International Journal of Innovative Teaching and Learning in Higher Education
Volume 5 • Issue 1

Hardware-Free Network Internals Exploration:

A Simulation-Based Approach for Online Computer Networking Course

Qian Liu, Rhode Island College, USA*
https://orcid.org/0000-0002-3314-8760

ABSTRACT

In undergraduate computer networking courses, the ideal scenario involves demonstrating network communications with multiple interconnected computers and a packet sniffer tool. However, practical challenges arise when attempting hands-on exercises, such as accessing or reconfiguring physical computers for online networking practice. Additionally, certain network concepts, like routing and switching, are typically discussed theoretically due to the limitations of observing external network packet transfers and the constraints faced by institutions in maintaining the necessary hardware for hands-on practice. This paper introduces a simulation-based approach to facilitate the teaching and learning of computer networking internals in an online environment, eliminating the need for dedicated hardware devices. The paper outlines various simulation activities and experiments designed to assist instructors in teaching and enable students to explore these internal networking concepts effectively.

KEYWORDS

Computer Networking, Network Internals, Online Environment, Routing and Switching, Simulation

INTRODUCTION

In computer networking courses at undergraduate level, the traditional teaching and learning method involves connecting several physical computers to form a small Local Area Network (LAN). Packet sniffer tools, such as Wireshark software (Wireshark Foundation, 2024), are then installed on those computers to capture networking packets within the small LAN, helping students comprehend networking abstractions in a visualized way. However, the internal processes of computer networking, such as how networking packets are forwarded within internal network locations and routed over the Internet, are usually discussed theoretically with little or no hands-on practice due to several reasons, including students cannot directly observe how networking packets are transferred across the internet due to security concerns and infrastructure limitations, and maintaining a dedicated LAN can be

DOI: 10.4018/IJITLHE.339002
*Corresponding Author

costly and resource-intensive for schools, making it challenging for some institutions to implement this traditional teaching and learning method.

As online teaching and learning has surged in recent years and has become the lifeline of education during the COronaVIrus Disease of 2019 (COVID-19) pandemic, the traditional methods of in-person instruction have faced significant hurdles. Even with the reopening of higher education institutions in many regions after the pandemic, a substantial number of courses continue to be offered online or in hybrid (half online half in-person) formats. In the authors' state, public universities and colleges offer many courses in these two modalities. This shift has presented unique challenges for teaching subjects like computer networking, which traditionally relies on hands-on labs and practical exercises involving physical network setups, but now becomes impractical in the online environment. In a traditional classroom setting, students can connect their computers to a dedicated LAN and engage in activities like packet sniffing, device configuration, and troubleshooting network issues firsthand. This immersive experience allows them to visualize and understand abstract networking concepts in a tangible way. However, replicating such a setup in the online environment is cumbersome, and rewiring devices remotely, let alone practicing those networking internals discussed above, is impossible. These limitations of online teaching and learning environments pose significant challenges for instructors teaching computer networking courses. Traditional labs and exercises must be redesigned or replaced with alternative approaches that can effectively simulate real-world network scenarios and provide students with the necessary hands-on experience. This necessitates the development of innovative online experiments and interactive simulations that can bridge the gap between theoretical knowledge and practical application.

This paper addresses this challenge by proposing a simulation-based approach for teaching and learning those network internals, such as routing and switching, Address Resolution Protocol (ARP) internals, and packet metadata alteration during transfer. We begin by conducting a comparative analysis of several popular simulators and identifying the one that demonstrably excels in simplifying and visually illustrating those network internals. This chosen simulator then serves as the foundation for our proposed approach, which is specifically designed to facilitate effective teaching and learning of network internals within the online environment. By seamlessly merging theoretical knowledge with hands-on experiments, this simulation-based approach empowers students to grasp network internals in online environment without requiring or accessing dedicated hardware devices, paving the way for a more accessible and flexible approach to computer networking education.

LEARNING OBJECTIVES AND SIMULATOR COMPARISON

Table 1 outlines the specific learning objectives related to the network internals our approach targets. In our undergraduate-level computer networking course, students learn and practice the layered architecture and protocols of the Internet model, gaining an understanding of network fundamentals, the Internet protocol suite, routing and forwarding principles, and the socket Application Programming Interface (API). Access to dedicated hardware devices for teaching and learning those network internals poses a significant challenge in both traditional LAN-based and online environments. Network simulators address these limitations by providing a versatile and accessible platform for visualizing these internal processes and enriching student learning.

We evaluated several free network simulators to identify the most suitable one for our simulation-based approach in online environments. Our focus lies on simulating functionalities within the link and network layers, addressing crucial network internals often overlooked for hands-on practice in traditional in-person courses. Students in traditional courses often take for granted the seamless operation of these "lower layer" internals, assuming they simply "work perfectly" all the time. A profound understanding of these foundational internals enhances comprehension of the entire network communication architecture, while also equipping students with essential skills for tackling

Table 1. Learning objectives related to the internal processes of computer networking

Network Internals	Learning Objectives
Address Resolution Protocol (ARP) internals	1. Understand ARP request/reply process. 2. Understand how Internet Protocol (IP) address translates to Medica Access Control (MAC) address. 3. Understand ARP cache. 4. Understand the information exchange for MAC address discovery.
Routing and switching	1. Learn how to interpret routing tables. 2. Create routing tables and forwarding tables. 3. Identify common routing/switching problems and resolve them.
Packet metadata alteration during transfer	1. Understand the structure and purpose of packet headers. 2. Trace and understand the modifications made to headers during transfer. 3. Evaluate how changes in headers reflects specific network behavior involved.

advanced concepts. This is evident in the use of similar addressing and forwarding mechanisms in high-performance networks like InfiniBand (IBTA, 2015).

The Graphical Network Simulator-3 (SolarWinds Worldwide, 2024) is one of the simulators that supports routing and switching configurations, requiring manual configuration of routing tables to activate connectivity among different subnets. It also captures networking traffic in its virtual network, enabling observation of packet metadata alterations during data transfer across subnets. In the real world, however, capturing packets in external networks is generally not possible, preventing similar practices in small LAN environments. Despite these advantages, GNS3 requires running actual router commands to construct routing tables, which might exceed course requirements and challenge student comprehension in online settings.

The Packet Tracer (Cisco System, 2023) is a network simulator designed by Cisco for its certificate training courses, like the Cisco Certified Network Associate (CCNA) certificate. Like GNS3, users can drag-and-drop elements like computer hosts, networking switches, and routers to build and configure a network in the online environment. The key difference between Packet Tracer and GNS3 lies in its automatic animation of networking traffic and detailed packet recording. While GNS3 also offers traffic capture, beginners may find it challenging to determine the chosen path. As network complexity grows, tracing expected packet flows through multiple potential paths in GNS3 can become unwieldy.

The Network Simulator-3 (NS-3) simulator (NS-3, 2023) is an open-source C++ software primarily used on Linux, requiring users to code using NS-3 proprietary interfaces to create simulation environments. While it lacks virtual router and switch entities, users can configure virtual routing tables for simulations involving multiple subnets. This can be done either automatically using the built-in method *Ipv4GlobalRoutingHelper::PopulateRoutingTables* or manually by adding routes to a static routing table using the *AddHostRouteTo* method. The latter approach, like configuring actual routing tables, requires specifying destination Internet Protocol (IP) prefixes and next-hop IP addresses, providing valuable hands-on routing practice for students. However, the NS-3's reliance on coding with proprietary interfaces necessitates a thorough understanding of its internals, potentially creating a steep learning curve, and burdening both instructors and students in online environments.

The Objective Modular Network Testbed in C++ (OMNeT++) simulator (OMNeT++, 2023), like the NS-3 simulator, requires users to follow and code using its proprietary interfaces to build simulations, but it also offers a graphical user interface, akin to the Packet Tracer and GNS3, for drag-and-drop network construction. It can animate traffic flow, enable manual routing table configuration, and facilitate troubleshooting of connectivity issues, similar to Packet Tracer. However, exploring network switch functionalities in the OMNeT++ poses challenges because: 1) network switches are generally self-learning entities so that the OMNeT++ does not make it open for configuration, and 2) it needs modifications to the simulator itself if one needs to observe switch internals. These factors, as with the NS-3, can burden students when practicing switching internals in online environments.

In the Optimum Network Performance (OPNET) Modeler (Riverbed Technology, 2022), there are two avenues for practicing routing internals, although it lacks control or observation of switching functionalities. First, akin to the NS-3, OPNET automatically constructs routing tables by default. Alternatively, users can configure static routes manually to establish communication between subnets. However, OPNET's limitations in animating traffic flows and providing diagnostic information in cases of routing configuration errors can hinder students' ability to effectively troubleshoot connectivity issues, especially in online learning environments.

Table 2 summarizes the compared network simulators, each capable of achieving the previously discussed learning objectives focused on the "lower networking layers". In our simulation-based approach, we leverage the Packet Tracer to provide students with extensive hands-on practice in this domain. Our choice is driven by several factors, including:

- Packet Tracer's intuitive drag-and-drop interface makes it beginner-friendly for online learning environments.
- Its comprehensive library of networking elements enables realistic network construction and exploration. Students can gain hands-on experience with device configuration without risking real-world network disruptions.
- Its built-in animation and traffic visualization capabilities facilitate real-time observation and troubleshooting.
- Its automatic traffic recording feature allows students to observe how packets are transferred among different subnets and learn how (and why) packet metadata are altered during transfer. In the real world however, it is not possible to observe how data are transferred in external networks.
- Students can engage in an interactive learning environment where they possess comprehensive control over network elements like computer hosts, switches, and routers. This allows them to actively configure network parameters, analyze simulated outcomes, and develop critical troubleshooting skills by identifying and resolving virtual network issues. Further expanding learning opportunities, students can introduce controlled, realistic network disruptions such as cable faults or port malfunctions, honing their diagnostic and problem-solving abilities within a safe online environment.
- Students can take an active role in their learning in the online environment, setting their own pace and navigating the simulator's features to acquire knowledge and develop skills specific to their needs, driving personalized learning outcomes.

SIMULATION-BASED APPROACH

Traditional networking courses at the undergraduate level often confine instruction on critical layer 2 (link layer) and layer 3 (network layer) protocols to theoretical explorations. This knowledge gap between theoretical understanding and practical application can hinder students' grasp of fundamental networking concepts. To bridge this gap, we present a novel educational framework utilizing the Packet Tracer simulator. This framework leverages a collaborative learning approach to empower instructors to facilitate active learning environments while enabling students to engage in immersive, step-by-step experimentation with the internal processes of computer networking.

Our framework specifically targets topics such as the Address Resolution Protocol (ARP) internals, switch self-learning and forwarding mechanisms, and routing table construction. These topics, often discussed in theoretical obscurity within traditional lectures, come alive through our interactive platform. Students actively experiment with virtual network configurations, observing firsthand the

Table 2. Free network simulator comparison

Simulator	Support ARP Simulation	Support Routing and Switching	Visualize Packet Metadata Alteration	Animate Traffic
Graphical Network Simulator-3 (GNS3)	Yes	Yes	Yes, needs manual record	Yes
Packet Tracer	Yes	Yes	Yes	Yes
Network Simulator-3 (NS-3)	Yes, needs coding	Yes	Yes, needs coding	No
Objective Modular Network Testbed in C++ (OMNeT++)	Yes	Yes	Yes, needs coding	Yes
Optimum Network Performance (OPNET)	No	Yes	No	No

dynamic relationships between protocols and observing the intricacies of packet forwarding and address resolution. This experiential learning fosters a deeper understanding of the "why" behind the "what", solidifying foundational knowledge and fostering confidence in practical application.

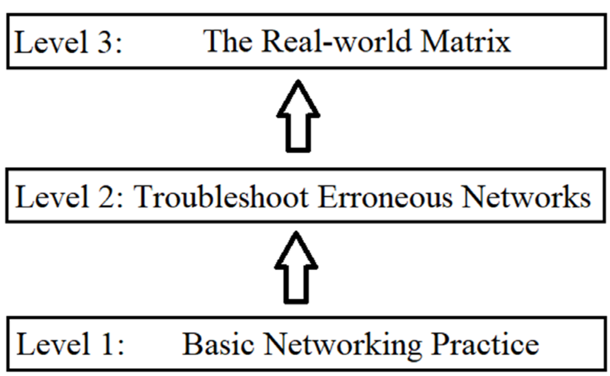
The framework is structured across three distinct levels, each meticulously designed to provide incremental complexity and challenge. As depicted in Figure 1, level 1 lays the groundwork with the fundamentals, level 2 introduces realistic device disruptions and troubleshooting scenarios, and level 3 culminates in open-ended exploration, encouraging students to creatively apply their acquired knowledge to design and manage networks and devices. This progressive learning journey ensures that students master basic concepts before delving into more intricate mechanisms, ultimately cultivating a comprehensive understanding of layer 2 and layer 3 operations.

This progressive learning structure assigns students evolving roles that mirror their growing expertise. As depicted in Figure 1, level 1, "Basic Networking Practice", casts them as "Network Observers". Students monitor simulated network traffic, observe protocol interactions, and gain foundational knowledge through guided experiments.

Upon mastering the fundamentals, students ascend to level 2, "Troubleshoot Erroneous Networks", where they transform into "Problem Solvers". Here, they encounter deliberately injected network malfunctions, such as routing loops or ARP poisoning. Using their honed skills, they diagnose these issues, implement corrective actions, and restore network stability. This level fosters critical thinking and problem-solving skills, preparing students for real-world challenges.

Finally, level 3, "Real-world Matrix", elevates students to "Creators of Networks", encouraging students to creatively apply their acquired knowledge to design complex network scenarios and manage virtual networks. This culminating stage empowers students to apply their acquired knowledge creatively, fostering confidence and competence in the face of diverse network challenges.

Figure 1. The progressive learning framework introduced in our approach



By progressively assigning these roles with increasing complexity, our framework provides a unique learning journey that mirrors the evolution of a network engineer's career. Students actively engage in the various facets of network management, solidifying their understanding and preparing them for real-world network administration and design.

BASIC NETWORKING PRACTICE

This initial stage immerses students in the intuitive graphical interface of the Packet Tracer, fostering their confidence in a virtual network environment. Drag-and-drop functionality simplifies network construction, making it an ideal springboard for mastering fundamental subnet management skills. We start with a simple network shown in Figure 2, a straightforward LAN topology serves as the platform for practicing subnet management. Students take ownership by assigning appropriate IP addresses to each host (shown as "PC-PT" in Figure 2). This hands-on learning allows them to grasp the practical implications of addressing schemes.

Next, students choose two hosts and initiate a ping test, simulating real-world communication. This triggers the generation of an ARP request packet (Figure 3), offering a clear visualization of the Address Resolution Protocol in action. Students can dissect the packet and witness the dynamic interaction between Internet Protocol (IP) address and Media Access Control (MAC) address.

Then, the learning journey continues as switches within the network actively learn the MAC addresses of connected devices, as shown in Figure 4, which depicts the forwarding table of Switch-1, which was shown in Figure 2. This provides students with valuable insights into switch self-learning and its impact on packet forwarding. Witnessing this process demystifies theoretical explanations by offering a tangible visual representation.

Next, students can observe and follow how switch flooding occurs within the simulation. By observing animated traffic flow throughout these stages, students gain a comprehensive understanding of switch operation and packet forwarding mechanisms. This interactive exploration transforms abstract concepts into practical knowledge, effectively bridging the gap between theoretical discussions and real-world application.

Figure 2. A simple local area network topology used in our simulation

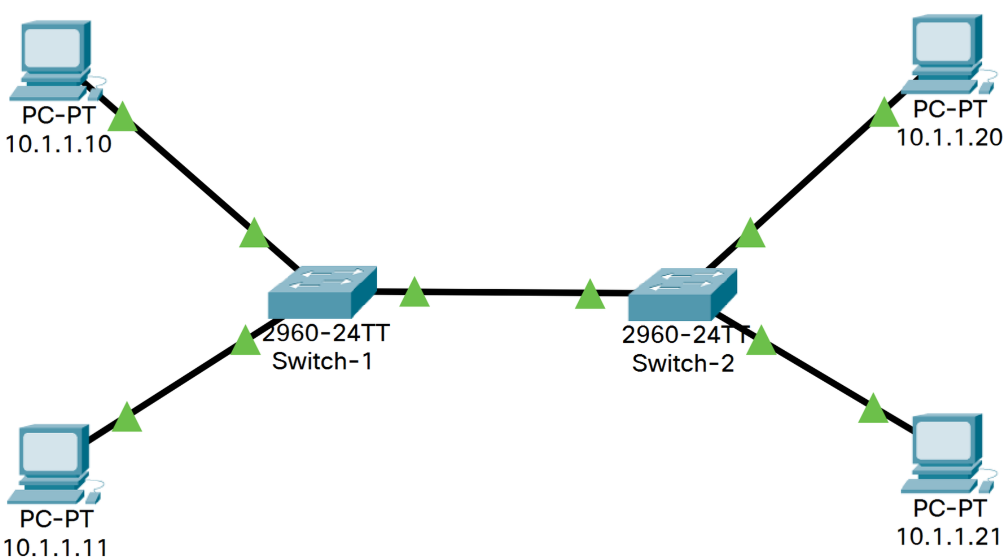


Figure 3. An address resolution protocol (ARP) request packet sent in our simulation

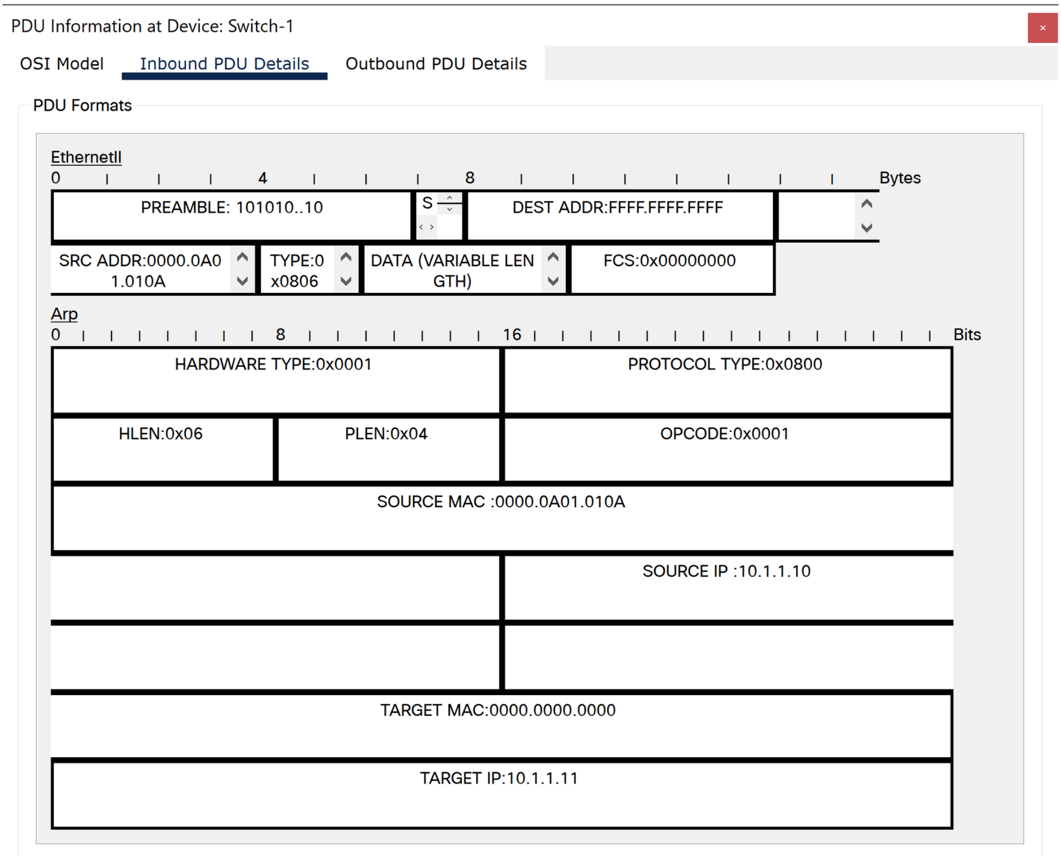


Figure 4. An example of a switch forwarding table shown in our simulation

```
Switch#show mac-address-table
          Mac Address Table
-------------------------------------------

Vlan    Mac Address       Type        Ports
----    -----------       --------    -----

   1    0000.0a01.010a    DYNAMIC     Fa0/1
   1    0000.0a01.010b    DYNAMIC     Fa0/2
   1    0000.0a01.010c    DYNAMIC     Fa0/3
   1    0000.0a01.010d    DYNAMIC     Fa0/3
```

Following their initial foray into subnet management, students progress to Level 1's next stage: navigating the inter-subnet routing. This module immerses them in the world of routers, tasked with manually configuring routing tables to connect different subnets and enable communication across networks. This hands-on exercise provides experience in establishing communication pathways among devices attached to each other. While configuring routing tables in a live network carries the risk of disrupting ongoing communication, the Packet Tracer presents a safe and controlled environment for experimentation. Unlike their real-world counterparts, students in the simulator enjoy the advantage of complete visibility. They can witness the entire data transfer process, from initial packet creation to delivery, and observe critical metadata changes in real-time. This level of transparency empowers them to readily identify and rectify configuration errors, pinpointing the root cause of any communication failures through the Packet Tracer's traffic animation.

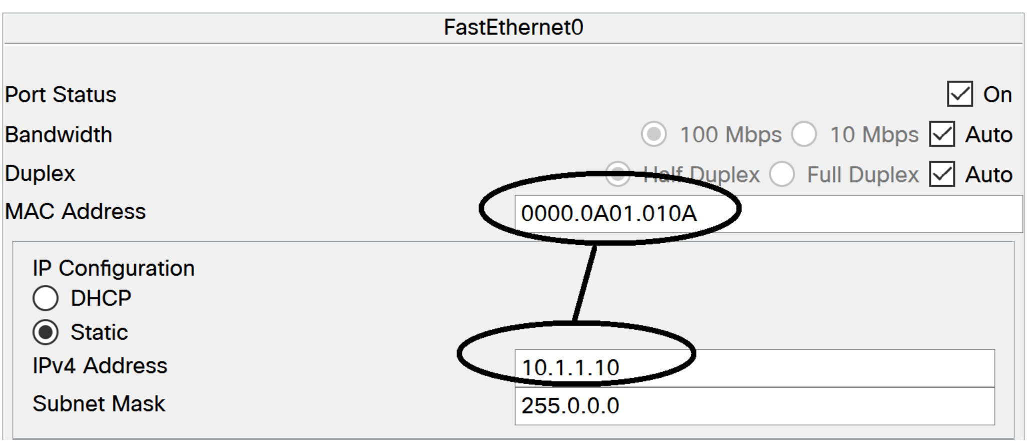
In the Packet Tracer and other simulators, a random MAC address is automatically generated and assigned to hosts and routers, but it is not convenient to examine the MAC/IP mappings nor follow the changes in packet metadata as it is difficult to associate an IP with a MAC. To overcome this obstacle, we have implemented the "*softMAC* conversion" mechanism to assign a "recognizable" MAC address, enabling users to readily link IP addresses with their corresponding MACs and streamline data flow analysis. The "*softMAC* conversion" simply prepends a 16-bit sequence of zeros to a given IPv4 address, thus generating a recognizable MAC address. For example, a host with an IP address *10.1.1.10* will have a MAC address *00-00-0A-01-01-0A* assigned. Figure 5 shows the host configuration window in the Packet Tracer when an IP address and a "*softMAC*" converted MAC address are assigned to a host.

The level 1 experiments presented above serve as a practical and adaptable template for institutions implementing our framework. Instructors can readily modify these exercises to align with their specific curriculum and student needs. Designed to foster familiarity with the Packet Tracer, these activities unveil the often-hidden inner workings of network communication. Through this immersive introductory stage, students develop confidence navigating the intricate world of network internals. This foundational understanding then paves the way for deeper exploration and mastery of progressively complex network operations in subsequent levels.

Troubleshoot Erroneous Networks

While network simulators like the Packet Tracer offer valuable controlled environments for learning, a potential limitation often cited (Ristov, et. al., 2015) is their tendency to present idealized scenarios where everything functions flawlessly. This, however, rarely reflects the realities of real-world networks, where unexpected errors and malfunctions can disrupt operations. Recognizing this crucial gap, level 2 of our framework intentionally introduces controlled network malfunctions within the Packet Tracer. These injected errors, meticulously mimicking real-world scenarios, provide students

Figure 5. Setting host medica access control (MAC) address in packet tracer by using softMAC mechanism



with opportunities to develop essential troubleshooting skills. By detecting and resolving these simulated malfunctions, students gain practical experience in:

- Problem identification: Analyzing network behavior, traffic flow, and error messages to pinpoint the root cause of the malfunction.
- Decision-making: Evaluating potential solutions and choosing the most appropriate course of action based on the specific context.
- Configuration correction: Applying their understanding of routing protocols, network protocols, and device configurations to rectify the issue and restore proper network operation.

Figure 6 illustrates an example of such a learning activity. Four subnets are connected to four different routers, but communication among them is deliberately disrupted due to strategically placed errors within the routing tables. Through this exercise, students can engage in the troubleshooting process: observing the problematic traffic, analyzing the root cause, and fixing the configurations. By incorporating such simulated malfunctions into the learning process, level 2 empowers students to transition from passive theoretical understanding to active problem-solving, building the confidence and skills needed to navigate the complexities of real-world networks.

To introduce link layer errors, students can deactivate a few switch ports in a well-functioned network (depicted in Figure 7) to simulate port failures in the real world. Students can then witness the procedure of the Spanning Tree Protocol (STP), track switch forwarding table updates, and explain changes in subsequent data transfer. In level 2, students change from “Observers” to “Problem Solvers” through these engaging activities, they can identify, diagnose, and troubleshoot networking errors that do not occur in an “ideal” simulation environment.

The Real-World Matrix

Building upon the foundational skills acquired in the previous levels, level 3 invites students to dive deeper into the complexities of network operations through scenario-based learning. As a springboard for this exploration, we introduce the ARP cache simulation, a critical element that significantly impacts network communication in diverse ways. First, students need to list several what-if scenarios (e.g., what if a switch ages out its forwarding table; what if a host clears its ARP cache). Then they need to combine those scenarios, systematically analyze these interconnected possibilities, and explore how they interact with each other and how the interaction affects subsequent communication, thus creating a What-if Matrix. The matrix serves as a visual roadmap, outlining various combinations

Figure 6. An example to deal with routing loops

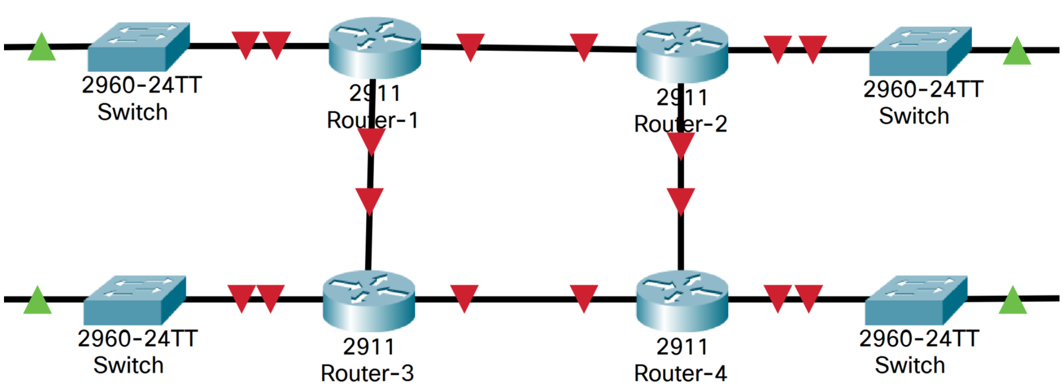
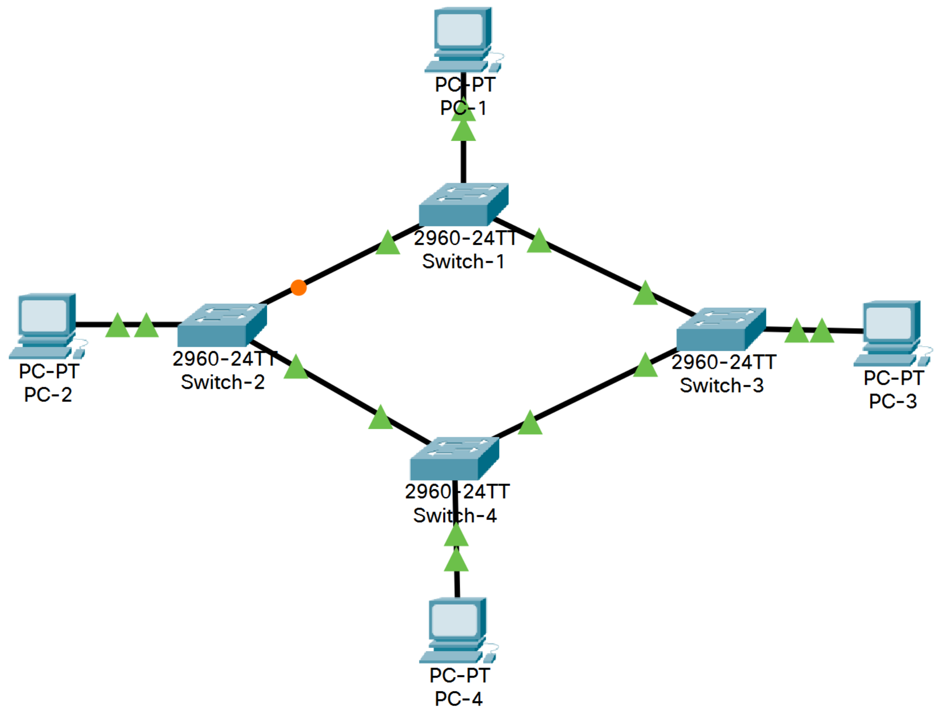


Figure 7. An example to simulate switch port failure



of potential events and their subsequent impact on network communication. Table 3 showcases an example of such a matrix, with rows representing different switch forwarding table scenarios and columns indicating host ARP cache situations.

A more complicated What-if Matrix can be built based on the network topology shown in Figure 6. This more intricate exercise goes beyond individual ARP cache simulations, delving into the dynamic interplay between router ARP caches, router ARP messages, and those exchanged between routers and hosts. Because students need to adjust many device settings at the right time and initiate packet transmission accurately, these hands-on practices challenge them to think critically and cultivate their understanding of network dynamics while solidifying their grasp of those network internals.

LEARNING RESULT DISCUSSION

Our computer networking course, geared towards computer science juniors, has been a mainstay in our curriculum for several semesters and taught by the same instructor. Through its evolution across

Table 3. An example of the what-if matrix (one host pings another in following 4 scenarios)

	Host Caches All MACs	**Host ARP Cache Has Aged Out**
Switch has learned all hosts' MAC	Predict network traffic behavior	Will (and how) ARP cache affect network communication?
Switch has aged out MAC tables	Predict switch self-learning and flooding procedure	Explain the difference between broadcasting and flooding

three distinct teaching and learning models, the course has adapted to changing circumstances. Prior to the COVID-19 pandemic, the course adhered to a traditional in-person format, which emphasized theoretical discussions of network internals like routing and switching. However, with the onset of the pandemic in Spring 2020, we transitioned to a hybrid half-in-person, half-online model, followed by a fully online model (offered twice during and after the pandemic), necessitating a shift in instructional methods. In response, we developed and implemented this proposed innovative simulation-based approach since Spring 2020, allowing students to actively engage with these network internals, complementing and enhancing their theoretical understanding.

We analyzed course assessment data from three distinct teaching and learning periods: traditional in-person, half-in-person/half-online, and fully online. This data included enrollment rate (ratio of enrolled students to class capacity), retention rate (course drop rate), and student overall performance (rate of grades A and A-). As shown in Table 4, the course maintained a consistently high enrollment rate across all periods, confirming its popularity among computer science students. Notably, even during the challenging Spring 2020 semester (half in-person, half online), when the sudden shift to online learning in the middle of the semester impacted many courses, students were confined at homes and many of them faced the online environment for the first time, they still demonstrated resilience and achieved good grades, though it was not as good as the previous semester. After that when students were getting used to the online environment, they showed significant progress in understanding those network internals. These findings suggested that our learning framework effectively supported student success in online environments.

To evaluate the effectiveness of our learning framework in addressing student weaknesses, we conducted pre- and post-framework assessments focusing on common error patterns in those network internals, shown in Table 5. These assessments, administered immediately following theoretical discussions and again after the completion of framework-based simulation practices, revealed significant improvements across all identified weak topics. For instance, the theoretically discussion alone failed to solidify students' understanding of how to assign an appropriate IP address given a subnet prefix, but the simulation environment offered a transformative experience, students were able to observe traffic errors resulting from an incorrect IP assignment. This was reflected in the striking reduction of errors from 23% in pre-simulation assessments to a mere 2%

Table 4. Course assessment data in different periods

Course Mode	Enrollment Rate	Drop Rate	Student Performance (Rate of A, A-)
Traditional in-person	100.0%	6.2%	73.9%
Half in-person, half online	91.6%	9.0%	70.0%
Fully online	95.8%	8.6%	80.5%

Table 5. Weak topics comparison before and after framework learning in online environment

Common Errors/Weak Topics	Before Simulation	After Simulation
Incorrect IP assignment by a given IP prefix	23%	2%
IP/MAC address confusion	9%	0%
Incorrect static routes	35%	16%
Switch self-learning	19%	9%
Switch flooding	22%	7%
ARP-IP-MAC interactions	26%	14%

in post-simulation assessments. The data in Table 5 demonstrated that the framework successfully fostered structured knowledge acquisition and enhanced skill mastery, solidifying its value in supporting student success.

Table 6 provides a detailed breakdown of student time allocation at each level within the online learning environment. Our computer networking course dedicated 6 weeks to the network and link layers (we used another 7 weeks covering the application layer, transport layer, and socket programming). At level 1, all students demonstrated proficiency by successfully completing the assigned experiments. Level 2 presented a slight hurdle for some students who encountered difficulties with debugging and root cause analysis. However, the majority successfully navigated the troubleshooting tests. Finally, level 3 witnessed a noteworthy achievement: 83.33% of student pairs effectively designed and implemented the matrix, showcasing their understanding of complex network and device internals. While the remaining groups accurately designed the matrix tables, they faced challenges in configuring scenarios to trigger specific behaviors.

Table 7 details the assessment methods and results for the learning objectives listed in Table 2. Our diverse assessment methods encompassed a range of activities, including quizzes for knowledge retention, engaging discussions for fostering critical thinking, hands-on labs for practical application, and collaborative paired projects for honing teamwork and communication skills. The effectiveness of our learning framework was demonstrably evident in the student performance data, indicating a deeper understanding of those network internals and the successful achievement of targeted learning objectives.

RELATED WORK

Network simulations and interactive learning models have become invaluable tools in computer science education, particularly in online settings. This is because they provide students with engaging, hands-on experience exploring complex network concepts, protocols, and mechanisms, often without the need for expensive hardware or real-world deployments. This section reviews existing resources and identifies limitations that our approach addresses.

Existing deployments of network simulators and introductions of learning models in computer networking courses primarily focus on application layer and transport layer protocols and mechanisms, for instance, TCP flow control and congestion control (Kurose & Ross, 2020; Holiday, 2003), TCP error control and connection management (Liu, 2019), and the comparison of network performance in different parameter settings (Fitigau & Toderean, 2012; Kulgachev & Jasani, 2010; Aburdene, Meng & Mokodean, 2004). Simulators like NS-3, OMNeT++, and OPNET enable experimentation with diverse transport protocols and network conditions.

Learning models based on Packet Tracer simulator are prevalent. (Ristov, Spasov & Gusev, 2015) integrates Packet Tracer with CCNA certification training in their computer networking course, but it

Table 6. Student engagement metrics

Activity	Average Time Spend (Weeks)	Completion Rate (%)
Level 1: subnet management	2/3	100.00%
Level 1: packet metadata changes	2/3	100.00%
Level 1: ARP basics	1/3	100.00%
Level 2: troubleshoot routing error	1	93.75%
Level 2: troubleshoot switching error	1	91.67%
Level 3: matrix (paired projects)	2	83.33%

Table 7. Learning objectives assessment

Learning Objectives	Assessment Activities	Performance Result (Below Expectation: 0-70%, at Expectation: 71-89%, Above Expectation: 90-100%)
Understand ARP request/reply process	Quiz & discussions	Above expectation (91%)
Understand how IP translates to MAC	Quiz & labs	At expectation (86%)
Understand ARP cache	Labs	At expectation (77%)
Understand the information exchange for MAC address discovery	Quiz & labs & projects	At expectation (73%)
Learn how to interpret routing tables and forwarding tables	Quiz & discussions	Above expectation (92%)
Create routing tables and forwarding tables	Quiz & labs	At expectation (81%)
Identify common routing problems and resolve them	labs	At expectation (86%)
Understand the structure and purpose of packet headers	Quiz & discussions	Above expectation (99%)
Trace and understand the modifications made to headers during transfer	labs	At expectation (82%)
Evaluate how changes in headers reflects specific network behavior involved	labs	At expectation (79%)

prioritizes industry-specific skills over comprehensive network understanding. Other learning models using Packet Tracer (Vijayalakshmi, Desai & Raikar, 2016; Zhang, Liang & Ma, 2012; Janitor, Jakab & Kniewald, 2010) offer basic introductions and application/transport layer exercises but neglect crucial network and link layer aspects. Further limitations can be observed in models focusing solely on router configuration via device commands (Javid, 2014; Noor, et al., 2018), or topics like Internet of Things (IoT) simulation (Gwangwava & Mubvirwi, 2021; Almalki, 2020; Chaudhari, Joshi, Joshi & Kumar, 2020; Gumina & Tang, 2021), or basic routing table manipulation, wireless, and firewall configurations (Allison, 2022; Abdrabou & Shakhatreh, 2021; Reddy, et. al, 2020). These models lack deeper exploration of network internals and complex scenarios in network and link layers.

Our approach distinguishes itself from existing learning models by addressing these limitations. We prioritize comprehensive coverage of network layers, encompassing the network and link layers alongside the transport layer. This allows students to develop a holistic understanding of network protocols, mechanisms, and interactions across all layers. Furthermore, our framework incorporates real-world network errors and complex scenarios, challenging students to apply their knowledge in practical settings. This fosters deeper learning and prepares students for real-world network engineering challenges.

CONCLUSION

This paper proposes an innovative, hardware-independent framework for effectively teaching and learning the intricacies of lower-layer computer networks in the online environment. Our framework applies a progressive learning strategy, focusing on key areas like ARP, switch self-learning and forwarding, routing table construction, and packet metadata usage/changes during transfer, which often lack sufficient hands-on practice in traditional coursework. Grasping these fundamental functionalities not only facilitates understanding of network communication architecture but also equips students with advanced skills.

The framework encompasses three levels, each featuring multiple meticulously designed experiments to guide students step-by-step through their learning journey. To enhance comprehension of packet metadata transformations, we introduce a novel "*softMAC conversion*" mechanism, providing clear visual feedback at each stage. This allows students to actively engage with the experiments, personalize their learning pace, and explore related topics concurrently.

Over the past 4 years, our framework has consistently demonstrated its effectiveness in boosting student performance, even within the online setting. These encouraging results underscore the framework's potential to revolutionize computer networks education and empower students with a deeper understanding of this critical foundational subject. Furthermore, our research presents several intriguing research implications for future work:

- Scalability and adaptability: Investigating the framework's potential for adaptation to boarder technical domains and diverse online learning environments, potentially accommodating larger student populations.
- Assessment and optimization: Developing robust methods to assess the framework's efficacy in various contexts and identify opportunities for further optimization in terms of experimental design, feedback mechanisms, and personalized learning pathways.
- Transfer of learning: Evaluating the long-term impact of the framework on students' ability to apply their newly acquired knowledge and skills to real-world networking scenarios and advanced coursework.

REFERENCES

Abdrabou, A., & Shakhatreh, W. (2021). Teaching Computer Networks to Electrical Engineering Students by a Lecture-based Course and a Lab Course: A Quantitative Analysis of Students' Perceptions. In *2021 The 6th International Conference on Information and Education Innovations*. ACM. doi:10.1145/3470716.3470721

Aburdene, M., Meng, X., & Mokodean, G. (2004). Analysis of Computer Networks Courses in Undergraduate Computer Science, Electrical Engineering and Information Science Programs. *Proceedings of the 2004 American Society for Engineering Education Annual Conference & Exposition*. doi:10.18260/1-2--13660

Allison, J. (2022). Simulation-Based Learning via Cisco Packet Tracer to Enhance the Teaching of Computer Networks. *Proceedings of the 27th ACM Conference on Innovation and Technology in Computer Science Education*. doi:10.1145/3502718.3524739

Almalki, F. (2020). Implementation of 5G IoT Based Smart Buildings using VLAN Configuration via Cisco Packet Tracer. *International Journal of Electronics Communication and Computer Engineering*, *11*(4), 56–67.

Chaudhari, R., Joshi, K. K., Joshi, N., & Kumar, M. (2020). Smart and secure home using IOT Simulations with Cisco Packet Tracer. *International Journal of Scientific Research in Computer Science*. *Engineering and Information Technology*, *6*(3), 88–93. Advance online publication. doi:10.32628/CSEIT206311

Cisco System. (2023). *Cisco Packet Tracer (version 8.2.1)* [Computer Software]. https://www.netacad.com/courses/packet-tracer

Fitigau, I., & Toderean, G. (2012). Performance analysis of TCP and UDP using OPNET simulator. *11th International Conference on development and application systems*.

Gumina, S., & Tang, H. (2021). Inspiring Student Creativity: Collaboration on a Network Design Using IoT Project. *Proceedings of the 21st Annual Conference on Information Technology Education*, 49–50. doi:10.1145/3450329.3478318

Gwangwava, N., & Mubvirwi, T. (2021). Design and Simulation of IoT Systems Using the Cisco Packet Tracer. *Advances in Internet of Thing*, *11*(2), 59–76. Advance online publication. doi:10.4236/ait.2021.112005

Holiday, M. (2003). Animation of computer networking concepts. *Journal of Educational Resources in Computing*, *3*(2). https://dl.acm.org/doi/10.1145/982753.982755

IBTA (InfiniBand Trade Association Std). (2015). *InfiniBand Architecture Specification Volume 1, Release 1.3* [Online]. https://www.infinibandta.org/ibta-specification/

Janitor, J., Jakab, F., & Kniewald, K. (2010). Visual Learning Tools for Teaching/Learning Computer Networks. *2010 Sixth International Conference on Networking and Services*, 351-355. https://dl.acm.org/doi/10.1109/ICNS.2010.55

Javid, S. (2014). Role of packet tracer in learning computer networks. *International Journal of Advanced Research in Computer and Communication Engineering*, *3*(5), 6508–6511.

Kulgachev, V., & Jasani, H. (2010). 802.11 Networks Performance Evaluation using OPNET. *Proceedings of the 2010 ACM conference on Information technology education*, 149-152. https://dl.acm.org/doi/abs/10.1145/1867651.1867690

Kurose, J., & Ross, K. (2020). *Computer Networking: A Top-Down Approach* (8th ed.). Pearson.

Liu, Q. (2019). An Integrated Framework for Learning Fundamentals in Computer Networks. *ASEE Computers in Education Journal, 10*(3).

Noor, M., Yayao, N., & Sulaiman, S. (2018). Effectiveness of using Cisco packet tracer as a learning tool: A case study of routing protocol. *International Journal of Information and Education Technology (IJIET)*, *8*(1), 11–16. Advance online publication. doi:10.18178/ijiet.2018.8.1.1004

NS-3. (2023). *NS-3 Network Simulator* [Computer Software]. https://www.nsnam.org/

OMNeT++. (2023). *OMNeT++ Discrete Event Simulator (version 6.0.2)* [Computer Software] https://omnetpp.org/

Reddy, P., Akram, P., Ramana, T., Ram, P., Raj, R., & Sharma, M. (2020). Configuration of Firewalls in Educational Organization LAB setup by using Cisco Packet Tracer. *2020 IEEE International Symposium on Sustainable Energy, Signal Processing and Cyber Security*, 1–6. doi:10.1109/iSSSC50941.2020.9358818

Ristov, S., Spasov, D., & Gusev, M. (2015). Successful Integration of Practical Cisco CCNA in the computer networks design course. *2015 IEEE Global Engineering Education Conference*, 694-703. doi:10.1109/EDUCON.2015.7096045

Riverbed Technology. (2022). *Riverbed OPNET Modeler (version 18.10.0)* [Computer Software]. https://support.riverbed.com/content/support/software/opnet-model/modeler.html

SolarWinds Worldwide. (2024). *Graphic Network Simulator 3 (version 2.2.45)* [Computer Software]. https://www.gns3.com/

Vijayalakshmi, M., Desai, P., & Raikar, M. (2016). Packet Tracer Simulation Tool as Pedagogy to Enhance Learning of Computer Network Concepts. *2016 IEEE 4th International Conference on MOOCs, Innovation and Technology in Education*, 71-76. doi:10.1109/MITE.2016.024

Wireshark Foundation. (2024). *Wireshark Network Protocol Analyzer (version 4.2.2)* [Computer Software]. https://www.wireshark.org/

Zhang, Y., Liang, R., & Ma, H. (2012). Teaching Innovation in Computer Network Course for undergraduate students with Packet Tracer. *IERI Procedia*, *2*(1), 504–510. doi:10.1016/j.ieri.2012.06.124