2026 ASEE ANNUAL CONFERENCE June 21–24, 2026 | Charlotte Convention Center, NC
Host Sponsor UNC Charlotte
Paper ID #51966

Learning Through Wiring: Teaching Real-World Networking and Security with a Raspberry Pi Cluster

Yamen Ziadeh, University of Missouri - Columbia
Dr. Fang Wang, University of Missouri - Columbia

Fang Wang is an Associate Teaching Professor in the Engineering and Information Technology Department at University of Missouri. Her research interests include the design, evaluation, and application of virtual and augmented reality, game-based and mobile technologies for healthcare and education, and computational modeling and simulation in engineering. Her research has been funded by the National Science Foundation and the Coulter Biomedical Accelerator Program.

Scottie Dean Murrell, University of Missouri - Columbia
Michael Robert Tompkins, University of Missouri - Columbia
Dr. Hani Salim P.E., University of Missouri - Columbia

Dr. Salim is a professor of Civil Engineering and served at the associate dean for academic programs and student success at the college of engineering between 2015 and 2019.

Learning Through Wiring: Teaching Real-World Networking and Security with a Raspberry Pi Cluster

Abstract

This paper introduces Pod-Pi, a scalable, low-cost, and fully sandboxed hardware laboratory environment designed to bridge the gap between network simulation and real-world implementation. While virtualization and simulation are common in IT education, they often obscure the physical complexities and unpredictable failure modes of production-grade hardware. The Pod-Pi architecture utilizes a Cisco router, a Cisco switch, and five Raspberry Pi devices to create an isolated "branch network" where students can safely configure routing, VLANs, firewall policies, and site-to-site VPNs. We employ a unique two-stage instructional model: students first complete a Packet Tracer simulation individually, followed by a collaborative, role-based lab on the Pod-Pi hardware. Evaluation through pre- and post-surveys and convergence-based performance checks demonstrate significant gains in student confidence and troubleshooting efficacy, particularly in complex security configurations. These results suggest that Pod-Pi offers a replicable framework for institutions seeking to provide authentic, hands-on networking experience without the high cost or security risks of traditional production infrastructure.

1. Introduction

Hands-on, hardware-supported projects are essential for deepening computing and security concepts beyond the capabilities of purely simulated environments. In IT education, Kim et al. [1] note that hands-on lab exercises are crucial for helping students observe how theoretical concepts apply to real systems, but the significant equipment costs often push curricula toward virtualization. While scalable, virtualization can obscure the physical complexities of how systems are actually connected and how they behave as separate, discrete devices. To address this, Khan et al. [2] describe a curriculum approach that “infuses” Raspberry Pi projects across multiple courses, providing depth through project-based tasks like implementing firewalls and intrusion detection systems on a real Linux platform. Supporting this demand for scalable, repeatable hands-on work, Ma et al. [3] emphasize the need for physical platforms that are manageable and easily replicable, explicitly contrasting recent simulation-heavy periods with a renewed requirement for in-person experimentation.

Within network security education specifically, prior research has demonstrated the value of isolated, team-based environments where students can practice realistic attack and defense workflows without risking campus infrastructure. Hill et al. report on an isolated network laboratory where students worked in persistent cooperative teams, highlighting the engagement and learning benefits of realistic, contained environments [4]. Similarly, Abler et al. present a hands-on internetwork security laboratory designed to complement theory with a model Internet environment and practical configuration work [5].

Collectively, these studies establish that while simulated environments offer essential conceptual foundations, physical interaction is necessary to master the physical-logical mappings and the

unpredictable realities of production systems. However, a critical gap remains: the existing literature often treats simulation and hardware as alternatives rather than integrated pedagogical stages, and there is a lack of structured comparison of learning gains between individual simulation and collaborative hardware execution. Furthermore, while prior work highlights the value of isolated labs, there is a notable absence of low-cost, modular hardware designs that incorporate the rapid, automated reset mechanisms required to support high student throughput in modern instructional settings. Our research addresses these voids by evaluating a dual-stage instructional model within a sandboxed pod architecture designed for both depth and operational scalability.

At the same time, networking and security courses face practical constraints that limit the use of real infrastructure. Institutional policies often restrict students' access to privileged configuration tasks, such as routing changes, VPN setup, firewall policy, and network monitoring, due to concerns about disruption and security. As a result, many courses rely heavily on simulations or controlled virtual environments, which may limit students' understanding of how real systems interact and persist in the real world.

Pod-Pi addresses these challenges by operating as a fully sandboxed, air-gapped hardware platform that allows for high-stakes configuration without risking institutional infrastructure. By combining Raspberry Pi endpoints with legacy-capable Cisco routing and switching equipment, we create a pod-based architecture that provides the tactile complexity of production systems while remaining safe for offline, collaborative use. This design enables a direct comparison between an individual simulation-first experience and a collaborative hardware lab under controlled, risk-free conditions. This study is guided by the following objectives:

1. System Design: Demonstrate a low-cost, scalable architecture for implementing high-stakes networking labs in resource-constrained or highly regulated institutional environments.

2. Evaluation: Evaluate the impact of a hardware-based "second stage" on student skill retention and configuration accuracy following a traditional simulation "first stage".

3. Student Outcome: Assess how collaborative, role-based interaction with physical hardware improves student confidence in troubleshooting and problem-solving compared to individual simulation work.

2. System Design

The project began with a practical question: how can students practice security-relevant configurations safely, with minimal dependency on campus IT systems, while still using real devices and real failure modes?

We translated that into constraints:

1. Fully isolated from campus networks and the internet during the lab.
2. Repeatable resets between teams, with minimal instructor overhead.
3. Low cost per seat and scalable across multiple teams.
4. Uses legacy-capable networking features common in the industry and common in older Cisco hardware.
5. Supports structured teamwork and meaningful convergence tests.

The final design is a pod that behaves like a small branch network. Students configure switching, routing, firewall policy, and VPN connectivity to a designated headquarters “HQ” endpoint. Success requires end-to-end function and team coordination.

System Architecture and Hardware

This section provides a technical overview of the Pod-Pi "branch network". Readers will learn about the specific Cisco and Raspberry Pi hardware components, network isolation and access models, and the logical topology used to create a realistic, sandboxed environment for applied security training.

(a) Pod-Pi Hardware Components

Figure 1 shows the Pod-Pi hardware, it is comprised of a Cisco Catalyst 8200 router, a Catalyst 9200 switch, five Raspberry Pi endpoints, and a dedicated wireless access point to create a fully isolated, resettable branch network environment.

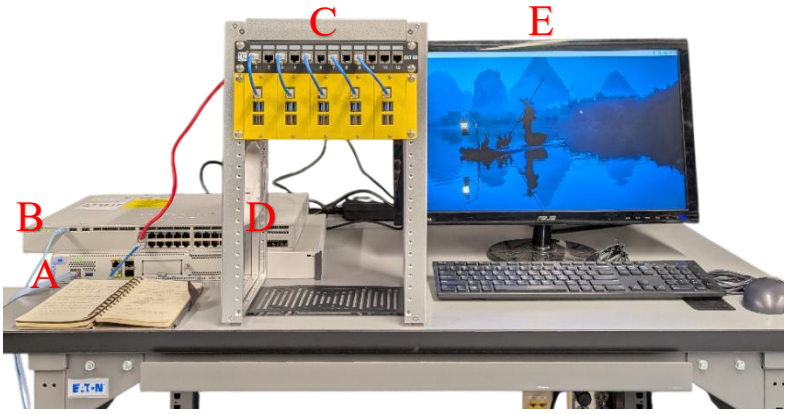


Figure 1. Physical Architecture and Integrated Setup of the Pod-Pi Laboratory Environment

- **[A]** Cisco Catalyst 8200 Router (the bottom unit on the left).
- **[B]** Cisco Catalyst 9200 Switch (the unit directly above the router).
- **[C]** Raspberry Pi Cluster (the yellow rack in the center).
- **[D]** Wireless Access Point (the white circular device mounted inside the frame).
- **[E]** Management Console (monitor and peripherals on the right).

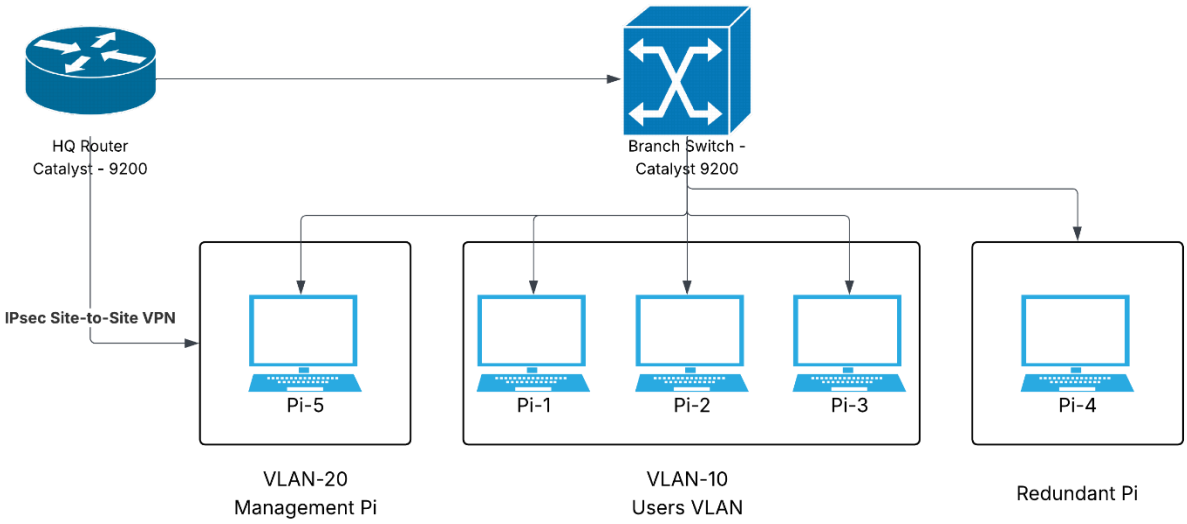


Figure 2. Logical Topology Diagram

Figure 2 maps out the Pod-Pi architecture, illustrating how the physical components are organized into a functional "branch network" for hands-on learning. A Cisco Catalyst 9200 Branch Switch acts as the central hub, distributing connections to five Raspberry Pi devices that serve as student endpoints and utility hosts. To simulate real-world networking, the architecture utilizes VLAN tagging to separate traffic, specifically, Pi-1, Pi-2, and Pi-3 are assigned to the Users VLAN (VLAN-10), while Pi-5 functions as the Management Pi in VLAN-20. The diagram also highlights the IPsec Site-to-Site VPN connection to the HQ Router and designates Pi-4 as a Redundant Pi, supporting the system's goal of ensuring high reliability and operational scalability through quick hardware resets.

Figure 3. Final Setup of the Lab

As shown in Figure 3, we have set up 4 Pod-Pi stations, with 12 students able to take the final exam simultaneously, with enough space to feel comfortable using their own laptops. The design is modular and intended to be replicated.

(b) Network isolation and access model

Students access the pod through a local wireless AP that is intentionally air-gapped from the campus network and the internet. This VNC-based setup creates a strictly controlled environment where interaction is limited to the provided lab devices.

By removing internet access, we intentionally eliminate relying on external step-by-step tutorials, forcing students to rely on their internalized knowledge and peer collaboration to resolve complex configuration tasks. This isolation is a critical pedagogical design choice: it ensures that the subsequent evaluation measures a student's authentic problem-solving capability and teamwork under the offline constraints typical of high-security production environments.

(c) Reproducibility and Redundancy

To ensure the system remains reliable across back-to-back lab sessions, we implemented a structured reset protocol centered on a Linux Bash shell script. Each Raspberry Pi is equipped with a local SSD containing a recovery partition. When triggered, the script stops active networking services and uses the dd utility to overwrite the primary system partition with a baseline "Golden Image".

While developing this part of the project, a significant bottleneck emerged: while the Raspberry Pi re-imaging is automated, resetting the Cisco Catalyst 8200 router and 9200 switch remains a manual process via the serial console. We found that the "write erase" and reload cycle on modern Cisco IOS-XE hardware takes significantly longer (approx. 5–7 minutes) than the Pi re-imaging (approx. 3 minutes) so the total reset time is between 8 and 10 minutes.

Early iterations faced "ghost connections" where the switch MAC tables retained entries from previous teams. We updated the protocol to include a physical power-cycle of the switch to ensure a clear state before the next lab session begins. The instructor then performs a final verification by running a "Clean Slate" ping sweep from a master utility Pi to confirm no unauthorized routing persistence exists.

(d) Curriculum Integration and Lab Design

Each team consists of three students assigned to specific, primary roles: the VPN Engineer, the Switch Engineer, and the Security Analyst. The VPN engineer role, specializing in site-to-site VPNs and managing identities, is a minute replica of a Security Architect in a networking company. The switch engineer's role simulates a Network Engineer's role, and the Security Analyst role matches what you see in Security Operations Centers (SOC) analysts/auditors. By assigning these defined roles, the lab mimics professional workflows where specialization while maintaining a shared responsibility for the end-to-end product. This model prepares students for

the collaborative realities of modern enterprise environments, where "siloed" knowledge is a primary point of failure.

Role Assignment and Rotation Strategy:
In the current study, roles were assigned at the start of the 2-hour lab session and remained fixed to ensure teams could reach the final convergence condition within the tight time limit. However, the design mitigates skill siloing by requiring Shared CLI Accountability. Every student must log into their respective Pi and execute specific verification commands (e.g., *show crypto isakmp sa* or *show ip interface brief*) to prove their portion of the network is functional.

This approach balances two competing pedagogical needs:

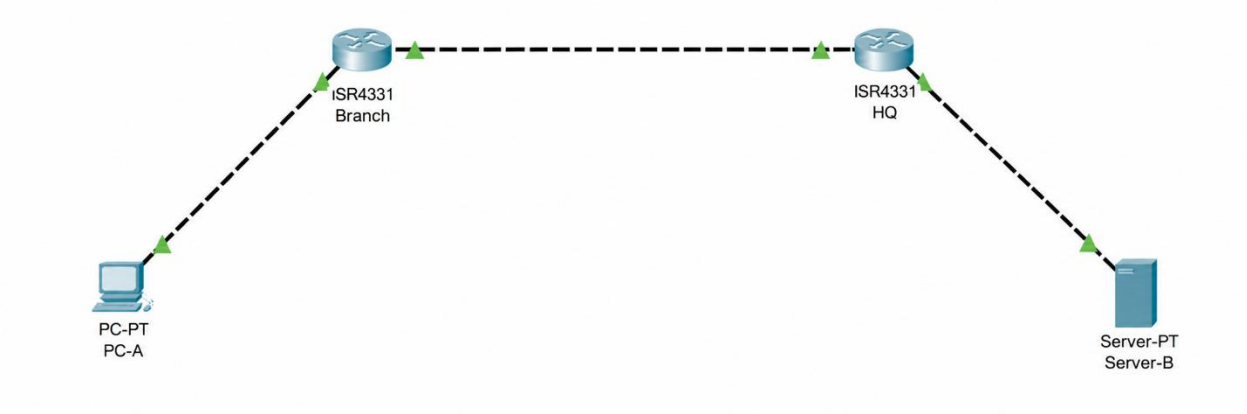
1. Depth: Fixed roles allow students to master one complex area (like Phase 2 IPsec commands) during the lab.

2. Breadth: To ensure holistic learning, future iterations of Pod-Pi will implement a mid-point role swap, where students move to a neighboring device after the initial connectivity is established, forcing them to troubleshoot a configuration they did not personally create.

To simulate real-world operations, we provided cheat sheets. These functioned like playbooks used in industry: specialized commands reminders, verification checklists, and troubleshooting methods. The intent was not to remove thinking, but to reduce irrelevant memory load so students focus on reasoning, debugging, and collaboration.

3. Experiment Study

The study was conducted in a college-level Network Security course, with Network Fundamentals as a prerequisite. Throughout the semester, students completed assignments primarily in Packet Tracer, and instruction emphasized Cisco-specific concepts and workflows. The course enrolled 48 students, but the lab room capacity allowed 12 students at a time. As a result, the hands-on Pod-Pi lab was delivered over four sessions, each lasting 2 hours and split across 2 days.
The students first completed the Packet Tracer simulation (Fig. 4). It’s designed to provide a simplified preview of the final hands-on lab by having students build the same basic two-site topology and validate end-to-end connectivity using a clear addressing plan and routing path. While also building simplified versions of both the site-to-site VPNs and the Zone-Based Firewall (ZBF), to give them an early sense of what the lab structure and success criteria might look like.



Device	Interface	IP Address	Subnet Mask	Default Gateway
Branch Router	Gi0/0 (LAN)	192.168.10.1	255.255.255.0	N/A
Branch Router	Gi0/1 (WAN)	10.1.1.1	255.255.255.252	N/A
HQ Router	Gi0/0 (LAN)	192.168.20.1	255.255.255.0	N/A
HQ Router	Gi0/1 (WAN)	10.1.1.2	255.255.255.252	N/A
PC-A	NIC	192.168.10.10	255.255.255.0	192.168.10.1
Server-B	NIC	192.168.20.10	255.255.255.0	192.168.20.1

Figure 4 Packet Tracer Experiment Design

After completing the Packet Tracer simulation assignment, students move to the hardware section of the lab, where the student teams must reach a final convergence condition that only occurs when the core network and security configuration are correct end-to-end. This includes correct VLAN and trunk setup, valid routing to the target network, properly applied security policies, a site-to-site VPN that is established and functional for the intended traffic, and a ZBF that is verifiably installed in place. Convergence is verified using a simple mechanism such as a Management Pi (Pi-4) server endpoint that is reachable only when the full path is correct, reached by a provided set of pings that need to be successful. Each student must independently demonstrate success from their assigned endpoint, which preserves individual accountability while still requiring a shared team outcome.

We used two surveys to evaluate outcomes across the two instructional stages. The pre-survey was administered after the Packet Tracer simulation lab, and the post-survey was administered after the hands-on Pod-Pi lab. Both instruments were implemented as anonymous Qualtrics surveys and consisted primarily of structured, close-ended items to support consistent scoring and comparison. Both surveys measure knowledge and confidence on a 1 to 10 scale across key categories such as real-world transfer, troubleshooting, firewall policy, and VPN skills.

4. Results

48 students participated in the lab, and 42 students were able to finish all tasks, with some of them only to some degree of completion, meaning not all pings were successful, but the required structure is up, and subsequently, partial credit was given to these sections. Out of all 48 pre- and post-means surveys, we received 30 responses from students.

The convergence test comprises 5 ping checks, each worth 50 points of the final grade, totaling 250 points. In case of a ping failing where it needs to be successful, the instructor inspects the configuration, and gives partial credit based on the level of completion of the problem at hand. Average grade was 226/250, or 90.4%.

We report pre- and post-means by category and the improvement per category. There are 5 categories, each with 2 to 3 questions, totaling 11 questions, each question measures self-reported confidence pertaining to security concepts on a scale from 1 to 10. The categories are:

- General Application and Incident Response: applying course learning to real settings, including how you would respond to real network/security problems.

  Q1: “I feel able to apply what I learned in this course to real life situations outside the lab or classroom”

  Q2: “I could use what I learned in this course to help diagnose and fix basic network problems in a real environment.”

  Q3: “I could use what I learned in this course to help improve the security of a small real network”

- Network Tracing and Design: mapping between logical design and physical implementation, including tracing, cabling, and topology alignment.

  Q4: “I can trace the path a packet takes through a rack by following cables, port labels, and status indicators from source to destination”

  Q5: “I can relate a logical network diagram, with subnets and VLANs, to the physical ports and cables on real equipment”

  Q6: “I could plan how to connect end devices, switches, and a router so that the physical cabling matches a given logical topology”

- Connectivity Troubleshooting and Testing: using a structured troubleshooting approach and validating behavior using simple tests.

  Q7: “If two devices in a small network cannot reach each other, I could outline a step by step plan to diagnose the problem”

  Q8: “I can use simple traffic such as pings, web requests, or SSH connections to test whether firewall or access control rules are working as intended”

- Firewall Policy Concepts: understanding and modifying firewall policy constructs (Zone-Based Firewall concepts).

Q9: “I understand the logical relationship between Zones, Class-Maps, and Policy-Maps, and I could modify an existing configuration to allow a new protocol (such as DNS or SSH) to pass through the firewall.”

- VPN Troubleshooting: verifying tunnel functionality and diagnosing “tunnel up but traffic not working.”

  Q10: “I can tell, using VPN status information on the router and simple tests like pings or web requests, whether a VPN connection is up and carrying the traffic it is supposed to carry”

  Q11: “If a VPN tunnel appears to be up but users at one site still cannot reach services at the other site, I could outline several concrete things to check to find the problem”

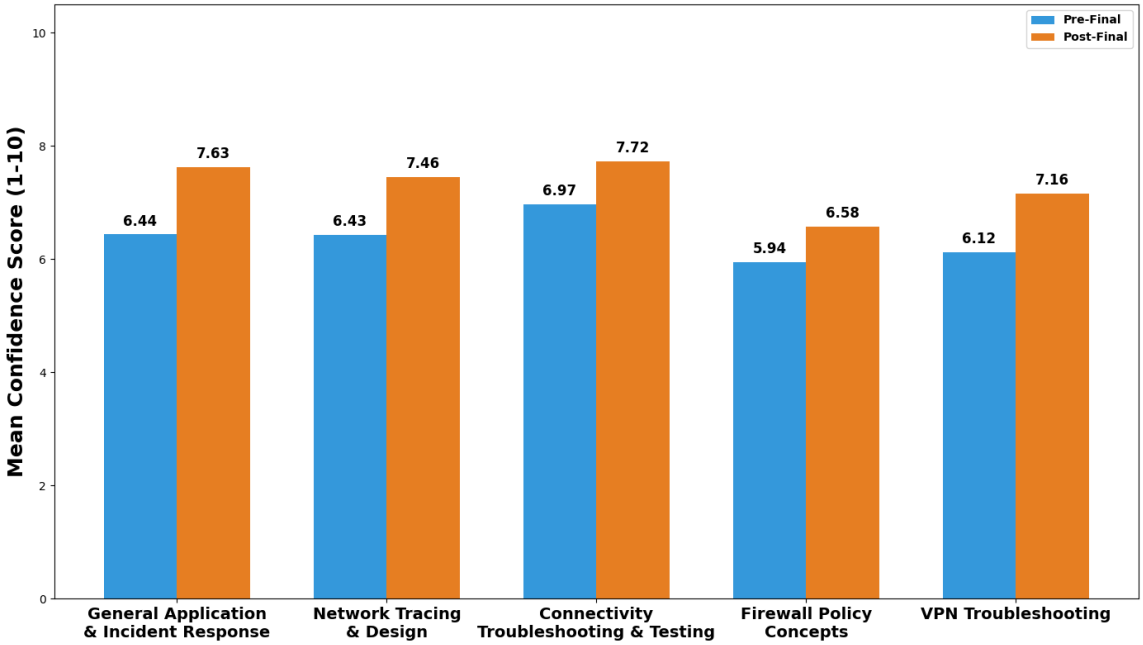


Figure 5. Pre vs. post means by category (1–10 scale)

Figure 5 compares the average self-reported knowledge scores for each category after the simulation (pre) versus after the hands-on Pod-Pi lab (post). Each bar represents the mean category score across all respondents, where higher values indicate stronger perceived ability in that skill area.

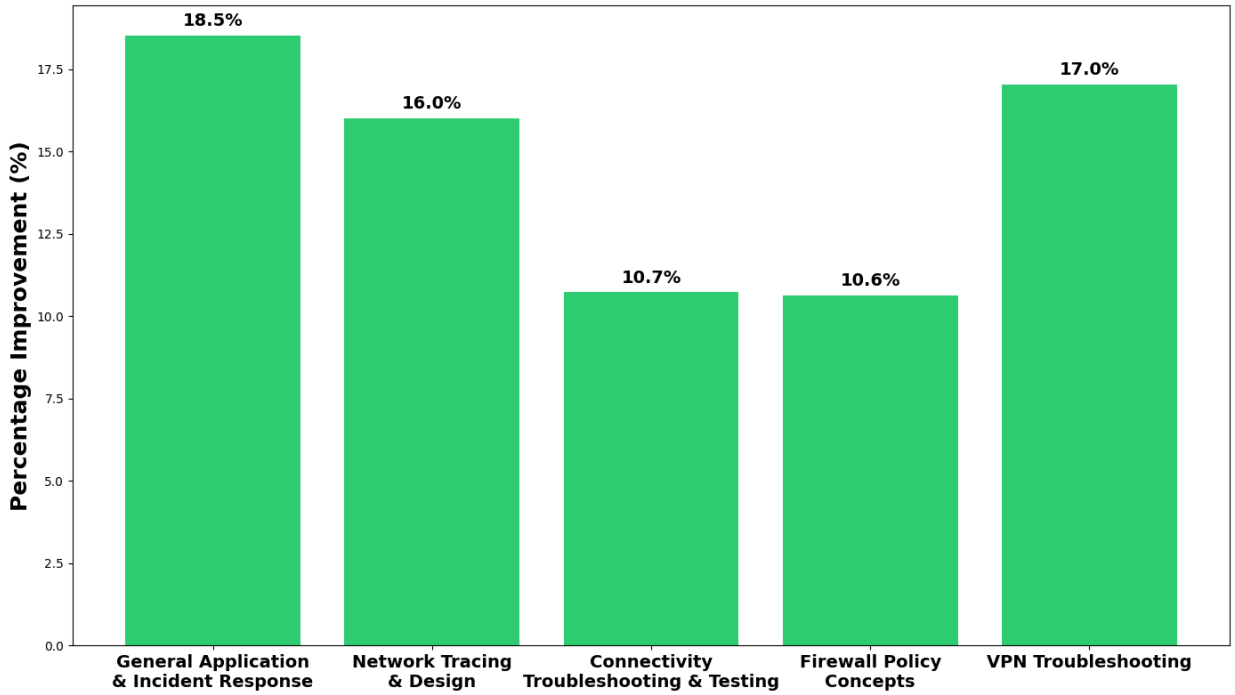


Figure 6. Improvement per category (post minus pre).

Figure 6 illustrates the mean score increase for each category from pre- to post-survey. Positive values represent the percentage of improvement in student confidence following the hands-on lab relative to the initial simulation stage.

We observed consistent post-survey increases across all five categories. The most significant gains occurred in General Application and Incident Response (18.5%), followed closely by VPN Troubleshooting (17.0%) and Network Tracing and Design (16.0%). These gains suggest that physical interaction with Cisco hardware helps students bridge the gap between theoretical configuration and real-world application. In the VPN category specifically, the hands-on environment forces students to move beyond "status up" indicators and perform concrete traffic verification, a distinction that is often lost in purely simulated environments.

The smallest gains were observed in Connectivity Troubleshooting and Testing (10.7%) and Firewall Policy Concepts (10.6%). The lower rate of improvement in Connectivity may stem from its reliance on fundamental concepts introduced earlier in the semester; under the time pressure of a final lab, students may prioritize complex new configurations over basic diagnostic steps. Similarly, the modest gains in Firewall Policy Concepts suggest that while students can follow logical maps, the transition to modifying live Zone-Based Firewall (ZBF) policies remains a steep learning curve.

These results provide a clear roadmap for iterating on the lab design. Future versions should focus on decomposing ZBF configuration into smaller, modular tasks and incorporating brief "diagnostic checkpoints" for fundamental connectivity earlier in the exercise. By reinforcing these core competencies before advancing to complex security policies, we expect to see more balanced learning gains across all domains without increasing the overall complexity of the lab.

During the hardware lab, students transitioned from individual task execution to high-intensity collaborative workflows. Several key behavioral themes emerged:

- Hypothesis-Driven Troubleshooting: Teams were observed discussing hypotheses and comparing verification outputs of their corresponding devices to ensure working in tandem and towards a common goal, rather than just typing whatever troubleshooting commands were given to them, they were in active conversation with each other to determine the next steps as well as tracing back earlier mistakes.

- Cooperative Verification: The collaborative environment was cited as a key factor in student confidence, as peers validated each other's CLI inputs to ensure the whole system reached convergence.

- Physical-to-Logical Mapping**:** Teams were observed tracing physical cables back to the switch to verify they were in the correct port. This "Layer 1" awareness, absent in software simulations, bridged the gap between abstract theory and physical reality.

- Resilience to Real-World Latency: Students learned to account for hardware convergence times (i.e. Real-world systems usually take a few seconds to minutes to update their state). This forced them to use verification commands (e.g., show ip interface brief) to monitor state changes rather than assuming immediate connectivity.

5. Discussion

The primary advantage of the Pod-Pi hardware implementation is the introduction of authentic constraints that are typically overlooked in virtual environments for the sake of simplicity. These include real-world interface behavior, link state changes, and the critical nature of configuration persistence. Unlike simulations, where success is often a binary "green light" physical hardware forces students to validate security policies and VPN functionality using live traffic and concrete verification outputs. Even with the assistance of instructional "cheat sheets," teams were required to reason through complex cause-and-effect relationships and coordinate changes across multiple physical devices.

The significant gains observed in General Application and Incident Response (18.5%) and VPN Troubleshooting (17%) skills likely stem from the mental shift needed for physical networking. In theory, a student may understand a VPN tunnel, but the hardware environment introduces the critical nuance that a "tunnel up" status does not guarantee traffic passing.

This pattern aligns with experiential learning theory, where the physical interaction with Cisco equipment forces students to reconcile their internal mental models with visible, often unpredictable, hardware flags. By forcing students to deal with “looks right but doesn’t work” the Pod-Pi environment bridges the gap between role memorization and the internal problem-solving required in production environments.

The Pod-Pi framework directly addresses national trends in IT and cybersecurity education that emphasize "work-ready" skills over purely theoretical knowledge. Our transition from

simulation-heavy instruction to a collaborative, role-based hardware model reflects the industry's shift toward team-centric operations, where industry professionals often work in conjunction with each other, rather than in isolated environments where their actions have little consequences on other people's work.

Study Limitations and Future Directions

While the results are promising, several limitations must be acknowledged to contextualize the findings:

- Self-Reported Measures: The evaluation relies heavily on anonymous surveys, which capture perceived confidence and ability rather than purely objective knowledge. Future work will implement rubric-based performance checks to provide a more standardized measure of skill acquisition, as well as a more robust simulation lab.

- Sample Size and Matching: With a sample of 48 students, individual-level comparisons were limited because pre- and post-surveys were not matched one-to-one because of the anonymous nature of the surveys.

- Device Variability: The use of legacy Cisco equipment introduces hardware-specific quirks that can lead to inconsistent student experiences. While we use reset automation to mitigate this, some "ghost states" or feature limitations remain unavoidable, the lab was designed to circumvent said limitations, but this also limited the complexity at which the lab is administered.

- Time Constraints: Smaller gains in general application and basic troubleshooting suggest that the high cognitive load of late-semester exams may overshadow fundamental concepts learned in earlier parts of the semester.

To address these, future iterations will collect objective, time-stamped data during the labs to analyze exactly how long teams spend in failure states before recovering. While collaboration was encouraged and fundamental concepts' understanding was required from all individual students, role specialization still persisted to some degree, and not all students touched all concepts of the exam, so we also plan to implement role rotation to ensure that students do not become overly specialized in one area at the expense of a holistic understanding of the network.

6. Conclusion

The Pod-Pi framework demonstrates that a realistic networking and security lab does not require prohibitively expensive enterprise-level equipment or reliance on restrictive campus infrastructures. By bridging the gap between isolated software simulations and high-stakes physical hardware, Pod-Pi forces students to navigate the messy reality of Layer 1 constraints and real-time device connectivity. This dual-stage approach offers a replicable blueprint for modernizing applied technical education.

To successfully implement or scale a Pod-Pi model, we offer the following strategies for academic departments:

- Scaling: Because the pods are self-contained and VNC-accessible, institutions can scale their lab capacity incrementally. Departments can deploy mobile pods that do not require dedicated lab space, not to mention that the Pod-Pi comes equipped with an uninterrupted power supply (UPS) allowing further flexibility to do security training in general-purpose classrooms, without worrying about wiring/powering devices.

- Curriculum-Wide Integration: Pod-Pi should not be treated as an isolated exercise. We recommend integrating these pods across the curriculum, starting with basic VLAN routing in introductory courses and graduating to complex ZBF and IPsec configurations in senior networking/cybersecurity courses.

The findings from this study suggest a necessary shift in how we prepare the next generation of the cyber workforce. Moving beyond traditional testing and learning methods, the Pod-Pi model emphasizes a robust pedagogical approach:

- Policy and Practice: Our results argue for the standardization of live hardware assessments and practice in IT education. Accreditation bodies and curriculum-designing individuals should look toward performance-based testing methods as a more accurate metric of workforce readiness than traditional multiple-choice exams.

- Pedagogical Research: The planned collection of time-stamped diagnostic data represents a significant leap forward for educational research. By analyzing the specific order of student commands and the duration of "failure states," researchers can identify exactly where conceptual bottlenecks occur, allowing for the design of more targeted instructional interventions.

- Workforce Readiness: By simulating the role-based pressures of a Network Operations Center (NOC) and a Security Operations Center (SOC), Pod-Pi prepares students for the collaborative nature of industry. Future work into role rotation will further ensure that this collaboration does not lead to over-specialization, creating well-rounded engineers capable of troubleshooting the entire network stack.

In summary, Pod-Pi serves as a stress test for both the student and the curriculum. It transforms the lab from a guided walkthrough into a dynamic environment where evidence, iteration, and teamwork are the only paths to success.

Acknowledgment

This work is in part supported by the MoExcels FY2025 Award “MU Engineering - CEEIT.”

References

[1] T.-H. Kim, R. Calix, and D. Patel, "Low Cost System for Laboratory-based Course in IT Education using Raspberry Pi," in Proc. 2019 ASEE Annu. Conf. & Expos., Tampa, FL, USA, Jun. 2019, p. 33073. doi: 10.18260/1-2--33073.

[2] F. Khan, M. K. Quweider, A. Qubbaj, E. Tomai, L. Zhang, and H. Lei, “Infusing Raspberry Pi in the Computer Science Curriculum for Enhanced Learning,” in Proc. 2020 ASEE Virtual Annu. Conf. Content Access, Virtual, Jun. 2020, p. 34828, doi: 10.18260/1-2--34828.

[3] L. Ma, J. Bartholomew, Y. Wang, and X. Li, "Development of a Raspberry PI-Controlled VEX Robot for a Robotics Technology Course," in Proc. 2023 ASEE Annu. Conf. & Expos., Baltimore, MD, USA, Jun. 2023, p. 43130. doi: 10.18260/1-2--43130.

[4] J. M. D. Hill, C. A. Carver, J. W. Humphries, and U. W. Pooch, "Using an isolated network laboratory to teach advanced networks and security," in Proc. 32nd SIGCSE Tech. Symp. Comput. Sci. Educ., Charlotte, NC, USA, Feb. 2001, pp. 36–40. doi: 10.1145/364447.364533.

[5] R. T. Abler, D. Contis, J. B. Grizzard, and H. L. Owen, "Georgia Tech Information Security Center Hands-On Network Security Laboratory," IEEE Trans. Educ., vol. 49, no. 1, pp. 82–87, Feb. 2006. doi: 10.1109/TE.2005.858403.