Network Simulator

A Tool for Visualizing Network Protocols and Behaviors

Submitted By

Sonia yadav & Tavaheed Tariq

2022BITE030 & 2022BITE008

DEPARTMENT OF INFORMATION TECHNOLOGY

NATIONAL INSTITUTE OF TECHNOLOGY SRINAGAR

March 2025

Contents

Chapter 1

Introduction

Network Simulator is a comprehensive software program created to enable the user to grasp network protocols and represent network activity visually. Computer networks have been growing more complex, and an interactive learning mechanism that can simulate the behavior of different network protocols and activities in a graphical representation is becoming a necessity. This simulator fulfills this requirement by offering a vehicle for simulating multiple network conditions and protocols. [4]

1.1 Project Scope

The Network Simulator is structured into two major parts. The Layer Simulation part addresses the simulation of data link and physical layers, enabling users to know how devices such as bridges, switches, and hubs work in a network. The Protocol Simulation part addresses network protocols like CRC, CSMA/CD, Stop-and-Wait, and Go-Back-N, offering visualizations of how they work.

Chapter 2

Features

The Network Simulator includes several features that make it a valuable tool for learning network concepts:

2.1 Layer Simulation

The Layer Simulation element simulates data link layer devices and physical layer devices including bridges, switches, and hubs. It delivers visualization of the topology of the network utilizing PyVis and NetworkX to ensure users gain good knowledge about the working of the devices. It incorporates real-time visualization of traffic flow through networks to ensure better comprehension by the users on the flow of data within a network. Besides, it also provides analysis of network performance parameters to provide users with information regarding the effectiveness of various network arrangements. [1, 3, 5, 6]

2.2 Protocol Simulation

The Protocol Simulation element applies different network protocols with feedback displays. The implementation of the Cyclic Redundancy Check has noise channels that illustrate error detection features. Visualizing Carrier Sense Multiple Access with Collision Detection indicates how devices contend for a shared medium and how they recover from collisions within a network. Stop-and-Wait Protocol illustrates an easy flow control technique with a feedback display explaining how acknowledgments are performed. The Go-Back-N Protocol has an implementation that provides a sliding window protocol with packet transmission visualization and acknowledgment procedures. [1, 2, 5, 6]

2.3 User Interface

The Network Simulator has an interactive web-based interface developed with Streamlit. [5] The interface gives users real-time visualizations of network operations to allow them to grasp how networks operate. Users can configure simulation parameters easily to test various network scenarios. The interface displays simulation results and metrics in a simple way, and it is easy for users to analyze network performance. The design emphasizes giving a seamless and intuitive experience for learning.

Chapter 3

System Architecture

The Network Simulator follows a modular architecture that separates different components of the system for better maintainability and extensibility.

3.1 Overall Architecture

The system architecture comprises a number of key components collaborating to offer an end-to-end simulation environment. The User Interface Module, which is developed based on Streamlit [5], manages user input and presents simulation outputs in a user-friendly manner. The Network Topology Module is used to generate and control network topologies based on NetworkX [2] for flexible network designs. The Visualization Module is responsible for the visualization of network topologies and protocol activities with PyVis [3], which offers intuitive visual feedback to users. The Protocol Simulation Module simulates various network protocols and their behaviors, enabling users to see how these protocols work. The Layer Simulation Module simulates the behavior of physical and data link layer devices, which informs users how these building blocks work within a network.

3.2 Component Interaction

The modules communicate with one another in a systematic way to create an unbroken simulation environment. The User Interface Module accepts input from the user and sends it to the respective simulation module so that user requests are well taken care of. The Network Topology Module establishes a network topology according to user requirements, creating the platform for simulations. The Protocol Simulation Module or Layer Simulation Module carries out operations on the network topology, mimicking network activity based on the chosen protocol or layer. The Visualization Module renders the network topology and protocol activities, providing users with a clear understanding of what is occurring within the simulation. Lastly, the User Interface Module presents the visualization and simulation outputs to the user, completing the cycle of interaction.

Chapter 4

Conclusion

The Network Simulator in this report offers a robust tool for visualization and comprehension of network protocols and behaviors. By providing interactive network layer and protocol simulations, the simulator is a valuable educational tool for students and professionals. Through the use of technologies like NetworkX [2], PyVis [3], and Streamlit [5], the simulator delivers a smooth and interactive user experience, while maintainability and extensibility are guaranteed through its modular design.

4.1 Future Work

In subsequent work, we will improve the simulator by adding Network Layer routing mechanisms, IP addressing, subnetting, and NAT. Transport Layer functions will include TCP (flow control, connection management, congestion control) and UDP. Application Layer protocols such as HTTP, FTP, DNS, and SMTP will also be added for realistic network modeling.

Bibliography

[1] Claude AI. Claude ai. https://claude.ai/, 2025. Accessed: 2025-03-24.

[2] NetworkX. Networkx documentation. https://networkx.org/, 2025.

[3] PyVis. Pyvis documentation. https://pyvis.readthedocs.io/en/latest/, 2025.

[4] sciencedirect. Network simulator. https://www.sciencedirect.com/topics/computer-science/network-simulator, 2025. Accessed : 2025-03-25.

[5] Streamlit. Streamlit documentation. https://docs.streamlit.io/, 2025.

[6] Mir Tasleem. Network simulator. https://github.com/Mir-Tasleem/Network-Simulator, 2025.