

Survey of Network Learning and Practicing Tools

Irina Maria Ciocan, Lorenz Lehle*

**Chair of Network Architectures and Services*

School of Computation, Information and Technology, Technical University of Munich, Germany

Email: irina.maria.ciocan@tum.de, lehle@net.in.tum.de

Abstract—Practical exercises and visualization play an important role in the learning process. In networking, they help students understand difficult and abstract concepts, and allow them to observe the behavior of protocols, configurations and devices in different scenarios.

This paper presents an analysis of four different tools: Cisco Packet Tracer, Graphical Network Simulator 3 (GNS3), Common Open Research Emulator (CORE) and plain orchestrating service (pos), together with its virtual clone vPos, with respect to their methodologies and three characteristics: realism, scalability and compatibility.

The analysis shows that no tool is universally superior, each tool presenting different trade-offs between the aforementioned characteristics.

Index Terms—Cisco Packet Tracer, GNS3, CORE, pos, vPos

1. Introduction

Networking courses often rely on practical exercises to help students understand how different protocols, configurations and network components behave. Since providing physical hardware for a large number of students is expensive, difficult to maintain and thus, unrealistic, many courses and certifications rely on tools that simulate or emulate the behavior of real network components [1]–[3]. As a result, the tools differ in how realistic, scalable and suitable for beginners or advanced users they are.

This paper analyzes several tools commonly used for network learning and practicing. On one hand, it considers tools such as Cisco Packet Tracer (Packet Tracer), GNS3 and CORE, which enable the creation of virtual, emulated or hybrid network environments [1], [3], [4]. On the other hand, it also examines pos, which is used to orchestrate physical hardware, and vpos, which is a virtualized version of the pos testbed [5].

The paper first introduces each tool and their respective methodologies. Subsequently, the analysis is structured according to a set of criteria relevant in an educational context, each accompanied by a brief explanation of its relevance.

2. Tools and their methodologies

To understand the functionality of each tool, this section first examines the objectives and use cases they were designed to address.

2.1. Cisco Packet Tracer

Packet Tracer is the network learning and practicing tool used in the Cisco Certified Network Associate (CCNA) certification. It is used throughout the course to teach topics such as network configuration, routing and the fundamentals of security, automation and programmability. [1], [2], [6], [7]

Packet Tracer uses visualization based learning to teach the basic concepts of network design and configuration, troubleshooting and protocol behavior. The workflow of Packet Tracer is based on the graphical construction of topologies, where users place virtual network devices and configure them [6]. Its GUI is shown in Figure 1a.

In a series of experiments conducted by Vijayalakshmi *et al.*, Packet Tracer was used to teach abstract concepts through visualization. As a result, students were able to better understand protocols such as ICMP, TCP, UDP, and application layer services like DNS, DHCP, HTTP, FTP, and E-mail. [2]

Additionally, Packet Tracer includes the *Activity Wizard* feature, which allows teachers to define a final, target network and an initial configuration which is provided to students. This gives teachers the possibility to automatically assess students' work. [1]

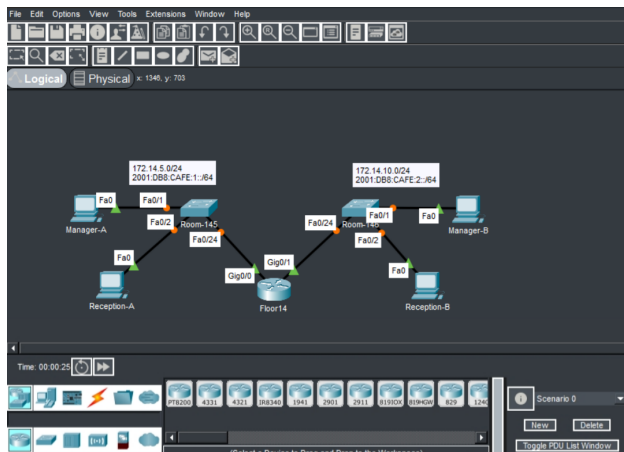
2.2. GNS3

GNS3 is used to emulate, configure, test and troubleshoot virtual and real networks. GNS3 allows users to run topologies ranging from only a few devices on a laptop, to many devices hosted on multiple servers or even hosted in the cloud. [3], [8]–[10]

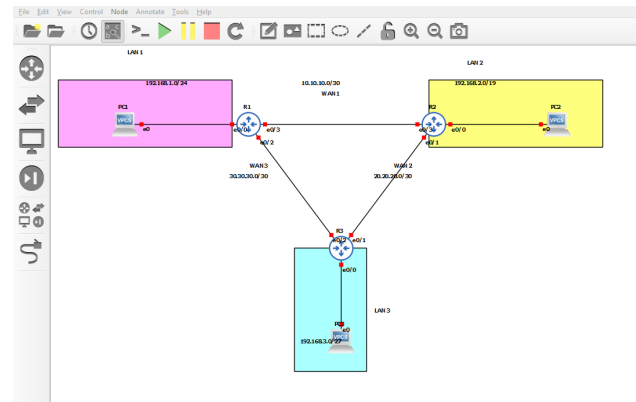
The methodology of GNS3 is based on the virtualization of real networks, allowing students to work with emulated network devices and real operating systems in a controlled environment. Similar to Packet Tracer, GNS3 has a visualization oriented workflow in which students draw topologies, as shown in Figure 1b. Students gain knowledge through creating topologies, testing their performance and analyzing their behavior. [8], [9]

2.3. CORE

CORE is a tool for building virtual networks. It is an efficient and scalable emulator that runs applications and protocols without modification. CORE has many applications, such as network and protocol research, testing, evaluating networking scenarios and increasing the size of physical test networks. [4], [11]



(a) Cisco Packet Tracer GUI



(b) GNS3 GUI

Figure 1: Side by side view of the graphical user interfaces of Cisco Packet Tracer (left) and GNS3 (right).

CORE is based on interactive experimentation. Similar to the workflows of Packet Tracer and GNS3, the CORE workflow starts with drawing nodes, network devices and links between them. After the user configures and instantiates the topology, the session enters the runtime phase. When the session is stopped, the user has the opportunity to save the log files and node data collected during the runtime phase. [4], [11]

While the graphical workflow is similar to Packet Tracer and GNS3, the underlying methodology differs. CORE is used for experimentation and protocol evaluation rather than visualization based learning. [4], [11]

2.4. pos

pos is a framework developed at the Chair of Network Architectures and Services, which focuses on incentivizing reproducible network experiments. Reproducibility is defined as the property that allows different people to get the same results on the same setup. The developers' aim is to create a tool that reduces the amount of work required for conducting reproducible experiments. [5]

vpos is a virtualized version of the pos framework that replicates the methodology of pos in a virtual environment. It enables the execution of pos experiments without requiring access to the physical testbed. [5]

The methodology of pos and vpos is based on an automated experimental workflow, which is divided into a setup, an experiment and an evaluation phase. [5]

During the setup phase, devices are allocated and booted via live images, to ensure a clean, identical state each time the experiment is executed. The configuration is then done via scripts only. [5]

Depending on which host has access to them and where they are used in the workflow, pos distinguishes between global variables (i.e., accessible from all experiment hosts), local variables (i.e., defined for each experiment host) and loop variables (i.e., shared across all experiment hosts, but continuously changed between different measurement runs). During the measurement phase, the script is executed once for each possible combination of loop variables to ensure full coverage. After the completion of

all runs, the collected data and outputs are stored in the result folder. [5]

During the evaluation phase, results are parsed and evaluated using scripts, with the metadata being attached automatically. pos can also generate plots and figures based on the collected data and it has the ability to automatically transform data into a publishable format (e.g. a repository or a website containing all collected information). [5]

This strict experimental methodology enforces reproducibility by design. [5]

3. Characteristics

In order to meaningfully compare the presented tools, this section analyzes them based on realism, scalability and compatibility, which are relevant for network learning and practicing.

3.1. Realism

Realism measures how accurately a certain tool replicates the behavior of a real network. It is an important trait to consider because it determines how close students can observe the aspects of real networks.

Packet Tracer uses software simulation to replicate the behavior of real world network components like routers and switches. Simulation offers many advantages, for example, it allows users to create large-scale topologies that would be difficult or impossible to create with physical hardware or with an emulator. It also facilitates remote access for both students and teachers and it allows students to make mistakes without breaking physical hardware. [1], [2], [6]

However, simulation also has disadvantages. Packet Tracer has limitations caused by the simplified protocol implementation. Some protocols and functions can present different errors, due to the fact that Packet Tracer's simulated devices cannot reproduce exactly how real devices behave. Some advanced features that are present in real devices are missing in Packet Tracer, which causes errors when using protocols like ICMPv6 and DHCPv6. Packet

Tracer is also unable to simulate load and congestion or to measure throughput and latency. [1], [9]

GNS3 follows a different approach by relying on emulation rather than simplified simulation. It supports all features available on physical routers and can emulate services like httpd, ftpd, telnetd and sshd. This allows it to produce traffic as if it were generated on a real network and to replicate the configuration and behavior of real networks more closely than Packet Tracer. [3], [9], [10]

In a study conducted by Gil *et al.*, student participants of the survey stated that GNS3 proved itself to be especially useful in helping them understand topics about communications and TCP/IP. When asked about the similarity to the real laboratory, the students gave it a score of 3/5, while the lecturer estimated the similarity to be up to 90%. [9] Similarly, the effectiveness of Packet Tracer as a learning tool has also been demonstrated in educational studies. Vijayalakshmi *et al.* observed that the use of Packet Tracer in networking exercises improved students' understanding of fundamental concepts, increasing the average quiz score from 65% to 85%. [12]

CORE achieves a high level of realism by offering a high fidelity emulation for layers 3 through 7 and a simplified simulation of layers 1 and 2. Similarly to GNS3, CORE can connect a live running emulation to physical hardware with the help of the RJ45 node and the Tunnel tool, allowing users to test protocols and applications in a hybrid environment. Emulated nodes can send and receive traffic from physical hardware, which allows them to interact in real time. [4], [11]

The RJ45 node represents an interface that connects the emulated network to a physical external device through a host network port. The limitations that this approach brings are the host's requirement for a physical port for each external device and the requirement that the host and the external devices be colocated. [11]

The Tunnel tool allows the connection of multiple emulations as well as the connection of an emulation to a physical device. Unlike RJ45, this tool does not require the dedication of a host's port for the connection and allows devices and CORE machines to be connected remotely. [11]

pos initializes physical devices with live-boot images and uses direct wiring between hosts to prevent unwanted influence of switches and hubs, which isolates the network

from non-investigated devices that could interfere with the experiment. The live images force the OS to repeatedly start from a well-defined state. Thus, pos reflects true hardware behavior. [5]

In a case study conducted by Gallenmüller *et al.*, the virtual clone of pos, vpos, showed significant performance differences, as shown in Figure 2. For the hardware testbed the forwarded packet rate increases linearly along the generated packet rate until it reaches the system's limits. In contrast, for the virtual testbed the forwarding rate already becomes unstable at relatively low generated rates, resulting in packet loss. This highlights the importance of realism when analyzing network behavior. [5]

However, teaching environments do not require the same level of measurement accuracy as scientific experiments. Therefore, the realism offered by vpos is sufficient for educational purposes.

3.2. Scalability and performance

In an educational context, scalability refers to either a tool's ability to handle topologies of different sizes or its ability to support multiple users. Considering scalability is relevant for assessing whether a tool is suitable for a class where collaborative work is required and whether it can effectively substitute hardware based networking laboratories.

As mentioned in Section 3.1, given that **Packet Tracer** is a simulation tool, it can be used to design large and complex topologies, which would be difficult or expensive to implement using physical hardware [2].

The tool is also equipped with a Multiuser feature, which facilitates the creation of a collaborative environment. In a test conducted by Andrew Smith and Colin Black, they showed that the Multiuser feature allows a shared topology to be accessed and controlled remotely by more than 40 users, up to 20 being active simultaneously. [7]

Because it can be ran on a single physical computer, **GNS3** maintains the accessibility provided by Packet Tracer, while also creating a more realistic virtual environment. Virtualization also allows for devices to be easily duplicated or moved across the topology. [8], [9]

However, GNS3 emulates real Cisco IOS images, which requires significantly more CPU and memory resources. As a result, its performance is limited by the capabilities of the host system, leading to lower scalability compared to Packet Tracer, while still offering greater scalability than laboratories based on physical hardware. [3], [8]–[10]

The performance of **CORE** depends on several different factors: hardware, OS version, active processes, network traffic and GUI usage. According to the documentation, it is more meaningful to research how much traffic CORE can handle, rather than how many nodes it can emulate, which highly depends on the workload of each node. [11]

On a single-CPU Xeon 3.0GHz, 2GB RAM server running Linux, CORE was able to instantiate more than 100 nodes. CORE nodes share the host's kernel and file system, and require a small amount of system resources, which leads to a high scalability. [11]

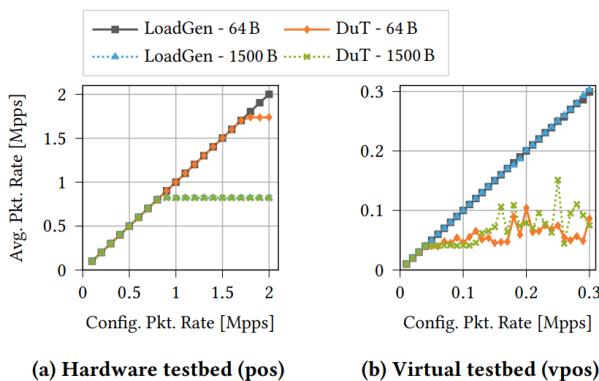


Figure 2: Performance comparison between pos and vpos. [5]

TABLE 1: Characteristics comparison between tools

	Packet Tracer	GNS3	CORE	pos
software	✓	✓	✓	vPos
simulation	✓			
emulation		✓	✓	
hardware			✓	✓
realism	low	moderate	high	very high
scalability	high	moderate	moderate	low
compatibility	low	medium	high	high

On the same server, it was able to handle a traffic of approximately 300.000 packets per second. This number indicates how often the system handled sending or receiving packets. The size of the packets and the number of hops that are emulated do not limit the performance of CORE. [11]

Additionally, CORE allows multiple emulations on different servers to be connected. A large emulation scenario divided across multiple devices can be controlled by the same GUI that can be run either on one of the emulation servers or on a dedicated device. [11]

When using **pos**, topologies have to be physically wired. This prevents them from being automatically created or recreated, and leads to a low topology scalability. [5]

When a multiuser testbed is operated, an integrated calendar system schedules access to experimental nodes and ensures that resources are reserved for the entire duration of an experiment. This enables multiple users to conduct independent experiments in parallel on different nodes while preventing simultaneous use of the same node. [5]

Unlike **pos**, **vpos** does not require nodes to be physically wired. As discussed in Section 3.1, **vpos** offers more scalability through virtualization, although it comes at the cost of reduced reliability compared to **pos**. [5]

As shown in Figure 2, the maximum forwarding performance of **pos** reaches approximately 1.750.000 pps for 64 B packets and approximately 800,000 pps for 1500 B packets, both exceeding the 300.000 packets per second achieved by CORE [5], [11]. **vpos** shows the lowest throughput, reaching approximately 40.000 pps before packet loss occurs, regardless of packet size [5].

3.3. Compatibility and flexibility

When designing a network and selecting a network tool, compatibility and flexibility are essential, as they determine which devices and technologies a tool supports.

Packet Tracer can simulate multiple routers, switches and wireless devices, but is limited to devices that are part of the Cisco ecosystem. An advantage of this approach is that Packet Tracer does not require users to upload real device images or operating systems. However, this approach brings the disadvantage of reduced device diversity, limiting the range of models a user can introduce in the topology. [1], [6], [13]

One major advantage that **GNS3** holds over Packet Tracer when it comes to compatibility is that it supports

the emulation of any Quick Emulator (QEMU) based devices, not only Cisco devices. GNS3 also being compatible with VMware allows virtual machines that operate either Windows or Linux to be integrated into GNS3 networks. [8], [10]

In an experiment conducted by Mohtasin *et al.*, it was concluded that a network realized with GNS3 can also be connected to physical hardware. This was possible both by creating a Loopback network adapter on the physical computer and by connecting it via an Ethernet cable to the machine running the GNS3 network. [8]

CORE features three types of nodes: CORE nodes, Docker nodes and Physical nodes [11]. CORE nodes can function as virtual hosts, routers, switches and wireless devices. They originally used the FreeBSD network stack virtualization and are currently backed by Linux network namespaces [4], [11]. Docker nodes allow users to configure nodes using predefined images and filesystems that CORE nodes do not provide [11]. Physical nodes, as discussed in section 3.1, can be connected locally via RJ45 nodes or remotely via Generic Routing Encapsulation (GRE) tunnels [11]. CORE emulates network devices by using Linux namespaces, each having an independent process environment and a private network stack [11]. Compared to Packet Tracer, which relies on predefined device models, CORE offers greater flexibility in configuring nodes and integrating additional software.

pos utilizes live-boot images for experiment nodes to ensure that every experiment begins from a consistent initial state. Despite this standardized starting point, the testbed gives users freedom to design and execute experiments using scripts and root access on experiment nodes. [5]

One of the objectives accomplished by **pos**, referred to as heterogeneity, is the development of a testbed adaptable to different devices. In order to achieve heterogeneity, **pos** implements two APIs: an initialization and a configuration interface. The initialization interface is used to reset and boot devices, for example via IPMI, while the configuration interface is used to configure devices and execute experiments, typically via SSH. To support devices that rely on APIs other than those already implemented, **pos** allows the implementation of additional initialization and configuration interfaces. [5]

4. Conclusion and future work

This paper shows that there is no single "best" tool for network learning and practicing. Instead, each user has to choose a tool based on the intended use case, the learning objective and their level of expertise. Each of the analyzed tools make different trade-offs between realism, scalability and compatibility, as shown in Table 1.

For beginners, tools like Packet Tracer and GNS3 are more appropriate, as they can be used to learn the fundamentals of networking. CORE and **pos** are more suitable for advanced users, since they require deeper knowledge and understanding of networking concepts.

The analyzed tools differ significantly in terms of realism. Packet Tracer relies on simulation, which leads to low realism. GNS3 and CORE reach higher levels of realism through emulation. However, none of the three

tools reach the level of realism provided by pos, as it operates directly on physical hardware.

When the goal is to model large-scale topologies, Packet Tracer and GNS3 provide better scalability than CORE or pos, and are better suited for scenarios in which limited hardware is available. CORE's scalability is limited due to its reliance on the host's resources, while pos offers the lowest scalability, as it depends on the availability of physical hardware. pos prioritizes realism and reproducibility over scalability and accessibility, with scalability being addressed through vPos.

In terms of compatibility, Packet Tracer is the most limited, as it is restricted to devices belonging to Cisco's ecosystem. GNS3 and CORE can integrate a broader spectrum of devices, while pos provides the highest level of compatibility, through its focus on heterogeneity.

Packet Tracer focuses on accessibility rather than realism, making it well suited for introductory courses. GNS3 achieves a higher level of realism while maintaining accessibility, allowing users to move towards more complex and realistic scenarios. CORE offers a strong balance between realism, scalability and compatibility, as it supports a wide range of software and can integrate external devices.

References

- [1] J. Uramová, P. Segeč, and M. Kontšek, "Best practise for creating Packet Tracer activities for distance learning and assessment of practical skills," *2019 International Conference on Emerging eLearning Technologies and Applications (ICETA)*, 2019.
- [2] M. Vijayalakshmi, P. Desai, and M. M. Raikar, "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 (MITE)*, 2016.
- [3] S. Liu, J. Liu, H. Wang, and M. Xian, "Feasibility analysis of network security teaching platform based on KVM and GNS3," *2019 International Conference on Information Technology and Computer Application (ITCA)*, 2019.
- [4] J. Ahrenholz, C. Danilov, T. R. Henderson, and J. H. Kim, "CORE: A real-time network emulator," *2008 IEEE Military Communications Conference (MILCOM)*, 2008.
- [5] S. Gallenmüller, J. Naab, I. Adam, and G. Carle, "The pos framework: A methodology and toolchain for reproducible network experiments," *Proceedings of the 17th International Conference on Emerging Networking Experiments and Technologies (CoNEXT '21)*, 2021.
- [6] J. Janitor, F. Jakab, and K. Kniewald, "Visual Learning Tools for Teaching/Learning Computer Networks: Cisco Networking Academy and Packet Tracer," *2010 Sixth International Conference on Networking and Services (ICNS)*, 2010.
- [7] A. Smith and C. Bluck, "Multiuser Collaborative Practical Learning Using Packet Tracer," *2010 Sixth International Conference on Networking and Services*, 2010.
- [8] R. Mohtasin, P. Prasad, A. Alsadoon, G. Zajko, A. Elchouemi, and A. K. Singh, "Development of a Virtualized Networking Lab using GNS3 and VMware Workstation," *2016 International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET)*, 2016.
- [9] P. Gil, G. J. Garcia, A. Delgado, R. M. Medina, A. Calderon, and P. Marti, "Computer Networks Virtualization with GNS3: Evaluating a Solution to Optimize Resources and Achieve Distance Learning," *2015 International Conference on Information Technology Based Higher Education and Training (ITHET)*, 2015.
- [10] "Gns3 documentation," <https://docs.gns3.com/docs/>, last accessed: 14.12.2025.
- [11] "Core documentation," <https://coreemu.github.io/core/index.html>, last accessed: 14.12.2025.
- [12] M. M. R. Vijayalakshmi M., Padmashree Desai, "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*, 2016.
- [13] A. Sharma, A. Kumar, and M. Kumar, "Comparison of Packet Tracer and EVE-NG Tools for Efficient Network Design," *2024 11th International Conference on Computing for Sustainable Global Development (INDIACom)*, 2024.