Tree Topology In Network Simulation: A Comparative Study of Packet Tracer and NS2

Akshara PB[1], Prof. Suranya Das [2]
[1, 2] Dept of Computer Science and Engineering
[1, 2] Government Engineering College, Wayanad Kerala, India

Abstract- *Tree topology is a fundamental net- work architecture that offers a hierarchical structure, making it suitable for large-scale networks, including enterprise environ- ments and academic institutions. This review paper presents a comparative analysis of tree topology implementation using two widely used network simulation tools: Cisco Packet Tracer and Network Simulator 2 (NS2). The study evaluates key aspects such as ease of use, simulation depth, scalability, performance analysis capabilities, and real-world applicability of both tools. Packet Tracer, with its graphical interface, is ideal for network configuration, troubleshooting, and educational training, pro- viding an intuitive environment for learners and professionals. In contrast, NS2 offers in-depth network modeling, traffic flow analysis, and statistical performance evaluation, making it more suitable for research-oriented simulations. The findings highlight the strengths and limitations of each tool, providing insights into which platform is best suited for different networking scenarios. This study serves as a reference for researchers, educators, and network professionals in selecting the most appropriate simulation tool based on their specific requirements.*

I. INTRODUCTION

Network topology plays a crucial role in determining the efficiency, scalability, and performance of communication net- works. Among various topologies, tree topology is widely used due to its hierarchical structure, which enables efficient data transmission and network management. It is commonly implemented in enterprise networks, academic institutions, and large-scale infrastructures, where structured connectivity is essential. Tree topology integrates the features of both star and bus topologies, ensuring better fault tolerance, scalability, and organized data flow.

To analyze and evaluate network performance before real- world deployment, network simulation tools are used exten- sively. Cisco Packet Tracer and Network Simulator 2 (NS2) are two widely adopted tools for network simulation and analysis. Packet Tracer, developed by Cisco, is a GUI-based simulation tool primarily used for learning network configurations, trou- bleshooting, and real-world device emulation. It provides a hands-on approach for students and professionals to practice network design.

On the other hand, NS2 is a discrete event network simu- lator that allows researchers to analyze network protocols, traffic flow, and performance metrics at a more granular level. Unlike Packet Tracer, NS2 relies on scripting (TCL) and provides detailed statistical outputs for research-based network simulations. This paper presents a comparative study of tree topology implementation in Packet Tracer and NS2, focusing on ease of implementation, simulation capabilities, performance evaluation, scalability, and real-world applicabil- ity. By analyzing their features, advantages, and limitations, this study aims to provide insights into choosing the most suitable simulation tool for different networking requirements, whether for educational, professional, or research purposes.

II. METHODOLOGY

A. *Comparison of Cisco Packet Tracer and Network Simulator 2 (NS2): Capabilities and Applications*

Difference between NS2 (Network Simulator 2) and Packet Tracer is that NS2 is a highly customizable, flexible network simulation platform that allows for detailed modeling of various network protocols and behaviors, while Packet Tracer is primarily designed for beginner network learning, focusing on a user-friendly interface to simulate basic Cisco network devices and configurations, particularly for studying for CCNA certifications; essentially, NS2 offers greater complexity and customization for advanced network research, while Packet Tracer prioritizes ease of use for basic network concepts.

Packet Tracer and NS2 have distinct differences in their usage and capabilities. In terms of ease of use, Packet Tracer provides a graphical user interface (GUI), which simplifies network design and configuration. This makes it ideal for beginners and practical networking exercises. NS2, in contrast, requires scripting knowledge and is more complex to use, but it offers a higher level of flexibility and customization. When it comes to simulation depth, Packet

Tracer focuses primarily on basic network simulations and real-world device configurations. It allows users to test and troubleshoot network setups using virtual Cisco devices. NS2, however, provides a more detailed approach, supporting in-depth simulations of network protocols, traffic flow, and packet-level interactions. For performance analysis, Packet Tracer offers basic tools for troubleshooting and network diagnostics, making it suitable for educational purposes. NS2, on the other hand, is designed for comprehensive performance analysis, providing statistical insights such as packet delivery ratio, latency, congestion control, and throughput evaluation.
Scalability is another key difference between the two tools. Packet Tracer is limited in scalability and is most effective for small to medium-sized networks. NS2 supports large- scale network simulations and is commonly used in research projects that require complex network modeling and evalu- ation. In terms of application, Packet Tracer is best suited for students, educators, and network professionals who need hands-on experience with network configurations. It is partic- ularly beneficial for those studying networking concepts and preparing for Cisco certification exams. NS2, however, is used primarily in research environments where detailed protocol analysis and network performance studies are required.

Visualization and Customization Packet Tracer provides real- time network visualization, allowing users to see network devices interact dynamically. It offers a drag-and-drop inter- face for designing network topologies, making it accessible to beginners. NS2, however, relies on the Network Animator (NAM) tool for packet movement visualization, which is less interactive but provides deeper insights into network opera- tions. Customization is another area where NS2 outperforms Packet Tracer. While Packet Tracer is limited to predefined Cisco devices and configurations, NS2 allows researchers to implement and test custom network protocols, making it a more flexible option for network experimentation.

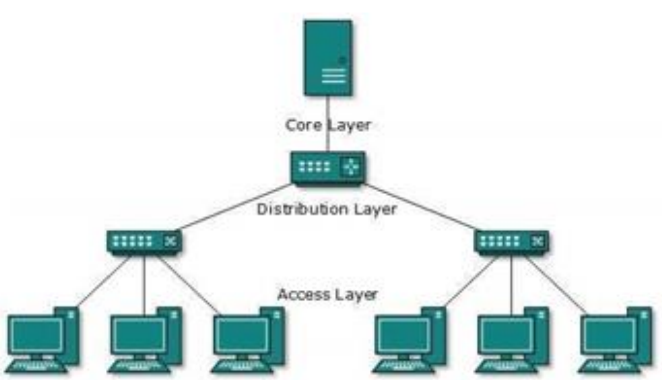

Fig. 1. Images of Heirarchical Topology

Also known as Hierarchical Topology, this is the most common form of network topology in use presently.This topology imitates as extended Star topology and inherits properties of bus topology. This topology divides the network in to multiple levels/layers of network. Mainly in LANs, a network is bifurcated into three types of network devices. The lowermost is access-layer where computers are attached. The middle layer is known as distribution layer, which works as mediator between upper layer and lower layer. The highest layer is known as core layer, and is central point of the network, i.e. root of the tree from which all nodes fork.

B. *Tree Topology Simulation in Cisco Packet Tracer: Struc- ture, Scalability, and Performance*

Tree topology is widely used in Cisco Packet Tracer for designing and simulating hierarchical network structures. In Packet Tracer, tree topology is implemented by interconnecting multiple switches and routers in a parent-child hierarchy, resembling an inverted tree structure. This topology consists of multiple layers: the core layer, distribution layer, and access layer, which allow efficient data transmission and network management. One of the primary advantages of using tree topology in Packet Tracer is the ease of network scalability. Additional devices can be integrated seamlessly without major reconfiguration. Moreover, this topology enhances network performance by distributing traffic across multiple links, reduc- ing congestion. However, tree topology in Packet Tracer also has some limitations, such as dependency on central devices (core switches/routers), which can create bottlenecks or single points of failure if redundancy is not properly implemented. Tree Topology in Cisco Packet Tracer Tree topology is widely used in Cisco Packet Tracer for designing and simulating hierarchical network structures. In Packet Tracer, tree topol- ogy is implemented by interconnecting multiple switches and routers in a parent-child hierarchy, resembling an inverted tree structure. This topology consists of multiple layers: the core layer, distribution layer, and access layer, which allow efficient data transmission and network management. One of the primary advantages of using tree topology in Packet Tracer is the ease of network scalability. Additional devices can be in- tegrated seamlessly without major reconfiguration. Moreover, this topology enhances network performance by distributing traffic across multiple links, reducing congestion. However, tree topology in Packet Tracer also has some limitations, such as dependency on central devices (core switches/routers), which can create bottlenecks or single points of failure if redundancy is not properly implemented.

To simulate tree topology in Packet Tracer, users can follow these steps: First, users need to add network devices, including routers, switches, and end devices, by selecting them from the device library and placing them onto the workspace. Once the devices are in place, the next step is to establish connec- tions between them using appropriate cables, such as copper straight-through or crossover, ensuring a hierarchical structure is maintained. After setting up the physical connections, users should configure IP addresses for routers and end devices to facilitate communication within the network. Proper IP address assignment ensures seamless data transfer between different network segments.

Following the IP configuration, routing protocols such as RIP, OSPF, or EIGRP should be enabled to optimize data forwarding and establish communication between different network nodes. Implementing these protocols enhances net- work efficiency and allows for dynamic routing updates. Finally, network connectivity must be tested using tools like the ping command and Packet Tracer's simulation mode. This step helps verify whether the network is functioning correctly and allows users to troubleshoot any potential issues. Cisco Packet Tracer allows users to analyze traffic flow, troubleshoot network issues, and experiment with different configurations to optimize tree topology networks. This makes it a valuable tool for both beginners learning networking concepts and professionals designing enterprise-level networks.

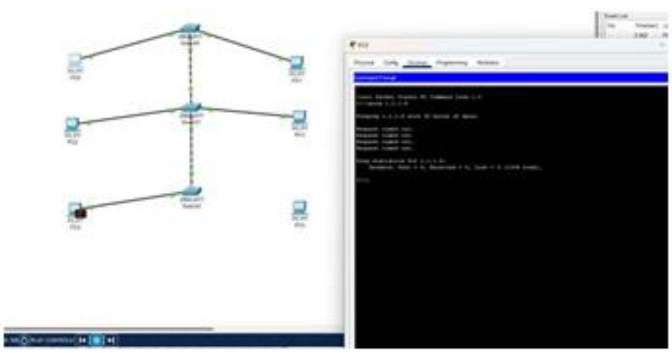

Fig. 2. Simulated Tree Topology in Cisco Packet Tracer

Fig. 2 provides a glimpse into a fundamental networking con- cept using Cisco Packet Tracer: testing network connectivity. The visual representation of the network, with its switches and PCs connected by green lines, gives a clear picture of the physical setup. However, the real action is happening in the command prompt window. By using the ping command, the user is essentially asking, "Can PC0 reach this other device on the network?" The successful replies confirm that communica- tion is flowing smoothly between PC0 and the device with the IP address 1.1.1.6. This seemingly simple test has significant implications in real-world networking. Network administrators use ping routinely to diagnose connectivity issues. If the pings had failed, it would signal a problem somewhere in the network – perhaps a faulty cable, a misconfigured switch, or an issue with the destination device itself. The ping statistics offer additional clues about the quality of the connection. The average round trip time, for example, can indicate the latency or delay in the network. The attempt to use the -t option, while unsuccessful in this case, highlights another important aspect of network troubleshooting: understanding the tools and their limitations. Different operating systems and network devices may have slightly different implementations of commands like ping. In this Packet Tracer simulation, the -t option might be unavailable to simplify the learning environment or to focus on specific concepts. Overall, this image encapsulates a key process in network management: verifying connectivity using the ping command. It also demonstrates the value of network simulation tools like Packet Tracer for learning and practicing networking skills in a safe and controlled environment.

Fig. 3 depicts a network simulation in Cisco Packet Tracer, showcasing a failed connectivity test. On the left, a network topology is displayed with three switches (Switch0, Switch1, and Switch2) interconnected and each connected to two PCs. All connections are indicated by green lines, suggesting they are physically active. However, on the right side, the command prompt of PC0 reveals a failed ping attempt to the IP address 1.1.1.6. The output shows four "Request timed out" messages, indicating that PC0 was unable to reach the destination. The ping statistics confirm this, reporting 100

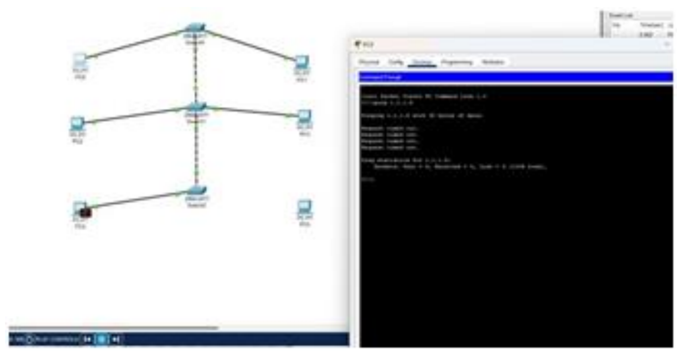

Fig. 3. Images of Heirarchical Topology

C. *Tree Topology Simulation in NS2: Hierarchical Network Modeling and Analysis*

Tree topology is a hierarchical network structure where nodes are arranged in a parent-child manner, similar to a tree. This type of network is commonly used in wired commu- nication networks, hierarchical routing, and structured data transmission systems. In Network Simulator 2 (NS2), we can implement a tree topology using TCL (Tool Command Lan- guage) by defining nodes, establishing duplex links between them, and visualizing the structure using NAM (Network Animator). The tree topology follows a structured approach, where a single root node connects to multiple child nodes, and each child node can further expand into additional sub-nodes, forming a multi-level hierarchical network. This topology is widely used in LAN (Local Area Networks), enterprise networks, and data centers where efficient routing and organi- zation of network traffic are essential. In this implementation, we define a simple three-level tree topology with one root node (n0), two intermediate nodes (n1, n2), and four leaf nodes (n3, n4, n5, n6). The root node serves as the central hub, connecting to its child nodes, which further branch out to additional nodes. To achieve this, we first create an instance of the NS2 simulator and open a NAM trace file to record the simulation. Next, we define a procedure called create link to simplify the process of adding duplex links between nodes. The duplex links represent bidirectional communication between nodes and are assigned a bandwidth of 1 Mbps, a delay of 10 ms, and a DropTail queue (a basic FIFO buffer). After defining this function, we create seven nodes, starting with the root node, followed by intermediate nodes, and finally the leaf nodes. These nodes are then interconnected to form the tree structure, with the root node at the top level, intermediate nodes at the second level, and leaf nodes at the third level.

Once the network topology is created, we schedule the simu- lation to end after 0.1 seconds and define a finish procedure to clean up the simulation, close the NAM trace file, and launch NAM to visualize the network topology. The simulation is then executed using the $ns run run command. When we run the script in the NS2 terminal, it generates a NAM output file (out.nam), which, when opened in NAM, displays the tree topology visualization. The expected output consists of a structured binary tree-like topology, where node n0 is at the top, branching into n1 and n2, which further split into n3, n4, n5, and n6, forming a three-level hierarchical network. This type of topology is useful in structured networks, allowing efficient communication and scalability.

To enhance the simulation, we can modify the script to add more levels to the tree, introduce network traffic (TCP, UDP, CBR) between nodes, or change network parameters such as bandwidth, delay, and queue management for performance analysis. By expanding the tree depth, we can create larger networks that closely resemble real-world hierarchical net- work structures, such as corporate networks, ISP backbone networks, or data centers. Additionally, traffic sources such as FTP (File Transfer Protocol), Constant Bit Rate (CBR), and TCP-based communication can be added to analyze network behavior under different conditions. The tree topology in NS2 provides a flexible and scalable framework for simulating hierarchical networks and studying their performance under various configurations.

When comparing the implementation of tree topology in Packet Tracer and NS2 (Network Simulator 2), both have their own advantages and limitations depending on the purpose of the simulation. Packet Tracer, developed by Cisco, is a GUI- based network simulation tool primarily designed for learning and practicing network configurations, device interactions, and protocol implementations in a visual environment. It allows users to create tree topology networks using routers, switches, and end devices while enabling real-time packet flow visual- ization, device configuration, and troubleshooting using Cisco CLI (Command Line Interface). This makes Packet Tracer an excellent tool for network administrators, students, and professionals who need a hands-on approach to understanding network behavior, VLANs, routing protocols, and network security. However, Packet Tracer is limited to Cisco-supported devices and protocols, and it lacks the ability to analyze complex network performance metrics, such as packet delay, congestion control, and queue management, making it less suitable for in-depth research and performance analysis.

On the other hand, NS2 is a command-line-based discrete event simulator that provides a more detailed, research- oriented simulation environment for network protocol testing, traffic analysis, and performance evaluation at a low level. Unlike Packet Tracer, which focuses on practical network- ing, NS2 allows users to define a tree topology using TCL scripting, create custom traffic flows (CBR, FTP, TCP, UDP), and analyze packet behavior over time. NS2 is widely used in academic research and network performance studies as it provides detailed statistical outputs, trace files, and simulation- based insights into network congestion, bandwidth utiliza- tion, and delay analysis. Additionally, NS2 integrates with NAM (Network Animator) for visualizing packet movements, making it useful for understanding network behavior over time rather than in real-time interactions like Packet Tracer. However, NS2 has a steeper learning curve, as it requires knowledge of scripting (TCL) and trace file analysis, and lacks a user-friendly GUI like Packet Tracer. In summary,

Packet Tracer is better suited for practical learning, network design, and device configuration, making it ideal for network engineers and students who need a hands-on understanding of networking concepts. Meanwhile, NS2 is more powerful for research, protocol analysis, and performance evaluation, making it a better choice for academic and research-based sim- ulations where low-level network behavior and performance metrics need to be studied in detail.

III. CONCLUSION

Tree topology remains a fundamental network structure, balancing scalability, efficiency, and hierarchical organization. Its widespread use in enterprise networks, educational institutions, and large-scale infrastructures highlights its significance in modern networking. This study provided a comparative analysis of tree topology implementation using Cisco Packet Tracer and Network Simulator 2 (NS2), shedding light on their respective capabilities, advantages, and limitations. Cisco Packet Tracer offers an intuitive, user- friendly platform for learning network configurations, device interactions, and troubleshooting in a simulated environment. Its real-time visualization and simplified approach make it an excellent tool for students, educators, and professionals seeking hands-on experience with network design. However, its limited scalability and lack of in-depth statistical analysis make it less suitable for advanced research applications.

Conversely, NS2 is a powerful simulation tool designed for in-depth network research and protocol analysis. With its scripting-based approach, it allows for precise modeling of network behaviors, performance metrics, and traffic analysis. Although it has a steep learning curve due to its reliance on TCL scripting and trace file evaluation, its ability to simulate large-scale networks with intricate traffic patterns makes it a valuable resource for researchers and network engineers. Ultimately, the choice between Packet Tracer and NS2 depends on the specific objectives of the user. If the goal is practical learning, real-world network configuration, and interactive troubleshooting, Packet Tracer is the preferred option. However, for in-depth performance analysis, protocol behavior evaluation, and large-scale network simulations, NS2 provides the necessary depth and flexibility.

REFERENCES

[1] M. N. De Miranda, D. R. Figueiredo, D. F. Luiz, C. B. Carvalho, and D. S. Menasche´, “Design and analysis of distributed tree growing algorithms,” *IEEE Access*, vol. 10, pp. 26580–26602, 2022.

[2] J. Ye, G. Fei, X. Zhai, and G. Hu, “Network topology inference based on subset structure fusion,” *IEEE Access*, vol. 8, pp. 194192–194205, 2020.

[3] Z. Jia, Y. Sun, Q. Liu, S. Dai, and C. Liu, “cretor: An sdn-based routing scheme for data centers with regular topologies,” *IEEE Access*, vol. 8, pp. 116866–116880, 2020.

[4] G. Fei, J. Ye, S. Wen, and G. Hu, “Network topology inference using higher-order statistical characteristics of end-to-end measured delays,” *IEEE Access*, vol. 8, pp. 59960–59975, 2020.

[5] Y. Zhang, Z. Zhang, W. Zhang, D. Mao, and Z. Rao, “Stop-probability- based network topology discovery method,” *IEICE Transactions on Communications*, vol. E107-B, no. 9, pp. 583–594, 2024.