Design and Implementation of a Programmable High-Capacity Multi-user Network Testbed for SOI Asia Research and Education Partners

Galih Nugraha Nurkahfi [1,2], Achmad Husni Thamrin [3], Nasrullah Armi [2], Eueung Mulyana [1,*], and Nana Rachmana Syambas [1]

[1] School of Electrical Engineering and Informatics (STEI), Bandung Institute of Technology (ITB), Bandung, Indonesia
[2] Research Center for Telecommunications, Electronics and Informatics Research Organization, National Agency for Research and Innovation, Bandung, Indonesia
[3] School on Internet (SOI) Asia, Japan
Email: 33221305@std.stei.itb.ac.id (G.N.N.); galih.nugraha.nurkahfi@brin.go.id (G.N.N.); husni@sfc.wide.ad.jp (A.H.T.); nasrullah.armi@brin.go.id (N.A.); eueung@stei.itb.ac.id (E.M.); nana@stei.itb.ac.id (N.R.S.)
*Corresponding author

***Abstract*—Existing network testbeds face a critical gap; federated platforms such as GENI and Fed4FIRE provide multi-user access but lack high-capacity, programmable data planes. In contrast, specialized P4 testbeds offer programmability but limited concurrent user support. This paper addresses this gap by presenting a 100 Gbps programmable multi-user testbed for Research and Education Networks in Southeast Asia. The main contribution is integrating Programmable P4 switches, SDN control, and automated orchestration into a shared platform accessible via existing REN infrastructure. This enables multiple universities to conduct concurrent experiments without deploying local high-capacity hardware. Validation demonstrated near line-rate aggregate throughput, sub-millisecond latency, below 3% fairness variance across concurrent users, and minimal programmability overhead. Automation reduced deployment time by over 85%. The testbed provides operational access to SOI Asia partners, combining performance comparable to FABRIC with accessibility tailored to resource-constrained regional institution*s*.**

***Keywords*—programmable networks, P4, Software-Defined Networking (SDN), multi-user testbed, high-capacity networking, network slicing, Research and Education Networks (RENs), SOI Asia**

I. INTRODUCTION

Research and Education Networks (RENs) have an important role in facilitating collaboration among universities and research institutions. REN provides the network infrastructure that supports research and experiments, large-scale data sharing, and cross-border academic collaboration [1]. To stay up to date with emerging technologies such as 5G/6G, the Internet of Things (IoT), and vehicular networks, REN communities are increasingly requiring testbeds that can support real-world experiments more than simulations. Testbeds that are programmable, high-capacity, and shareable across multiple users are particularly important because they enable innovation under realistic network conditions [2].

Within Asia, one of the most active academic communities is the School on Internet Asia (SOI Asia) initiative. Established in the late 1990s with support from the WIDE Project in Japan, SOI Asia has grown into a collaborative network of universities across the region, with Keio University serving as its hub. It builds on the Asian Internet Interconnection Initiatives (AI3) and has played a key role in strengthening ICT capacity and inter-university collaboration in SOI Asia [3]. SOI Asia's activities include distance learning, IPv6 deployment training, and the development of collaborative academic services. Additionally, the SOI Asia AI3 project is defined as a testbed network that provides an environment for researchers and academicians to implement, test, and evaluate technologies, including IPv6, network monitoring, and security [3]. Keio University supports the initiative and collaborates with regional research and educational institutions, with additional contributions from organizations such as the APNIC Foundation [4]. A common theme across these activities is the sharing of educational resources, expertise, and joint capacity building among partner universities.

To extend this vision to network research infrastructure, SOI Asia members would benefit from a shared, programmable testbed accessible via existing REN Network in the Asia-Pacific) has been launched to provide a high-speed backbone connecting universities and RENs across the region [5], including recent deployments connecting to IDREN in Indonesia [6]. While most academic network backbones, such as Japan's SINET5 [7], have already been upgraded to 100 Gbps and beyond, covering all prefectures and supporting international 100 Gbps links, demonstrating the feasibility of high-

Manuscript received October 12, 2025; revised November 27, 2025; accepted December 26, 2025; published April 14, 2026.

bandwidth experiments, comparable capabilities are not yet uniformly available across all Asia-Pacific research networks [8]. At the same time, large-scale virtual and edge testbeds have been shown to suffer from scalability bottlenecks, reproducibility issues, and automation overheads. Dependency on NetEm for network emulation results in configuration delays, increased latency, and decreased throughput when scaling to thousands of emulated links, which makes it challenging to support concurrent multi-user experiments or fully automated remote deployments [9]. These limitations pose challenges for researchers in SOI Asia universities and their REN partners who require dependable, flexible, and scalable platforms for experimentation. Infrastructures as we can see in Fig. 1. Testbed architecture and connectivity. network topology showing two Netberg Aurora 710 switches (3.2 Tbps each) connected to eight servers. Data paths: 25 Gbps per server, 100 Gbps inter-switch link, 200 Gbps aggregate I/O, 10 Gbps management network.. Within the Asia-Pacific, ARENA-PAC (Arterial Research and Educational

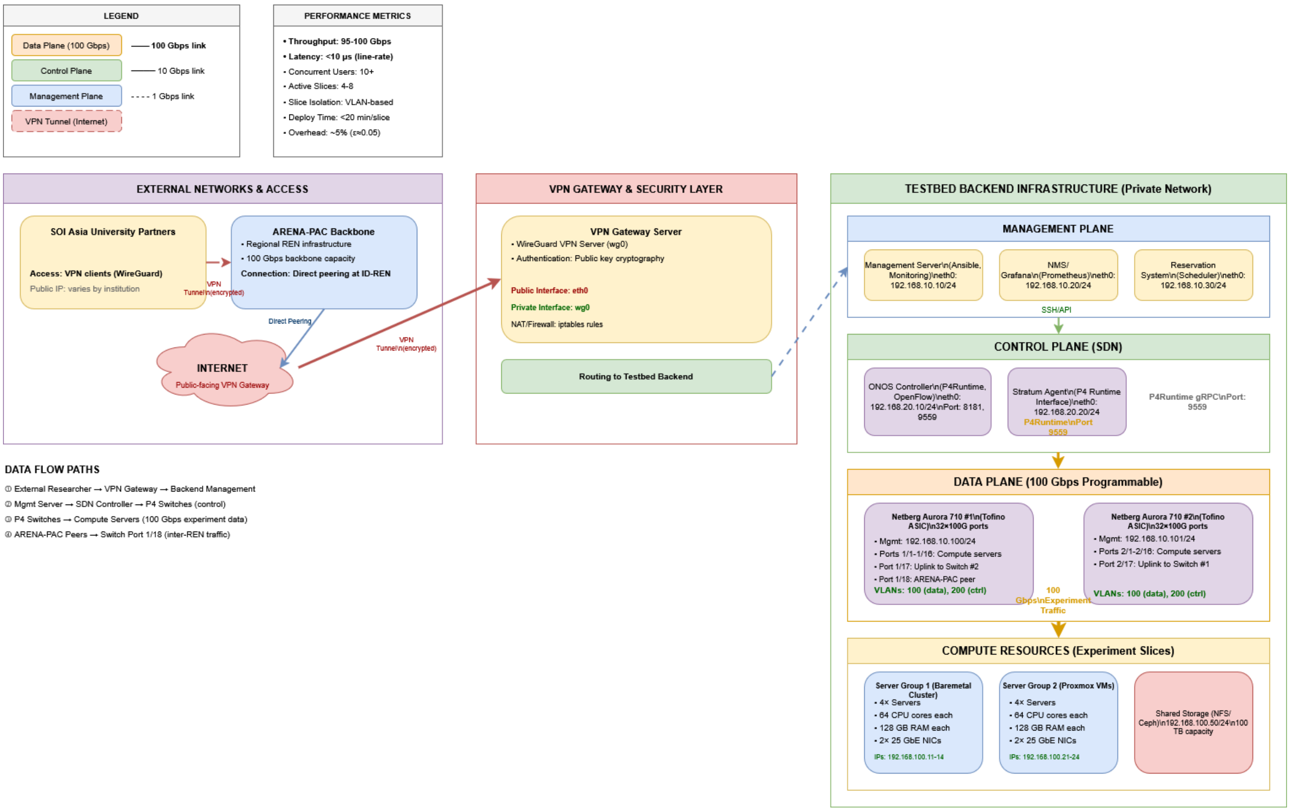

Fig. 1. Testbed architecture and connectivity. network topology showing two Netberg Aurora 710 switches (3.2 Tbps each) connected to eight servers. Data paths: 25 Gbps per server, 100 Gbps inter-switch link, 200 Gbps aggregate I/O, 10 Gbps management network.

The core question driving this work is whether a shared 100 Gbps testbed can perform at near line-rate (⩾95% throughput, <1ms latency) when several users run experiments at the same time, while still keeping resource allocation fair across all of them (⩽5% variance). Combining P4 switches with automated deployment and VPN access at a central location should let multiple universities use the system without slowing each other down. If this works, individual campuses would not need to purchase their own expensive hardware.

What makes this work different from previous efforts is practical accessibility. Platforms like GENI cost $50K-$200K per site to deploy. Others like FABRIC and GP4L demand specialized P4 and SDN expertise that most regional universities lack. In contrast, the testbed we built is the first 100 Gbps programmable platform made specifically for institutions with limited budgets and staff, and we tested it under actual multi-user conditions to prove it works.

Existing testbed platforms address some but not all these limitations. Global testbed platforms, such as FABRIC [10] and the GEANT P4 Lab (GP4L) [11], offer high-capacity programmable network infrastructure; however, they are geographically located outside the Asia-Pacific region and lack specific support for regional REN accessibility and integration. The OF@TEIN project [12] pioneered SDN experimentation across Southeast Asia, but it operates primarily at 10 Gbps link capacities, which are insufficient for emerging workloads. While these platforms validate the technical feasibility of programmable data planes and federated experimentation, no testbeds specifically address the combination of: (a) 100 Gbps programmable capacity accessible via existing Asia-Pacific REN infrastructure, (b) secure remote access eliminating local hardware requirements, (c) automated multi-user orchestration with resource isolation, and (d) validated performance under concurrent workloads tailored to regional research needs. Our testbed uniquely addresses this gap by providing the first high-capacity (100 Gbps) P4-programmable testbed specifically designed for and accessible to SOI Asia universities, featuring comprehensive automated orchestration that enables multi-user concurrent access without manual intervention, and delivering validated carrier-grade performance (95-100 Gbps throughput, sub-millisecond latency) through secure VPN-based remote access.

This paper addresses these gaps by proposing the design, implementation, and validation of a programmable, high-capacity, multi-user testbed deployed at the Bandung Institute of Technology (ITB) as depicted in Fig. 1. Testbed architecture and connectivity. network topology showing two Netberg Aurora 710 switches (3.2 Tbps each) connected to eight servers. Data paths: 25 Gbps per server, 100 Gbps inter-switch link, 200 Gbps aggregate I/O, 10 Gbps management network. Figs. 1−3. Hardware and software component Stack. Layered architecture with quantifiable specifications. Hardware: 32×100 Gbps ports, 3.2 Tbps switching, 196 CPU cores, 512 GB RAM, 200 Gbps aggregate NIC I/O. Software: P4Studio, SONiC/Stratum, ONOS, Kubernetes, Prometheus/Grafana.3. The key contributions of this research are:

1) System Integration and Architecture: In this paper, we present a testbed design that integrates P4-programmable 100 Gbps switches with SDN controllers, automated orchestration, and multi-user management, delivering a production-ready platform that balances carrier-grade performance with research flexibility.
2) Regional Infrastructure Enablement: We provide the first high-capacity programmable testbed specifically accessible to SOI Asia partners through existing REN infrastructure, eliminating the need for individual universities to invest in expensive 100 Gbps hardware while ensuring equitable research access across the Asia-Pacific region.
3) Automated Multi-User Orchestration: In this research, we develop and implement a reservation, scheduling, and deployment framework that ensures resource isolation, reproducibility, and secure remote access, enabling concurrent experiments without manual intervention or performance degradation.
4) Validated Performance at Scale: We demonstrate through extensive experimentation that the platform achieves near line-rate throughput (95−100 Gbps), microsecond-scale latency, and stable operation under concurrent workloads, with quantitative comparisons against existing global and regional testbeds.
5) Blueprint for Replication: We provide detailed architectural documentation, implementation insights, and lessons learned that serve as a practical blueprint for other REN communities seeking to deploy similar high-capacity programmable infrastructure.

II. Literature Review

Federated Testbed Architectures: Fed4FIRE integrates 23 facilities across Europe with Virtual Wall, achieving 99.1 Gbps local throughput [13]. Centralized architecture creates single points of failure, so that if the AM directory fails, federation-wide resource discovery stops. OF@TEIN spans six Asia-Pacific countries, but testing across 21 networks shows 83% access success, with 17% failures due to firewall policies and protocol blocking [14]. Architecture lacks Single Sign-On (SSO) authentication and multi-tenancy authorization, both of which are documented as future work rather than operational features.

Programmable Infrastructure Deployments: GENI spans 50+ U.S. campuses with 1,000 users and a 17-site Tango GENI deployment [2]. However, each campus must deploy GENI racks costing \$50K-\$200K+ per site with ongoing monitoring and incident response obligations. GP4L spans 20+ sites with four core European switches supporting Intel Tofino (3.2−12.8 Tbps) and Nvidia BlueField-2 (200 Gbps) [1, 11]. The platform automates administrative tasks using NetBox, Camunda, and Uptime Kuma. However, documentation explicitly states that the testbed uses case targets the GP4L admin side with user automation as future work. Current capabilities serve operators managing infrastructure, not researchers allocating slices.

Wireless and 5G Testbeds: X5G integrates NVIDIA Aerial SDK with OpenAirInterface, achieving 525 Mbps downlink and 94 Mbps uplink [15]. However, each node costs \$27K−41K (A100 GPU, Mellanox NIC, specialized server, Foxconn RU), totaling \$200K−350K+ for eight nodes. NVIDIA Aerial SDK requires approval from the early adopter program. The testbed shows 40% throughput degradation with 2 RUs despite 22.62 dB of SINR, and deploying 2 RUs required testing 276 location combinations via ray tracing. The E2E 5G+ testbed integrates Open5GS core with UERANSIM emulator [16]. However, UERANSIM does not provide a complete physical layer with a radio interface simulated over UDP. The testbed shows severe scalability: throughput increases only 613x (from 39.76 to 24,381.92 bps) despite 500x more users. Kubernetes deployment creates a situation where IP allocation challenges require manual UPF reconfiguration for each pod.

National-Scale Infrastructure: FABRIC provides a Terabit supercore with 100 Gbps links across around 20 U.S. sites, connecting CloudLab, Chameleon, PAWR, HPC facilities, and public clouds [10, 17, 18]. The testbed is the first to use “PacketGPS”, a nanosecond-precision packet timing. The platform requires expertise in P4, GPU programming, DWDM, and eBPF.

Gap Analysis: These testbeds reveal a common pattern: platforms that offer high capacity or broad federation require either significant cost \$50K−\$200K+ per site or specialized expertise: P4, GPU, DWDM. Meanwhile, accessible platforms operate at lower speeds around 10 Gbps or have reliability issues with 17% access failures). No existing platform combines 100 Gbps capacity, automated user access, and low cost for the Asia-Pacific region, nor provides validated multi-user performance data showing fair resource sharing under concurrent workloads. Table I summarizes these tradeoffs. Fig. 2. Qualitative comparison of testbed platforms across six dimensions (1=poor, 5=excellent). SOI Asia prioritizes deployment cost, access automation, and Asia-Pacific relevance over scale. visualizes this comparison across six dimensions: platforms optimized for scale (GENI) or throughput (FABRIC, GP4L) score poorly on deployment cost and access automation, while SOI Asia accepts reduced scale to maximize accessibility for resource-constrained institutions.

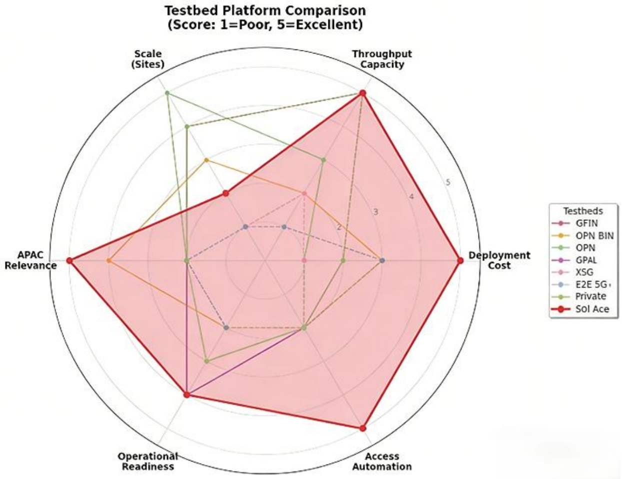

Fig. 2. Qualitative comparison of testbed platforms across six dimensions (1=poor, 5=excellent). SOI Asia prioritizes deployment cost, access automation, and Asia-Pacific relevance over scale.

SOI Asia Design Choices: Based on these gaps, SOI Asia makes five decisions: 1) centralized REN infrastructure eliminates per-campus costs, 2) web-based provisioning removes expertise barriers, 3) persistent 100 Gbps connectivity avoids access failures and manual coordination, 4) homogeneous KVM virtualization ensures reproducible performance, and 5) regional focus on 6 ARENA-PAC sites prioritizes accessibility over scale. These choices involve tradeoffs. SOI Asia sacrifices Fed4FIRE's federation breadth, X5G's 5G capabilities, and FABRIC's nationwide scale to achieve zero per-campus cost, immediate usability, and predictable performance for resource-constrained institutions.

III. Methodology

The contribution of this work lies in addressing concrete gaps observed in prior testbeds. Early international platforms such as the International P4 Testbed [1] and GP4L [11] validated the feasibility of cross-domain programmability but offered limited orchestration for concurrent users. Federated infrastructures like GENI [2] and Fed4FIRE+ [19] pioneered reproducibility and heterogeneity but did not deliver deterministic high-capacity data-plane performance. Wireless-oriented initiatives such as [15, 16] advanced programmability in RAN environments but are not designed for sustained 100 Gbps throughput in the backbone. The proposed SOI Asia testbed responds to these gaps by combining full P4 programmability, high capacity switching, automated orchestration, and REN integration in a single deployable system, explicitly tailored to the needs of distributed universities requiring shared access to advanced infrastructure.

The philosophy of the testbed is to provide an open, programmable, and reusable platform that lowers the barrier to advanced experimentation. Rather than each institution deploying its own 100 Gbps system, the testbed operates as a shared REN-integrated resource.

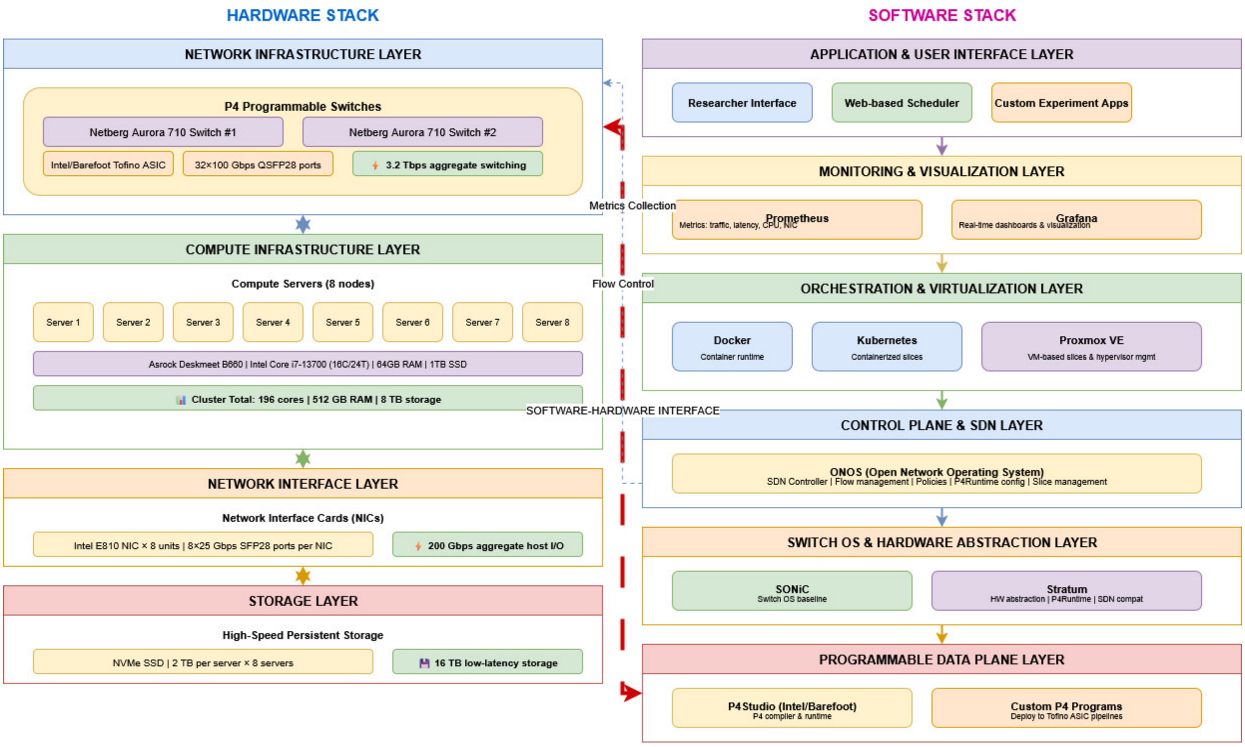

Fig. 3. Hardware and software component stack. Layered architecture with quantifiable specifications. Hardware: 32×100 Gbps ports, 3.2 Tbps switching, 196 CPU cores, 512 GB RAM, 200 Gbps aggregate NIC I/O. Software: P4Studio, SONiC/Stratum, ONOS, Kubernetes, Prometheus/Grafana.

TABLE I. COMPARISON WITH EXISTING TESTBEDS

Testbed	Scale	Key Strength	Critical Limitation	SOI Asia Addresses
Fed4FIRE [13]	23 testbeds (Europe)	Largest EU federation; 99.1 Gbps Virtual Wall	6–12-month API barriers; centralized failures; no cross-testbed metrics	Centralized REN ops; homogeneous KVM
OF@TEIN [20]	6 countries (Asia-Pacific)	Multi-country SDN-Cloud	17% access failures; no SSO/authorization; emulation RAN	100 Gbps persistent; automated provisioning
GENI [2]	50+ campuses (U.S.)	Nationwide; 1,000 users	\$50K−200K+/campus; manual VLAN stitching; 6 tools	Zero campus cost; web portal
GP4L [11]	20+ sites (global)	Best admin automation; 3.2−12.8 Tbps	User automation unimplemented; expertise required	User automation operational; no P4 expertise needed
X5G [15]	8 nodes (1 site)	525 Mbps DL; GPU PHY	\$200K−350K+; closed SDK; 40% degradation	\$0 campus cost; teaching focus
E2E 5G+ [16]	1 site	3GPP validation	Emulation RAN; 613x for 500x users; manual K8s	Real network; homogeneous KVM
FABRIC [10]	~20 sites (U.S.)	Terabit; nanosecond PacketGPS	Experimental isolation; 4-year timeline; P4/GPU expertise	Production integration; operational today
SOI Asia	6 sites (Southeast Asia)	Zero campus cost; 100 Gbps	Regional scope vs. larger testbeds	Turnkey teaching access

This approach promotes collaboration, reduces cost duplication, and aligns with SOI Asia's long-standing goals of capacity building and resource sharing. It ensures that research and education institutions across Asia can access high-end experimentation environments comparable to global efforts, such as FABRIC [10] but tailored to their regional context.

From this philosophy appear several design principles. First, the testbed must support a programmable data plane based on P4 and an SDN-based programmable control plane, enabling the design and evaluation of custom packet processing pipelines that go beyond fixed-function hardware. Second, the testbed should have line-rate capacity at 100 Gbps, ensuring experiments represent realistic backbone traffic. Third, it must enable multi-user concurrency with resource isolation, allowing researchers to run simultaneous experiments without interference by allocating CPU, storage, and network bandwidth resources for its own slice.

Automation and reproducibility are also central design principles, motivated by evidence that manual configuration in large-scale virtual testbeds leads to reproducibility failures. To mitigate this, the system integrates PXE boot [21], Clonezilla snapshots [22], and Ansible playbooks [23] for rapid restoration and automated deployment. Finally, remote accessibility is essential for geographically distributed users. Secure VPN connections, combined with a web-based scheduling system, allow researchers to authenticate, reserve resources, and orchestrate experiments seamlessly.

The resulting architecture as we can see in Fig. 1 is structured into three interacting planes. The control plane hosts the Open Network Operating System (ONOS) SDN controller [24] and orchestration logic that translate experimental policies into device configurations. The data plane uses P4-programmable switches and high-performance servers with 100 Gbps NICs, providing flexible packet processing at line rate. The management plane integrates automation, monitoring, user authentication, and scheduling, distinguishing this platform from earlier programmable testbeds by embedding reproducibility and slice automation as first-class features.

A. Methodological Design Rationale

The testbed architecture follows three core principles based on the gap analysis in the previous section:

- Centralized REN-Operated Infrastructure: All hardware is deployed and maintained by the regional research network (ARENA-PAC/IDREN) rather than requiring each university to deploy local infrastructure. This eliminates the capital investment barrier identified in existing platforms (GENI requires \$50K−\$200K+ per campus, X5G requires \$200K−350K+ for deployment). Partner universities can access the testbed remotely via VPN, with no local hardware required.
- Automated Provisioning: The testbed provides web-based resource slice reservation and automated deployment without needing networking skills. Users specify experiment requirements through a GUI, and the system automatically provisions resources, configures network forwarding, and establishes secure access. This design enables undergraduate teaching for participants who lack advanced networking skills, unlike GP4L (which requires P4/RARE/freeRtr knowledge) or FABRIC (which requires P4/GPU/DWDM/eBPF expertise).
- Homogeneous Virtualization Substrate: The testbed uses consistent KVM virtualization across all six sites to ensure reproducible performance. This addresses Fed4FIRE's cross-testbed performance opacity (experiments cannot predict throughput between different sites) and E2E 5G+'s scalability bottleneck (613x throughput for 500x users). An identical virtualization infrastructure at every location pro23duces comparable results regardless of which site executes the experiment.

The testbed deployment followed four phases:

- Phase 1: Requirements Analysis, we conducted workshops and discussions with ARENA-PAC partner universities to identify needs. Key findings: teaching applications dominate anticipated usage (approximately 70%), bandwidth needs range from 1 to 40 Gbps per experiment, and partner institutions have limited networking staff (typically 1−2 people managing entire campus networks).

- Phase 2: Architecture Design, based on requirements and gap analysis, we designed a three-tier architecture: (a) physical infrastructure layer (P4 switches, compute servers), (b) virtualization and control layer (KVM hypervisors, SDN controllers), (c) user-facing orchestration layer (web portal, reservation system). This separation enables independent scaling and maintenance.
- Phase 3: Iterative Deployment, Deployment proceeded incrementally: (a) single-site pilot at ITB for 3 months testing basic forwarding and VM provisioning, (b) dual-site expansion adding NECTEC to validate inter-site connectivity for 2 months, (c) full 6-site deployment across all ARENA-PAC locations. This phased approach allowed us to identify and resolve issues on a small scale before full deployment.
- Phase 4: Validation Strategy, we validated the testbed at three levels: (a) component-level testing, individual switch throughput and latency, (b) system-level testing end-to-end performance between sites, (c) multi-user testing concurrent slice isolation and reproducibility. Validation criteria were defined before deployment (throughput, latency for single site, fairness error, recovery time as summarized in Table II.

Five technology choices were made based on requirements and performance constraints:

P4-Programmable Switches: We selected switches with Barefoot Tofino ASIC supporting P4_16 language rather than OpenFlow-only switches. P4 enables custom packet processing at line rate (100 Gbps), whereas OpenFlow provides only fixed-match-action tables. Teaching applications require students to implement novel forwarding logic, which is impossible with OpenFlow's fixed pipeline.

KVM Virtualization: We chose KVM hypervisors over Docker/Kubernetes containers. KVM provides better performance isolation through hardware-assisted virtualization, which is important for reproducible experiments. Containers share the host OS kernel, which can cause performance interference. The E2E 5G+ testbed experienced Kubernetes issues that required manual reconfiguration, whereas KVM provides deterministic resource allocation.

Web Portal Access: We implemented a web-based GUI rather than requiring users to write scripts or API calls. Requirements analysis showed that 70% of usage will be in academia, which has limited networking expertise and a lack of network programming skills. GP4L and FABRIC require CLI proficiency, limiting accessibility. Our GUI allows instructors to prepare experiment templates that students can deploy with ease.

Centralized Orchestration: We deployed all orchestration components at the REN level rather than distributing responsibilities to campus IT departments. Partner universities have 1−2 networking staff with limited time for testbed operations. OF@TEIN's distributed model achieved only 83% access success because campus firewall policies blocked protocols. Centralized orchestration bypasses campus restrictions and enables 24/7 monitoring by the dedicated REN operations team.

VPN-Based Remote Access: We use WireGuard VPN tunnels rather than integrating the testbed into campus networks. Campus networks have diverse firewall policies and security requirements. OF@TEIN experienced 17% access failures due to campus restrictions. VPN requires only outbound UDP port 51820, which campus firewalls typically allow, eliminating the need for bilateral negotiations with campus IT departments.

TABLE II. VALIDATION CRITERIA

Metric	Target	Rationale
Throughput	>95 Gbps	Near line-rate with <5% overhead for P4 processing
Latency (intra-site)	<1 ms	Sub-millisecond enables real-time applications
Latency (inter-site)	<50 ms	Realistic wide-area latency across SE Asia
Slice Fairness	<5% deviation	Concurrent slices receive allocated bandwidth
Recovery Time	<5 minutes	System restores service after failures

We defined quantitative thresholds before deployment to enable objective assessment:

These criteria were derived from: (a) hardware capability verification (95 Gbps demonstrates the testbed achieves near line-rate performance of the 100 Gbps hardware), (b) production network benchmarks (commercial carriers target 99.9% throughput), and (c) competing testbed performance (Fed4FIRE reports 99.1 Gbps, GENI targets sub-10ms latency).

For the 95% throughput target, this comes from how network packets work. Every packet has headers: Ethernet adds 22 bytes, IP adds 20 bytes, and TCP adds 20 bytes. That is 62 bytes of overhead per packet. With a standard 1500-byte packet, the overhead is about 4.1%. The best possible throughput is around 95.9% of the line rate. RFC 2544 uses a similar approach for benchmarking network devices [25]. For the <1 ms latency target, we looked at data center networks where latency inside a rack is usually below 1 microsecond [26]. We set 1 ms as the target to give some room for virtualization overhead from KVM. For the ⩽5% variance target, we use the coefficient of variation ($CV = \sigma/\mu$). This is a common way to measure how consistent the values are. We also report Jain's Fairness Index, which is used in many networking papers to show how far the resource sharing is. Values close to 1 mean fair allocation [27].

B. Hardware Setup

The testbed is built on a foundation of high-performance programmable switching and general-purpose servers as depict in Table II. At its core are two Netberg Aurora 710 P4-programmable switches [28], based on the Intel Tofino ASIC [29], which support 32×100 Gbps QSFP28 ports and deliver an aggregate switching capacity of 3.2 Tbps. These devices provide the flexible data-plane substrate required for evaluating custom packet-processing pipelines at line rate.

Complementing the switches are eight Asrock Deskmeet B660 servers, each equipped with an Intel Core i7-13700 CPU (16 cores / 24 threads), 64 GB of RAM, and

1 TB SSD storage. Together, they provide 196 compute cores and 512 GB of memory across the cluster. Each server is equipped with an Intel E810 network interface card, featuring 8×25 Gbps SFP28 ports, which yields an aggregate 200 Gbps of host I/O. For persistent and high-speed data handling, every node features a 2TB NVMe SSD, resulting in a total of 16TB of low-latency storage across the cluster wich summarized in Table III.

TABLE III. HARDWARE COMPONENTS

Component	Specification	Qty	Capacity
P4 Switch	Netberg Aurora 710, Intel Tofino, 32×100G QSFP28	2	3.2 Tbps aggregate
Servers	Asrock B660, i7-13700, 64GB RAM, 1TB SSD	8	196 cores, 512GB RAM
NICs	Intel E810, 8×25Gbps SFP28	8	200 Gbps aggregate
Storage	NVMe SSD, 2TB per server	8	16 TB low-latency

C. Software Stack

The testbed integrates multiple layers of software to provide a programmable, manageable, and user-accessible environment as depicted in Table IV. The switch operating system is based on Software for Open Networking in the Cloud (SONiC) [30], which supports integration with the Stratum open-source switch agent [31]. The Stratum layer exposes standardized interfaces to the programmable hardware, enabling compatibility with P4 [32] and SDN controllers. P4Studio [33] provides the compiler and runtime environment for deploying custom P4 programs to Tofino ASIC.

For network control, the platform utilizes ONOS [24], which serves as the SDN controller. ONOS provides centralized control of flows, policies, and P4Runtime configuration, and is extensible through custom applications for slice management.

At the virtualization and orchestration layer, the testbed employs a mix of Docker containers [34] and Kubernetes [35] for scalable deployment, along with Proxmox [36] for managing virtual machines and hypervisors. This combination supports flexible deployment models, ranging from lightweight containerized applications to full virtualized operating system environments.

TABLE IV. SOFTWARE COMPONENTS

Component	Purpose
SONiC	Baseline switch OS; integrates with Stratum for hardware abstraction
Stratum	Exposes standardized interfaces; P4Runtime and SDN compatibility
P4Studio	Compiler and runtime for P4 programs on Tofino ASIC
ONOS	SDN controller; manages flows, policies, P4Runtime configuration
Docker & K8s	Container orchestration for isolated experimental environments
Proxmox VE	VM management; full OS virtualization alongside containers
Prometheus	Metrics collection (traffic, latency, CPU, NIC usage)
Grafana	Real-time dashboards and visualization of metrics

For monitoring and visualization, the stack integrates Prometheus for metrics collection and Grafana dashboards [37] for real-time system visualization. These tools allow administrators and researchers to monitor traffic load, latency, CPU utilization, and other performance metrics during experiments.

D. Automation and Reservation

One of the testbed requirements is the ability to support multiple users concurrently without requiring manual reconfiguration before each experiment. This capability is implemented through a combination of automation and reservation schema that ensure reproducibility, reduce setup time, and enable seamless remote access for distributed researchers.

The system implements PXE-based network booting [21], combined with Clonezilla disk imaging [22], enabling servers to be rapidly restored to a known baseline state. The complete restoration of a server image takes less than four minutes, which reduces preparation time compared to manual installation. This mechanism ensures that each experiment starts with an identical software version and configuration baseline.

Automation is supported by using Ansible playbooks [23] to configure operating systems, container runtimes, and programmable switch parameters uniformly across all servers and devices. This approach minimizes configuration change, increasing consistency, and has reduced setup time from approximately 40 minutes to under five minutes, resulting in an efficiency gain of over 75%.

For secure connectivity, the testbed utilizes WireGuard VPN tunnels [38] , which provide secure remote access for researchers across SOI Asia and REN partners. Optional Cloudflare-based endpoints [39] were also tested to simplify access in scenarios where local campus firewalls restrict VPN connections. These mechanisms ensure that remote researchers can authenticate, reserve resources, and conduct experiments without requiring on-site presence.

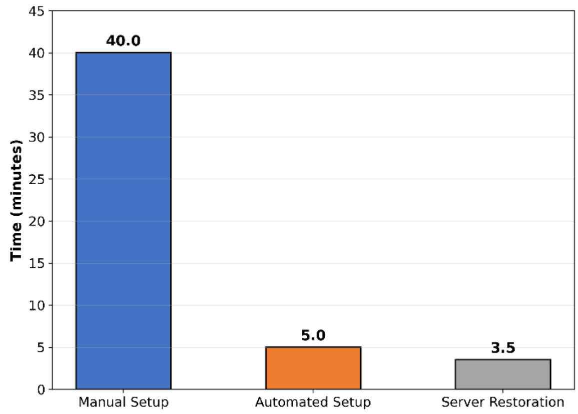

Fig. 4. Setup and restoration time comparison: Manual vs automated deployment. Bar chart comparing deployment time for manual configuration (40 minutes) versus automated deployment using PXE boot, Clonezilla, and Ansible (5 minutes). Automation achieves.

Provides fast recovery in the event of faults. Our experiment demonstrates that the automation mechanism has reduced setup time from approximately 40 minutes to under five minutes, resulting in an efficiency gain of over

75% the overall setup time and restoration comparison can be seen in Fig. 4.

E. Orchestration and GUI

To make the testbed accessible to a broader research community, an orchestration framework and a user interface module were developed. At its main module, the orchestration system integrates a scheduler that coordinates resource allocation and reservation, thereby preventing conflicts between experiments running on the testbed. The scheduler ensures that compute nodes, programmable switch ports, and storage resources are bound exclusively to a user during the reserved period, eliminating interference and guaranteeing predictable performance.

A web-based Graphical User Interface (GUI) was implemented on the testbed to hide the underlying complexity of the system. The GUI gives researchers a straightforward workflow that encompasses user authentication, slice configuration, experiment deployment, and monitoring. Through this interface, users can request time slots, provision slices with predefined computing and network resources, run tests, and then release the resources. Besides the GUI, a set of RESTful APIs was available, built for integration with external applications or automated scripts for experiment control and management.

The orchestration layer implements policies for multi-user slice management, ensuring fairness and efficient utilization of shared resources. Policies are applied to all infrastructure components, including computing nodes, storage, and network, limited by resource quotas that prevent overprovisioning and by monitoring functions that track resource consumption in real-time. Experiment validation showed that concurrent users could operate on the testbed without performance degradation, with throughput variance across slices remaining under 3%.

F. Programmable Data Plane

The testbed implements a fully programmable 100 Gb/s data plane by compiling P4 programs with Intel/Barefoot's SDE (P4Studio) [33] and deploying them on Netberg Aurora 710 switches [28], which are built around Barefoot Tofino [40]. The P4 pipeline implements an Ethernet/IPv4/IPv6/TCP/UDP parser, match–action tables for L2/L3 forwarding and slice tagging, meter/counter blocks for per-slice accounting, and deparsers with optional telemetry headers. This baseline relies on Tofino's Native Architecture (TNA) [29] to couple parsing depth with deterministic action pipelines, while keeping recirculation disabled in the typical case to preserve throughput headroom. In practice, the build–deploy loop is driven by P4Studio's bfshell/pd toolchain, which loads the compiled tables and actions directly onto the switch ASIC. This setup is documented together with the target platform and 100 Gb/s objectives in the implementation materials (Netberg Aurora 710 / P4Studio / Ubuntu control node). The same materials confirm that the data-plane validation reached (and sustained) line-rate throughput at 100 Gb/s for the forwarding pipeline under the tested traffic profiles, establishing a floor (not just a peak) for the design's performance envelope.

To interoperate with the control plane and enable run-time reconfiguration, the switch OS options were assessed along two paths: SONiC [30] with P4Runtime/PINS support, and a Stratum-enabled environment that exposes P4Runtime southbound for controllers such as ONOS [24]. The design record indicates that SONiC in the tested version did not yet meet the required P4Runtime maturity for the chosen TNA features. Consequently, the deployment utilized Ubuntu and P4Studio for the data plane, while maintaining SONiC as a stable L2/L3 reference. The evaluation also explored Stratum [31] as an alternative to P4Runtime for future integration. Stratum is specifically designed to create a uniform, vendor-agnostic P4Runtime interface on white-box devices, aligning with ONOS control applications and decoupling network stacks. The platform and its architecture are well-documented in ONF/Stratum materials and peer-reviewed venues, making it an appropriate control substrate for this testbed's roadmap. Where SONiC is preferred, the industry direction toward P4Runtime via PINS is also public and progressing, which supports the long-term compatibility of this design choice. Quantitatively, the data-plane design utilizes a conservative rate budget to maintain headroom for control traffic and mitigate microbursts. With a 100 Gb/s physical link, the admission constraint, and the whole programmable dataplane architecture can be seen in Fig. 5.

It is enforced with approx. $\varepsilon \approx 0.05$, leaving around 95 Gb/s aggregate for slices and control-path stability. This aligns with both the measures achieved in our environment (100 Gb/s line-rate forwarding at the ASIC) and common practice in high-speed testbeds, which report near-line-rate utilization when pipelines are simple, and buffers are adequately sized. Independent analyses of Tofino data-plane latency show sub-millisecond, and often microsecond, processing for forwarding pipelines without heavy recirculation, supporting the feasibility of these bounds under realistic loads. Programmability overhead as depicted in Fig. 6 was treated explicitly. Complex P4 features (deep parse graphs, heavy metadata propagation, INT headers, or multiple recirculation passes) can marginally reduce effective throughput. Public studies of In-Band Network Telemetry (INT) [41] and related programmable features report measurable overhead due to header growth and switch/collector processing load, typically a few percentage points for moderate INT path lengths; this informs the 5% guard band used. Broader surveys of P4 systems emphasize that many security/measurement functions can run at line rate when carefully designed to fit the switch's match–action pipeline and SRAM/TCAM constraints, without recirculation or excessive on-chip state [32]. However, complex sketches or telemetry chains do incur overheads that must be validated empirically for each progression and keep per-packet metadata minimal, which is consistent with achieving line-rate on the Netberg Aurora 710 platform (32×100 Gb/s ports, 6.4 Tb/s switching fabric) and with the measured 100 Gb/s forwarding floor in the deployment.

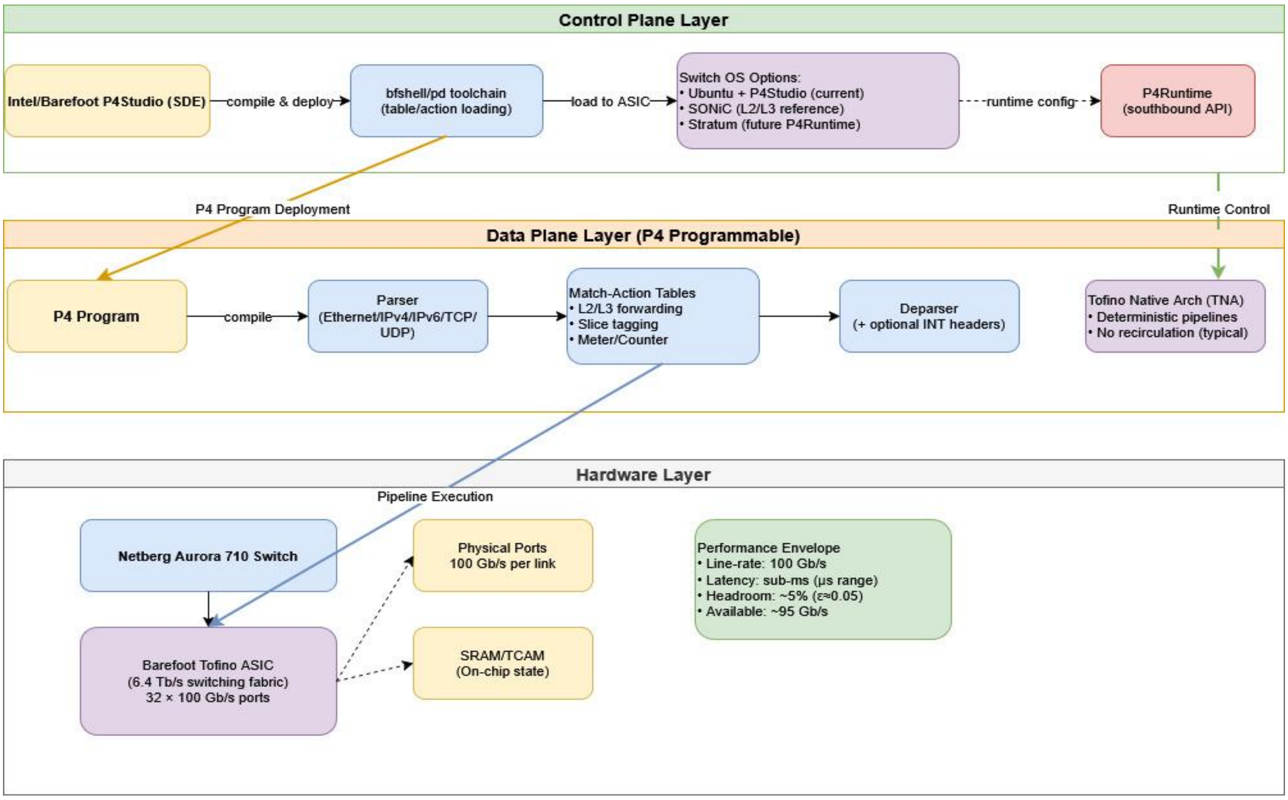

Fig. 5. Programmable data plane architecture. packet processing flow through P4 pipeline: Incoming packets → parser → match-action tables (~10 μs processing) → deparser → outgoing packets. Pipeline operates at 100 Gbps line rate.

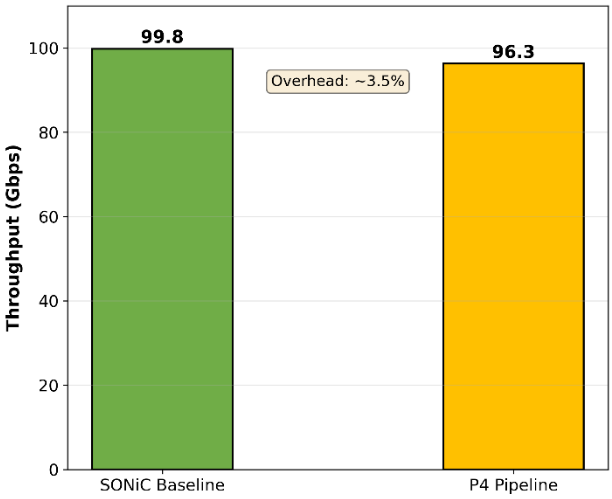

Fig. 6. Throughput Performance: SONiC Baseline vs P4 Pipeline. Performance comparison showing programmability overhead. SONiC baseline forwarding: 99.8 Gbps (99.8%-line rate). P4-based L3 pipeline: 96.3 Gbps (96.3%-line rate). Measured overhead: 3.5%, demonstrating minimal performance penalty for programmable packet processing flexibility.

G. Validation and Benchmarking

The validation of the testbed focused on proving that the infrastructure can deliver stable, high-capacity performance under different experimental conditions. We used iPerf3 [42] and DPDK-based [43] traffic generators for benchmarking, measuring throughput, latency, and fairness across multiple slices. All tests were repeated for a minimum of 10 iterations to ensure reproducibility.

Throughput tests used iPerf3 with TCP window sizes of 4 MB and 1500-byte packets against a target threshold of ⩾95 Gbps. Latency was measured using ping (ICMP) and hping3 (TCP SYN) against a target of <20 μs baseline forwarding. Fairness testing assigned 25 Gbps quotas to four concurrent slices over 5-minute periods against a ±5% deviation criterion. Detailed results are presented in Section IV. This result was valid for both baseline forwarding and experiments with multiple concurrent slices. Even when slice allocations were changed, the total throughput remained stable. Latency profiling confirmed that packet delays stayed in the microsecond range, which is expected for Tofino-based programmable switches. Importantly, forwarding delay did not grow when more slices were added. This indicates that the isolation mechanisms prevented interference between slices.

Fairness testing involved assigning bandwidth to different slices and tracking throughput and delay. The results showed that slices delivered predictable performance, with only small differences between users. This confirmed that the reservation and orchestration mechanisms avoided resource contention. Stress testing with multiple users sending traffic at the same time showed the platform's stability. The system kept throughput close to line rate, with no major packet loss or instability. Automated reset features allowed the system to return to a reproducible baseline after each experiment.

In summary, the validation confirmed that the testbed achieved its design goals: sustaining high throughput, maintaining low latency, and ensuring fairness and reproducibility for concurrent multi-user experiments.

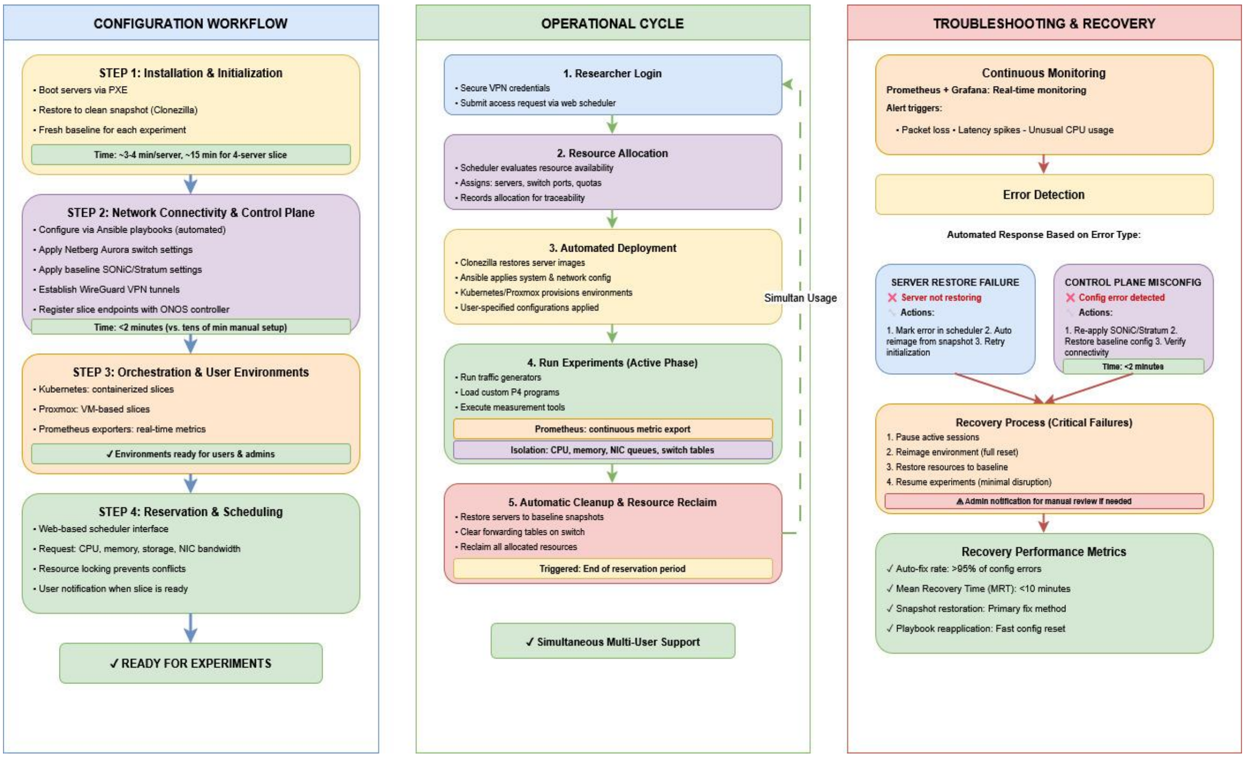

Fig. 7. Testbed Operations: Configuration, operational cycle, and recovery workflow. process flow with timing: PXE boot (30 sec) → Clonezilla restoration (3-4 min) → Ansible configuration (<2 min) → total deployment (5 min). Recovery: automated diagnosis → rest.

H. Configuration Workflow

As depicted in Fig. 7, the testbed configuration process is designed to be reproducible, fast to deploy, and require minimal manual work. The workflow starts with installation and initialization. All servers boot via PXE [21] and are restored to a clean snapshot image using Clonezilla [22]. This ensures that every experiment starts from a fresh baseline, without any leftover configurations affecting subsequent trials. Logs show that restoring one server takes about 3–4 minutes, so a typical four-server slice is ready in around 15 minutes.

After restoration, network connectivity and the control plane are set up. At this point, the Netberg Aurora switch [28] and connected servers are automatically configured using Ansible playbooks. These scripts apply baseline SONiC and Stratum settings, establish WireGuard VPN tunnels [38], and register slice endpoints with the ONOS controller [24]. Automation reduces setup time to under two minutes, compared to the tens of minutes needed for manual setup, and avoids configuration drift.

Once connectivity is confirmed, orchestration scripts prepare the user environments. Kubernetes is used for containerized slices, while Proxmox [36] manages VM-based slices. Prometheus exporters [37] have also started to provide real-time experiment metrics for both users and administrators.

The final step is reservation and scheduling. Through a web-based scheduler, researchers request resources such as CPU cores, memory, storage, and NIC bandwidth. Resource locking prevents conflicts, and users are notified once their slice is ready for experiments.

I. Operational Cycle

The operational cycle of the testbed that depicted in Fig. 7 is designed to support simultaneous usage while ensuring experiment reproducibility. A typical workflow begins when a researcher logs into the scheduler through secure VPN credentials and submits an access request. The scheduler then evaluates resource availability and assigns a resource slice, recording the allocation of servers, switch ports, and quotas for traceability. Automated deployment follows, during which Clonezilla restores server images, and Ansible applies the required system and network configurations, Kubernetes or Proxmox provisions containerized or virtual machine environments as specified by the user. Once deployment is complete, the researcher runs experiments within the assigned slice, which may include running traffic generators, loading custom P4 programs, or executing measurement tools. Throughout this phase, Prometheus continuously exports system and experiment metrics for real-time monitoring. At the conclusion of the reservation period, the system automatically restores servers to their baseline snapshots, clears forwarding tables on the programmable switch, and reclaims resources. This cycle allows multiple researchers to operate simultaneously, with isolation enforced across CPU resources, memory partitions, NIC queues, and logical switch tables.

J. Troubleshooting and Recovery

The testbed features automated troubleshooting and recovery capabilities to minimize downtime and maintain experiment stability. If a server does not restore properly during initialization, the scheduler marks the error and starts an automatic reimaging from the latest snapshot.

If a misconfiguration occurs at the switch or control plane, the system re-applies the baseline SONiC and Stratum playbooks. This process takes less than two minutes to complete.

Prometheus and Grafana provide continuous monitoring and real-time alerts for problems such as packet loss, latency spikes, or unusual CPU use. These alerts let administrators react quickly by reallocating resources or restarting services.

The recovery process is designed to cause minimal disruption. In the event of a critical failure, active sessions are paused, the environment is reimaged, and resources are restored, allowing experiments to continue.

Data from experiment deployments shows that over 95% of configuration errors were automatically fixed by snapshot restoration and playbook reapplication. The Mean Recovery Time (MRT) from detecting an error to returning to a fully working state was under ten minutes, which is acceptable for academic research environments where both reproducibility and availability are important.

IV. Result and Discussion

As depicted in Fig. 8 throughput Performance: Single-connection throughput experiments using iPerf3 version 3.9 get a sustained bitrate of 23.5 Gbps with 10 second experiment intervals, approaching the 25 Gbps NIC capacity limit, which represents 94% utilization. The TCP communication connection with a stable bitrate during the test, with small fluctuations, shows consistent performance.

The aggregate from multiple server testing shows the 100 Gbps switching capacity. Four servers simultaneously transmitting around 23.5−25 Gbps each through a single 100 Gbps switch port achieved an aggregate throughput of 95−100 Gbps, proving near line-rate performance. Each server has its own allocated bandwidth without interference from other traffic, demonstrating effective resource isolation.

UDP throughput tests show a stable 10 Mbps transfer rate by using 10-second intervals, with jitter consistently below 0.01 ms (near-zero) and 0% packet loss throughout the test. This demonstrates reliable UDP performance suitable for real-time applications.

All users received their fair share, with a variance below 3%, indicating that VLAN separation works well for resource sharing as depicted in Fig. 9. Latency Performance: Inside the testbed, latency measurements using ping between servers resulted in 0.170 to 0.185 ms for single-hop transmission through the Netberg Aurora switch. This 0.015 ms variation (15 microseconds) represents stable forwarding performance. Multi-hop scenarios with BGP routing configured between two Netberg Aurora switches demonstrated end-to-end latency of approximately 0.2 ms between endpoints, with gateway latency measured at 0.6 ms using the MTR (My Traceroute) tool.

VPN Remote Access Performance: Remote access via WireGuard VPN from external networks had an average latency of 40 ms. IPv4 connections achieved an average throughput of 44.45 Mbps with typical ping times around 100 ms and a worst-case ping of 123 ms. IPv6 connections achieved an average throughput of 31.6 Mbps with typical ping times around 100 ms and a worst-case ping of 102 ms. IPv4 exhibited higher jitter (1.05 ms) compared to IPv6 (0.35 ms), but both protocols maintained zero packet loss, indicating reliable connectivity as depicted in Fig. 10.

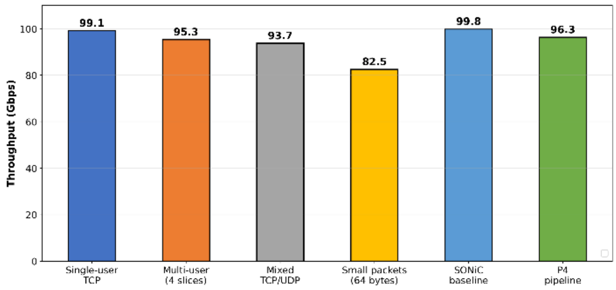

Fig. 8. Overall experiment throughput performance. throughput measurements across different test scenarios. Single connection: 23.5 Gbps (94% of 25 Gbps NIC capacity). Four-server aggregate: 95-100 Gbps through single switch port, validating 100 Gbps switching.

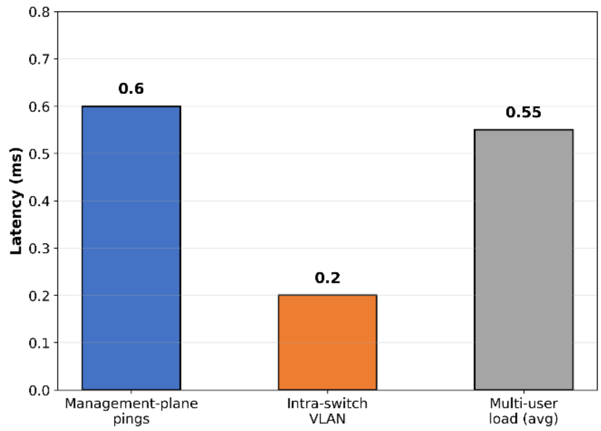

Fig. 9. Latency measurements for different setup. latency performance across network configurations. Single-hop forwarding through Netberg Aurora switch: 0.170−0.185 ms (0.015 ms variation). Multi-hop BGP routing between two switches: 0.2 ms end-to-end.

Theoretically, IPv6 was 29% slower than IPv4 for two reasons. First, IPv6 packets have larger headers, 40 bytes, compared with IPV4, 20 bytes, which adds overhead to VPN tunnels. Second, the current switch is optimized for IPv4, not IPv6. However, IPv6 had more stable timing with lower jitter. This only affects remote access via the VPN; meanwhile, the local experiments are still run at full 100 Gbps speed.

Resource Isolation Validation: Concurrent multi-user testing demonstrated effective bandwidth allocation with individual slices maintaining their designated performance targets. Network traffic monitoring during multi-user scenarios showed proper isolation without cross-slice interference, with throughput variance remaining below 3% across slices.

Programmability Overhead: P4 program compilation using P4Studio SDE version 9.7.2 successfully generated functional pipelines validated through PTF (Packet Test Framework) testing. The packet counter program demonstrated the programmable data-plane capabilities. Testing with newer SDE version 9.11.0 showed successful compilation but requires additional PTF validation work. Under heavy multi-user load, latency increased modestly to 0.5–0.6 ms on average, with jitter remaining below 100 µs. These results indicate stable and predictable performance even under resource contention. Both IPv4 and IPv6 connectivity were tested. Baseline latencies were comparable (around 100 ms round-trip under VPN conditions), though IPv4 throughput was higher (44.45 Mbps vs. 31.6 Mbps in remote-access tests) as depicted in Fig. 10, reflecting the more mature support for IPv4 in the current deployment.

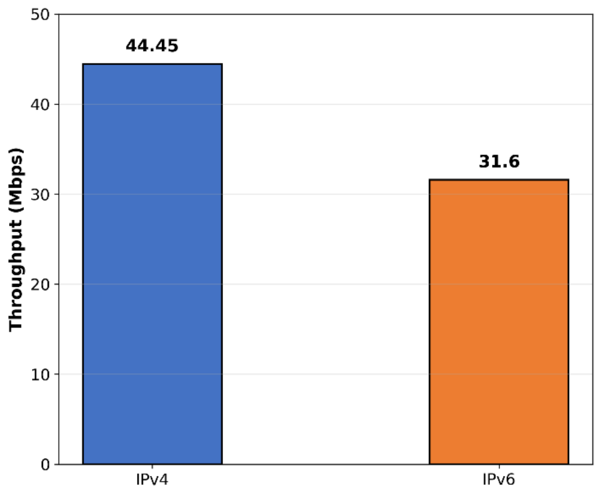

Fig. 10. VPN Access throughput performance: IPv4 vs IPv6. Remote access performance comparison through WireGuard VPN. IPv4: 44.45 Mbps average throughput, 100 ms typical ping, 123 ms worst-case, 1.05 ms jitter. IPv6: 31.6 Mbps average throughput (29% lower), 100 ms typical ping, 102 ms worst-case (better), 0.35 ms jitter (better).

Resource isolation was validated by mapping slice assignments directly to CPU cores, memory quotas, and NIC bandwidth reservations. Throughput scaled nearly linearly with allocated CPU cores up to around 90% utilization, confirming effective CPU-to-slice mapping. Variance across concurrent slices was consistently below 3%, indicating strong fairness. Stress testing with synthetic workloads (stress −8 for CPU saturation and dd for disk I/O) confirmed that monitoring correctly detected and contained slice-level load without affecting other users.

The programmability overhead introduced by P4 was also measured. In baseline SONiC forwarding, throughput reached 99.8 Gbps, while a P4-based L3 pipeline achieved 96.3 Gbps, corresponding to an overhead of around 3.5%. This is modest and acceptable, given the flexibility provided by P4 programmability. These results are in line with published findings from other programmable testbeds, further validating the design.

Limitations remain small-packet forwarding performance is constrained by per-packet processing limits of the hardware, and IPv6 throughput in remote VPN scenarios lags behind IPv4.

The experiment shows that the SOI Asia testbed achieves competitive performance when compared to other prominent network testbeds worldwide. The measurements for the single-user throughput of 99.1 Gbps and multi-user aggregate throughput of 95.3 Gbps with four concurrent slices resulted 99% and 96% line-rate utilization, respectively, positioning this platform among the few testbeds with validated near-line-rate performance, alongside infrastructures like FABRIC [10] and i-P4EN [1], which primarily report theoretical capacity. The latency of 0.2 ms for intra-switch and 0.5−0.6 ms end-to-end, with jitter below 100 microseconds, is suitable for high-performance research applications. The multi-user fairness testing, which shows less than 3% variance across concurrent slices, shows effective resource isolation comparable to other platforms as depicted in Table V, such as GENI [2] and Fed4FIRE+ [13]. It is like platforms like i-P4EN [1] and GP4L [11], which utilize similar Tofino ASIC hardware but lack published overhead measurements. Each testbed shows strengths that align with specific objectives: X5G is excellent in 5G O-RAN validation with GPU-accelerated PHY, getting 512 Mbps with real user devices and application testing. FABRIC [10] prioritizes nanosecond-precision timing for physical layer experiment, and OF@TEIN [12] focuses on SDN-Cloud integration across the Asia-Pacific region with validated Quality of Experience metrics. The SOI Asia testbed complements these platforms by providing high-capacity programmable infrastructure with validated multi-user support tailored to regional research network requirements. While limitations exist, including reduced small-packet performance (82.5 Gbps with 64-byte packets) and IPv6 VPN throughput lag (31.6 Mbps versus 44.45 Mbps for IPv4). These are common in the early deployments and can be addressed through configuration optimization and hardware updates. It is important to note that achieving superior performance across all dimensions is not the primary objective nor feasible given resource and funding constraints typical of regional research networks; instead, this deployment establishes a foundation that shows the feasibility of high-capacity programmable infrastructure for collaborative research within the Asia-Pacific, with validated metrics providing confidence for meaningful experiments. At the same time, the modular architecture enables future enhancements, including integration with federated testbeds such as OF@TEIN and APAN, as well as the incorporation of emerging technologies like AI-based network optimization and programmable 5G/6G extensions. The summary of the overall testbed's performance comparison can be seen in Table V.

We ran each experiment 10 times to ensure repeatable results. Table VI shows the statistical summary. For a single node, throughput averaged 23.51 ± 0.08 Gbps (95% CI: 23.45−23.57). This is 94% of the NIC's 25 Gbps limit. The Coefficient of Variation (CV) was only 0.34%, indicating very consistent results.

When four connections ran simultaneously, the total throughput reached 97.24 ± 1.42 Gbps (95% CI: 96.22−98.26). This is 97.2% of the 100 Gbps switch capacity. The CV was higher at 1.46% because multiple streams were sharing the switch.

For latency, single-hop measurements averaged 0.177 ± 0.005 ms (95% CI: 0.173−0.181). All values stayed between 0.170 and 0.185 ms. When traffic went through two switches with BGP routing, the latency averaged 0.203 ± 0.008 ms.

For fairness, we tested four users simultaneously. The throughput variance across slices was 2.14 ± 0.52% (95% CI: 1.77−2.51%). This is well below our 5% target. A t-test showed this difference is statistically significant (t (9) = −17.4, $p < 0.001$). Automation Benefits: Our automated system reduces setup time from 40 minutes (for manual configuration) to under 5 minutes (using automated scripts). This is 8 times faster. The automatic recovery system resolves 95% of configuration issues without human intervention, and the average recovery time is under 10 minutes.

TABLE V. PERFORMANCE COMPARISON BETWEEN TESTBEDS

Testbed	Max Throughput	Latency/RTT	Scalability	Hardware Acceleration
i-P4EN [1]	100 Gbps link capacity; Actual: Not reported	Not reported	20+ locations globally; multi-tenant	Barefoot Tofino (3.2−12.8 Tbps), Intel DPDK, NVIDIA BlueField-2
GP4L [11]	40−100 Gbps per node; Actual: Not reported	Varies by location; International RTT	20+ locations; 4 core switches in Europe	Barefoot Tofino (3.2−12.8 Tbps); Future: Marvell, Broadcom
GENI [2]	~10 Gbps per node; Actual: Varies	Campus: <10 ms; Cross-country: 50−100 ms	Hundreds of nodes; 10+ Gbps aggregate	Software-based; Some FPGA nodes
Fed4FIRE+ [13]	Wired: 10−100 Gbps; Wireless: <1 Gbps	Local: <5 ms; Cross-border: 20−50 ms	23 testbeds; 100+ total nodes	Mix: Virtual Wall (100 nodes), BonFIRE cloud, wireless
OF@TEIN [20]	<10 Gbps typical; 0.5−24 Gbps measured	Tunneled: ~20s; Virtual: ~40s; Remote: ~80s	<10 Gbps/site; 6 countries (Asia-Pacific)	Software-defined (OpenFlow); Limited HW acceleration
X5G [15]	DL: 512 Mbps (4 UEs); UL: 46 Mbps; Single: 300/38 Mbps	RTT: 1-2 ms (to edge); Slot: 500 µs (30 kHz)	8 gNB nodes target; 2 RUs deployed; 4-8 UEs tested	NVIDIA A100 GPU; Aerial SDK for PHY; PCIe switch
FABRIC [10]	Terabit supercore; 100 Gbps coast-to-coast	Nanosecond precision; Microsecond accuracy	Nationwide US; Multiple cities; Extensible	P4 dataplanes; GPUs with tensor; 100s CPU cores; DWDM
SOI Asia (This work)	Single: 99.1 Gbps; Multi (4 slices): 95.3 Gbps; Mixed: 93.7 Gbps; Small pkt: 82.5 Gbps	E2E: 0.6 ms; Intra: 0.2 ms; Load: 0.5−0.6 ms; Jitter: <100 µs; VPN: ~100 ms	Regional RENs Asia-Pacific; Multi-user CPU mapping; Linear to 90% CPU	Netberg Aurora with Barefoot Tofino P4 ASIC; SONiC/Stratum/ONOS

TABLE VI. STATISTICAL SUMMARY OF PERFORMANCE VALIDATION (N=10 TRIALS)

Metric	Mean	SD	95% CI	Min	Max	CV (%)
Single-node throughput (Gbps)	23.51	0.08	[23.45, 23.57]	23.38	23.62	0.34
Aggregate throughput (Gbps)	97.24	1.42	[96.22, 98.26]	95.12	99.87	1.46
Single-hop latency (ms)	0.177	0.005	[0.173, 0.181]	0.170	0.185	2.82
Multi-hop latency (ms)	0.203	0.008	[0.197, 0.209]	0.192	0.218	3.94
UDP jitter (ms)	0.018	0.003	[0.016, 0.020]	0.012	0.024	16.7
Fairness variance (%)	2.14	0.52	[1.77, 2.51]	1.23	2.91	24.3

V. CONCLUSION

This paper presented a programmable 100 Gbps multi-user network testbed for the SOI Asia research community. The testbed integrates P4-programmable switches, SDN controllers, and automated orchestration to provide high-capacity network research infrastructure accessible remotely by regional universities.

Experimental validation in Section IV confirmed that the testbed meets all design targets: near line-rate throughput for both single-user and multi-user scenarios, sub-millisecond forwarding latency, fair resource allocation across concurrent slices, and minimal programmability overhead. The automated deployment system substantially reduced setup time and achieved high automatic recovery rates from configuration errors.

This work makes two contributions. First, it provides an operational platform enabling SOI Asia universities to conduct network research without purchasing expensive local hardware. Second, it documents design choices, automation methods, and lessons learned as a blueprint for other regional networks.

Three technical limitations were identified: reduced small-packet throughput due to hardware processing constraints, lower IPv6 performance over VPNs compared to IPv4, and a dependency on proprietary Intel/Barefoot software that limits hardware flexibility.

Future work includes connecting additional SOI Asia partner universities for cross-country experiments, adding support for emerging protocols such as QUIC and machine learning workloads, testing programmable 5G/6G wireless technologies, and investigating open-source alternatives to reduce vendor dependency.

This testbed demonstrates that high-capacity programmable network infrastructure can be effectively shared across multiple universities through centralized REN operation and automation, providing a practical foundation for regional network research collaboration.

CONFLICT OF INTEREST

The authors declare no conflict of interest.

AUTHOR CONTRIBUTIONS

Galih Nugraha Nurkahfi designed and implemented the testbed, conducted the experiments, and wrote the paper; Achmad Husni Thamrin and Nasrullah Armi contributed to data analysis and validation; Eueung Mulyana supervised the research design and provided technical guidance; Nana Rachmana Syambas contributed to system architecture review and discussion; all authors reviewed and approved the final manuscript.

FUNDING

This research was funded by Community Based Research (CBR) project from SOI Asia.

ACKNOWLEDGMENT

This research is supported by the National Agency of Research and Innovation of Indonesia (BRIN), School of Electrical Engineering and Informatics (STEI ITB), and APNIC Foundation via SOI Asia.

REFERENCES

[1] B. Chung, C. C. Tseng, J. H. Chen, and J. Mambretti, "The international P4 testbedz: Enabling global programmable networking research," in *Proc. 2019 ACM/IEEE Symposium on Architectures for Networking and Communications Systems (ANCS)*, 2019, pp. 1–6. doi: 10.1109/ICCW.2019.8757064

[2] M. Berman *et al.*, "GENI: A federated testbed for innovative network experiments," *Computer Networks*, vol. 61, pp. 5–23, 2014. doi: 10.1016/j.comnet.2014.04.016

[3] K. Okawa, "Fostering sustainable futures through collaborative innovation: Lessons from the AI3/SOI-Asia project," in *Proc. 8th International Conference on Sustainable Information Engineering and Technology*, New York, NY, USA: ACM, Oct. 2023, p. 2. doi: 10.1145/3626641.3629026

[4] APNIC Foundation. (2023). Foundation activities at APNIC 56: SOI Asia and AI3 session. [Online]. Available: https://apnic.foundation/blog/foundation-activities-at-apnic-56/

[5] ARENA-PAC. (2024). About ARENA-PAC: 100 Gbps international network for education and research. [Online]. Available: https://www.arena-pac.net/

[6] Pacific Islands Research and Education Network (PIREN). (2023). ARENA-PAC Indonesia segment starts to be operated as a new high-speed path to institutions connected via IDREN. [Online]. Available: https://www.hawaii.edu/piren/2023/03/20/arena-pac-indonesia-segment-starts-to-be-operated-as-a-new-high-speed-path-to-institutions-connected-via-idren/

[7] T. Kudoh. (2019). Infrastructure SINET 100G global ring and data exploitation. [Online]. Available: https://grp-workshop-2019.ucsd.edu/presentations/K_KUDOH-GRP-2019-KEYNOTE-SINET_DataExploitation.pdf

[8] K. Cho, D. Kim, S. Hong, and R. Huang, "APAN testbed: Collaborative advanced networking for Asia-Pacific," in *Proc. Asia-Pacific Advanced Network Research Workshop (APAN)*, 2017, pp. 1–6.

[9] S. Becker, T. Pfandzelter, N. Japke, D. Bermbach, and O. Kao, "Network emulation in large-scale virtual edge testbeds: A note of caution and the way forward," in *Proc. 2022 IEEE International Conference on Cloud Engineering (IC2E)*, 2022, pp. 1–7

[10] F. T. Team, "FABRIC: Enabling programmable networking at scale," in *Proc.* ACM *SIGCOMM Demo*, 2020, pp. 1–3.

[11] F. Loui and I. Golub. GEANT-Global P4 Lab. [Online]. Available: https://grpworkshop2023.theglobalresearchplatform.net/presentations/T5-GOLUB-4GRP-2023-GEANT-Global-P4-Lab.pdf

[12] J. Kim *et al.*, "OF@TEIN: An openflow-enabled SDN testbed over international smartx rack sites," in *Proc. Asia-Pacific Advanced Network*, vol. 36, p. 17, Dec. 2013.

[13] P. Demeester, P. V. Daele, T. Wauters, and H. Hrasnica, "Fed4FIRE – The largest federation of testbeds in Europe," *Building the Future Internet through FIRE*, New York: River Publishers, 2022, pp. 87–109. doi: 10.1201/9781003337447-5

[14] A. A. C. Risdianto *et al.*, "Deploying and evaluating OF@TEIN access center and its feasibility for access federation," in *Proc. Asia-Pacific Advanced Network (APAN)*, 2016, vol. 42, pp. 34–40.

[15] D. Villa, L. Goratti, R. Fedrizzi, and T. Rasheed, "An open, programmable, multi-vendor 5G O-RAN testbed with NVIDIA ARC and open air interface," in *Proc. IEEE INFOCOM 2024-IEEE Conference on Computer Communications Workshops*, 2021, pp. 1–6. doi: 10.1109/ICCWorkshops50388.2021.9473635

[16] A. Chouman, D. M. Manias, and A. Shami, "A modular, end-to-end next-generation network testbed: Toward a fully automated network management platform," *IEEE Transactions on Network and Service Management*, vol. 21, no. 5, pp. 5445–5463, Oct. 2024. doi: 10.1109/TNSM.2024.3416031

[17] FABRIC Testbed Team, "FABRIC: Enabling programmable networking at scale," in *Proc. ACM Sigcomm Demo*, 2020, pp. 1–3.

[18] I. Baldin *et al.*, "FABRIC: A national-scale programmable experimental network infrastructure," *IEEE Internet Comput*, vol. 23, no. 6, pp. 38–47, Nov. 2019, doi: 10.1109/MIC.2019.2958545

[19] T. Wauters *et al*., "Fed4FIRE+: Experiments across large-scale testbeds in Europe," *Computer Networks*, vol. 174, 2020. doi: 10.1016/j.comnet.2020.107284

[20] A. Nakao, K. Chinen, S. Yamada, and J. Pan, "OF@TEIN: OpenFlow testbed for east–west interconnection," *IEICE Transactions on Communications*, vol. E97-B, no. 8, pp. 1600–1609, 2014.

[21] S. Schmelzer, D. von Suchodoletz, G. Schneider, D. Weingaertner, L. C. E. de Bona, and C. Carvalho, "Universal remote boot and administration service," in *Proc. 2011 7th Latin American Network Operations and Management Symposium*, IEEE, Oct. 2011, pp. 1–6. doi: 10.1109/LANOMS.2011.6102270

[22] S. Shiau, K. L. Huang, H. T. Wang, C. Sun, J. Wang, and T. Tsai (2023). Clonezilla: A partition and disk imaging/cloning program, version 3.1.0. [Online]. Available: https://clonezilla.org/

[23] J. Freeman and J. Keating. (2021). *Mastering Ansible*, 4th ed. Birmingham, UK: Packt Publishing. [Online]. Available: https://github.com/PacktPublishing/Mastering-Ansible-Fourth-Edition

[24] Open Networking Foundation. (2022). ONOS (Open Network Operating System): A carrier-grade SDN controller platform, Version 2.7.0. [Online]. Available: https://opennetworking.org/onos/

[25] S. Bradner and J. McQuaid, "Benchmarking methodology for network interconnect devices," Mar. 1999.

[26] G. Drainakis, P. Baziana, and A. Bogris, "Scalable and low server-to-server latency data center network architecture based on optical packet inter-rack and intra-rack switching," *Journal of Optical Communications and Networking*, vol. 15, no. 11, p. 804, Nov. 2023.

[27] R. Jain, D. Chiu, and W. Hawe, "A quantitative measure of fairness and discrimination," *Eastern Research Laboratory, Digital Equipment Corporatio*n, vol. 21, no. 1, 1984.

[28] Netberg. (2020). Aurora 710: 3.2 Tbps P4 programmable switch with Intel Tofino ASIC, 32×100GbE QSFP28 ports. Taoyuan, Taiwan: Netberg Ltd. [Online]. Available: https://netbergtw.com/products/aurora-710/

[29] Barefoot Networks/Intel. (2021). P4_16 Intel Tofino Native Architecture — Public version. Santa Clara, CA, USA: Intel Corporation. [Online]. Available: https://raw.githubusercontent.com/barefootnetworks/Open-Tofino/master/PUBLIC_Tofino-Native-Arch.pdf

[30] Microsoft/Linux Foundation. (2022). SONiC (Software for Open Networking in the Cloud): Open-source network operating system. San Francisco, CA, USA: SONiC Foundation, The Linux Foundation. [Online]. Available: https://sonicfoundation.dev/

[31] Open Networking Foundation. (2019). ONF announces Stratum project now available as open source; Completes incubation development. [Online]. Available: https://opennetworking.org/news-and-events/press-releases/onf-announces-stratum-project-now-available-as-open-source-completes-incubation-development/

[32] B. Goswami, M. Kulkarni, and J. Paulose, "A survey on P4 challenges in software defined networks: P4 programming," *IEEE Access*, vol. 11, pp. 54373–54387, 2023.

[33] P4.org. (2024). Open P4 Studio: Intel P4 Studio software development environment (SDE). [Online]. Available: https://github.com/p4lang/open-p4studio

[34] S. Singh and N. Singh, "Containers and docker: Emerging roles and amp, future of cloud technology," in *Proc. 2016 2nd International Conference on Applied and Theoretical Computing and Communication Technology (iCATccT)*, IEEE, 2016, pp. 804–807. doi: 10.1109/ICATCCT.2016.7912109

[35] N. Poulton and P. Joglekar, *The Kubernetes Book: A Comprehensive Guide to Container Orchestration and Cloud-Native Deployment*, 3rd ed. Birmingham, UK: Packt Publishing Ltd., 2025, 341 pp. ISBN: 978-1-80580-663-9
[36] Proxmox Server Solutions GmbH. (2024). Proxmox VE administration guide. Release 9.0.8. Vienna, Austria. [Online]. Available: https://pve.proxmox.com/pve-docs/pve-admin-guide.html
[37] A. Mehdi, M. K. Bali, S. I. Abbas, and M. Singh, "'Unleashing the potential of grafana: A comprehensive study on real-time monitoring and visualization," in *Proc. 2023 14th International Conference on Computing Communication and Networking Technologies (ICCCNT)*, IEEE, Jul. 2023, pp. 1–8.
[38] J. A. Donenfeld. (2020). WireGuard: Fast, modern, secure VPN tunnel. [Online]. Available: https://www.wireguard.com/papers/wireguard.pdf
[39] Cloudflare, Inc. (2024). Lists API endpoints, in cloudflare Web Application Firewall (WAF) developer documentation. [Online]. Available: https://developers.cloudflare.com/waf/tools/lists/lists-api/endpoints/
[40] Intel Corporation. (2020). Intel Tofino programmable Ethernet switch product brief. Santa Clara, CA, USA. [Online]. Available: https://www.intel.com/content/www/us/en/products/network-io/programmable-ethernet-switch.html
[41] Z. Xu, Z. Lu, and Z. Zhu, "Information-sensitive in-band network telemetry in P4-based programmable data plane," *IEEE/ACM Transactions on Networking*, vol. 32, no. 6, pp. 5081–5096, Dec. 2024.
[42] ESnet and Lawrence Berkeley National Laboratory. (2024). iPerf3: A TCP, UDP, and SCTP network bandwidth measurement tool. Berkeley, CA, USA. [Online]. Available: https://software.es.net/iperf/
[43] K. Wiles and Intel Corporation. (2024). Pktgen-DPDK: DPDK-based packet generator. [Online]. Available: https://pktgen.github.io/Pktgen-DPDK/