VFabric: A Digital Twin Emulator for Core Switching Equipment

WANG Qianglin[1], ZHANG Xiaoning[1], YANG Yi[1],
FAN Chenyu[1], YUE Yangyang[2], WU Wei[2], DUAN Wei[2]

(1. School of Information and Communication Engineering, University of Electronic Science and Technology of China, Chengdu 610000, China;
2. ZTE Corporation, Shenzhen 518057, China)

DOI: 10.12142/ZTECOM.202501012

https://kns.cnki.net/kcms/detail/34.1294.TN.20250319.0908.002.html, published online March 19, 2025

Manuscript received: 2023-10-24

Abstract: The proliferation of heterogeneous networks, such as the Internet of Things (IoT), unmanned aerial vehicle (UAV) networks, and edge networks, has increased the complexity of network operation and administration, driving the emergence of digital twin networks (DTNs) that create digital-physical network mappings. While DTNs enable performance analysis through emulation testbeds, current research focuses on network-level systems, neglecting equipment-level emulation of critical components like core switches and routers. To address this issue, we propose vFabric (short for virtual switch), a digital twin emulator for high-capacity core switching equipment. This solution implements virtual switching and network processor (NP) chip models through specialized processes, deployable on single or distributed servers via socket communication. The vFabric emulator can realize the accurate emulation for the core switching equipment with 720 ports and 100 Gbit/s per port on the largest scale. To our knowledge, this represents the first digital twin emulation framework specifically designed for large-capacity core switching equipment in communication networks.

Keywords: digital twin network; core switch/router; sockets; network emulation

Citation (Format 1): WANG Q L, ZHANG X N, YANG Y, et al. VFabric: a digital twin emulator for core switching equipment [J]. ZTE Communications, 2025, 23(1): 90 - 100. DOI: 10.12142/ZTECOM.202501012

Citation (Format 2): Q. L. Wang, X. N. Zhang, Y. Yang, et al., "VFabric: a digital twin emulator for core switching equipment," *ZTE Communications*, vol. 23, no. 1, pp. 90 - 100, Mar. 2025. doi: 10.12142/ZTECOM.202501012.

1 Introduction

In recent years, with the fast development of information and communication technology (ICT), such as big data, cloud computing, and artificial intelligence (AI), there has been a rise in emerging heterogeneous communication networks, such as the Internet of Things (IoT)[1], unmanned aerial vehicle (UAV) networks[2], and edge networks[3]. These different network forms have complex operation and administration requirements. For example, UAV networks should effectively adapt to dynamic topology due to UAV flight and satisfy the quality of service (QoS) requirements of various types of traffic[4]. Meanwhile, the number of nodes or mobile devices connected to communication networks is increasing explosively, which brings scalability and flexibility challenges. In general, current communication networks have become increasingly complex and difficult to operate and manage.

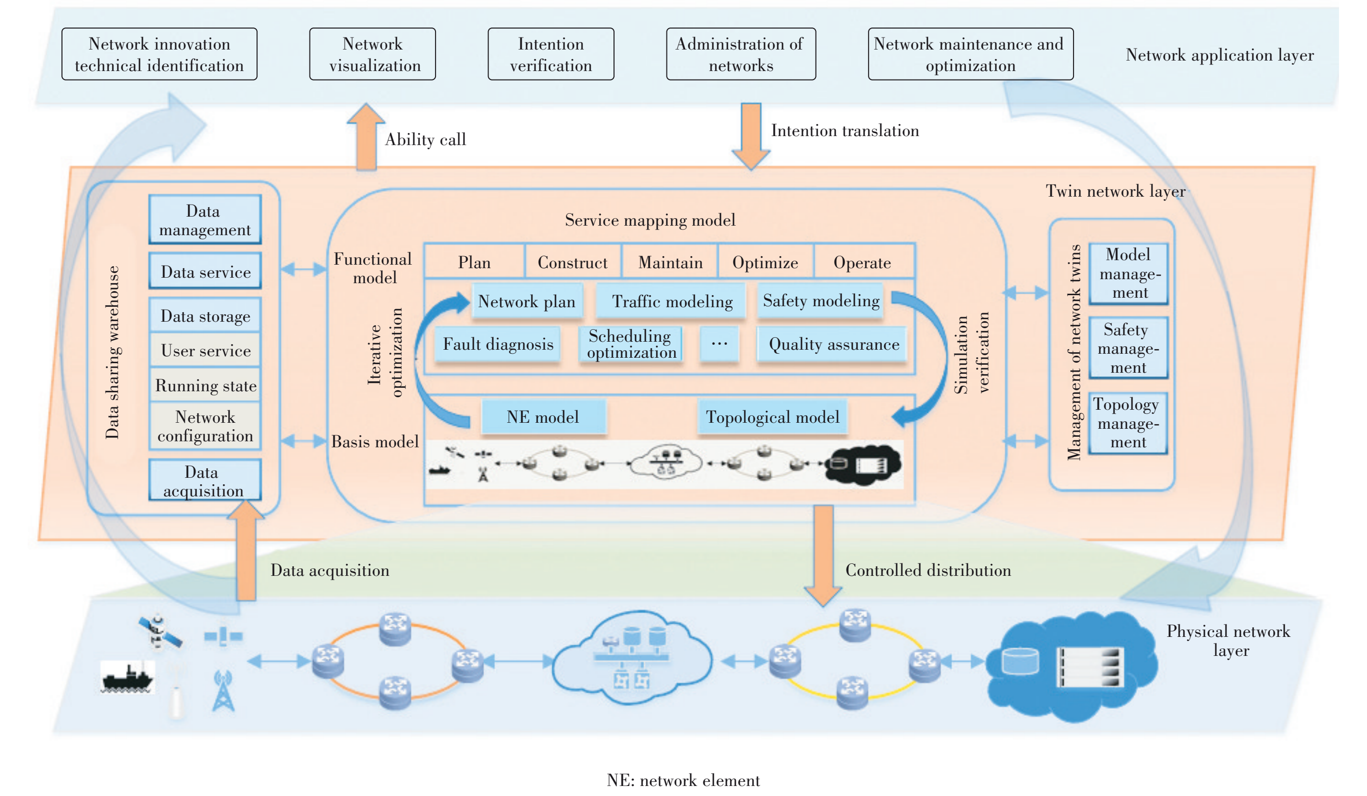
To solve this problem, the digital twin network (DTN) technology has been introduced to facilitate the effective management of communication networks[5]. The concept of digital twin (DT) has been developed in many industries for decades. Today, the DT technology has been widely applied in a large variety of domains, including smart manufacturing Industry 4.0[6], aviation[7], healthcare[8], communication networks[9], and smart grid systems[10]. The basic idea of DT is a digital representation or virtual model of a single physical object. It is a system that focuses on producing a virtual model of a physical entity with high fidelity. Such a system needs to be intelligent and persistently evolving[11]. A DT system generally contains three main modules: a physical object in physical space, a virtual object in virtual space, and the data connection between the two spaces. Leveraging the DTN technology, a high-fidelity emulation system is established for efficiently controlling and managing dynamic and complex communication networks. The DTN emulation system allows network operators to analyze and forecast network performance, develop network solutions, precisely pinpoint network failures, and upgrade networks to accommodate the demands of a growing user base and the integration of new technologies[12].

This work was supported in part by the National Natural Science Foundation of China (NSFC) under Grant Nos. 62171085, 62272428, 62001087, U20A20156, and 61871097 and the ZTE Industry-University-Institute Cooperation Funds under Grant No. HC-CN-20220722010.

According to the definition and four core elements of the digital twin network[13], it can be designed as a "three-layer three-domain dual-closed-loop" architecture (Fig. 1). The three layers include the physical network layer, the twin network layer, and the network application layer, while the three domains correspond to the data domain, model domain, and management domain of the twin network layer. These domains are respectively implemented by the data-sharing repository, service mapping model, and network twin management subsystems. Meanwhile, the "dual-closed-loop" refers to the inner closed-loop optimization based on the service mapping model and the outer closed-loop control, feedback, and optimization based on the three-layer architecture. The physical network layer is a component of the digital twin network, where various network elements exchange network data and control information with the network twin body through the twin southbound interface. The twin network layer is the hallmark of the digital twin network system, containing three critical subsystems: the data sharing repository, service mapping model, and network twin management subsystems. The network application layer controls the digital twin network. Network applications input requirements to the twin network layer through the twin northbound interface and deploy services in the twin network layer via model instances.

In recent years, some research works have been conducted on applying digital twin technology in the field of communication networks (i. e., DTN). DTN is a key enabler for efficient management in communication networks. In particular, the virtual models of DTN can reflect the dynamic characteristics of physical communication networks (e.g., dynamic network topology, growing traffic flows, or devices)[14]. Thus, the network administrators can effectively manage the network considering the network dynamics. For example, the network administrators can easily perform network planning and complete traffic engineering with the help of DTN technology. Therefore, the DTN can accurately forecast the future network state and provide optimal solutions. On the other hand, current DTN research works always study the entire network system (such as 6G networks, vehicular networks, and the IoT)[15] and do not investigate the virtual model of communication network equipment. In fact, how to build an accurate DT virtual model for communication network equipment, especially high-capacity core network switches or routers, is an important issue to be studied. Since the core switch or router has a large-capacity (i.e., a large number of ports, and each port with at least a rate of 100 Gbit/s), achieving high-fidelity emulation for large-capacity communication equipment is a great technical challenge.

To address this issue, we develop a digital twin emulator for large-capacity core switching equipment, called vFabric



NE: network element

Figure 1. Architecture of digital twin network

(short for "virtual switch"), which enables performance troubleshooting and optimization. When a specific system module fails in practice, the corresponding module in vFabric can be analyzed to identify possible faults, quickly locate the problem, and provide fault diagnosis. At the same time, when a system performance bottleneck occurs, various design schemes can be verified and tested in vFabric, and their performance, reliability, and efficiency can be evaluated to ensure the selection of the optimal design. We use the vFabric emulator to simulate the Clos network inside the core switching equipment and build virtual models for switching chips and network processor (NP) chips, which are key components of the core switch/router. For small-scale core switching equipment, the vFabric emulator is implemented on a single physical server. For large-scale core switching equipment, the vFabric emulator is distributed across multiple physical servers interconnected via sockets. Our vFabric emulator achieves accurate emulation of core switching equipment with up to 720 ports and 100 Gbit/s per port on the largest scale. To the best of our knowledge, this is the first study on DT emulation for large-capacity core switching equipment in the field of communication networks. Our work addresses the gap in digital twin applications for communication devices and pioneers the application of DT to communication devices.

The remainder of this paper is organized as follows. Section 2 reviews related work. Section 3 presents the system model and technical challenges, while Section 4 details the design of vFabric. Section 5 presents the implementation of vFabric and Section 6 provides the emulation results. Section 7 concludes the paper.

2 Related Work

Digital twins have been applied in various domains. Here, we only review important areas such as the Internet of Vehicles (IoV), edge networks, and 6G.

1) IoV: The digital twin-assisted decision-making framework for the IoV leverages the integration of communication, sensing, and computing to enhance vehicle collaboration. FU et al. [16] introduced a digital twin technology concept that maps vehicles and roadside infrastructure from physical space to cyberspace to form simulated and reconstructed virtual entities. Ref. [16] also discussed a multi-agent system (MAS) approach to modeling connected autonomous driving, using artificial intelligence algorithms such as deep reinforcement learning (DRL) to enable decision-making. It highlighted the limitations of traditional multi-agent deep reinforcement learning (MADRL) methods, where agents are not connected with each other, and emphasized the importance of enabling agent-to-agent communications. QIN et al. [17] investigated pricing strategies and resource management between vehicles and mobile edge network (MEC) servers when combining digital twins and MEC in the IoV. Ref. [17] also established a dynamic digital twin for the air-assisted IoV to capture time-varying resource supply and demand, enabling unified resource scheduling and allocation.

2) Edge networks: PILLAI et al.[18] implemented a DT system in vehicular networks to enhance edge computing capabilities. The DT collected data from roadside units (RSUs) and optimized task offloading and resource allocation for efficient system load management. DAI et al.[19] integrated digital twin technology into vehicular edge computing networks to improve network management and offloading efficiency. The proposed adaptive digital twin-enabled network utilized virtual representations of the physical network, and a deep reinforcement learning-based offloading scheme was designed to minimize latency[19]. DAI et al. [20] explored a DTN-assisted MEC system, aiming to maximize the number of service requests served by MECs or minimize the load on the cloud. GUO et al. [21] focused on utilizing digital twin technology to enhance the management efficiency of physical entities in edge computing networks. The proposed mechanism included a time-frequency correlation-based activity estimation model and a chaotic particle swarm optimization algorithm for network sensing edge deployment.

3) 6G: NJOKU et al.[22] explored the potential application of digital twin technology in 6G communication systems. They emphasized the need for innovative architectures and enabling technologies to meet the demanding requirements of 6G systems. TAO et al. [23] proposed a software-defined DTN architecture with virtualization for adaptive 6G service response. They also introduced a deep reinforcement learning-based resource orchestration algorithm to optimize service quality. LU et al.[24] focused on integrating digital twin technology with edge networks to address the challenges in building 6G networks with ubiquitous connectivity, low latency, and enhanced edge intelligence.

4) Data communication networks: WEI et al. [25] discussed data-driven routing, a typical network function under the DTN framework, and demonstrated the potential of DTNs to solve traditional network problems. In SDN-based networks, RAJ et al.[26] proposed a data representation-based DTN architecture that integrates knowledge graphs (KGs) for data modeling and storage. ONO[27] et al. presented a scheme called Area-Controlled Mobile Ad-Hoc Networking (AMoND). The digital twin used in AMoND focuses on managing node location information and does not need to fully replicate real-world environments on a computer.

While the related works presented above focus on building DTN models for entire network systems (such as 6G and edge networks) without considering the virtual modeling of communication network equipment, our work specifically addresses high-fidelity emulation of large-capacity core switching equipment. To our knowledge, this is the first study on developing a DT model for core switches/routers.

3 System Model and Technical Challenges

As an important transmission and forwarding device, large-capacity core switching equipment (e.g., the core routers Huawei NetEngine 8000[28], Cicso 8000[29], and T8000[30], and the core switch ZXR10 9900/9900-S[31]) has been widely deployed in various scenarios including backbone networks, campus networks, data center networks, etc. To satisfy the requirement of high bandwidth and intelligent management, large-capacity core switching equipment has the following characteristics: high reliability, scalability, and performance. These features enable enhanced network bandwidth, elimination of bottlenecks, congestion mitigation, and support for diverse traffic interfaces[32]. Therefore, the performance of large-capacity core switching equipment directly impacts the stability and QoS of the whole communication network.

3.1 Logic Model of Large-Capacity Core Switching Equipment

Generally, large-capacity core switching equipment comprises four fundamental modules (Fig. 2): the input module, output module, switching fabric, and control module. The switching fabric is composed of multiple basic switching units[33]. The first three modules constitute the data plane of switching equipment, while the last module belongs to the control plane. The data plane is responsible for packet forwarding, while the control plane is responsible for generating the data path in the switching fabric. During packet forwarding in large-capacity core switching equipment, the control module first calculates the routing path of the incoming packets, i.e., generating the Routing Information Base (RIB). Subsequently, the Forwarding Information Base (FIB) is generated from the RIB and installed in basic switching units of the switching fabric module. This ensures accurate transmission of incoming packets from the input port to the output port through the switching fabric. Inside large-capacity core switching equipment, the switching fabric and its corresponding scheduling algorithm play an important role in switching performance (throughput and delay). At present, the frequently used switching fabrics are single-stage crossbar[34] and multistage Clos networks[35]. Since the number of ports and the read/writing rate of shared memory limit the performance of single-stage crossbar, the multistage Clos network is more broadly used than the single-stage crossbar. When the number of ports increases, compared with the single-stage crossbar, the multistage Clos network can effectively reduce the number of crossover nodes by an order of magnitude. In a multistage Clos network, two basic switching units are connected by only one link, but multiple paths exist between the arbitrary input port and output port[35]. Therefore, the multistage Clos network can support multipath transmission and achieve load balancing for traffic. On the other hand, in the multistage Clos network, the basic switching units of each stage have the same scale (the same number of input and output ports), which means it has good scalability. Thus, small-scale basic switching units can be used to construct a large-scale/capacity Clos network. In summary, the multistage Clos network has the following advantages: modularity, non-blocking with multiple paths, and good scalability. These features make it a preferred choice for commercial off-the-shelf (COTS) core routers or switches[36].

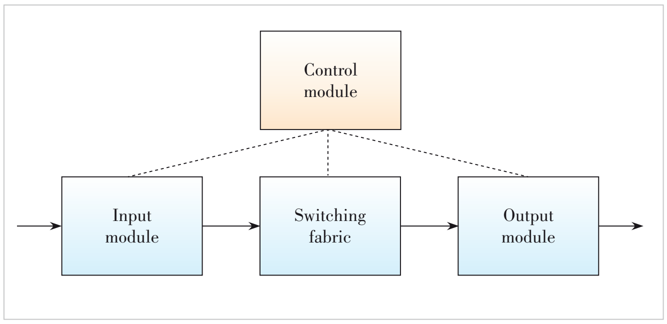


Figure 2. Logic model of large-capacity switching equipment

The three-stage Clos network consists of an input stage, an intermediate stage, and an output stage. The input stage comprises k n×m crossbars, where n denotes the number of input ports of each crossbar and m represents the number of output ports of each crossbar. The intermediate stage consists of m k×k crossbars, where k denotes the number of input and output ports of each crossbar. The output stage is composed of k m×n crossbars, where m denotes the number of input ports of each crossbar and n represents the number of output ports of each crossbar. Therefore, the three-stage Clos network can be denoted as $C(n, m, k)$.

3.2 Hardware Model of Large-Capacity Core Switching Equipment

Based on the logic model of large-capacity core switching equipment, we further illustrate its hardware model. The hardware model of core switching equipment is composed of a main control board, a service board, and a switching board. The main control board corresponds to the control module in the logic model, which generates the routing path inside the switching equipment by the control software, such as the Open Shortest Path First (OSPF) protocol, running on the Linux operating system, and sends the FIB to the switching chips on the data plane. The main control board communicates with the service board and the switching board with socket network communication over Ethernet. The service board comprises the NP chip, the interface chip, and the switching chip. The main work of the NP chip is to perform traffic flow scheduling and QoS management. It is important that each port in the service board is bidirectional, serving as both an input and an output port. The switching board consists solely of the switching chip, which is responsible for high-speed switching. Key components of the switching chip include the Parser (for parsing packet headers), the Forwarding Table (for recording packet forwarding paths), and the Buffer (for storing packets during switching), among others.

3.3 Technical Challenges for DT Model

Since large-capacity core switching equipment is a key enabler for various types of communication networks, e.g., backbone networks, campus networks, and data center networks, it is necessary to develop a DT model for such equipment[37]. This allows network researchers and operators to test and verify the new technology in the DT model and obtain accurate emulation results. Undoubtedly, the DT model for core switching equipment can help design network optimization algorithms, analyze network performance, and forecast network status under new traffic patterns.

The main technical challenge in developing a DT model for high-capacity core switching equipment is accurately emulating the high bandwidth and large traffic environment of core routers/switches. Since the core switching equipment comprises a large number of service boards and switching boards (normally having 20 ports, each supporting 100 Gbit/s at least), its DT model needs to emulate large volumes of traffic, which brings a great challenge for the existing simulation tools (e.g., OPNET, OMNeT++, and NS-3). Another important issue is the scalability of the core switching equipment. By combining varying numbers of service boards and switching boards, larger-scale core switching equipment can be created, which enhances the emulation difficulty of the DT model.

4 Design of VFabric

In this paper, to address the aforementioned challenges, we propose vFabric, a digital twin emulator for large-capacity core switching equipment. The name “vFabric” is derived from “virtual switch”, reflecting its purpose. Our vFabric is a distributed architecture with high scalability and accuracy. It not only simulates the details of the core switch, but also enhances computational capacity by deploying across multiple servers as the scale of hardware equipment increases.

4.1 Testbed Architecture

The large-capacity core switching equipment consists of a main control board, a service board, and a switching board. In this study, we develop vFabric on two servers. The main control board and the switching board are deployed on Server A, while the service board is deployed on Server B. In vFabric, the NP and switching processes emulate the functionality of the chips on actual network cards. Server A supports a maximum scale of 32 switching processes, while Server B supports up to 36 NP processes and 32 switching processes. Each NP process is connected to 20 port threads, each capable of sending and receiving packets at 100 Gbit/s. In vFabric, the packet transmission route is as follows: packets are generated from the ports and sequentially handled by the NP processes in Server A, the switching processes in Server A, the switching processes in Server B, the switching processes in Server A, the NP processes in Server A, and finally the port threads. All switching processes utilize round-robin scheduling to transmit the packets (Fig. 3).

The NP and switching processes have a similar architecture, consisting of a shell and a core (Fig. 4). The shell primarily implements data forwarding, while the core can execute packet processing modules for packet manipulation. In the

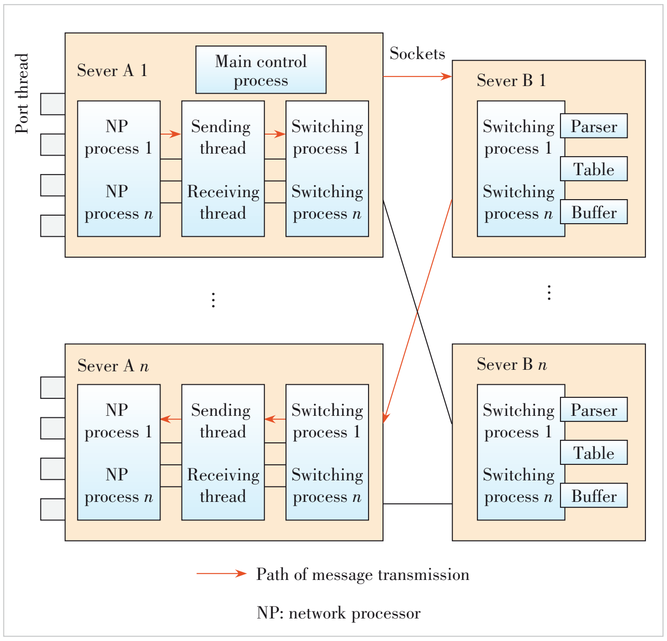


Figure 3. Framework of the digital twin model

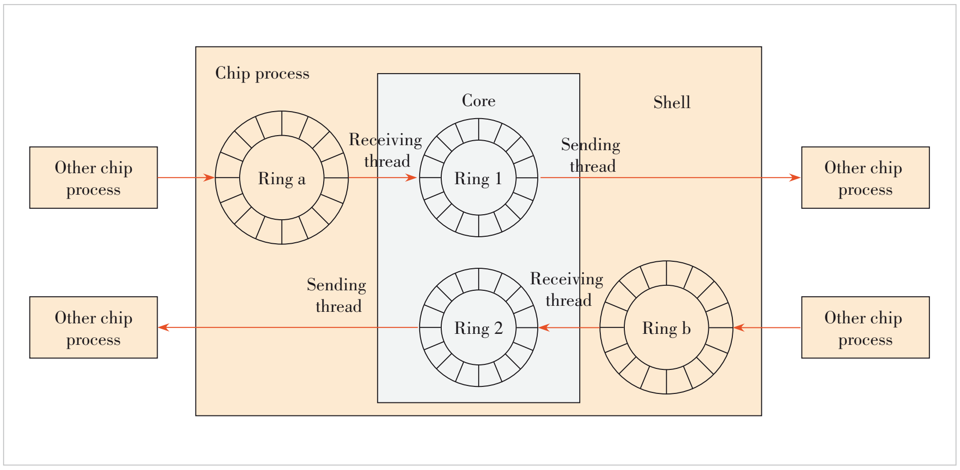


Figure 4. Framework of the process chip

shell, there are two queues (Ring a and Ring b) and two threads (receiving and sending threads). The core also contains two queues (Ring 1 and Ring 2). The receiving thread transfers packets between the shell and core through different queues, while the sending thread transfers packets to other chip processes.

4.2 Inter-Process Communication

The simulation process in vFabric involves a continuous flow of packet transmission and reception, which may impede the simulation speed and rapidly consume computation resources. To address this issue, we pre-configure memory based on simulation patterns and adopt a high-speed inter-process communication solution based on a multi-threading mechanism. This solution is designed to simulate the communication process between chips.

For small-scale deployments, vFabric can operate on a single server. We utilize the ring queue based on shared memory to facilitate communication between chip processes. The ring queue is a data structure that connects the front and rear ends circularly, following the first-in-first-out (FIFO) principle. It utilizes a linear array for storing data and offers simple data organization and efficient management.

In this paper, we present the implementation of high-speed data exchange between chip processes utilizing a producer-consumer model, as illustrated in Fig. 5. The model involves two threads: the producer and the consumer, which interact by reading from and writing to shared memory. The producer and consumer need to perform mutual exclusion operations to ensure data accuracy and safety.

For the ring queue, the sending thread acts as the producer, while the receiving thread serves as the consumer. Ensuring sole control over the circular queue is of utmost importance, permitting manipulation by a solitary thread exclusively at any given time. Specifically, when multiple producers write to the ring queue at the same time, only one thread is allowed to write to the queue. Consumers should also adhere to this principle in order to maintain data integrity and prevent conflicts.

On a large scale, vFabric is deployed across multiple servers, utilizing sockets for communication between multiple servers. The socket is a widely used network communication technology based on the Transmission Control Protocol/Internet Protocol (TCP/IP) protocol. It provides two endpoints for bidirectional host-to-host interaction. In modern networks, sockets have various applications and can facilitate data transmission, control, and management through various protocols, meeting the requirements of different scenarios.

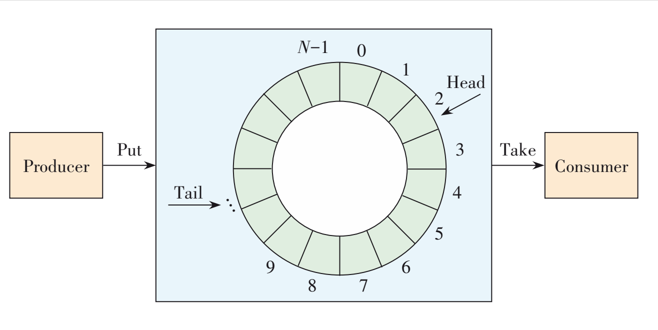


Figure 5. Framework of the producer-consumer model

4.3 Synchronization Mechanism

The vFabric platform utilizes a distributed simulation system across multiple servers. In a distributed system, time is a pivotal concept. Simultaneously, the chip processes also rely on time for synchronized interaction.

Currently, commonly used time synchronization methods in distributed simulation systems include the Network Time Protocol (NTP) and the Berkeley algorithm[38]. NTP is used to synchronize the time of various nodes in a computer network to ensure time consistency and accuracy. In contrast, the Berkeley algorithm achieves time synchronization in distributed systems by selecting a reference node to provide accurate time information and synchronizing the clocks of other nodes through communication with the reference node.

However, these techniques have limitations. NTP is not suitable for all types of distributed systems, and the Berkeley algorithm requires a central server, which may introduce a single point of failure. Therefore, we propose a time synchronization method for large-scale distributed simulation systems based on multi-level management (Fig. 6).

The time synchronization framework consists of three roles:

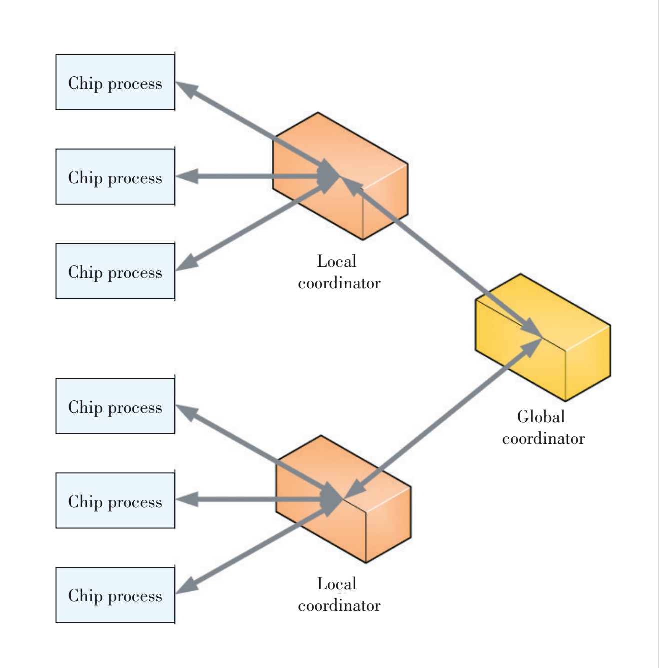


Figure 6. Framework of the synchronization module

the global coordinator, local coordinator, and member node. First, all member nodes send timestamps to the local coordinator node after each time slice and then enter a blocked state. The local coordinator node receives the timestamps from all the member nodes within its management range, sends the timestamps to the global coordinator node, and enters a blocked state. Upon receiving the timestamps from all local coordinator nodes, the global coordinator node advances the global clock and notifies all local coordinator nodes to continue execution. After receiving the response, the member nodes continue their computational tasks, completing time synchronization. This process of time synchronization among the three types of nodes continues iteratively until the simulation concludes.

4.4 Exception Handling

To emulate real-world scenarios involving the process of a chip being uploaded and offloaded due to various factors such as equipment failure, maintenance, and updates, vFabric incorporates periodic online and offline operations on the chip processes.

These operations often lead to fluctuations in system load, such as some nodes becoming overloaded while others remaining underutilized, which can impact the performance of the simulation system. To address such issues and ensure stability and efficiency in the cluster simulation system, effective handling of device online and offline processes is required to achieve traffic load balancing. Load balancing involves distributing the workload evenly across the nodes in the simulation cluster, thereby optimizing resource utilization and preventing any individual node from becoming overloaded.

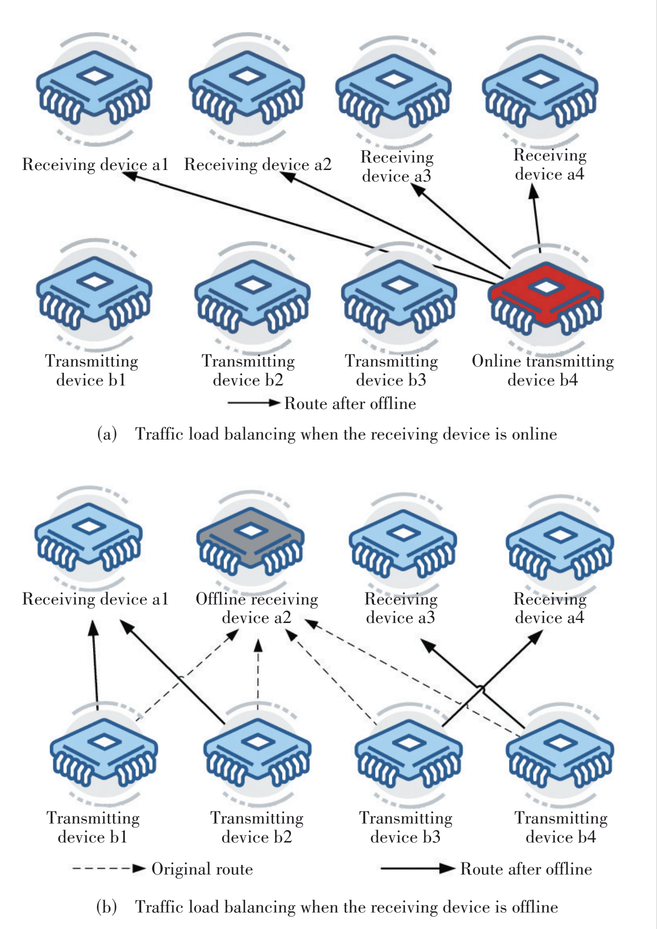
To address the challenges, this paper proposes a dynamic traffic load-balancing algorithm based on real-time device status information (e.g., cache utilization and CPU occupancy). By leveraging the real-time status information, the algorithm intelligently selects appropriate forwarding paths and dynamically reschedules data packets based on the actual device conditions. When a chip comes online, the packets are rerouted among all the chips in the system (Fig. 7a). This ensures that the new chip can participate in the packet forwarding process. On the other hand, when a chip goes offline, the packets originally residing in that chip are evenly distributed to other available chips (Fig. 7b). This ensures maximum transmission throughput to avoid packet loss while maximizing system performance.

5 Implementation of VFabric

In this section, we present the implementation of vFabric in detail. We develop vFabric on the Linux platform through C++.

5.1 Implementation of Synchronization

In this paper, we establish a synchronization mechanism



(a) Traffic load balancing when the receiving device is online

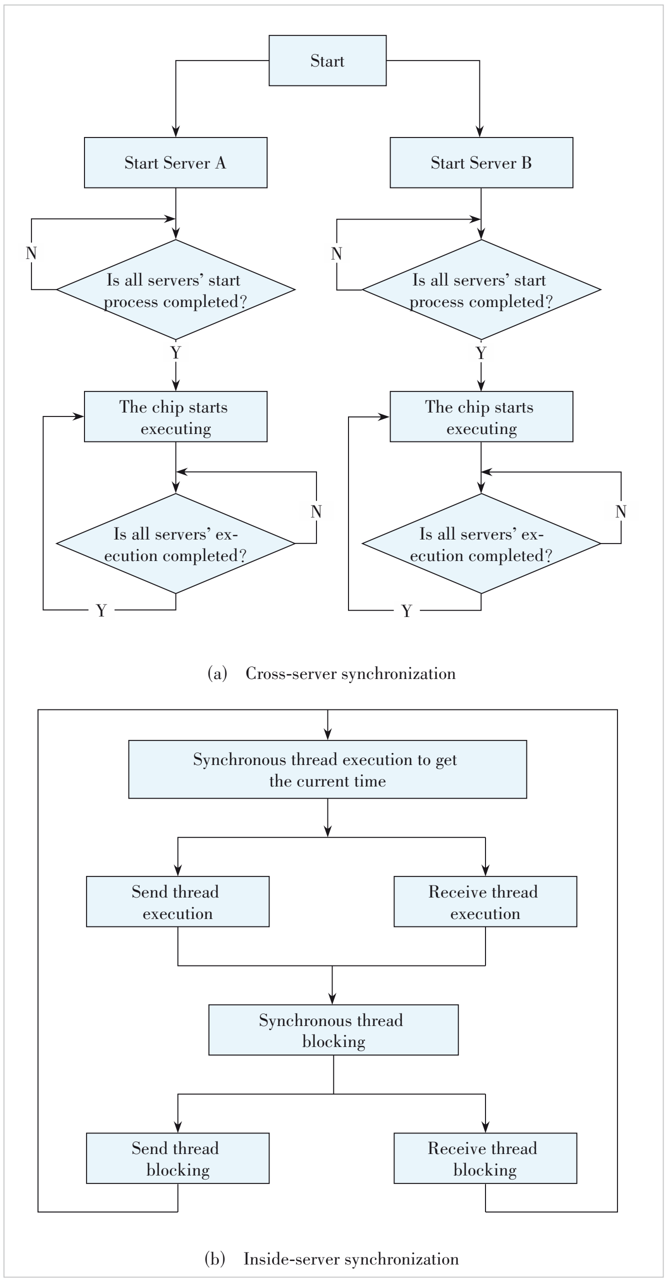
(b) Traffic load balancing when the receiving device is offline

Figure 7. Schematic diagrams of online and offline processes

based on Redis and semaphores. Redis[39] is a mainstream non-relational database that is distributed and scalable, with high-performance. The semaphore, a mechanism for synchronous control among multiple threads or processes, coordinates the access sequence among different threads and processes to avoid data inconsistency. We deploy a Redis database on each server and connect them to form a distributed cluster. Each server can obtain the current time through the Redis database and perform corresponding time synchronization operations.

In the synchronization process between servers, each server contains a global synchronization thread. Each server is executed alternately with a global synchronization thread until the simulation concludes (Fig. 8a).

In the synchronization process within the server, the threads inside the chip (such as receiving and sending threads) and the synchronization thread alternately execute the P(wait) operation and V(signal) operation on the synchronization semaphore (Fig. 8b). P(wait) operation can block process execution, and V(signal) operation can resume process execution. With each alternation, the system time increases by one time slice until the simulation concludes.


Start
Start Server A
Start Server B
N
Is all servers' start process completed?
Y
The chip starts executing
Is all servers' ex-ecution completed?
N
Y
Synchronous thread execution to get the current time
Send thread execution
Receive thread execution
Synchronous thread blocking
Send thread blocking
Receive thread blocking

(a) Cross-server synchronization

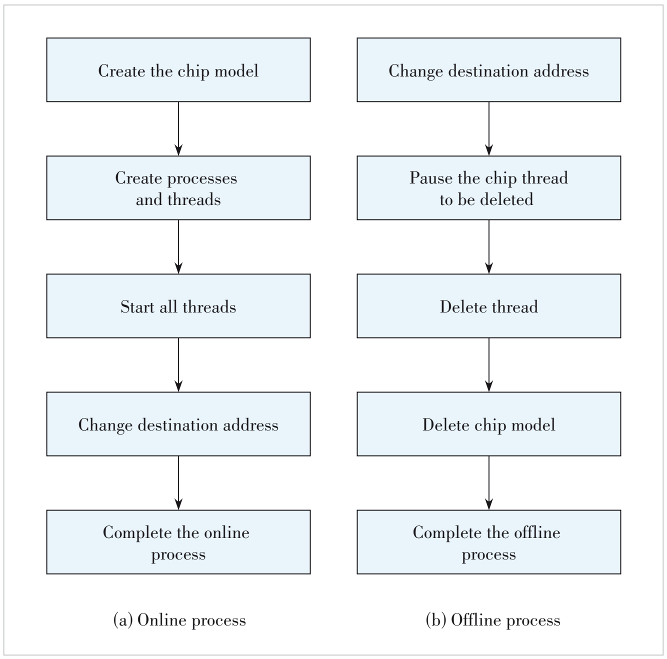
(b) Inside-server synchronization

Figure 8. Flow charts of synchronization

5.2 Implementation of Exception Handling

During both online and offline processes of the chips, it is crucial to effectively handle related events such as routing planning, process management, process creation, and termination.

During the online process (Fig. 9a), the following steps are executed: 1) The required models (e.g., variables and pointers) are created; 2) once the models are created, the related processes and threads are initialized and blocked; 3) the synchronization thread starts alongside other chip threads; 4) the destination addresses of the packets are modified to ensure load balancing.


Create the chip model
Create processes and threads
Start all threads
Change destination address
Complete the online process
Change destination address
Pause the chip thread to be deleted
Delete thread
Delete chip model
Complete the offline process

(a) Online process

(b) Offline process

Figure 9. Flow charts of exception handling

During the offline process (Fig. 9b), the following steps need to be taken: 1) The destination addresses of the packets are modified to prevent data loss; 2) while waiting for the synchronization thread to execute, the required chip threads are paused and removed; 3) the chip model is deleted to complete the device offline process.

The following is a typical process that a system performs over time. At the beginning of the offline process, the system time is recorded as t_1, and the destination addresses of the packets are immediately updated. If the destination chip of a packet has already gone offline, the destination address is changed to another available chip. After waiting until the system time reaches t_{1+nT} , the detection of destination addresses of internal chip packets is stopped. At this point, there are no more packets in the system with a destination address belonging to a chip that has already gone offline.

Here, t_1 is any time and T in t_{1+nT} represents the size of the time slice, and n is a variable that can be modified based on the server's performance. This variable can be determined according to specific circumstances to ensure that the detection and forwarding of internal chip packets are completed within an appropriate time frame, effectively preventing the processing of packets destined for offline chips.

6 Performance Evaluation

In this section, we conduct extensive experiments to evaluate the performance of vFabric and collect statistics as the scale gradually increases.

6.1 Experiment Settings

1) Platform: The evaluation platform is a workstation carrying an Intel(R) Xeon(R) Silver 4210R CPU (each has 80

cores). The RAM of the workstation is 260 GB and the operating system is Linux Ubuntu 20.10. The version of Redis is 6.0.8.

2) Data scale: As mentioned earlier, the servers are divided into two types (A and B). Server A can support a maximum of 36 NP processes and 32 switching processes, while Server B has a maximum of 32 switching processes. Each NP is connected to 20 100 Gbit/s port threads. We simulate the state of a real device forwarding 10 ms of traffic, while the total operation takes slightly longer than 10 ms. The additional time is allocated after 10 ms to ensure that all packets are fully transmitted. The delay depends on system characteristics and simulation requirements. Each port sends a packet of varying length every 25 ns. Table 1 summarizes the proportion of packets of different lengths. For example, the proportion of 64-byte messages is 449/1 000.

6.2 Results and Analysis

In the vFabric, each port generates 400 000 packets within a time interval of 10 ms. Table 2 summarizes the number of packets forwarded by different chips in a large-scale scenario.

The key indicators of the vFabric are the simulation time and packet loss rate. The simulation time of the system increases gradually with scale. For small-scale scenarios (4 NP processes and 32 switching processes), the total simulation time is approximately 100 s. For medium-scale scenarios (36 NP processes and 32 switching processes), the total simulation time is approximately 1 000 s. For large-scale scenarios (one Server A and one Server B), the total simulation time is approximately 4 800 s. These are generally in line with the expected requirements.

Different synchronization algorithms have been introduced above. We compare the time taken by two synchronization methods, the Berkeley algorithm and the multi-level management, as shown in Fig. 10. The multi-level management approach we adopted has a significant optimization effect. Specifically, the designed total simulation time is designed to be 10 ms, but in practice, it slightly exceeds this duration. Additional time is allocated after 10 ms to ensure all packets are fully transmitted. The length of this delay depends on the system characteristics and simulation requirements. Experimental results (Fig. 11) show that packet loss occurs when the simulation time is insufficient, and the packet loss rate gradually decreases as the simulation time increases. However, an excessively long simulation time leads to prolonged simulation duration. In the vFabric, the total simulation time is set to 10.1 ms.

Table 1. Proportion of packets of different lengths

Packet Length/B	Weight
64	449
130	160
260	200
577	80
1 518	110
9 000	1

Table 2. Number of packets forwarded by different chips

Type of Device	Number of Packets
Port	400 000
NP	8 000 000
Switch	9 000 000
Total server	28 800 000

NP: network processor

7 Conclusions

In this paper, we present a large-capacity core switching equipment digital twin platform. The simulation platform primarily consists of NP chips and switching chips, with a simu-

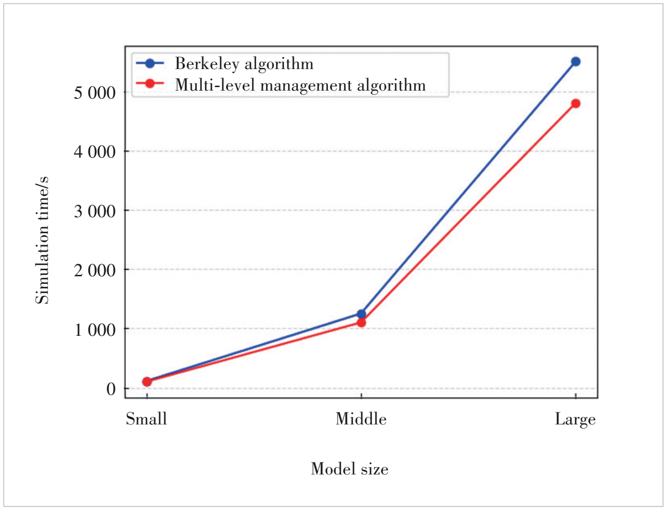


Figure 10. Simulation time of different synchronization algorithms

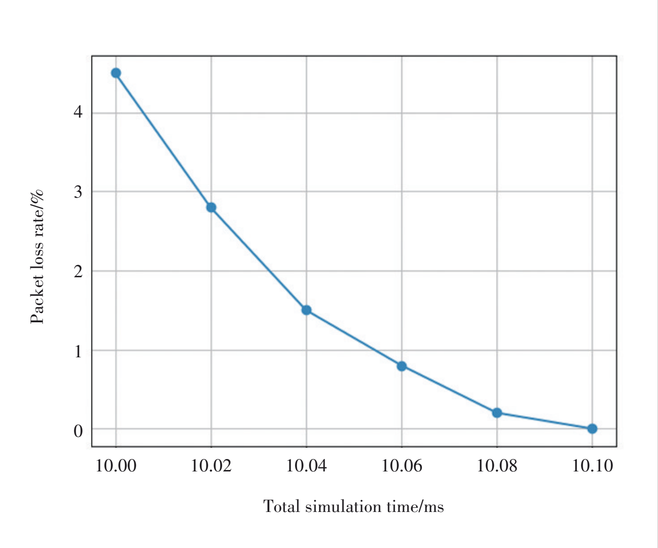


Figure 11. Change of the packet loss rate with time

lation time of 10 ms. The simulation results demonstrate the platform’s efficient and reliable simulation capabilities. It accurately replicates the operational state of large-capacity core switching equipment. Moreover, we have successfully implemented time synchronization and the ability to dynamically bring chip processes online and offline, further enhancing the platform’s functionality. The digital twin platform offers valuable applications in diagnosing and troubleshooting network failures. It assists engineers in promptly identifying and resolving issues, thereby enhancing the maintainability and manageability of the network. In the future, we will further enhance the platform to meet more complex and diverse simulation requirements.

References

[1] GÜNDOĞAN C, AMSÜSS C, SCHMIDT T C, et al. Content object security in the Internet of Things: challenges, prospects, and emerging solutions [J]. IEEE transactions on network and service management, 2022, 19(1): 538 – 553. DOI: 10.1109/TNSM.2021.3099902

[2] SHANG B D, LIU L J, MA J C, et al. Unmanned aerial vehicle meets vehicle-to-everything in secure communications [J]. IEEE communications magazine, 2019, 57(10): 98 – 103. DOI: 10.1109/MCOM.001.1900170

[3] YAN S C, ZHAO Z Y, GAO D Q, et al. Research on fuzzy location method of power communication network fault using digital twin model based on weight coefficient [C]//Proc. IEEE 6th Conference on Energy Internet and Energy System Integration. IEEE, 2022: 1779 – 1783. DOI: 10.1109/EI256261.2022.10116230

[4] CHEN M L, SHAO J, GUO S X, et al. Convoy_DTN: a security interaction engine design for digital twin network [C]//Proc. IEEE Globecom Workshops (GC Wkshps). IEEE, 2021. DOI: 10.1109/gcwkshps52748.2021.9682031

[5] WU Y W, ZHANG K, ZHANG Y. Digital twin networks: a survey [J]. IEEE Internet of Things journal, 2021, 8(18): 13789 – 13804. DOI: 10.1109/JIOT.2021.3079510

[6] RAZA M, KUMAR P M, HUNG D V, et al. A digital twin framework for industry 4.0 enabling next-gen manufacturing [C]//Proc. 9th International Conference on Industrial Technology and Management (ICITM). IEEE, 2020: 73 – 77. DOI: 10.1109/ICITM48982.2020.9080395

[7] LIU J, YANG H, NIU H C, et al. Digital twin civil aviation research airport for aircraft security and environment protection [C]//Proc. IEEE 4th International Conference on Civil Aviation Safety and Information Technology (ICCASIT). IEEE, 2022: 408 – 412. DOI: 10.1109/ICCASIT55263.2022.9986522

[8] SHRIVASTAVA M, CHUGH R, GOCHHAIT S, et al. A review on digital twin technology in healthcare [C]//Proc. International Conference on Innovative Data Communication Technologies and Application (ICIDCA). IEEE, 2023: 741 – 745. DOI: 10.1109/ICIDCA56705.2023.10099646

[9] CHEN J H, DENG R Q, GUO Y Y, et al. Research on network management technology of power line carrier communication in low-voltage distribution network based on digital twin [C]//Proc. 7th International Conference on Computer and Communications (ICCC). IEEE, 2021: 2112 – 2116. DOI: 10.1109/ICCC54389.2021.9674420

[10] RASHEED A, SAN O, KVAMSDAL T. Digital twin: values, challenges and enablers from a modeling perspective [J]. IEEE access, 2020, 8: 21980 – 22012. DOI: 10.1109/ACCESS.2020.2970143

[11] ZHAO Z Y, YAN S C, GAO D Q, et al. Research on digital twin network architecture for power grid telecommunication system [C]//Proc. IEEE 6th Conference on Energy Internet and Energy System Integration. IEEE, 2022: 1868 – 1874. DOI: 10.1109/EI256261.2022.10116995

[12] EL MARAI O, TALEB T, SONG J. Roads infrastructure digital twin: a step toward smarter cities realization [J]. IEEE network, 2021, 35(2): 136 – 143. DOI: 10.1109/MNET.011.2000398

[13] ZHU Y H, CHEN D Y, ZHOU C, et al. A knowledge graph based construction method for Digital Twin Network [C]//Proc. IEEE 1st International Conference on Digital Twins and Parallel Intelligence (DTPI). IEEE, 2021. DOI: 10.1109/dtpi52967.2021.9540177

[14] AUTIOSALO J, SIEGEL J, TAMMI K. Twinbase: open-source server software for the digital twin web [J]. IEEE access, 2021, 9: 140779 – 140798. DOI: 10.1109/ACCESS.2021.3119487

[15] KURUVATTI N P, HABIBI M A, PARTANI S, et al. Empowering 6G communication systems with digital twin technology: a comprehensive survey [J]. IEEE access, 2022, 10: 112158 – 112186. DOI: 10.1109/ACCESS.2022.3215493

[16] FU X Y, YUAN Q, LIU S F, et al. Communication-efficient decision-making of digital twin assisted Internet of vehicles: a hierarchical multi-agent reinforcement learning approach [J]. China communications, 2023, 20(3): 55 – 68. DOI: 10.23919/JCC.2023.03.005

[17] QIN W B, ZHANG C, YAO H P, et al. Stackelberg game-based offloading strategy for digital twin in Internet of vehicles [C]//Proc. International Wireless Communications and Mobile Computing (IWCMC). IEEE, 2023: 1365 – 1370. DOI: 10.1109/IWCMC58020.2023.10182450

[18] PILLAI R, BABBAR H. Digital twin for edge computing in smart vehicular systems [C]//Proc. International Conference on Advancement in Computation & Computer Technologies (InCACCT). IEEE, 2023: 1 – 5. DOI: 10.1109/InCACCT57535.2023.10141784

[19] DAI Y Y, ZHANG Y. Adaptive digital twin for vehicular edge computing and networks [J]. Journal of communications and information networks, 2022, 7(1): 48 – 59. DOI: 10.23919/JCIN.2022.9745481

[20] DAI C G, YANG K, DENG C J. A service placement algorithm based on merkle tree in MEC systems assisted by digital twin networks [C]//Proc. IEEE 21st International Conference on Ubiquitous Computing and Communications (IUCC/CIT/DSCI/SmartCNS). IEEE, 2022: 37 – 43. DOI: 10.1109/IUCC-CIT-DSCI-SmartCNS57392.2022.00020

[21] GUO Y, ZHUANG Y, LI X, et al. Time-frequency correlated network sensing edge deployment for digital twin [C]//Proc. 4th International Academic Exchange Conference on Science and Technology Innovation (IAECST). IEEE, 2022: 1183 – 1187. DOI: 10.1109/IAECST57965.2022.10062032

[22] NJOKU J N, NKORO E C, MEDINA R M, et al. Leveraging digital twin technology for battery management: a case study review [J]. IEEE access, 2022, 13: 21382 – 21412, DOI: 10.1109/ACCESS.2025.3531833

[23] TAO Y H, WU J, LIN X, et al. DRL-driven digital twin function virtualization for adaptive service response in 6G networks [J]. IEEE networking letters, 2023, 5(2): 125 – 129. DOI: 10.1109/LNET.2023.3269766

[24] LU Y L, MAHARJAN S, ZHANG Y. Adaptive edge association for wireless digital twin networks in 6G [J]. IEEE Internet of Things journal, 2021, 8(22): 16219 – 16230. DOI: 10.1109/JIOT.2021.3098508

[25] WEI Z Y, WANG S T, LI D, et al. Data-driven routing: a typical application of digital twin network [C]//Proc. IEEE 1st International Conference on Digital Twins and Parallel Intelligence (DTPI). IEEE, 2021. DOI: 10.1109/dtpi52967.2021.9540073

[26] RAJ D R R, SHAIK T A, HIRWE A, et al. Building a digital twin network of SDN using knowledge graphs [J]. IEEE access, 2023, 11: 63092 – 63106. DOI: 10.1109/ACCESS.2023.3288813

[27] ONO S, YAMAZAKI T, MIYOSHI T, et al. AMoND: area-controlled mobile ad-hoc networking with digital twin [J]. IEEE access, 2023, 11: 85224 – 85236. DOI: 10.1109/ACCESS.2023.3304374

[28] Huawei. NetEngine 8000 series router [EB/OL]. [2023-06-15]. https://e.huawei.com/cn/products/routers/ne8000

[29] Cisco. NAT order of operation [EB/OL]. [2023-06-15]. https://www.cisco.com/c/zhcn/products/routers/8000-series-routers/index.html

[30] ZTE. ZXR10 6800 multi-service router [EB/OL]. [2023-06-15]. https://www.zte.com.cn/china/products/bearer/router/ZXR10-6800.html

[31] ZTE. ZXR10 8900E series core switch [EB/OL]. [2023-06-15]. https://www.zte.com.cn/china/products/bearer/Ethernet-Switch/8900E-CH.html

[32] JIN S, ZHANG Z B, CHAKRABARTY K, et al. Hierarchical symbol-based health-status analysis using time-series data in a core router system [J]. IEEE transactions on computer-aided design of integrated circuits and systems, 2020, 39(3): 700 – 713. DOI: 10.1109/TCAD.2018.2890681

[33] JIN S, ZHANG Z B, CHAKRABARTY K, et al. Self-learning health-status analysis for a core router system [C]//Proc. IEEE International Test Conference (ITC). IEEE, 2018: 1 – 10. DOI: 10.1109/TEST.2018.8624712

[34] WU C C, QIAO L F, CHEN Q H. Design of a 640-gbps two-stage switch fabric for satellite on-board switches [J]. IEEE access, 2020, 8: 68725 – 68735. DOI: 10.1109/ACCESS.2020.2986300

[35] GLABOWSKI M, LEITGEB E, SOBIERAJ M, et al. Analytical modeling of switching fabrics of elastic optical networks [J]. IEEE access, 2020, 8: 193462 – 193477. DOI: 10.1109/ACCESS.2020.3033186

[36] MADUREIRA A L R, ARAÚJO F R C, ARAÚJO G B, et al. NDN fabric: where the software-defined networking meets the content-centric model [J]. IEEE transactions on network and service management, 2021, 18(1): 374 – 387. DOI: 10.1109/TNSM.2020.3044038

[37] ALMASAN P, FERRIOL-GALMES M, PAILLISSE J, et al. Network digital twin: context, enabling technologies, and opportunities [J]. IEEE communications magazine, 2022, 60(11): 22 – 27. DOI: 10.1109/mcom.001.2200012

[38] FANG Z H, GAO Y. Delay compensated one-way time synchronization in distributed wireless sensor networks [J]. IEEE wireless communications letters, 2022, 11(10): 2021 – 2025. DOI: 10.1109/LWC.2022.3189744

[39] BEN SEGHIER N, KAZAR O. Performance benchmarking and comparison of NoSQL databases: Redis vs MongoDB vs Cassandra using YCSB tool [C]//Proc. International Conference on Recent Advances in Mathematics and Informatics (ICRAMI). IEEE, 2021. DOI: 10.1109/icrami52622.2021.9585956

Biographies

WANG Qianglin received his MS degree in electronic information from University of Electronic Science and Technology of China in 2024. He joined Alibaba Corporation in 2024 and is now a network engineer there.

ZHANG Xiaoning (xnzhang@uestc.edu.cn) received his MS and PhD degrees in electronic information from the School of Communication and Information Engineering, University of Electronic Science and Technology of China (UESTC) in 2005 and 2007, respectively. Since 2007, he has been a professor with UESTC. His current research interests include software-defined networking and virtualization of network functions.

YANG Yi received his MS degree in electronic information from University of Electronic Science and Technology of China in 2024. He joined Tianyi Cloud Corporation in 2024.

FAN Chenyu received her MS degree in electronic information from University of Electronic Science and Technology of China in 2024. She joined Ericsson-Cloud Corporation in 2024.

YUE Yangyang is a senior engineer at ZTE Corporation. She is engaged in application software development for wired communications.

WU Wei is an engineer at ZTE Corporation. He is engaged in application software development for wired communications. His primary research focuses on large-scale router penetration testing, performance simulation, and functional development.

DUAN Wei is a senior R&D chief engineer at ZTE Corporation. He is engaged in the key technology research of IP networks and intelligent computing center networks. He has applied for more than 30 patents.