performance, flexible data model, and good horizontal scalability. Unlike traditional relational databases that use predefined table structures, MongoDB' s document storage model is based on JSON-formatted BSON documents [12], which can more flexibly adapt to diverse and dynamically changing data requirements, highly aligning with the needs of this system.

[Figure 3: see original paper]

2. IP Management Scheme Design

2.1 Control Network IP Address Management Scheme

Network devices in the CSNS accelerator adopt static IP configuration. In the previous management scheme, administrators allocated assignable IP segments to each subsystem, and the respective responsible persons would assign IP addresses to their devices independently. This approach had two main problems: First, network administrators could not uniformly manage IP information, making it difficult to directly locate devices when network problems occurred. Second, due to management differences among subsystems, incomplete IP information records accumulated over long-term operation, increasing the risk of IP conflicts.

To solve these problems, this paper implements a unified management mechanism for control network IP addresses, primarily including IP address allocation, information query, usage application, and approval functions.

Administrators pre-import lists of network segments, switches, ports, and assignable IPs into the system. These lists contain preset network information, such as network segment names and corresponding VLAN IDs, each switch' s

name, management IP, and location, and cable numbers corresponding to each port. Before adding new devices to the control network, users must submit a usage application for an unassigned IP address on the IP overview page. The application can only proceed after administrator approval. During the application process, users can view the pre-assigned subsystem for each IP address, thereby maintaining the use of uniformly allocated network segments and IP ranges by each subsystem. The IP address application form supports filling in device name, management information, hostname, purpose, subsystem used, device manufacturer, model, and connected switch name and port information, comprehensively covering the device information required for network management. Among these, device management information is only visible to the applicant, while other information can be viewed by all personnel.

If the filled information does not meet network management requirements, administrators will reject the application to ensure that all devices connected to the control network have complete maintenance information. Considering that users often cannot determine the switch name and port when applying for IP addresses, the system provides a one-click function to update application information based on collected dynamic information after the device is connected to the network. The complete IP address application process is shown in Figure 4 [Figure 4: see original paper].

[Figure 4: see original paper]

2.2 Database and Interface Design

For data storage design, this paper has designed six collections in MongoDB for storing and managing static data related to IP addresses. These collections are used to store IP information, application information, network segment information, location information, switch information, and switch port information. The collections are associated through common fields such as IP address, VLAN ID, switch IP and port connected to the device, and location name. Taking the request IP list as an example, the relationships between collections are shown in Figure 5 [Figure 5: see original paper].

[Figure 5: see original paper]

This system has designed and implemented seven user operation pages, including IP overview, personal IP query, switch information, port information, network segment information, administrator approval, and data import pages. The system finely divides access and operation permissions for different pages based on user privilege levels: the administrator approval and data import interfaces are only available to administrators, with main functions being reviewing user applications and batch data import; the personal IP query page is limited to logged-in users for viewing and modifying personal network device information; the IP overview, switch information, port information, and network segment information pages are open to all users but provide differentiated functions based on user type. Unlogged users can query basic control network operation infor-

mation, logged-in users can submit IP address usage applications, and administrators have the highest privilege to modify any IP information and execute network management operations.

When users access the IP overview page, the backend service program returns a list of available IPs in the control network and related information using MongoDB' s aggregation method according to the relationships shown in Figure 5. The system combines the $match stage in the aggregation pipeline to implement searches by various attributes such as VLAN, IP address, and device name [13]. The IP overview user interface is shown in Figure 6 [Figure 6: see original paper].

[Figure 6: see original paper]

Through the design that combines network information and device information, this system not only achieves unified management of IP addresses but also significantly improves the integrity and traceability of network information. It effectively reduces the risk of IP conflicts while providing convenience for network troubleshooting, greatly enhancing the management efficiency and security of the CSNS accelerator control network.

3. Dynamic Network Information

In accelerator network management, the dynamic network information that needs to be queried mainly includes two parts: device online information and location information. Online information includes device online status, online records, offline records, and MAC address change records, which help administrators determine whether devices are connected to the control network and whether device replacement has occurred. Location information refers to the physical location of network devices, including the connected switch name, port, and equipment hall.

Some tools already exist on the market for collecting dynamic network information, such as open-source network monitoring software like Netdisco, NeDi, and LibreNMS [14], as well as commercial network management software like eSight. These tools can provide good device discovery and IP tracking functions but lack flexibility in custom data collection and have limited configuration capabilities for network devices.

To address the limitations of existing tools, this paper has developed a customized dynamic network information collection program. Customized development offers several key advantages: First, it achieves seamless integration with the existing system architecture. The dynamic network information collected by this system needs to work collaboratively with other functional modules such as IP address management. Through customized development, the collection program directly stores data into the MongoDB database, sharing a unified data structure and access interface with other modules, improving inter-module coupling and data consistency, and providing a solid foundation for subsequent

data analysis and presentation. Second, customized development provides more flexible configuration capabilities. For example, in the network access control module, the system needs to implement differentiated configuration strategies for switches from different manufacturers. Finally, customized development ensures long-term maintainability and scalability of the system. When the network environment changes, the system can quickly respond to new requirements, such as adding support for new device types or adjusting data collection strategies.

3.1 Online Information

This paper has developed two modules to obtain and maintain device online information: an ARP table collection module and an ARP automatic scanning module. These two modules work collaboratively to ensure the system can accurately and efficiently monitor the online status of devices in the network. The following sections detail the implementation principles and working mechanisms of these modules.

There are three main methods for obtaining device online status: The first common approach uses the ICMP protocol to periodically probe target devices. This method is simple to implement but increases network load, and some devices may disable ICMP responses due to security policies, leading to inaccurate data. The second approach is based on network traffic analysis. Its advantage is that it doesn' t require active probing, reducing network interference, but it' s complex to implement, requires dedicated hardware or software deployment, and has difficulty detecting silent devices. The third approach is based on monitoring the core switch' s ARP table. This method doesn' t require active probing of each device, avoids generating additional network traffic, only needs access to the core switch without additional hardware, and is not limited by device type or operating system. The main disadvantage of ARP table-based monitoring is that real-time performance is limited by the ARP table aging time configured in the core switch. For example, the core switches used in this paper have a default aging time of 20 minutes, meaning the fastest update time for the system to obtain IP online status is 20 minutes. Nevertheless, considering that device status changes in control networks typically occur infrequently and most devices remain online for extended periods once connected, this update cycle can still effectively capture most device status changes. Therefore, this time interval meets the requirements of control network management.

Based on the above analysis, this paper adopts the ARP table-based monitoring scheme. This approach not only has broad applicability and can adapt to various accelerator control network environments but also minimizes impact on network performance while effectively monitoring network device status.

The Address Resolution Protocol (ARP) is a key protocol in the TCP/IP protocol suite responsible for mapping IP addresses to physical addresses [15]. When a device needs to communicate with other devices, the communication process follows different mechanisms depending on the target device' s location: For

communication within the same network segment, the sender first queries the local ARP cache; if no corresponding record exists, it sends an ARP request broadcast, and the target device responds with an ARP reply containing its MAC address. For cross-network segment communication, data packets are forwarded through gateways, and ARP interactions between devices and gateways are recorded in the core switch. These interaction processes enable the core switch to accumulate and maintain a dynamic table of IP-MAC mapping relationships for active devices in the network, covering device information connected to various network segments.

The system executes the core switch ARP table collection module every 20 minutes. This module uses the Netmiko library [16] to connect to the core switch via SSH, executes commands to view ARP table entries, and parses the returned text through TextFSM [17]. When an IP address exists in the ARP table and the MAC address is not "Incomplete," the system marks the IP as online; otherwise, it is marked as offline.

To store online status, this paper has designed three MongoDB collections: sw_{{{core}}}*{{arp}}}*, *logs*{{{ip}}}*{{online}}}*, *and logs*{{{mac}}}*{{change}}}*. *After each collection, the system writes online IPs and their corresponding MAC addresses into the sw*{{{core}}}*{{arp}}} collection, where the stored IP addresses represent currently online IPs. Simultaneously, the system compares the current collection data with the previous collection data. If the online status changes (from online to offline or offline to online), the change is written to the logs*{{{ip}}}*{{online}}} collection. If the MAC address mapped to an IP address changes, the change is written to the logs*{{{mac}}}_{{change}}} collection.

To avoid missing silent devices that have no data interaction after connecting to the network, the system executes the core switch ARP automatic scanning module once daily to prompt the core switch to include information for all devices connected to the network in its dynamic ARP table. The ARP table collection scheme is shown in Figure 7 [Figure 7: see original paper]. The program sequentially executes ARP scanning tasks in each network segment. These tasks trigger the core switch' s ARP automatic scanning function, which sends ARP request packets to IP addresses in the network segment where the interface IP address resides, thereby learning the corresponding ARP table entries. When ARP table entries expire, the switch initiates a re-learning mechanism. Once device information is recorded in the core switch's ARP table, as long as the device maintains network connection, the switch will maintain and update these entries through periodic ARP requests, making frequent ARP automatic scanning operations unnecessary. Considering that ARP automatic scanning consumes significant resources on the core switch, scanning operations are scheduled during the early morning hours when the network is relatively idle.

[Figure 7: see original paper]

Based on the collected ARP table data, the system supports querying device real-time online status, MAC addresses, online/offline records, and MAC address

change records through the user interface. The network status change record query page is shown in Figure 8 [Figure 8: see original paper]. Combined with the IP address application information described earlier, online status can be divided into four situations: online, unassigned but online, assigned but offline, and offline. The "unassigned but online" status is abnormal, indicating that an unauthorized device has connected to the control network without approval. In the IP overview page shown in Figure 6, the IP list for these four statuses can be filtered through combined selection of status and allocation dropdown boxes.

The method of monitoring online information based on the core switch ARP table is characterized by high efficiency, low interference, and broad applicability. This approach can not only identify normal device connections and disconnections but also detect unauthorized device access, providing strong support for network security management.

[Figure 8: see original paper]

3.2 Location Information

Network devices in large accelerators are widely distributed across multiple areas, involving numerous equipment management personnel. Due to the typically long construction-to-operation cycle of accelerators and personnel handover factors, accurately locating equipment has become a common challenge in past operation and maintenance processes. To solve this problem, this paper achieves precise location of any network device by collecting dynamic network information and combining it with static information entered by administrators. This method can pinpoint device location to the specific equipment hall, connected switch name, port number, and network cable number, as shown in Figure 9 [Figure 9: see original paper].

[Figure 9: see original paper]

This paper has developed a collection module that automatically collects interface information and MAC address tables from the core switch and all access switches daily. By analyzing the switch interface information and MAC address tables, the system can obtain the connection status, VLAN ID, and MAC addresses of devices connected to each switch port. Combined with the IP-MAC address mapping relationships contained in the core switch' s ARP table, the system can precisely locate the connection relationship between network devices and specific switch ports. Furthermore, since administrators have already entered the equipment hall where each switch is located and the corresponding network cable numbers for each switch port during switch information maintenance, the system can further determine the specific physical location of network devices and their connected network cable numbers. The logic of associating IP addresses with locations through dynamic information in MongoDB is shown in Figure 10 [Figure 10: see original paper], which illustrates two types of data stored in the system: the IP list, switch information, and port information are static information entered by administrators, while the core switch ARP table,

core switch interface information, and access switch MAC tables are dynamic information stored by the collection module.

[Figure 10: see original paper]

This combination of dynamic and static information ensures the accuracy and real-time nature of location information, solving the problem of difficult equipment location tracking in traditional management. The system provides a complete mapping from IP addresses to specific physical locations, improving the accuracy of network device management.

4. Network Access Control Scheme Design

4.1 Network Access Control Scheme Design

In conventional network management, there are three main authentication methods for network access control [18]: 802.1X authentication, MAC authentication, and Portal authentication. 802.1X is an IEEE standard port-based network access control protocol primarily used for authentication in wired and wireless LAN environments. MAC authentication is an authentication method that controls users' network access permissions based on port and MAC address, requiring no client software installation on users' part. Portal authentication, also commonly called Web authentication, is an authentication method based on Web pages.

However, most devices in accelerator control networks, such as power supplies, oscilloscopes, and serial servers, typically do not support 802.1X authentication or Portal authentication because these devices cannot install authentication software or browsers. Considering the requirement for 24-hour uninterrupted accelerator operation, adopting MAC authentication would require re-authentication when devices are replaced, potentially affecting normal accelerator operation.

To address this special requirement, this paper designs an access control scheme based on IP and port binding. The advantage of this scheme is that when devices are replaced, no re-binding is required; they can access the network simply by configuring the same IP address, ensuring both control network security and controllability while guaranteeing continuous accelerator operation.

4.2 Technical Implementation of IP and Port Binding

The implementation scheme for IP and port binding is closely related to switch hardware, with significant differences in function support and configuration methods among devices from different manufacturers. To address this diversity, this paper has developed customized program modules to flexibly adapt to equipment from multiple manufacturers. For example, the CSNS accelerator control network currently uses access switches from Huawei and Ruijie, requiring differentiated configuration strategies.

For Huawei switches, this paper utilizes the IP Source Guard (IPSG) function.

IPSG is a source IP address filtering technology based on Layer 2 interfaces. By enabling IPSG on switch interfaces on the user access side, the switch can inspect IP packets entering the interface and discard packets from illegal hosts, effectively preventing unauthorized devices from accessing the control network.

For Ruijie switches, although they also support IPSG functionality, their IPSG rules require simultaneous binding of IP, MAC, and port, which cannot meet the requirement of binding only IP and port [19]. Therefore, this paper adopts the port security function to implement IP and port binding. This function restricts unauthorized device access to the control network by limiting the source IP addresses allowed to enter switch ports. It should be noted that Huawei switches also support port security functionality but only support MAC and port binding, not IP and port binding.

The system' s IP and port binding process is shown in Figure 11 [Figure 11: see original paper]. To adapt to the characteristics of different manufacturers' devices, this paper has developed modular hardware interaction components for executing various switch operations. When administrators submit an IP and port binding request on the user interface, the system first sends the request to the Node.js backend service. The Node.js service verifies whether the requesting user has administrator privileges. After verification, the request is forwarded to the Flask service, which then calls the corresponding hardware interaction module based on the request type. This module is responsible for establishing connections with the appropriate access switches and executing binding operations. After completing the binding operation, the hardware interaction module reads the configuration data back from the switch and writes this data into the MongoDB database. Finally, the execution results are returned to the user interface, providing administrators with real-time operation feedback.

[Figure 11: see original paper]

In addition to IP and port binding functionality, this paper has also implemented port enable management and port VLAN ID modification functions. These functions follow the same process architecture, differing only in the hardware interaction module called, and therefore will not be elaborated further.

Through this innovative access control scheme, this paper effectively solves the access authentication problem for special devices in accelerator control networks, improving the flexibility and security of network management while ensuring continuous and stable accelerator operation.

5. Conclusion

This paper has designed and implemented a comprehensive control network management system for the CSNS-II accelerator to address the challenges faced by large-scale accelerator control network management, achieving the following objectives: (1) The unified IP address management mechanism effectively resolves network resource conflicts. This system establishes a unified IP ad-

dress application, approval, and allocation mechanism, and records detailed device information for each IP, providing complete foundational data support for subsequent network maintenance and troubleshooting. (2) The integration of dynamic network information significantly improves operational efficiency. By collecting dynamic switch information, this system achieves real-time monitoring of device online status and supports precise location of any network device' s physical position, greatly improving network troubleshooting efficiency. (3) The IP-port binding network access control mechanism effectively enhances network security. Addressing the special requirements of accelerator control networks, this system implements an access control scheme based on IP and port binding, ensuring both network security and continuous stable accelerator operation.

This system has been successfully deployed and is operating stably in the CSNS accelerator control network, effectively resolving long-standing issues such as IP conflicts, difficult device location, and uncontrolled network access. This not only establishes a solid foundation for CSNS-II network management but also helps prevent potential network risks and ensures safe and stable accelerator operation. Furthermore, the system' s design concepts and implementation methods provide replicable experience for network management in other large-scale accelerator facilities.

References

[1] Wang Sheng, Fu Shinian, Qu Huamin, et al. Development and Commissioning for High-intensity Proton Accelerator of China Spallation Neutron Source [J]. *Atomic Energy Science and Technology*, 2022, 56(09): 1747-1759. DOI: 10.7538/yzk.2022.youxian.0591.

[2] Zhang Y, Jin D, Zhu P, et al. The accelerator control system of CSNS[J]. *Radiation Detection Technology and Methods*, 2020, 4: 478-491. DOI: 10.1007/s41605-020-00203-y.

[3] Geng Q, Huang X. VRRP load balance technology simulation practice based on GNS3[C]. *MATEC Web of Conferences*. EDP Sciences, 2018, 228: 03012.

[4] Jia Juan, Wang Binqiang, Yang Shuai. Research and Implementation of a High-availability Method for Core Routers Based on VRRP[J]. *Application of Electronic Technique*, 2007, 33(2): 110-112. DOI:10.3969/j.issn.0258-7998.2007.02.036.

[5] Mehdizadeha A, Suinggia K, Mohammadpoorb M, et al. Virtual local area network (VLAN): Segmentation and security[C]. *The Third International Conference on Computing Technology and Information Management (ICCTIM2017)*. 2017: 78-89.

[6] Yu H, Qin T, Cui L, et al. Design of the network system for Hefei advanced light facility[J]. *Radiation Detection Technology and Methods*, 2025: 1-8. DOI: 10.1007/s41605-025-00541-9.

[7] Vue.js guide[EB/OL]. https://vuejs.org/guide/introduction.html, 2025-3-10.

[8] PrimeVue homepage[EB/OL]. https://primevue.org, 2025-3-10.

[9] Tilkov S, Vinoski S. Node.js: Using JavaScript to build high-performance network programs[J]. *IEEE Internet Computing*, 2010, 14(6): 80-83. DOI: 10.1109/MIC.2010.145.

[10] Mazin A A, Abidin H Z, Mazalan L, et al. Network Automation Using Python Programming to Interact with Multiple Third-Party Network Devices[C] *2023 10th International Conference on Information Technology, Computer, and Electrical Engineering (ICITACEE)*. IEEE, 2023: 59-64. DOI: 10.1109/ICITACEE58587.2023.10277400.

[11] Győrödi C, Győrödi R, Pecherle G, et al. A comparative study: MongoDB vs. MySQL[C] *2015 13th international conference on engineering of modern electric systems (EMES)*. IEEE, 2015: 1-6. DOI: 10.1109/EMES.2015.7158433.

[12] Gu Y, Wang X, Shen S, et al. Analysis of data storage mechanism in NoSQL database MongoDB[C] *2015 IEEE International Conference on Consumer Electronics-Taiwan*. IEEE, 2015: 70-71. DOI: 10.1109/ICCE-TW.2015.7217036.

[13] MongoDB manual[EB/OL]. https://www.mongodb.com/docs/manual/aggregation/, 2025-3-17.

[14] Qin T, Li C, Sun S A, et al. A device information-centered accelerator control network management system[J]. *Radiation Detection Technology and Methods*, 2024, 8(3): 1342-1358. DOI: 10.1007/s41605-024-00459-8.

[15] Plummer D C. An ethernet address resolution protocol[EB/OL]. https://www.rfc-editor.org/rfc/rfc826, 2025-3-17.

[16] Netmiko homepage[EB/OL]. https://ktbyers.github.io/netmiko/, 2025-3-17.

[17] TextFSM homepage[EB/OL]. https://github.com/google/textfsm, 2025-3-17.

[18] 锐捷交换产品实施一本通 [EB/OL]. https://www.ruijie.com.cn/fw/wd/57629/, 2025-3-24.

Note: Figure translations are in progress. See original paper for figures.