

Software for Remote Configuration of Network Devices (SRCND)



STUDENTS

Huzaifa Tahir	2019-CS-114
Zeeshan Ahmad	2019-CS-111
Ata-Ul-Wahab	2019-CS-115

Supervisor

Engr. Muhammad Kashan Basit
Lecturer

DEPARTMENT OF COMPUTER SCIENCE
MNS UNIVERSITY OF ENGINEERING AND TECHNOLOGY, MULTAN
(2023)

Software for Remote Configuration of Network Devices

Authors

Huzaifa Tahir

Zeeshan Ahmad

Ata-ul-Wahab

Student's Registration Numbers

2019-CS-114

2019-CS-111

2019-CS-115

A thesis submitted in partial fulfillment of the requirements for the degree of

B.S Computer Science

Thesis Supervisor:

Eng. M Kashan Basit

Lecturer Computer Science Department

External Examiner Signature: _____

Thesis Supervisor Signature: _____

DEPARTMENT OF COMPUTER SCIENCE
MNS UNIVERSITY OF ENGINEERING AND TECHNOLOGY, MULTAN
(2023)

CERTIFICATE OF APPROVAL

It is certified that the project work titled “**Software for Remote Configuration of Network Devices**” carried out by **Zeeshan Ahmad**(2019cs111), **Huzaifa Tahir**(2019cs114) and **Ata-Ul-Wahab**(2019cs115), under the supervision of “**Eng. M Kashan Basit**” at Muhammad Nawaz Sharif University of Engineering & Technology, Multan. It is fully adequate, in scope and in quality, as a thesis for the degree of B.S Computer Sciences.

Supervisor: -----

Dr./Lecturer/Assistant Professor
Dept. of Computer Science
Muhammad Nawaz Sharif
University of Engineering & Technology, Multan

FYP Convenor: -----

Dr./Lecturer/Assistant Professor
Dept. of Computer Science
Muhammad Nawaz Sharif
University of Engineering & Technology, Multan

External Examiner: -----

Dr./Lecturer/Assistant Professor
Dept. of Computer Science
Muhammad Nawaz Sharif
University of Engineering & Technology, Multan

Head of Department: -----

Dr./Assistant Professor
Dept. of Computer Science
Muhammad Nawaz Sharif
University of Engineering & Technology, Multan

DEPARTMENT OF COMPUTER SCIENCE
MNS UNIVERSITY OF ENGINEERING AND TECHNOLOGY, MULTAN

DECLARATION

It is declared that this is an original piece of my own work, except where otherwise acknowledged in text and references. This work has not been submitted in any form for another degree or diploma at any university or other institution for tertiary education and shall not be submitted by me in future for obtaining any degree from this or any other University or Institution.

Huzaifa Tahir
2019cs114

Zeeshan Ahmad
2019cs111

Ata-Ul-Wahab
2019cs115

August 2023

ACKNOWLEDGEMENT

In the name of **Allah**, the Creator of us all, who is most Beneficent and Merciful. He blessed the human with senses for better understanding of nature and enable all of us to traverse his creations and acquire knowledge.

I thanked Allah Almighty who kept His endless blessings during my research work and helped me in my difficult times.

We thank our **Prophet Muhammad (PBUH)**, who delivered the knowledge to mankind and enable them for better understanding of life.

It was a great opportunity to work under supervision of our respected Supervisor **Engr Kashan Basit**, whose understanding nature, wisdom and humble behavior helped me a lot to complete this work.

Our acknowledgement would be incomplete without considering the help and cooperating behavior of the Head of Department of Computer Science, **Dr. Samina Naz**.

In the end, special thanks to the people in term of moral and emotional support, our family and friends, whose prayers help me to reach this point where I am today and all others who played their significant role in this journey.

ABSTRACT

The software for network devices remote configuration is a powerful and user-friendly application designed to streamline the management and configuration of network infrastructure. With an intuitive graphical user interface (GUI), users can remotely connect to Cisco routers, switches, and MLS devices, eliminating the need for physical access. The software supports configuration templates, error handling, and logging, making it an essential tool for network administrators in various sectors, including IT, telecommunications, and data centers.

TABLE OF CONTENTS

Declaration.....	iii
Acknowledgment.....	iv
Abstract.....	v
Table of Contents.....	vi
List of Figures.....	ix
List of Acronyms/Abbreviations.....	x

Chapter 1

Introduction.....	1
1.1 Overview.....	2
1.2 Statement of Problem.....	2
1.3 Purpose of the research/project.....	2
1.4 Applications of the research.....	3
1.5 Theoretical bases and organization.....	4
1.6 Summary.....	5

Chapter 2

Literature review	7
2.1 Network Device Configuration Challenges	8
2.1.1 Complexity and Prone to Errors	8
2.1.2 Time-Consuming and Inefficient for Large-Scale Configurations:.....	8
2.1.3 Limited Accessibility for Novice Users:.....	8
2.1.4 Vendor-Specific Command Sets:.....	8
2.2) Benefits of Graphical User Interface (GUI) in Network Management.....	8
2.2.1 Intuitive and User-Friendly Design	8
2.2.2 Visual Representation of Network Topology	8
2.2.3 Streamlined Configuration Workflow	8
2.2.4 Interaction and Ease of Use	8
2.3 Importance of Automation in Network Configuration	9
2.3.1 Reducing Human Errors and Improving Consistency	10
2.3.2 Speeding Up Configuration Deployment.....	10
2.3.3 Handling Complex Configurations	10

2.3.4	Standardization and Compliance	<u>10</u>
2.4	Security Considerations in GUI-Based Configurations	<u>11</u>
2.5	Related Technologies	<u>11</u>
2.5.1	Previous CLI-Based Systems	<u>11</u>
2.5.1.1	<i>Introduction of CLI-Based Systems</i>	<u>11</u>
2.5.1.2	<i>Challenges with CLI-Based Systems</i>	<u>12</u>
2.5.1.3	<i>Manual Configuration Complexity</i>	<u>12</u>
2.5.1.4	<i>CLI-Based System as the Precursor</i>	<u>12</u>
2.6	Limitations of CLI-Based Systems and the Need for GUI-Based Systems	<u>12</u>
2.7	Summary	<u>13</u>

Chapter 3

Tools and techniques	<u>10</u>
3.1 Network Simulation Tools	<u>16</u>
3.1.1 GNS3 (Graphical Network Simulator 3)	<u>16</u>
3.1.2 VMware	<u>16</u>
3.2 Secure Connection Tools	<u>16</u>
3.2.1 Secure CRT	<u>16</u>
3.2.1.1 <i>Compatibility</i>	<u>17</u>
3.3 Cisco IOU (IOS on Unix) Images	<u>17</u>
3.4 Software Development Tools	<u>17</u>
3.4.1 PyCharm	<u>17</u>
3.4.1.1 Feature	<u>18</u>
3.4.2 Python	<u>18</u>
3.4.2.1 Readability and Simplicity	<u>19</u>
3.4.2.2 Rapid Development	<u>19</u>
3.4.2.3 Network Automation Proficiency	<u>19</u>
3.4.2.4 Cross-Platform Compatibility	<u>19</u>
3.4.2.5 GUI Development Capabilities	<u>19</u>
3.4.2.6 Seamless Third-Party Integration	<u>19</u>
3.4.3 Tkinter	<u>19</u>
3.5 APIs (Application Programming Interfaces)	<u>21</u>
3.5.1 Netmiko	<u>21</u>
3.5.2 Telnetlib	<u>24</u>

Chapter 4

Implementation and Results	27
4.1 Login	28
4.2 TELNET	29
4.2.1 TELNET Protocol	<u>29</u>
4.3 SSH	30
4.3.1 Benifits of SSH	<u>30</u>
4.4 Software Modules	31
4.4.1 Basic Configuration Module	<u>34</u>
4.4.2 Interfaces Module	<u>37</u>
4.4.3 IP Address Module	39
4.4.4 DHCP Module	<u>40</u>
4.4.5 DNS Module	<u>43</u>
4.4.6 Static Routing Module	<u>45</u>
4.4.7 Dynamic Routnig Module	<u>47</u>
4.4.8 ACL Module	<u>50</u>
4.4.9 VLAN Module	<u>52</u>
4.4.10 VTP Module	<u>54</u>
4.4.11 SwitchPort Module	<u>56</u>
4.4.12 NAT Module	<u>59</u>
4.5 Supported Vendor & Devices	62
4.5.1 Supported Vendor Cisco	<u>62</u>
4.5.2 Supported Devices	<u>62</u>

Chapter 5

Results and Analysis	12
5.1 Features of the "SRCND"	<u>64</u>
5.2 Limitations of the Software	<u>65</u>
5.2.1 Vendor Specificity	<u>65</u>
5.2.2 Lack of Real-Time Monitoring	<u>66</u>
5.2.3 Network Complexity	<u>66</u>
5.3 Conclusion	<u>66</u>
5.4 Future work	<u>67</u>
References	68

LIST OF FIGURES

Number	Page
Figure 4.1 Login page	29
Fig 4.2 SSH Config.....	30
Fig 4.3 Software Main Interface.....	31
Fig 4.4 Basic config> System.....	34
Fig 4.5 Basic Config> User.....	35
Fig 4.6 Device Management.....	36
Fig 4.7 Interface.....	37
Fig 4.8 Virtual Interface.....	38
Fig 4.9 IP Address.	39
Fig 4.10 DHCP Server (a).....	40
Fig 4.10 DHCP Server (b).....	41
Fig 4.11 DNS server.....	41
Fig 4.12 DNS Client.....	44
Fig 4.13 Static Routing(a).....	45
Fig 4.14 Routing Protocol RIP.....	47
Fig 4.15 Routing Protocol EIGRP.....	48
Fig 4.16 Routing Protocol OSPF.....	48
Fig 4.17 ACL.....	50
Fig 4.18 VLAN.....	52
Fig 4.19 VTP.....	54
Fig 4.20 Switch Port Access.....	56
Fig 4.21 Switch Port Trunk.....	57
Fig 4.22 Switch Port Dynamic.....	57
Fig 4.23 Static NAT.....	59
Fig 4.24 Dynamic NAT.....	60
Fig 4.25 PAT.....	61

LIST OF ACRONYMS

MNS:	Muhammad Nawaz Sharif
UET:	University of Engineering and TechnologyS
FYP:	Final Year Project
B.S:	Bachelors of Sciences
HOD:	Head of Department
SRCND:	Software for Remote Configuration of Network Devices
SSH:	Secure Shell
Telnet:	Telecommunication Network
IP:	Internet Protocol
DHCP:	Dynamic Host Configuration Protocol
DNS:	Domain Name System
VLAN:	Virtual Local Area Network
VTP:	VLAN Trunking Protocol
SVI:	Switched Virtual Interface
ROS:	Routing and Operating System
RIP:	Routing Information Protocol
EIGRP:	Enhanced Interior Gateway Routing Protocol
OSPF:	Open Shortest Path First
ACL:	Access Control List
EnPassword:	Enable Password
MOTD:	Message of the Day
NAT:	Network Address Translation
SNAT:	Static Network Address Translation

DNAT: Dynamic Network Address Translation

PAT: Port Address Translation

CHAPTER 1
INTRODUCTION

INTRODUCTION

1.1) Overview

The "Software for Remote Configuration of Network Devices" project is a pioneering endeavor that seeks to streamline the intricate process of configuring networking devices, such as Cisco routers, switches, and MLS, by introducing a user-friendly graphical user interface (GUI). The primary objective of this project is to automate the execution of complex networking commands through the backend, alleviating the need for users to memorize and manually input command-line instructions.

1.2) Statement of Problem

Network device configuration poses significant challenges to network administrators and users, owing to the following complexities:

- **Learning Curve:**
Configuring networking devices involves an extensive array of commands, each with specific syntax and options. This steep learning curve can be overwhelming, particularly for newcomers to networking.
- **Error-Prone:**
Manually entering command-line instructions is susceptible to human errors. Mistakes during configuration can lead to misconfigurations that disrupt network operations and jeopardize network security.
- **Efficiency Concerns:**
Configuring multiple devices individually using the command-line interface can be time-consuming and inefficient, especially in large-scale network setups.

The "Software for Remote Configuration of Network Devices" project aims to address these challenges by providing a novel solution that empowers both novice and experienced administrators to configure networking devices more effectively.

1.3) Purpose of the Research/Project

The core purpose of this research project is to conceive and develop an innovative software tool that simplifies the process of configuring network devices. By implementing a user-friendly GUI, users will interact with network devices through a familiar interface, abstracting

the underlying complexity of command-line configurations. This approach endeavors to achieve the following key objectives:

- **Ease of Use:**

The software will feature an intuitive GUI, guiding users through the configuration process with clear and user-friendly elements, such as buttons, drop-down menus, and input fields.

- **Accessibility:**

The project aims to make network device configuration accessible to novice users with minimal technical expertise. By eliminating the need for extensive training in command-line interfaces, users can perform configurations more confidently.

- **Automation:**

The software will automate complex backend commands, enabling the efficient execution of configuration tasks. This automation will not only save time for experienced administrators but also ensure accuracy and consistency in configurations.

- **Enhanced Network Management:**

By providing an accessible and efficient solution to configure multiple network devices, the project seeks to streamline network management processes. This will result in optimized network performance and improved resource utilization.

1.4) Applications of the Research

The research project's applications extend to diverse user groups and network scenarios, including:

- **Empowering Novice Users:**

The user-friendly GUI will empower users with limited technical knowledge to configure network devices confidently. Democratizing network management in this manner will lead to increased participation and enhanced operational efficiency.

- **Enhancing Efficiency for Experienced Administrators:**

Even experienced administrators will benefit from the software's automation capabilities, as it simplifies the process of configuring devices and allows them to focus on higher-level network tasks.

- **Streamlining Large-Scale Network Management**

In enterprise environments managing numerous network devices, the software's ability to handle multiple devices efficiently will significantly reduce administrative overhead, resulting in improved network performance.

1.5) Theoretical Bases and Organization

The theoretical foundations of the project revolve around network device configuration principles, GUI design concepts, and automation techniques:

- **Network Device Configuration Principles:**

An in-depth understanding of configuring different networking devices, including routers, switches, and MLS, is essential to develop an effective software solution.

- **GUI Design Concepts:**

Designing an intuitive and visually appealing graphical user interface that abstracts complex commands into user-friendly elements is crucial to the software's success.

- **Automation Techniques:**

Implementing a robust backend engine that translates GUI inputs into the appropriate networking commands will automate the configuration process and minimize manual intervention.

The project will be organized into the following stages:

1. **Requirements Gathering:** Identify the needs and expectations of potential users, including both novices and experienced administrators, to define the software's functionalities and design.
2. **GUI Design and Development:** Create an interactive and user-friendly GUI that intuitively represents various networking tasks and simplifies configuration procedures.
3. **Backend Implementation:** Develop the backend engine responsible for interpreting GUI inputs and executing the corresponding networking commands based on the selected device and configuration.
4. **Device Compatibility Testing:** Conduct extensive testing to ensure the software is compatible with a wide range of networking devices, making it applicable to various network setups.

5. **Security Implementation:** Prioritize security by implementing authentication and authorization mechanisms to safeguard sensitive network configurations from unauthorized access.
6. **Documentation and Support:** Provide comprehensive documentation, user guides, and troubleshooting resources to assist users in understanding and effectively using the software.

1.6) Who Will Use the Software and Their Benefits:

1. Network Administrators:

The software empowers network administrators by simplifying the configuration process. It eliminates the need to memorize complex command-line instructions, enabling administrators to efficiently manage network devices with greater accuracy and reduced errors.

2. Network Engineers:

Experienced network engineers can leverage the software to expedite routine configuration tasks. Its modular approach and advanced configuration capabilities provide a streamlined interface to manage complex networking setups.

3. Individuals with Basic Networking Knowledge:

Even those with limited networking expertise can benefit from the software. Its user-friendly graphical interface and simplified configuration options make it accessible to individuals who possess fundamental networking knowledge.

4. Organizations with Limited Resources:

Small to medium-sized organizations lacking the budget to hire highly skilled network engineers can utilize the software. It democratizes network management, enabling organizations to maintain their network devices without extensive technical staff.

1.7) Summary

In summary, the "Software for Remote Configuration of Network Devices" project endeavors to revolutionize network management by offering a GUI-based software tool that simplifies and automates the configuration of networking devices. By abstracting complex commands into an intuitive user interface, the project seeks to enhance accessibility for novice users and improve efficiency for experienced administrators. The project's theoretical foundations and

organized development process will ensure a successful outcome, empowering users to configure network devices efficiently and effectively.

CHAPTER 2

Literature Review

Literature Review

2.1) Network Device Configuration Challenges

Network device configuration is a critical aspect of network management, ensuring that routers, switches, and other devices operate optimally within a network. Traditionally, network administrators relied on Command-Line Interface (CLI) commands to configure devices. However, this approach presents several challenges:

2.1.1) Complexity and Prone to Errors:

- CLI commands often have complex syntax and numerous parameters, making them challenging to master.
- Manual entry of commands is error-prone, leading to misconfigurations and potential network disruptions.

2.1.2) Time-Consuming and Inefficient for Large-Scale Configurations:

- Configuring multiple devices individually using CLI is time-consuming and inefficient, especially in large-scale networks.
- The repetitive nature of manual configurations increases the risk of inconsistencies and delays.

2.1.3) Limited Accessibility for Novice Users:

- CLI configurations require specialized technical knowledge, making it difficult for less experienced users to manage devices effectively.
- Novice users may struggle with memorizing and understanding the vast array of commands.

2.1.4) Vendor-Specific Command Sets:

- Different network device vendors often have their own set of CLI commands, leading to vendor lock-in and complexity in managing heterogeneous networks.
- Administrators need to learn and switch between vendor-specific commands when working with different devices.

2.2) Benefits of Graphical User Interface (GUI) in Network Management

In network management, a Graphical User Interface (GUI) is a user-friendly visual interface that allows administrators to interact with network devices through buttons, icons, and menus instead of using complex command lines.

2.2.1) Intuitive and User-Friendly Design:

- GUI interfaces are designed with simplicity in mind, making it easy for users to navigate and understand the network configuration process.
- Users can perform actions by clicking buttons and selecting options, which reduces the need for memorizing and typing commands.

2.2.2) Visual Representation of Network Topology:

- GUIs often provide graphical representations of the network's topology, allowing administrators to see how devices are connected and how data flows through the network.
- This visual representation helps in understanding the network's structure and identifying potential bottlenecks or points of failure.

2.2.3) Streamlined Configuration Workflow:

- GUIs guide users through the configuration process step by step, ensuring that necessary settings are not overlooked.
- Users can easily view and edit configurations, making it convenient to apply changes and verify settings.

2.2.4) Interaction and Ease of Use:

- GUIs offer interactive features, such as drag-and-drop functionality, sliders, and tooltips, enhancing the overall user experience.
- Users can customize the interface according to their preferences, making it more user-centric.

2.3) Importance of Automation in Network Configuration

Automation in network configuration refers to the use of software tools and scripts to perform repetitive and complex tasks automatically. It plays a vital role in enhancing network management by reducing manual efforts and improving efficiency.

2.3.1) Reducing Human Errors and Improving Consistency:

- Manual configurations are prone to human errors, leading to network downtime and security vulnerabilities.
- Automation ensures consistent and error-free configurations, as tasks are performed precisely according to predefined rules.

2.3.2) Speeding Up Configuration Deployment:

- Automating configuration tasks significantly reduces the time required to deploy changes across multiple devices.
- Administrators can update configurations simultaneously on multiple devices, saving valuable time and effort.

2.3.3) Handling Complex Configurations:

- Networks often have complex configurations involving numerous devices and settings.
- Automation can efficiently handle these complexities, making it easier to manage large-scale and intricate networks.

2.3.4) Standardization and Compliance:

- Automation allows administrators to enforce standardized configurations, ensuring adherence to best practices and compliance requirements.
- This reduces the risk of misconfigurations and enhances network security and stability.
- By incorporating automation techniques into the "Software for Remote Configuration of Network Devices," the project aims to simplify the configuration process, eliminate manual errors, and expedite network changes. The combination of GUI-based interaction and backend automation will empower administrators to manage network devices efficiently and with greater reliability.

2.4) Security Considerations in GUI-Based Configurations

By incorporating the SSH protocol into the "Software for Remote Configuration of Network Devices," we ensure a secure and encrypted communication channel between the GUI interface and the network devices. SSH provides a strong level of security and authentication, mitigating the risk of unauthorized access and data tampering during configuration tasks. The implementation of SSH in conjunction with other security measures will ensure that the software maintains the highest standards of security for both the administrators and the network infrastructure.

2.5) Related Technologies

In this section, we explore related technologies that can enhance the development and functionality of the "Software for Remote Configuration of Network Devices" project. While our project focuses on GUI-based network configuration automation, it is essential to acknowledge the previous CLI-based systems, which laid the groundwork for our current work.

2.5.1) Previous CLI-Based Systems

Before the advent of GUI-based configuration tools and automation frameworks, network device configuration was primarily performed through Command-Line Interface (CLI) commands. CLI-based systems served as the foundation for network management and played a significant role in the evolution of network configuration techniques.

2.5.1.1) Introduction of CLI-Based Systems:

- CLI-based systems were among the earliest methods used to configure network devices, dating back to the early days of networking.
- Administrators interacted with network devices by entering commands in a text-based interface, which was the standard approach for configuring routers, switches, and other network equipment.

2.5.1.2) Challenges with CLI-Based Systems:

- Despite their significance, CLI-based systems presented challenges for administrators, especially those new to networking or lacking extensive technical expertise.

- Memorizing numerous commands and their syntax proved to be a barrier for novice users, leading to potential errors in configurations.

2.5.1.3) Manual Configuration Complexity:

- CLI configurations often involved complex sequences of commands to set up and manage network devices.
- Administrators needed to enter commands for individual devices, making large-scale configurations time-consuming and prone to inconsistencies.

2.5.1.4) CLI-Based System as the Precursor:

- It is important to recognize that the CLI-based systems laid the groundwork for modern network management practices.
- These early systems demonstrated the need for more user-friendly and efficient approaches to network configuration.

2.6) Limitations of CLI-Based Systems and the Need for GUI-Based Systems

CLI-based systems, while fundamental in the evolution of network management, present several limitations that have driven the need for more user-friendly GUI-based systems. The limitations of CLI-based systems include:

2.6.1) Steep Learning Curve:

CLI commands often have complex syntax and multiple parameters, making them challenging to master, especially for novice users or those without extensive technical knowledge. Learning and memorizing the vast array of commands can be time-consuming and error-prone.

2.6.2) Error-Prone Configuration:

Due to the manual nature of CLI configuration, human errors are common. A misplaced command or incorrect parameter can lead to misconfigurations, resulting in network disruptions and security vulnerabilities.

2.6.3) Time-Consuming Configuration Process:

Configuring multiple devices individually using CLI commands can be tedious and time-consuming, particularly in large-scale networks. Repetitive tasks can lead to inconsistencies and delays in deploying changes across the network.

2.6.4) Vendor-Specific Command Sets:

Different network device vendors often have their own proprietary set of CLI commands, leading to vendor lock-in and complexity when managing heterogeneous networks. Administrators need to switch between vendor-specific commands for each device.

2.6.5) Limited Accessibility for Novice Users:

CLI-based systems demand specialized technical expertise, making them less accessible to novice users or administrators with limited experience in networking. This hinders their ability to efficiently manage network devices.

2.6.6) Lack of Visual Representation:

CLI interfaces lack visual representations of network topology, making it challenging for administrators to visualize the network's structure and identify potential connectivity issues or performance bottlenecks.

2.6.7) Repetitive Configuration Tasks:

Configuring multiple devices with the same settings using CLI commands requires repetitive manual entry, increasing the likelihood of errors and time wastage.

2.7) Summary:

Chapter 2 provided a comprehensive literature review on network device configuration challenges and the evolution of network management techniques. It explored the limitations of Command-Line Interface (CLI) based systems, which include complexity, error-proneness, time-consuming manual configuration, and limited accessibility for novice users.

To address these limitations, Graphical User Interface (GUI) based systems emerged, offering user-friendly and visual interfaces that simplify configuration workflows. GUI interfaces provide a more intuitive way to interact with network devices, offering visual representations of network topology and streamlining configuration tasks.

Moreover, automation techniques, such as Ansible and Puppet, further enhanced network management by reducing human errors, handling complex configurations, and enforcing standardization. Integrating GUI with automation frameworks created a powerful synergy, enabling administrators to efficiently manage multiple devices simultaneously.

The literature review emphasized the significance of security considerations in GUI-based configuration tools. Secure authentication, data encryption, and access control mechanisms ensure the confidentiality and integrity of sensitive network configurations.

In conclusion, the transition from CLI to GUI-based configuration tools and automation frameworks marked a significant advancement in network management, enabling more accessible and efficient network device configuration. The "Software for Remote Configuration of Network Devices" project builds upon this progress, aiming to provide a secure, user-friendly, and automated solution for network administrators.

CHAPTER 3

Tools and Technologies Used in Development

Tools and Technologies

This chapter provides an in-depth overview of the tools and technologies utilized during the development of the "Software for Remote Configuration of Network Devices." The selection of appropriate tools and technologies plays a pivotal role in ensuring the project's success by enabling efficient development, secure connections, and user-friendly interfaces.

3.1) Network Simulation Tools

In the initial stages of development, network simulation tools were employed to create virtualized network environments for testing and validation. These tools included:

3.1.1) GNS3 (Graphical Network Simulator 3):

GNS3 allowed the creation of complex network topologies using virtualized routers, switches, and other network devices. It provided a platform to simulate and test network configurations before deploying them on actual devices.

3.1.2) VMware:

VMware was used to set up virtual machines for hosting different operating systems, allowing for comprehensive testing and development in isolated environments.

3.2) Secure Connection Tools

Establishing secure connections for configuring network devices is a paramount consideration.

The following tools facilitated secure connections:

3.2.1) Secure CRT:

- It was first released in the autumn of 1995 by VanDyke Software. Originally released as a premium version of CRT with support for SSH encryption, Secure CRT later absorbed the CRT product entirely. The program is part of a line of networking software which includes Secure FX, a file transfer client with SSL capability, and VShell, an SSH server.
- Secure CRT and Secure FX can be started from within each other and use a combined host information list. A separately-sold pack of command-line tools (e.g., scp, modeled after the Unix command of the same name) for use with VShell is also sold by the company. All offerings are commercial ware.

3.2.1.1) Compatibility

Secure CRT runs on Windows XP, Windows Vista and Windows 7, Windows 8, Windows 10 and Windows 11. It also runs on the Windows Server series of operating systems.[10] For Windows Vista and later, a 64-bit version is available for download. Secure CRT provided secure remote access to network devices using protocols like SSH and Telnet. It ensured encrypted communication and secure authentication, safeguarding sensitive configuration data.

3.3) Cisco IOU (IOS on Unix) Images:

Cisco IOU images enabled the emulation of Cisco network devices for testing purposes. This allowed for safe and controlled experimentation without risking the integrity of physical devices.

3.4) Software Development Tools:

The development of the software's user interface and backend functionality relied on a range of software development tools:

3.4.1) PyCharm:

PyCharm, an Integrated Development Environment (IDE), was used for writing, testing, and debugging Python code. Its user-friendly interface and features enhanced code productivity.

PyCharm is an integrated development environment (IDE) used for programming in Python. It provides code analysis, a graphical debugger, an integrated unit tester, integration with version control systems, and supports web development with Django. PyCharm is developed by the Czech company JetBrains.

It is cross-platform, working on Microsoft Windows, macOS and Linux. PyCharm has a Professional Edition, released under a proprietary license and a Community Edition released under the Apache License. PyCharm Community Edition is less extensive than the Professional Edition.

3.4.1.1) Feature:

1. **Intelligent Code Editor:** PyCharm provides an intelligent code editor with features like code completion, code analysis, and auto-suggestions. It helps developers write code faster and with fewer errors.
2. **Code Navigation and Search:** With powerful navigation tools, PyCharm allows developers to quickly jump between different parts of the code, search for specific functions or classes, and navigate through the project structure.
3. **Integrated Debugger:** PyCharm's debugger helps identify and fix issues in the code. Developers can set breakpoints, inspect variables, step through the code, and understand the flow of execution.
4. **Version Control Integration:** PyCharm seamlessly integrates with popular version control systems like Git, Mercurial, and Subversion. This allows developers to manage code changes, commits, branches, and merges directly within the IDE.
5. **Project Management:** Developers can manage and organize their projects effectively using PyCharm's project management features. It supports virtual environments, package management, and project-specific settings.
6. **Intelligent Code Analysis:** PyCharm's static code analysis highlights potential errors, suggests improvements, and adheres to Python's best practices. It helps maintain code quality and consistency.

3.4.2) Python:

The software was developed using the Python programming language due to its versatility, extensive libraries, and community support.

Python was chosen as the programming language for developing the "Software for Remote Configuration of Network Devices" due to its versatility, ease of use, and strong capabilities in network automation and scripting. The decision to use Python was driven by several key factors:

3.4.2.1) Readability and Simplicity:

Python's clean and readable syntax makes it an ideal choice for developers of all levels. The code is easily understandable, which is crucial when working on collaborative projects or maintaining the software over time.

3.4.2.2) Rapid Development:

Python's simplicity accelerates the development process, allowing for quicker implementation of features. The software's capabilities were efficiently translated into code, leading to reduced development time and faster deployment.

3.4.2.3) Network Automation Proficiency:

Python's prominence in network automation is underscored by its ability to streamline scripting and automation tasks. Leveraging Python, the software orchestrates complex configurations with ease, enhancing efficiency and accuracy.

3.4.2.4) Cross-Platform Compatibility:

Python's cross-platform compatibility ensures the software functions seamlessly across various operating systems. This compatibility maximizes accessibility for users, regardless of their preferred environment.

3.4.2.5) GUI Development Capabilities:

Python, coupled with Tkinter, excels in graphical user interface (GUI) development. This combination enabled the creation of an intuitive and visually appealing interface, enhancing user interaction.

3.4.2.6) Seamless Third-Party Integration:

Python's integration capabilities enable effective communication with external systems, APIs, and languages. This versatility ensures smooth interactions with diverse networking devices and protocols.

3.4.3) Tkinter:

Tkinter, a Python GUI library, enabled the creation of the user-friendly graphical interface of the software. It provided the framework for designing windows, buttons, forms, and other interactive elements.

Tkinter is a standard graphical user interface (GUI) library in Python that allows developers to create user interfaces for desktop applications. It provides a set of tools and widgets that enable the creation of windows, buttons, forms, text boxes, and various interactive elements in a user-friendly manner.

In our software project, we choose to use Tkinter for the user interface design. Here's why Tkinter was selected and its benefits for our software

3.4.3.1) Benefits and why we used Thinker

Here are the benefits of of using thinker

i. *Ease of Use:*

Tkinter is user-friendly and relatively easy to learn, making it suitable for developers with varying levels of experience. Its simple and intuitive syntax allows you to create interactive interfaces without requiring extensive coding skills.

ii. *Cross-Platform Compatibility:*

Tkinter is a cross-platform library, which means that the user interface you design using Tkinter will work consistently on different operating systems such as Windows, macOS, and Linux.

iii. *Wide Range of Widgets:*

Tkinter offers a wide range of built-in widgets, including buttons, labels, text boxes, radio buttons, check buttons, and more. This extensive widget library enables you to create diverse and interactive user interfaces.

iv. *Customization:*

Tkinter allows you to customize the appearance of widgets by specifying attributes such as colors, fonts, and sizes. This customization capability enables you to create visually appealing and coherent user interfaces.

v. *Event Handling:*

Tkinter provides event-driven programming, allowing you to associate functions with user actions like button clicks or menu selections. This enables you to create responsive interfaces that react to user interactions.

vi. *Integration with Python:*

Since Tkinter is part of the standard Python library, you don't need to install any additional packages to use it. It's readily available for use once you have Python installed.

vii. *Rapid Development:*

Tkinter's simplicity and ready-to-use widgets facilitate rapid development of graphical interfaces, saving development time and effort.

viii. Community Support:

Tkinter has a large and active community of developers, which means that you can find a wealth of tutorials, documentation, and resources to aid in your development process. responsive interfaces that react to user interactions.

3.5) APIs (Application Programming Interfaces)

APIs played a pivotal role in enhancing the software's capabilities by enabling seamless interaction with network devices:

3.5.1) Netmiko

Netmiko is a Python library that simplifies the automation and management of network devices by providing a unified interface to interact with various network devices, such as routers and switches, using SSH (Secure Shell) or Telnet protocols. It abstracts the complexity of handling different device types and protocols, allowing developers to focus on automating network configurations and tasks.

3.5.1.1) Reasons for Using Netmiko in our software

- i. **Vendor Agnostic Automation:**
Netmiko abstracts the differences in command syntax and behavior across various network device vendors, such as Cisco, Juniper, and more. This allows your software to interact with devices from different vendors using a consistent set of Python commands, streamlining automation efforts.
- ii. **Unified Interface:**
Netmiko provides a standardized API for automating network tasks. This means you don't need to develop separate scripts or functions for each device type, reducing complexity and increasing code maintainability.
- iii. **SSH and Telnet Support:**
Netmiko supports both SSH and Telnet protocols for connecting to network devices. SSH ensures secure communication by encrypting data, while Telnet is suitable for non-sensitive operations. This flexibility enables your software to accommodate a wide range of device capabilities.
- iv. **Configuration Consistency:**
Netmiko enables you to automate configuration changes across multiple devices, ensuring consistency and reducing the risk of configuration errors caused by manual inputs.

- v. **Automation Efficiency:**
With Netmiko, you can automate repetitive tasks like device configurations, backups, and updates. This improves efficiency and minimizes the potential for human errors associated with manual operations.
 - vi. **Scriptable Interactions:**
Netmiko allows you to script complex interactions with network devices. For example, you can create scripts to configure VLANs, set up routing protocols, or perform network diagnostics across multiple devices simultaneously.
 - vii. **Error Handling:**
Netmiko includes robust error handling mechanisms, allowing your software to gracefully handle connection errors, timeouts, and exceptions that may occur during device interactions.
 - viii. **Rapid Development:**
By utilizing Netmiko's pre-built functions, you can expedite the development process and focus on higher-level logic and features of our software.
 - ix. **Remote Configuration:**
Netmiko facilitates remote configuration tasks, allowing your software to configure devices from a central location, eliminating the need for manual configuration at each device.
 - x. **Enhanced Network Management:**
Netmiko empowers your software to perform a wide range of network management tasks, such as updating access control lists, modifying routing tables, and configuring network services.

In our software project, Netmiko plays a crucial role in enhancing the functionality of the "Software for Remote Configuration of Network Devices"
- ### 3.5.1.2) Benefits and advantage
- i. **Unified Interface:**

Netmiko abstracts the differences between different network device types and vendor-specific command variations. This means you can use a consistent set of Python commands to interact with devices from various vendors, like Cisco, Juniper, and more.

ii. **Automation:**

Netmiko enables you to automate network device configuration tasks, reducing the manual effort required for repetitive tasks. This can significantly improve operational efficiency and reduce the likelihood of human errors.

iii. **Secure Communication:**

Netmiko supports SSH and Telnet, allowing you to establish secure connections to network devices. SSH provides encrypted communication, ensuring the confidentiality and integrity of sensitive configuration data.

iv. **Scripting and Batch Processing:**

With Netmiko, you can write scripts to perform a series of commands across multiple devices, which is particularly useful for batch processing and device updates.

v. **Device Agnostic:**

You can use Netmiko to interact with a wide range of network devices, regardless of the underlying operating system or vendor. This versatility is beneficial in heterogeneous network environments.

vi. **Error Handling:**

Netmiko includes error handling mechanisms that help you handle exceptions and errors that might occur during interactions with network devices. This enhances the robustness and reliability of your automation scripts.

vii. **Consistency and Standardization:**

By using Netmiko, you can enforce consistent configurations across multiple devices, promoting standardization and reducing configuration discrepancies.

viii. **Flexibility:**

Netmiko provides the flexibility to send both single commands and command lists, enabling you to perform a variety of configuration tasks.

ix. **Community Support:**

Netmiko has a strong community of users and developers, which means you can find extensive documentation, tutorials, and resources to support your development efforts. In our software project, integrating Netmiko empowers your software to communicate with network devices securely, automate configuration tasks, and provide a seamless user experience. It plays a pivotal role in achieving the automation and remote configuration objectives of your project.

3.5.2) Telnetlib

Telnetlib is a Python library that enables communication with network devices using the Telnet protocol. While Telnet is less secure compared to SSH (Secure Shell), it can still be used for non-sensitive operations and interactions with devices that only support Telnet. In your project, Telnetlib might have been chosen as an alternative or as an additional option for device communication, providing flexibility based on user and device requirements.

3.5.2.1) Reasons for Using Telnetlib:

i. Compatibility:

Some older or less advanced network devices might only support Telnet for remote communication. Using Telnetlib allows your software to interact with a broader range of devices, catering to a wider user base.

ii. Non-Critical Operations:

While Telnet is not recommended for sensitive data transmission due to its lack of encryption, it can be suitable for non-critical operations such as retrieving device information or performing simple tasks that don't involve confidential data.

iii. Testing and Development:

During the development and testing phases, Telnetlib can be used for experimentation and debugging purposes. It provides a way to interact with devices without requiring SSH access.

3.5.2.2) How Telnetlib Works:

Importing the Library:

To use Telnetlib, you need to import the library in your Python script using the `import telnetlib` statement.

i. Establishing a Connection:

You initiate a Telnet connection to a network device by creating an instance of the Telnet class and specifying the device's IP address and port.

ii. Authentication:

You can provide authentication credentials to the device using the `read_until` and `write` methods. However, remember that the credentials are sent in plain text, so using Telnet for sensitive operations is not recommended.

iii. Interacting with the Device:

Once connected, you can use the `read_until` and `write` methods to send commands to the device and receive its responses.

iv. Closing the Connection:

After completing the desired operations, you should close the Telnet connection using the `close` method.

3.5.2.3) Benefits and Considerations:**i. Flexibility:**

Telnetlib provides an alternative communication method, allowing your software to interact with devices that support Telnet.

ii. Testing and Debugging:

Telnetlib can be valuable during development and testing, enabling you to validate your software's functionality without requiring SSH access.

iii. Limited Security:

Telnet communication is not secure, as data is transmitted in plain text. As such, it's best suited for non-sensitive operations.

iv. Compatibility:

Some legacy devices might not support SSH, making Telnetlib a practical solution for communicating with them.

It's important to note that while Telnetlib offers compatibility benefits, using SSH whenever possible is recommended due to its enhanced security features. When using Telnetlib, exercise caution and ensure that sensitive data or operations are not performed over Telnet connections.

CHAPTER 4

Implementation and Results

Implementation and results

4.1) LOGIN:

First interaction of user with LOGIN Panel. Here, Software need IP ADDRESS, Username, Password, Privilege configuration mode password of the device, Select the Protocol that you configured on your device to access it remotely. If you are configure a new device you need to use the commands mentioned below.

Your default credentials are:

- Connect To: IP Address of your device
- Username: abc
- Password: 123
- EnPassword: 123
- Telnet

```
conf
int
!int
no
!ip      add      192.168.33.150      255.255.255.0
ip      add      dhcp
no      shut
exit

username      abc      password      123
enable      password      123

line      vty      0      4
transport      input      all
login      local
password      123
exit
```

Here, you must be careful while selecting the interfaces and assign IP address to your device for the remote access. ! mark is use in the line for comment.

Default, configuration mentioned Above. For the ease of User While Login You just need to enter IP address and select the protocol to access the device. All other entries already filled with default Values.

4.2) TELNET:

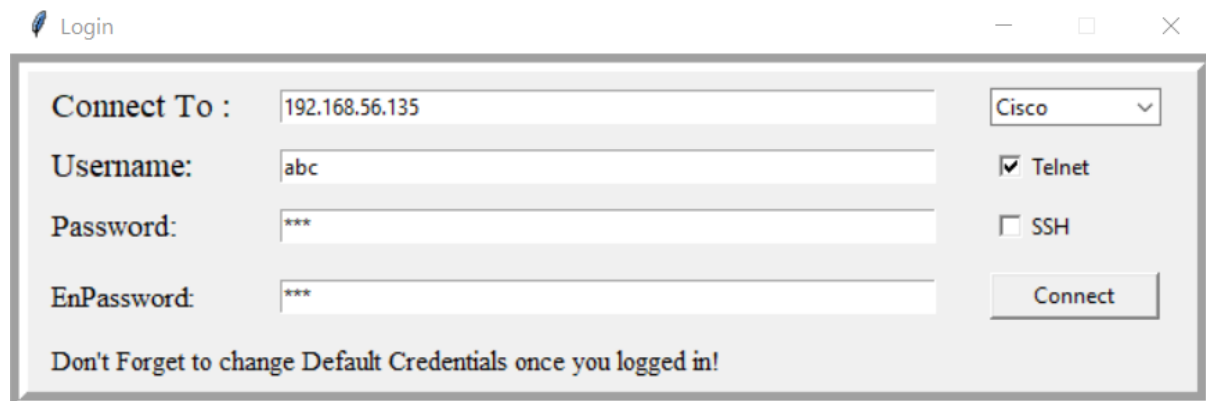


Fig 4.1 Login page

The "Telnet" protocol is a network communication protocol that allows remote access to network devices over a network connection. While it was widely used in the past for remote device management, Telnet is considered less secure compared to modern alternatives like SSH (Secure Shell) due to its lack of encryption. However, understanding the basics of Telnet can provide insights into its historical significance and its potential use cases in certain scenarios.

4.2.1) Telnet Protocol:

- Remote Access:

Telnet enables users to establish a remote terminal session with a network device, such as a router or switch, over a TCP/IP network.

- Clear Text Communication:

One of the main drawbacks of Telnet is that all communication, including usernames, passwords, and commands, is transmitted in clear text. This lack of encryption exposes sensitive information to potential eavesdropping and unauthorized access.

- Port Number:

Telnet typically operates on port 23. A client establishes a connection to the Telnet server (network device) on port 23 to initiate the communication.

4.3) SSH:

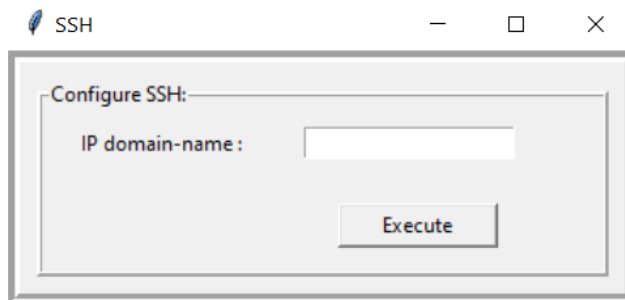


Fig 4.2 SSH Config

SSH (Secure Shell) is a widely used cryptographic network protocol that provides a secure way to access and manage network devices remotely over an unsecured network. Unlike protocols like Telnet that transmit data in clear text, SSH encrypts the communication between the client and the server, ensuring confidentiality, integrity, and authentication.

4.3.1) Benefits of SSH:

1. **Security:** One of the primary advantages of SSH is its strong security features. It uses encryption to protect the data being transmitted between the client and the server, making it significantly more secure than protocols like Telnet.
2. **Encryption:** SSH employs various encryption algorithms to ensure that sensitive information, including usernames, passwords, and commands, remains confidential and protected from eavesdropping and unauthorized access.
3. **Authentication:** SSH supports multiple methods of authentication, including password-based authentication and public key authentication. Public key authentication, in particular, enhances security by eliminating the need to transmit passwords over the network.
4. **Data Integrity:** SSH ensures that the data exchanged between the client and the server is not tampered with during transmission. This prevents attackers from modifying the data in transit.
5. **Port Forwarding:** SSH allows for secure port forwarding, also known as SSH tunneling. This feature enables users to securely access services on remote servers that might not be directly accessible from the local network.

In our "Software for Remote Configuration of Network Devices," integrating SSH as the primary protocol for device communication enhances security and ensures that sensitive configuration tasks are performed in a secure and encrypted manner. The benefits of SSH align with the goal of your software to provide a secure and user-friendly solution for remote configuration and management of network devices.

4.4) SOFTWARE Modules

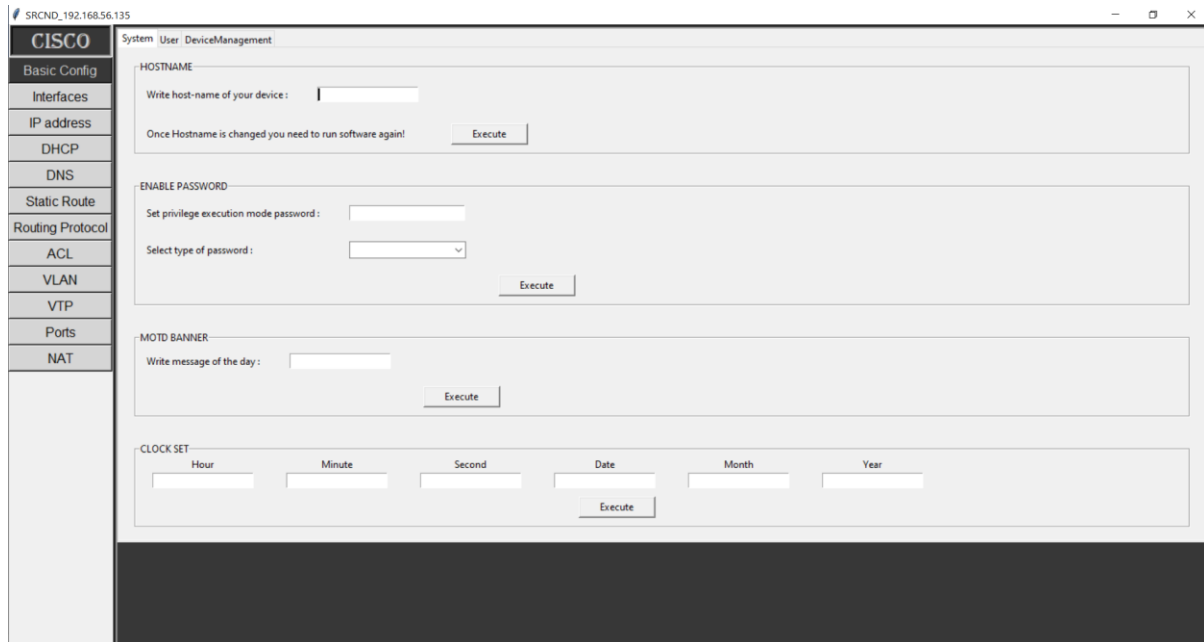


Fig 4.3 Software Main Interface

In this software we have a variety of modules such that:

4.4.1 Basic Configuration

- a) System
 - i) Hostname
 - ii) Enable Password
 - iii) MOTD banner
 - iv) Clock set
- b) User
 - i) Add user
 - ii) Remove user
- c) Device Management
 - i) Save
 - ii) Backup
 - iii) Factory Reset

4.4.2 Interfaces

- a) Interface Configuration

- i) Interface behavior
 - ii) Serial interface
 - iii) Default interface
 - iv) Switchport selection
- b) Virtual Interface
 - i) Loopback Interface
 - ii) Sub-interface
 - iii) SVI
- 4.4.3 IP address
- 4.4.4 DHCP
 - a) DHCP Server
 - i) Server Enable
 - ii) Server disable
 - b) DHCP Client
- 4.4.5 DNS
 - a) DNS SERVER
 - b) DNS Client
- 4.4.6 Static Routing
 - a) Static route
 - b) Default route
- 4.4.7 Dynamic Routing
 - a) RIP
 - b) EIGRP
 - c) OSPF
- 4.4.8 ACL
 - a) Define Access list globally
 - b) Apply Access list on interface
- 4.4.9 VLAN
- 4.4.10 VTP
- 4.4.11 Switchports
 - a) Access
 - i) Data

- ii) voice
- b) Trunk
 - i) Trunk with allowed VLAN
- c) Dynamic
 - i) Desirable
 - ii) Auto

4.4.12 NAT

- a) Static NAT
 - i) Inside & Outside Interfaces
 - ii) Public & Private IP addresses
- b) Dynamic NAT
 - i) Inside & Outside Interfaces
 - ii) Public & Private IP addresses
 - iii) Apply Access list
- c) PAT
 - i) Inside & Outside Interfaces
 - ii) Public & Private IP addresses
 - iii) Apply Access list

4.4.1) Basic Configuration Module

The "Basic Configuration" module serves as the foundation for initiating and managing crucial aspects of network devices using the "Software for Remote Configuration of Network Devices." This module enables users to establish essential device settings, enhance security measures, and ensure effective device management.

Fig 4.4 Basic config > System

a) System:

i) Hostname:

The hostname is a unique identifier for the network device within the network. Users can effortlessly set or modify the hostname to reflect the device's purpose or location. This helps network administrators and engineers recognize devices easily.

ii) Enable Password:

The enable password is a critical security component that safeguards access to the device's privileged mode. By setting an enable password, only authorized users with the correct password can access and configure privileged settings.

iii) MOTD Banner:

The MOTD (Message of the Day) banner allows administrators to display custom messages or warnings to users who access the device. This feature can convey important announcements, policies, or legal disclaimers.

iv) Clock set:

Ensuring accurate time synchronization across network devices is essential for logging and troubleshooting. The clock set feature enables users to synchronize the device's clock with a central time source, ensuring consistent and accurate timestamps.

b) User:

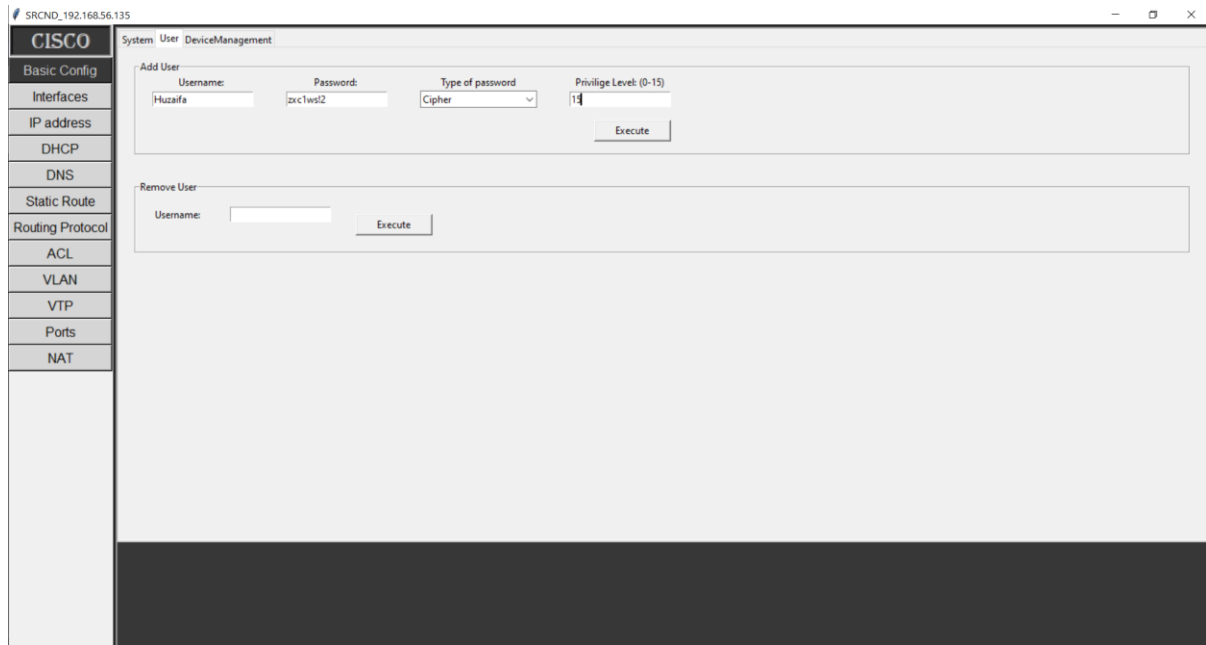


Fig 4.5 Basic Config > User.

i) Add User:

This functionality empowers administrators to create new user accounts on the device. Users can be assigned specific privileges and authentication credentials. The addition of user accounts contributes to effective access control and accountability.

ii) Remove User:

Network administrators have the capability to remove user accounts that are no longer required. This ensures that the list of authorized users remains current and that access is revoked for users who no longer require it.

c) Device Management:

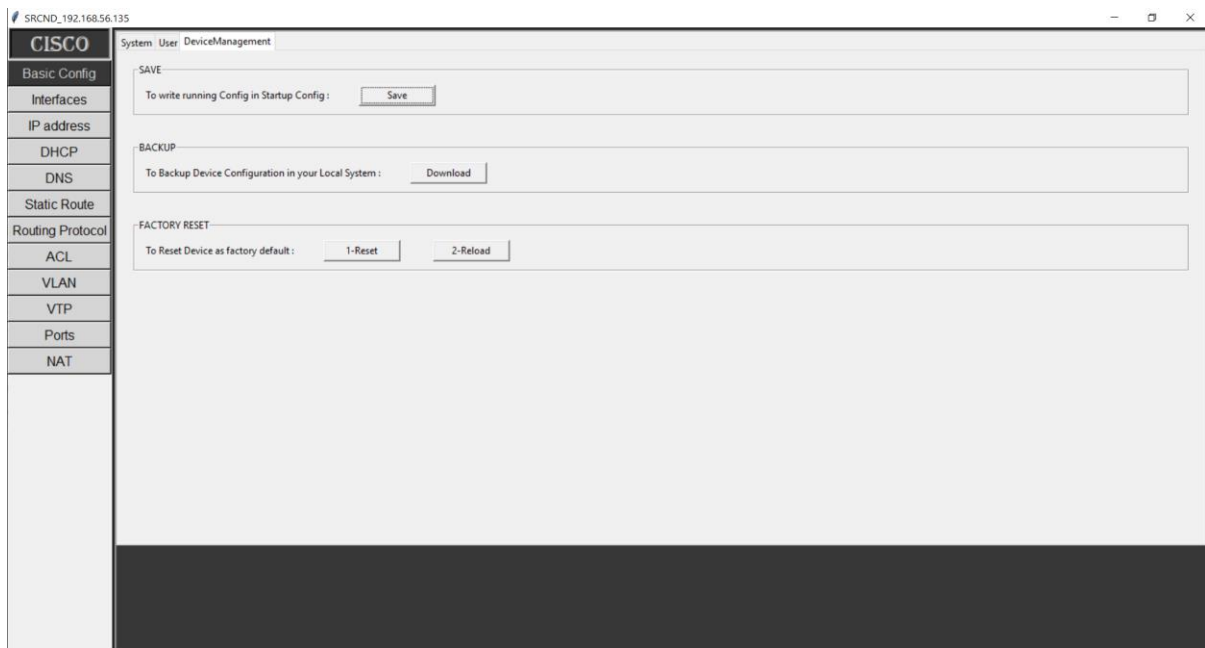


Fig 4.6 Device Management

i) Save:

By utilizing the "Save" feature, users can persistently store the current device configuration in non-volatile memory. This ensures that changes made to the device's configuration are retained even after a reboot.

ii) Backup:

The "Backup" functionality enables users to create backup copies of the device's configuration. These backups serve as a safety net in case of accidental configuration changes, hardware failures, or the need for disaster recovery.

iii) Factory Reset:

In situations where the device's configuration needs to be reset to its factory default settings, the "Factory Reset" feature can be initiated. This restores the device to its original state and is useful for troubleshooting or preparing devices for reassignment.

By encompassing these functionalities within the "Basic Configuration" module, the software simplifies the setup and management of essential device parameters. This module ensures that devices are configured correctly, operate securely, and can be effectively managed throughout their lifecycle.

4.4.2) Interfaces Module

The "Interfaces" module is pivotal for network administrators and engineers using the "Software for Remote Configuration of Network Devices." This module empowers users to manage and configure various types of interfaces, both physical and virtual, across network devices. This comprehensive module facilitates efficient network connectivity and communication.

b) Interface configuration

Fig 4.7 Interface

i) Interface Behavior:

The "Interface Behavior" feature allows users to customize the behavior of interfaces according to the specific requirements of the network. Configurable parameters might include link speed, duplex mode, and flow control settings.

ii) Serial Interface:

Serial interfaces are crucial for connecting devices over serial communication links. This feature enables users to configure serial interfaces with relevant settings such as baud rate, data bits, stop bits, and parity.

iii) Default Interface:

The "Default Interface" option is useful for designating a specific interface as the default for certain operations. For instance, the default interface for outbound traffic might be configured to ensure smooth communication.

iv) Switchport Selection:

In scenarios involving Ethernet interfaces, users can configure switchport settings. This includes options for configuring access or trunk modes, allowed VLANs, and native VLANs for switchports.

b) Virtual Interface:

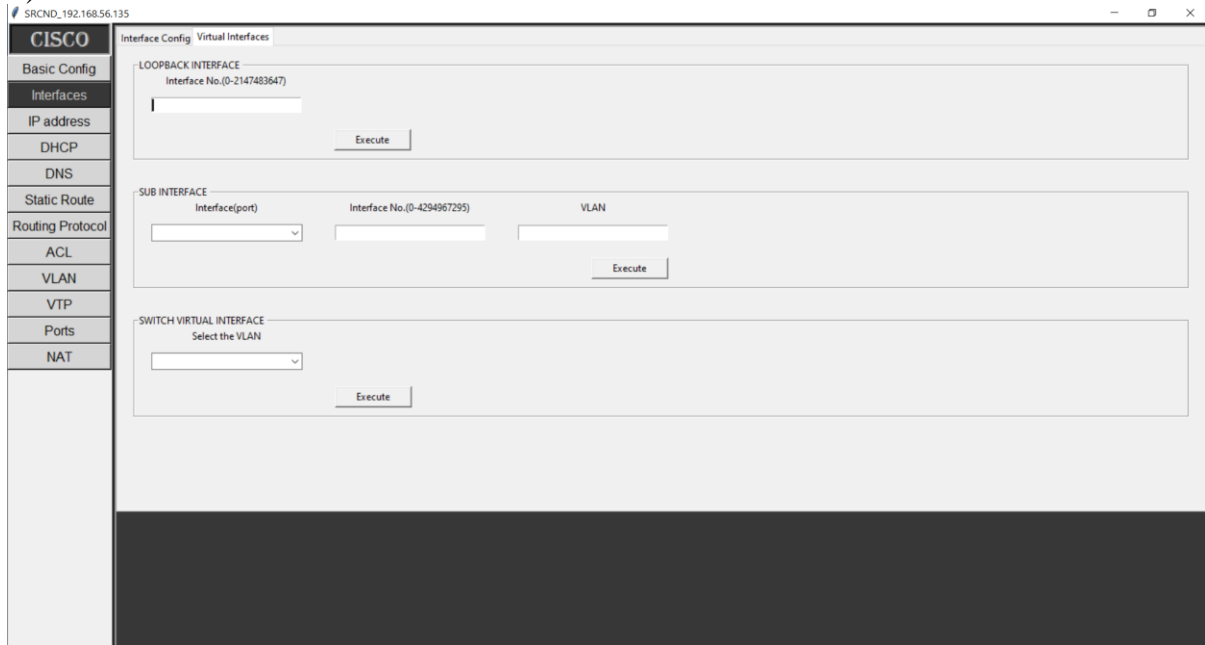


Fig 4.8 Virtual Interface.

i) Loopback Interface:

Loopback interfaces are virtual interfaces that are often used for testing and troubleshooting. Users can configure loopback interfaces with specific IP addresses, aiding in network diagnostics.

ii) Sub-interface:

Sub-interfaces enable the segmentation of physical interfaces into multiple logical interfaces, each with its own distinct characteristics. Users can configure sub-interfaces to separate network traffic.

iii) SVI (Switch Virtual Interface):

SVIs are virtual interfaces associated with VLANs on Layer 2 switches. Users can configure SVIs to allow Layer 3 routing between VLANs, enhancing network segmentation and connectivity.

By providing these functionalities within the "Interfaces" module, the software simplifies the process of configuring various types of interfaces across network devices. This module empowers users to tailor interface settings to meet network demands, enhance connectivity, and ensure efficient communication within the network infrastructure.

4.4.3) IP Address Module

The "IP Address" module within the "Software for Remote Configuration of Network Devices" is pivotal for establishing and managing IP addressing schemes across network devices. IP addresses are the foundation of network communication, and this module enables users to configure and maintain these addresses efficiently.

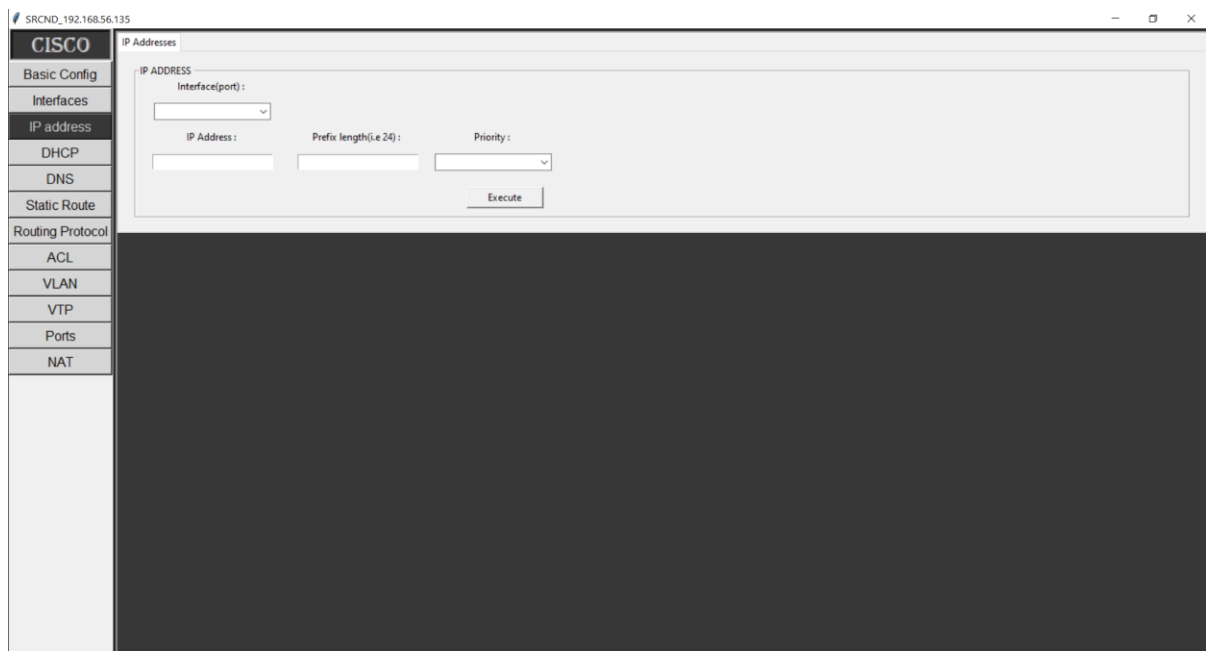


Fig 4.9 IP Address.

Functionality:

This module offers features that enable network administrators and engineers to manage IP addresses effectively:

IP Address Configuration: Users can configure IP addresses on various interfaces, allowing devices to communicate within the network. This involves specifying the IP address, subnet mask, and potentially other related settings.

4.4.4) DHCP Module

The "Dynamic Host Configuration Protocol (DHCP)" module within the "Software for Remote Configuration of Network Devices" provides essential tools for automating the assignment of IP addresses and network configuration settings. This module streamlines the management of IP addresses and enhances network connectivity.

a) DHCP SERVER:

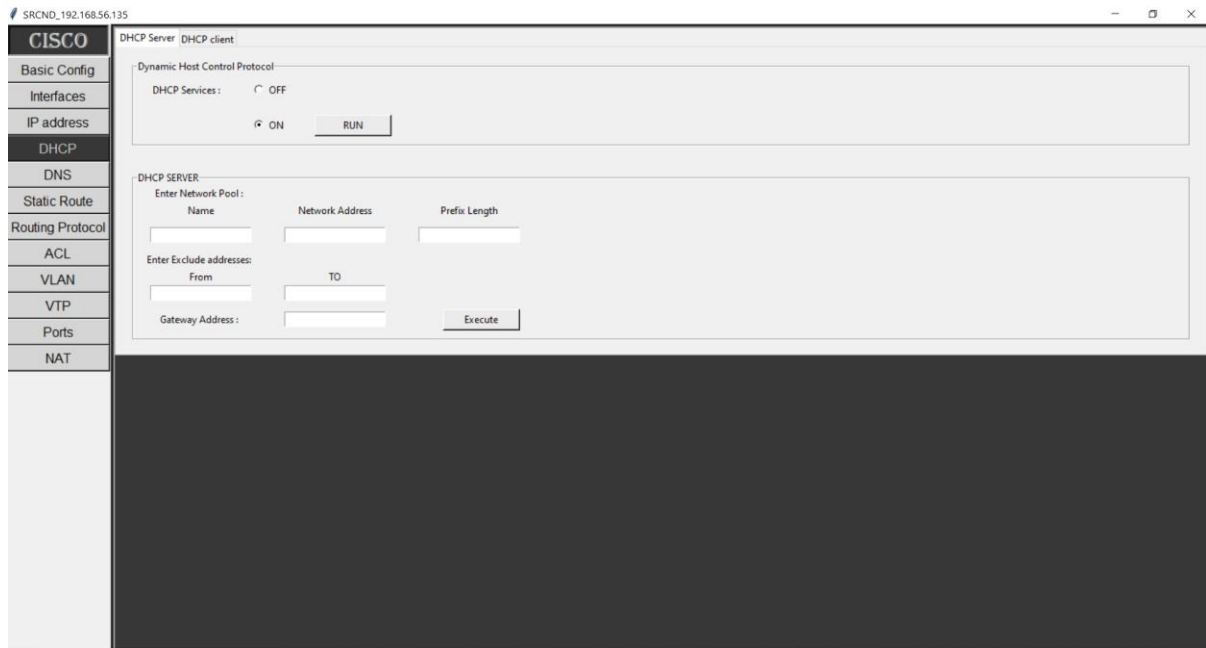


Fig 4.10 DHCP Server (a).

i) Server Enable:

Enabling the DHCP server functionality allows the network device to dynamically assign IP addresses and related configuration details to client devices. When enabled, the device acts as a DHCP server, simplifying the IP address allocation process.

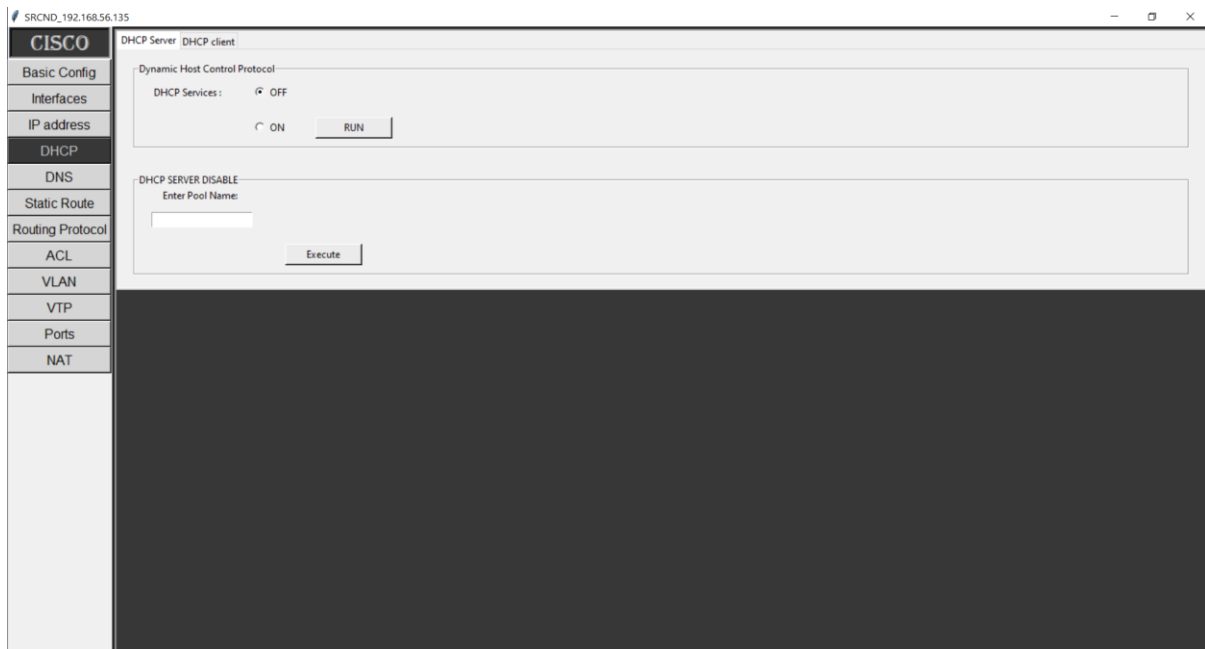


Fig 4.10 DHCP Server (b).

ii) Server Disable:

Disabling the DHCP server functionality suspends the automatic assignment of IP addresses. This might be useful in scenarios where static IP address assignment is preferred, or when troubleshooting DHCP-related issues.

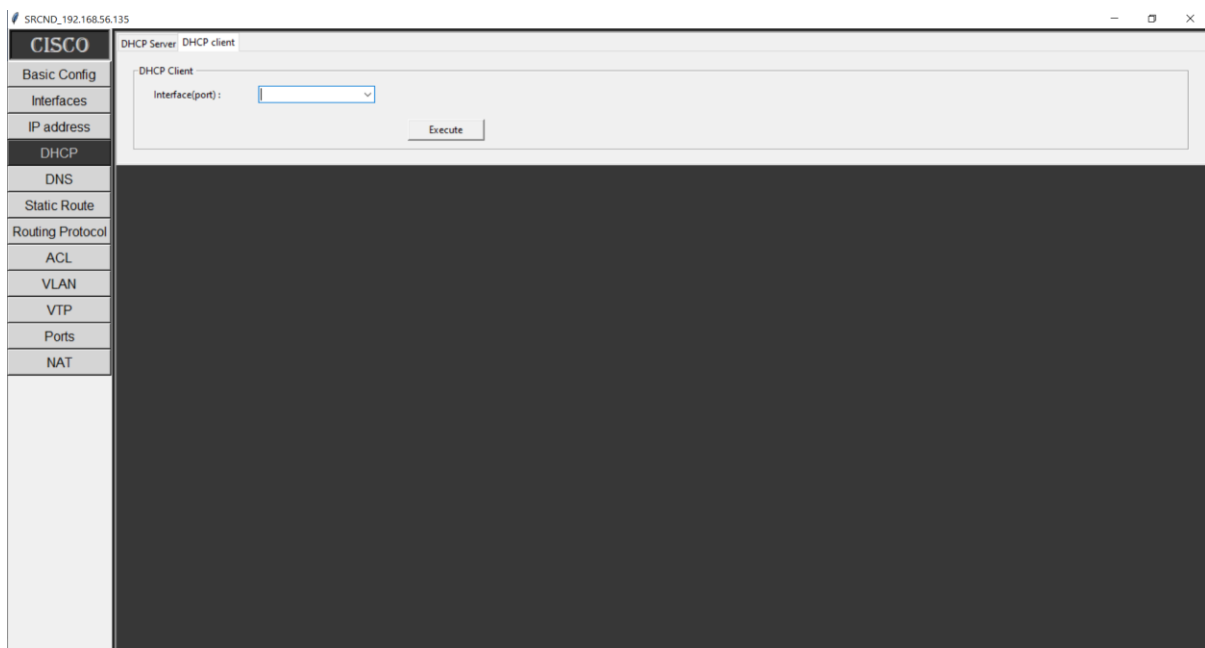
b) DHCP CLIENT:

Fig 4.11 DHCP Client.

b) DHCP Client:

The DHCP client functionality pertains to the network device itself, allowing it to request and obtain IP addresses and configuration settings from a DHCP server within the network. The device becomes a DHCP client when it requires an IP address dynamically.

Benefits:

The "DHCP" module offers several advantages for network administrators and engineers:

1. **Efficient IP Address Management:** DHCP eliminates the manual process of assigning IP addresses to devices, leading to efficient IP address management and allocation.
2. **Reduced Configuration Errors:** Automation through DHCP reduces the chances of human errors in IP address assignment, leading to smoother network operations.
3. **Network Scalability:** DHCP is particularly useful in large networks where manual IP address management becomes impractical. It facilitates the addition of new devices without the need for extensive manual configurations.
4. **IP Address Pooling:** DHCP maintains a pool of available IP addresses, ensuring that devices are assigned unique addresses without conflicts.

By incorporating the "DHCP" module, your software enhances network efficiency by automating IP address assignment and management. This module reduces administrative workload, minimizes configuration errors, and supports seamless network expansion.

4.4.5 DNS Module

The "Domain Name System (DNS)" module within the "Software for Remote Configuration of Network Devices" is crucial for translating human-readable domain names into IP addresses, enabling seamless communication across the internet and local networks. This module empowers users to manage DNS settings, ensuring efficient and accurate name resolution.

a) DNS Server:

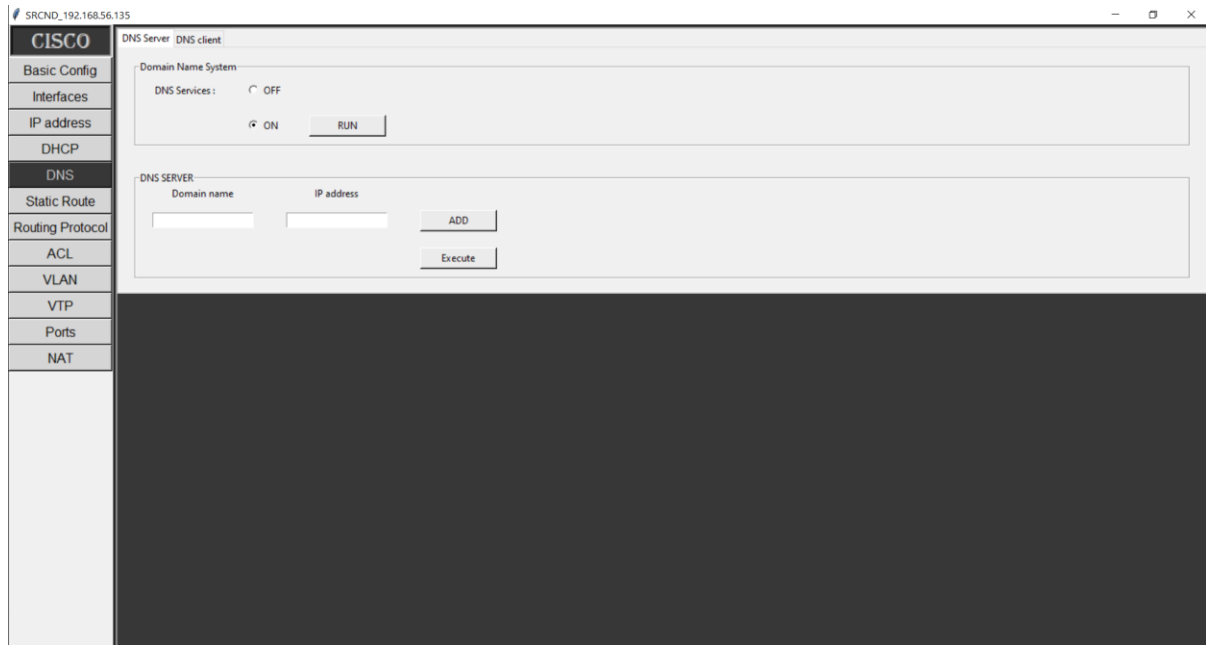


Fig 4.11 DNS server.

The DNS server functionality enables the network device to act as a DNS server, responsible for translating domain names into corresponding IP addresses. Users can configure and manage DNS server settings within this sub-module.

b) DNS Client:

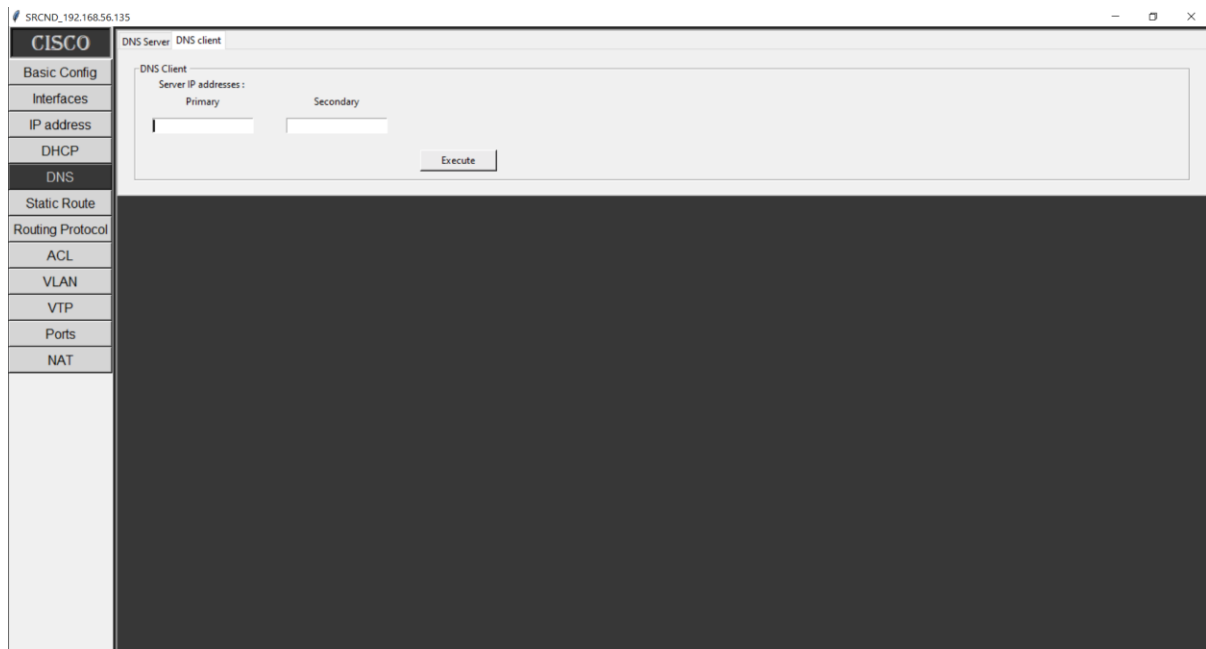


Fig 4.12 DNS Client.

The DNS client functionality allows the network device itself to request DNS resolution services from external DNS servers. This is essential for enabling the device to translate domain names into IP addresses, facilitating communication with other devices and resources on the network and the internet.

By incorporating the "DNS" module, our software contributes to efficient network communication and resource access by managing the translation of domain names to IP addresses. This module enhances user experience, supports network scalability, and ensures accurate and reliable name resolution across the network.

4.4.6) Static Routing Module

The "Static Routing" module within the "Software for Remote Configuration of Network Devices" empowers network administrators and engineers to configure routing tables manually, ensuring efficient data forwarding within the network infrastructure. This module is essential for controlling how data packets are directed from source to destination.

a) Static Route:

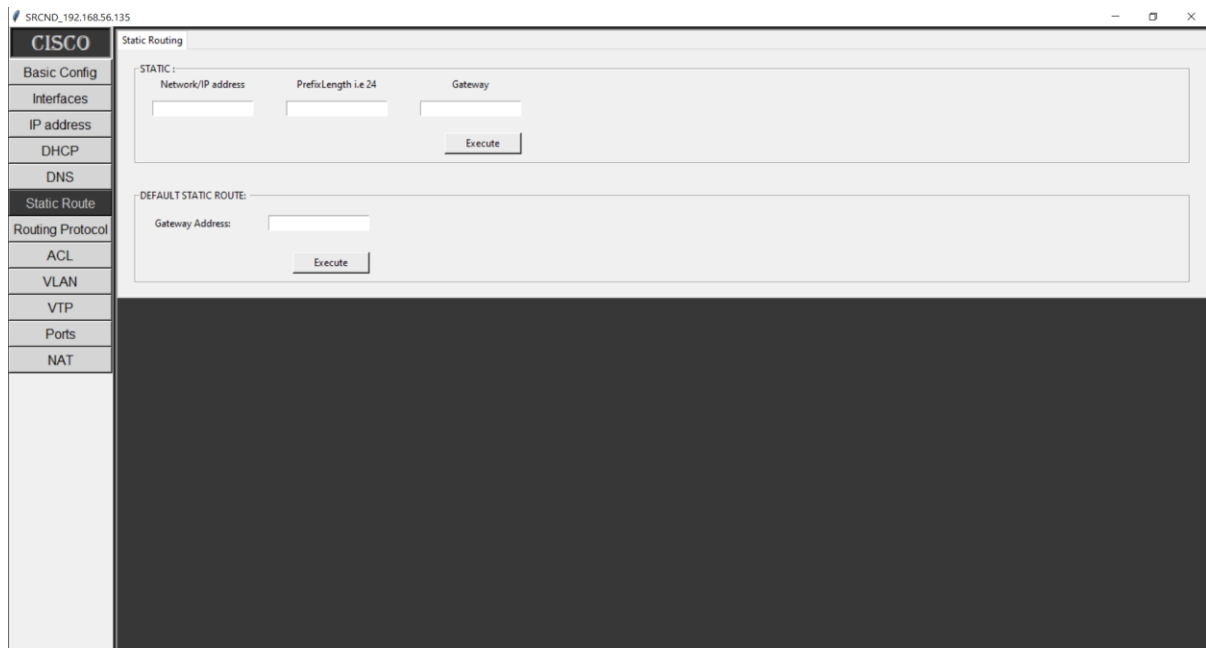
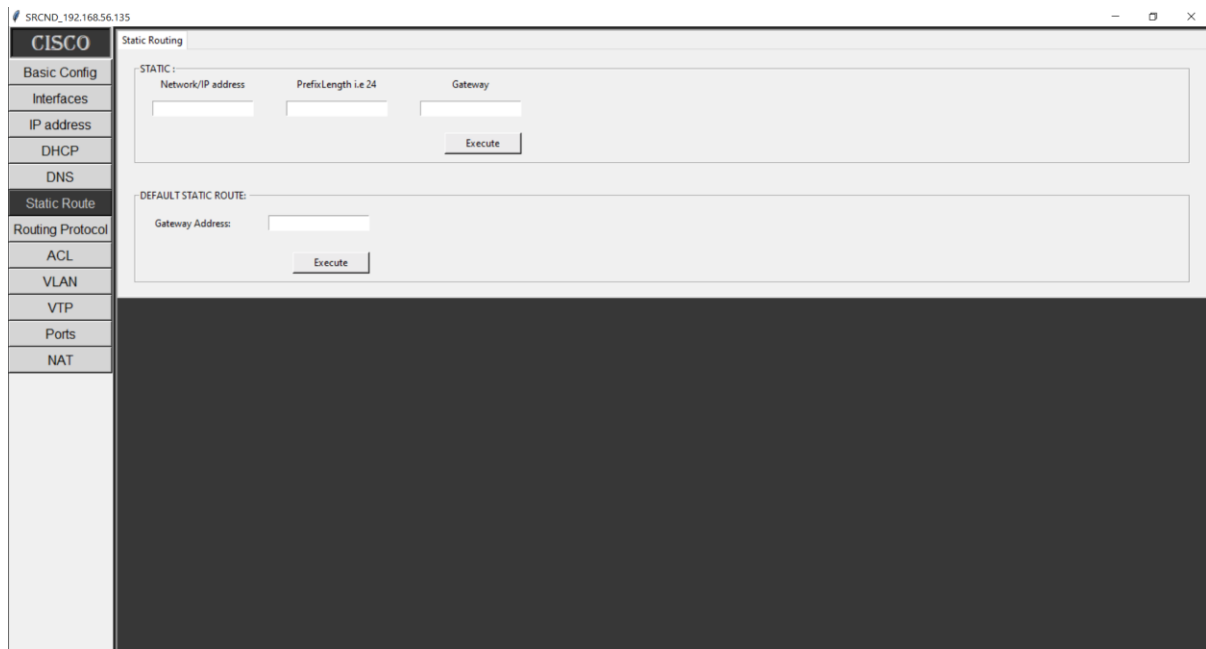
The screenshot shows a web-based configuration interface for a Cisco device. On the left is a vertical sidebar with a 'CISCO' logo at the top and a list of configuration categories: Basic Config, Interfaces, IP address, DHCP, DNS, Static Route (which is highlighted), Routing Protocol, ACL, VLAN, VTP, Ports, and NAT. The main area is titled 'Static Routing'. It contains two sections: 'STATIC' and 'DEFAULT STATIC ROUTE'. The 'STATIC' section has three input fields labeled 'Network/IP address', 'PrefixLength 1..24', and 'Gateway', followed by an 'Execute' button. The 'DEFAULT STATIC ROUTE' section has a single input field labeled 'Gateway Address' and an 'Execute' button. The bottom half of the main area is a large, dark grey rectangular block.

Fig 4.13(a) Static Routing.

Static routing involves configuring routes manually in the routing table of network devices. Administrators can define specific routes for forwarding data packets to destinations that are not covered by dynamic routing protocols. Static routes remain constant and do not change automatically.

b) Default Static Route:



The image shows a screenshot of the Cisco Packet Tracer Static Routing configuration window. The window has a sidebar on the left with a menu of configuration options: Basic Config, Interfaces, IP address, DHCP, DNS, Static Route (highlighted), Routing Protocol, ACL, VLAN, VTP, Ports, and NAT. The main area is titled 'Static Routing' and contains two sections. The first section, 'STATIC:', has three input fields: 'Network/IP address', 'Prefix Length i.e. 24', and 'Gateway', followed by an 'Execute' button. The second section, 'DEFAULT STATIC ROUTE:', has a 'Gateway Address' input field and an 'Execute' button. The bottom half of the window is a large, dark gray area, likely for displaying the routing table or command output.

Fig 4.13(b) Default Static Routing.

A default route (also known as the default gateway) serves as the path that network devices use to forward data packets when no specific matching route is found in the routing table. It directs traffic to a predetermined next-hop IP address, usually the router that connects the local network to external networks.

4.4.7) Dynamic Routing Module

The "Dynamic Routing" module within the "Software for Remote Configuration of Network Devices" introduces automation to the process of determining the optimal paths for data packets within the network. This module enables network administrators and engineers to configure and manage dynamic routing protocols, enhancing network efficiency and adaptability.

a) RIP (Routing Information Protocol):

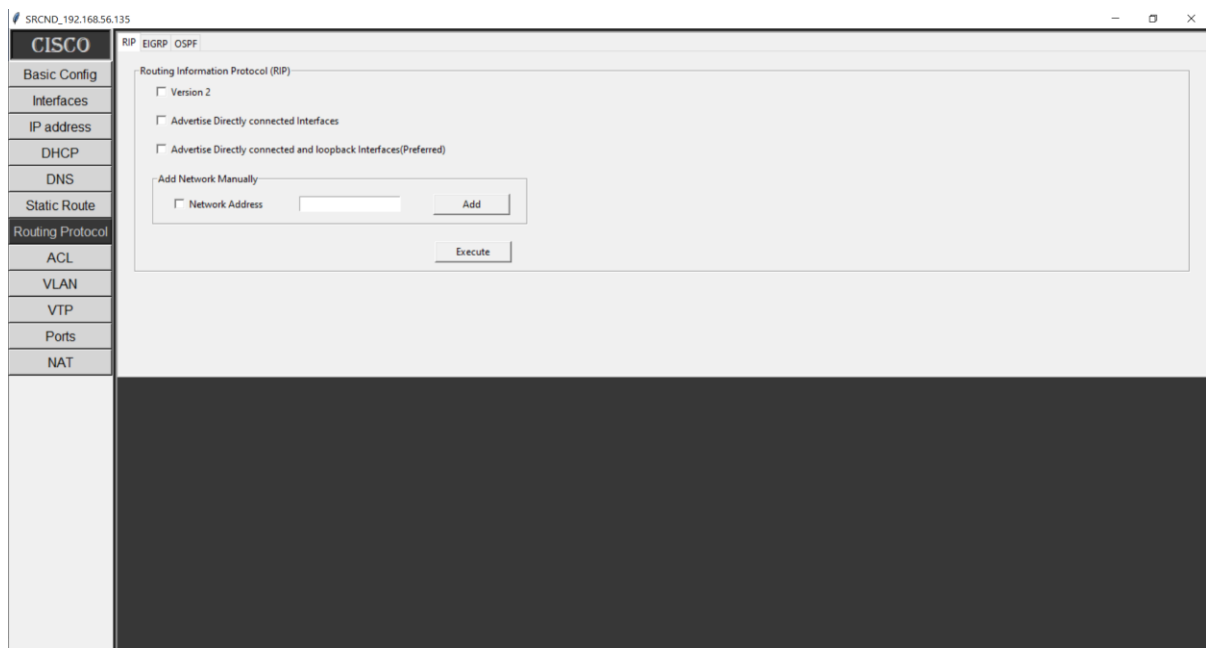
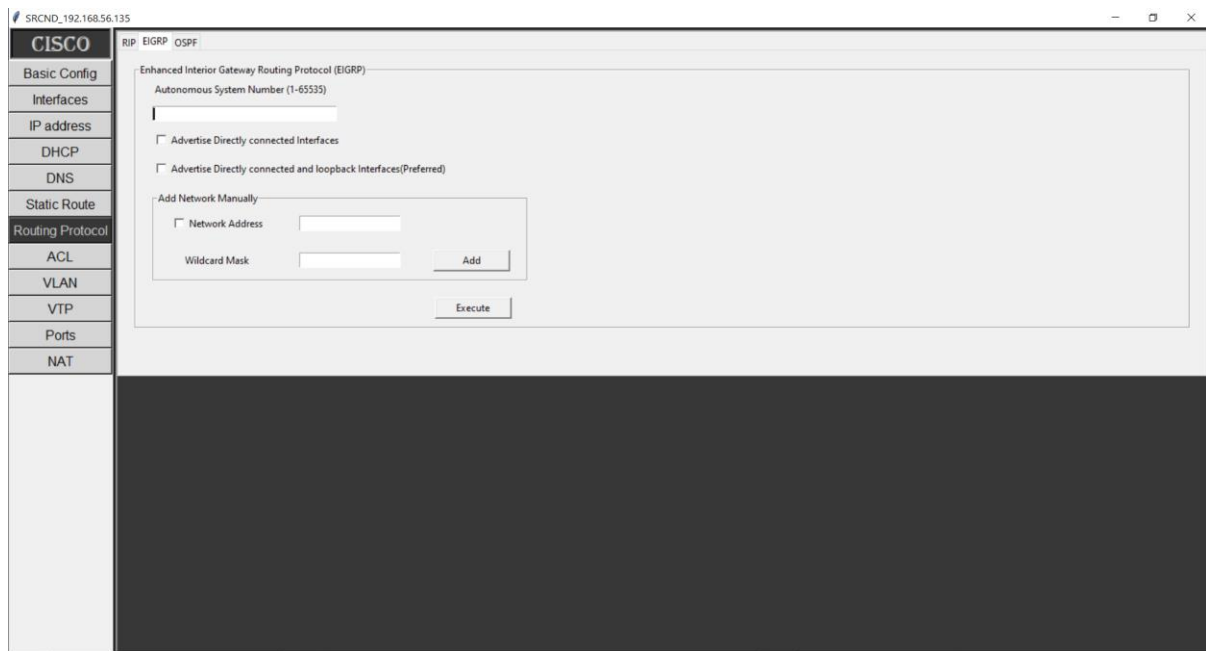


Fig 4.14 Routing Protocol RIP

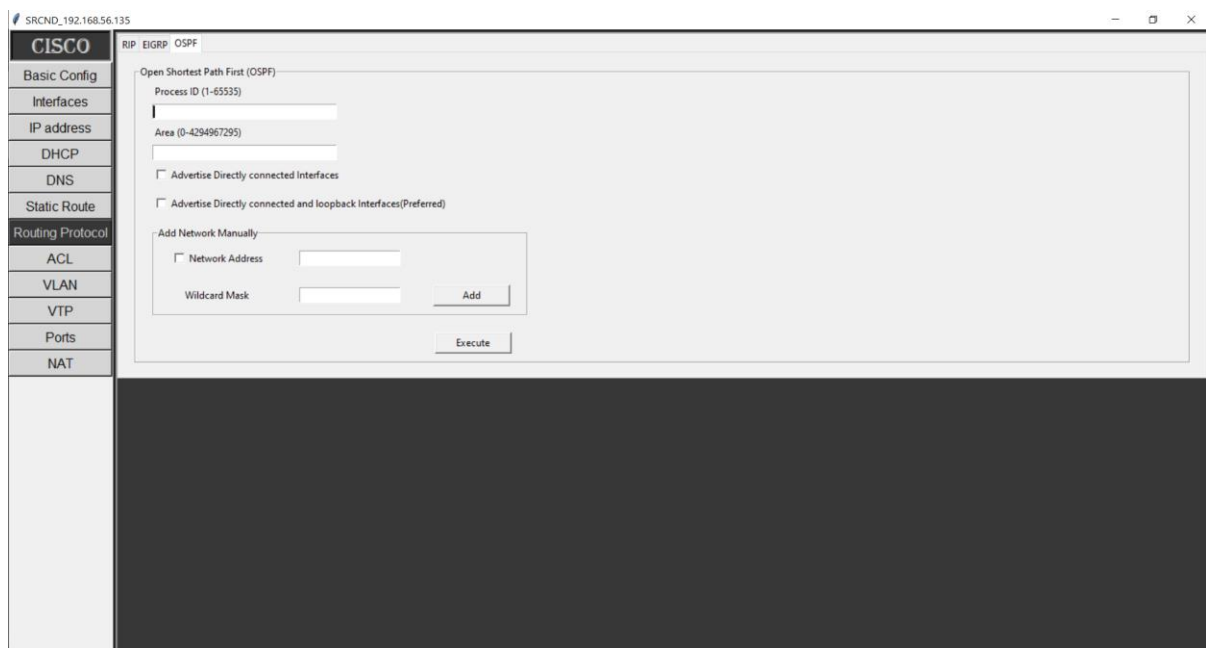
RIP is a distance-vector routing protocol that uses hop count as a metric to determine the best path for data packets. It periodically exchanges routing updates with neighboring routers, allowing routers to learn about network topology changes.

b) EIGRP (Enhanced Interior Gateway Routing Protocol):

*Fig 4.15 Routing Protocol EIGRP*

EIGRP is a Cisco proprietary routing protocol that combines features of both distance-vector and link-state routing protocols. It uses bandwidth and delay as metrics for path selection and supports rapid convergence.

c) OSPF (Open Shortest Path First):

*Fig 4.16 Routing Protocol OSPF*

OSPF is a link-state routing protocol that considers various factors, such as bandwidth and cost, to calculate the best path for data packets. It forms a detailed and accurate map of the network topology and converges quickly.

Benefits:

The "Dynamic Routing" module offers several advantages for network administrators and engineers:

1. **Efficient Path Selection:**

Dynamic routing protocols automatically adapt to changes in network topology, ensuring that data packets follow the most efficient paths.

2. **Quick Convergence:**

Dynamic routing protocols detect network changes and converge rapidly, minimizing disruptions to network connectivity.

3. **Load Balancing:**

Dynamic routing protocols distribute traffic among multiple paths, optimizing network resource utilization and preventing congestion.

4. **Scalability:**

Dynamic routing protocols handle network growth by automatically updating routing tables in response to topology changes.

5. **Adaptability:**

Changes in network topology are automatically accommodated, ensuring that the network continues to operate effectively.

4.4.8) Access Control Lists (ACL) Module:

The "Access Control Lists (ACL)" module within the "Software for Remote Configuration of Network Devices" empowers network administrators and engineers to control and manage traffic flow based on specific criteria. ACLs serve as a vital security and traffic management mechanism within the network.

Fig 4.17 ACL

a) Define Access List Globally:

This feature enables users to define access control lists (ACLs) that can be applied across the network globally. An ACL specifies rules that determine which traffic is allowed and which is denied based on criteria such as source IP address, destination IP address, port numbers, and more.

b) Apply Access List on Interface:

Network administrators can apply ACLs directly to specific interfaces. This allows for granular control over the types of traffic that can enter or exit a particular interface, enhancing security and optimizing network performance.

Benefits:

The "ACL" module offers several advantages for network administrators and engineers:

1. **Security Enhancement:** ACLs serve as a powerful security mechanism, allowing administrators to permit or deny specific types of traffic, protecting the network from unauthorized access and potential threats.

2. **Traffic Control:** By defining and applying ACLs, administrators can regulate the flow of traffic within the network, ensuring that only authorized traffic is allowed and that unwanted or malicious traffic is blocked.
3. **Network Segmentation:** ACLs support network segmentation by enabling administrators to control communication between different segments, enhancing security and preventing unauthorized access.
4. **Resource Optimization:** ACLs help optimize network resources by preventing unnecessary or unwanted traffic from consuming bandwidth and network resources.

By including the "ACL" module in our software, we provide users with the means to strengthen network security, optimize traffic management, and ensure compliance with network policies. This module enhances overall network integrity by controlling the flow of data according to defined rules and criteria.

4.4.9) VLAN Module

The "VLAN" module within the "Software for Remote Configuration of Network Devices" empowers network administrators and engineers to create and manage Virtual Local Area Networks (VLANs), enabling logical segmentation of the network into separate broadcast domains. This module enhances network efficiency, security, and management.

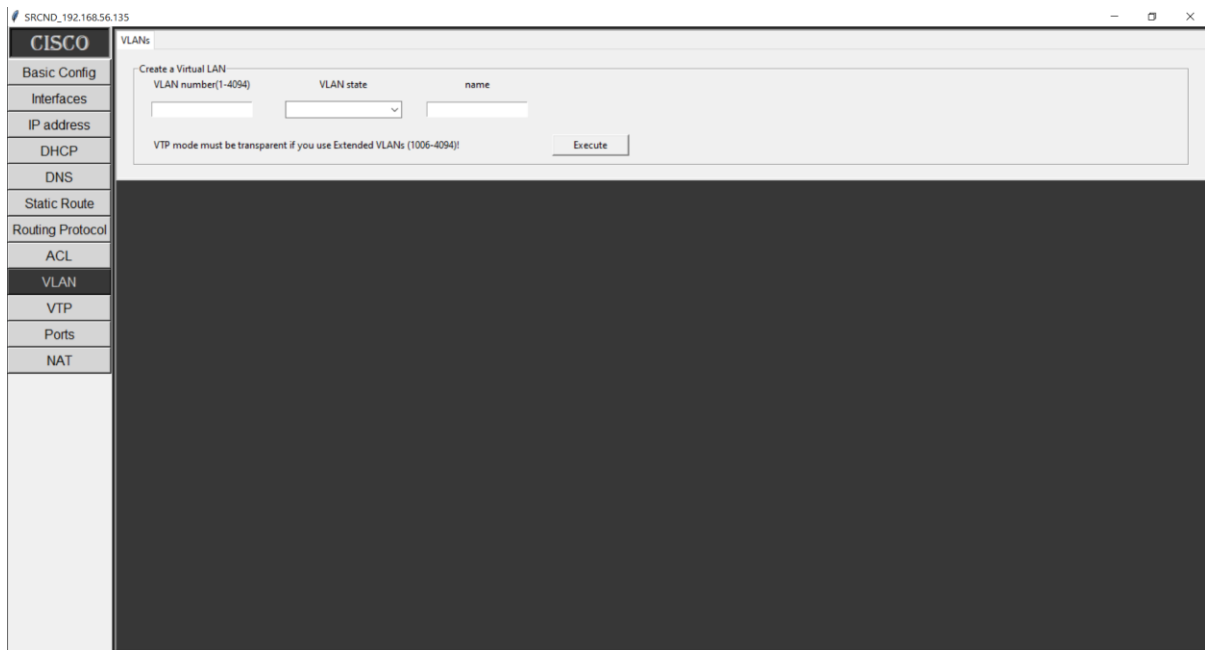


Fig 4.18 VLAN

Functionality:

This module offers features that allow users to configure and manage VLANs effectively:

- **VLAN Creation:** Users can create new VLANs, assigning them unique identifiers and names for easy recognition.
- **VLAN Membership:** Administrators can assign network devices to specific VLANs, determining which devices belong to which broadcast domains.
- **VLAN Trunking:** The module enables the configuration of trunk ports, allowing the transmission of multiple VLAN traffic over a single link.

Benefits:

The "VLAN" module offers several advantages for network administrators and engineers:

1. **Network Segmentation:** VLANs segment the network into logical groups, reducing broadcast domains and enhancing overall network performance.
2. **Broadcast Control:** By confining broadcast traffic within specific VLANs, network congestion due to excessive broadcasts is minimized.

3. **Enhanced Security:** VLANs can be used to isolate sensitive or critical network segments, preventing unauthorized access and enhancing security.
4. **Flexibility:** VLANs allow for the logical grouping of devices based on their function, department, or location, providing flexibility in network design.
5. **Resource Optimization:** VLANs optimize network resources by directing traffic only to relevant segments, reducing unnecessary data transmission.

By incorporating the "VLAN" module, our software enables administrators to create and manage VLANs effectively, improving network efficiency, security, and scalability. This module enhances the design and organization of the network infrastructure by offering logical segmentation and traffic control capabilities.

4.4.10) VLAN Trunking Protocol (VTP) Module

The "VLAN Trunking Protocol (VTP)" module within the "Software for Remote Configuration of Network Devices" enables network administrators and engineers to manage VLAN configurations across Cisco network devices. VTP simplifies the process of propagating and synchronizing VLAN information, ensuring consistent VLAN configurations throughout the network.

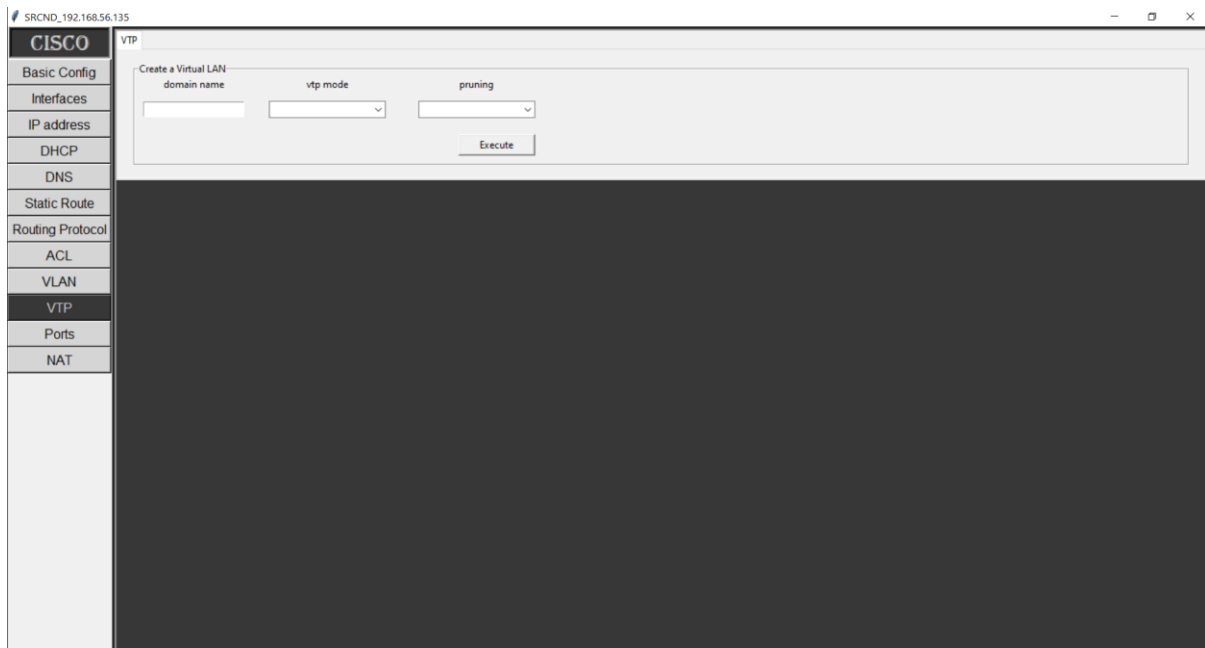


Fig 4.19 VTP

Functionality:

This module offers features that facilitate the management of VLANs using VTP:

- **VTP Domain Configuration:** Users can configure a VTP domain, specifying a common name for the domain to ensure that devices communicate and synchronize VLAN information effectively.
- **VTP Mode Selection:** Administrators can choose the operational mode of each device in the VTP domain, including server, client, or transparent mode.
- **VLAN Creation and Deletion:** Users can create and delete VLANs, and these changes will be automatically propagated to other devices in the VTP domain.

Benefits:

The "VTP" module offers several advantages for network administrators and engineers

1. **Simplified VLAN Management:** VTP automates the distribution of VLAN configuration changes, saving time and effort compared to manual configuration on each device.

2. **Consistency:** VTP ensures that VLAN configurations remain consistent across devices within the same VTP domain, minimizing configuration discrepancies.
3. **Ease of Deployment:** When adding or removing VLANs, administrators can make changes on a single device, and VTP will automatically update other devices.
4. **Scalability:** VTP is particularly useful in larger networks with numerous devices and VLANs, as it streamlines VLAN management.
5. **Reduced Configuration Errors:** By centralizing VLAN configuration updates, VTP reduces the risk of human errors that can occur when configuring VLANs individually on each device.
6. **Efficient Resource Allocation:** VTP aids in efficient resource allocation by ensuring that devices are aware of the same VLAN configuration.
7. **Rapid Network Changes:** VTP expedites the process of implementing network-wide VLAN changes, making network adjustments faster and more consistent.

4.4.11) Switchports Module

The "Switchports" module within the "Software for Remote Configuration of Network Devices" offers a comprehensive set of features for configuring various types of switchports. This module empowers network administrators and engineers to optimize data transmission and connectivity within the network by configuring switchports according to specific requirements.

a) Access Ports:

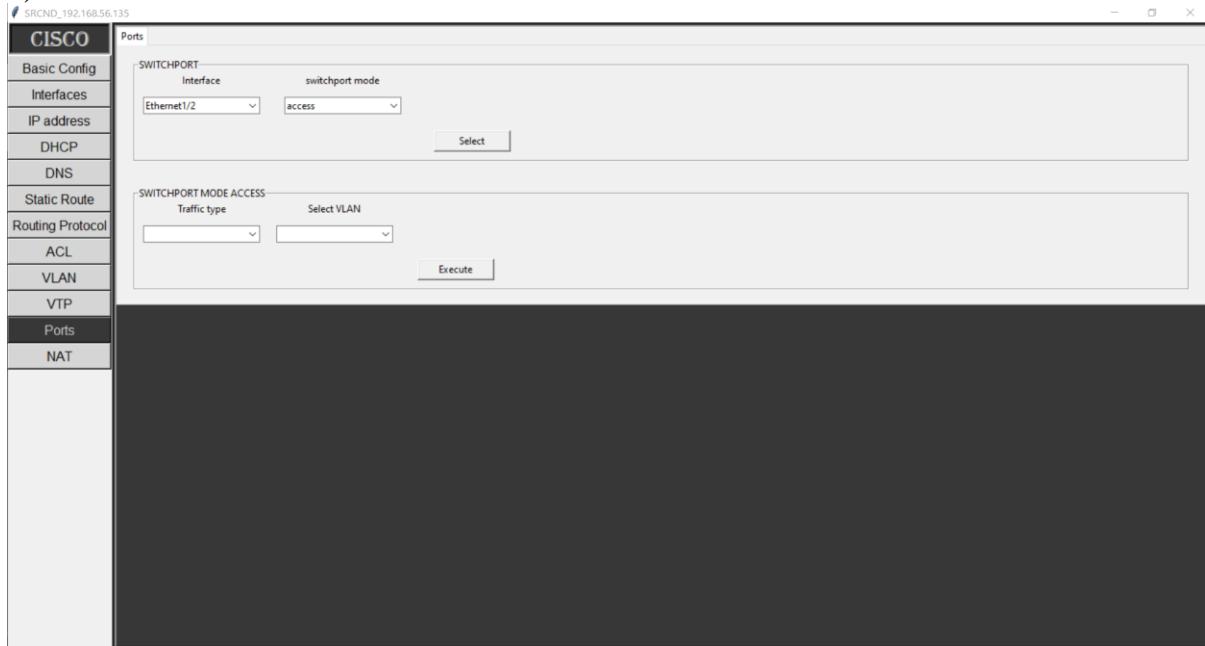


Fig 4.20 Switch Port Access

i) Data Access Port:

This feature allows users to configure an access port for regular data traffic. Devices connected to an access port belong to a specific VLAN, ensuring efficient data transmission within a designated broadcast domain.

ii) Voice Access Port:

Users can configure a voice access port to support Voice over IP (VoIP) traffic. This type of port is optimized for voice traffic and can be associated with a dedicated VLAN for VoIP devices.

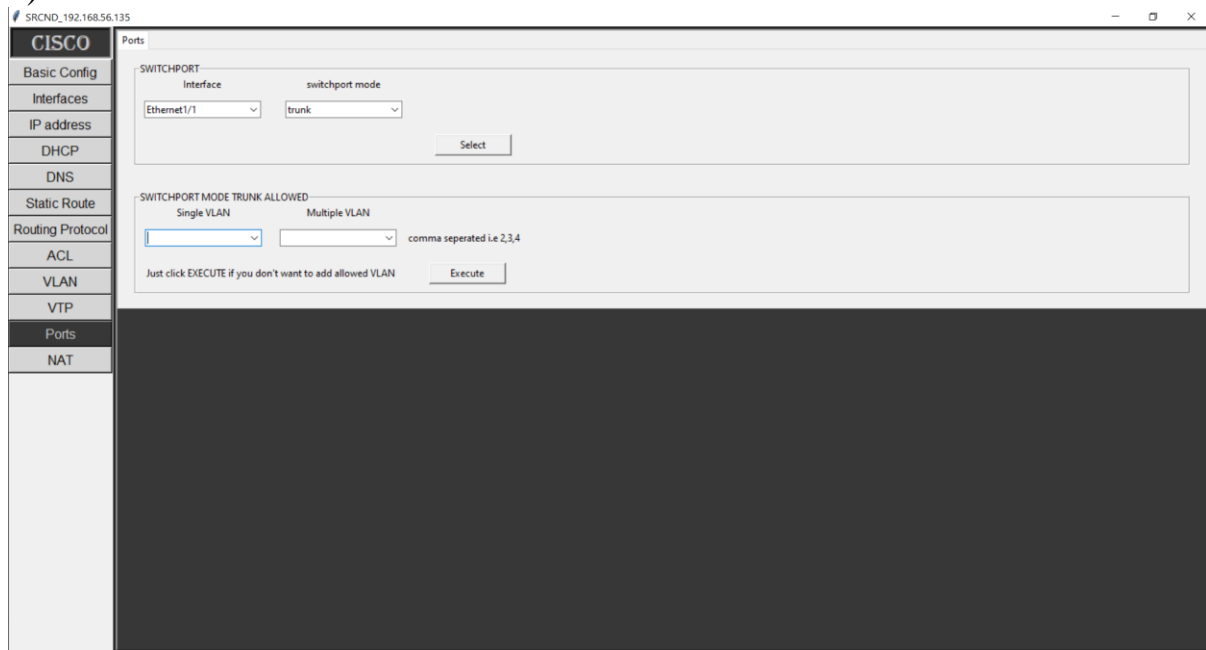
b) Trunk Ports:

Fig 4.21 Switch Port Trunk

i) Trunk Port with Allowed VLANs:

This feature enables users to configure a trunk port that can carry traffic for multiple VLANs. Administrators can specify which VLANs are allowed to traverse the trunk link, optimizing network resource utilization

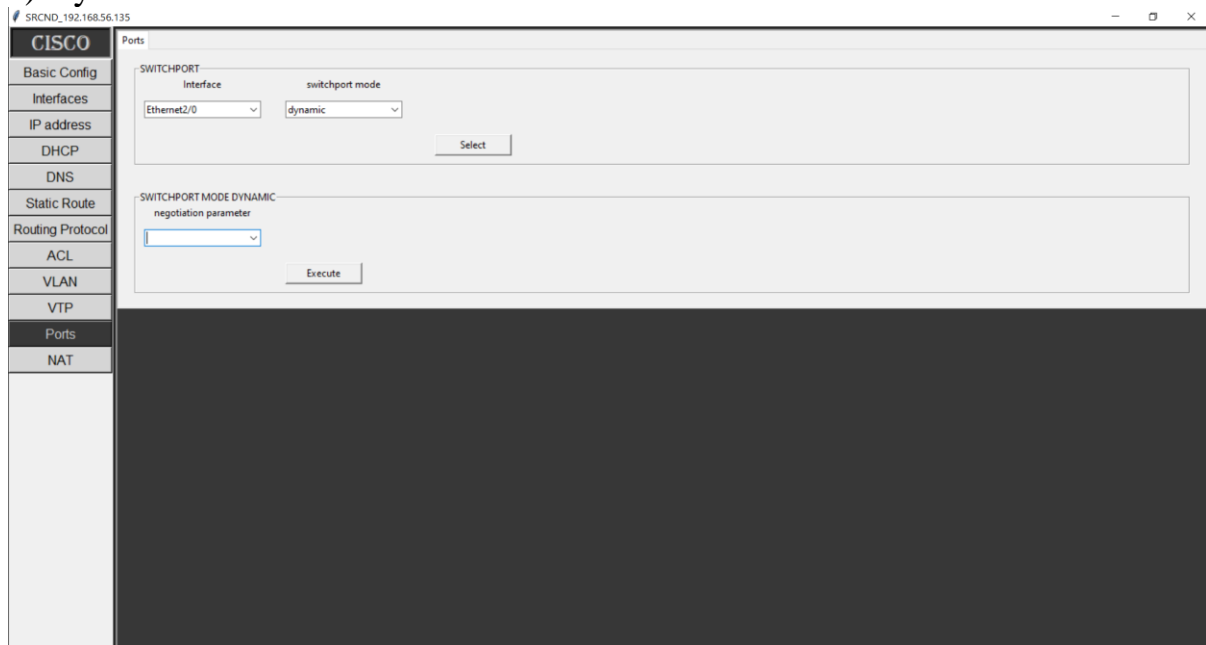
c) Dynamic Ports:

Fig 4.22 Switch Port Dynamic

i) Desirable Dynamic Port:

This dynamic port configuration mode allows the device to actively negotiate the trunking status with the neighboring device. If the neighboring device supports trunking, the link becomes a trunk port.

ii) Auto Dynamic Port:

In this dynamic port configuration mode, the device passively waits for the neighboring device to initiate trunking negotiation. If the neighboring device starts trunking negotiation, the link becomes a trunk port.

By including the "Switchports" module in our software, we provide users with versatile tools for configuring different types of switchports. This module enhances network performance, connectivity, and flexibility by catering to the specific needs of data, voice, and trunk traffic.

4.4.12) Network Address Translation (NAT) Module:

The "Network Address Translation (NAT)" module within the "Software for Remote Configuration of Network Devices" empowers network administrators and engineers to manage the translation of IP addresses, enabling efficient communication between devices within the network and external networks, such as the internet.

a) Static NAT:

The screenshot displays the Cisco NAT PAT configuration window. On the left is a sidebar with a menu including: Basic Config, Interfaces, IP address, DHCP, DNS, Static Route, Routing Protocol, ACL, VLAN, VTP, Ports, and NAT (which is currently selected). The main area is titled 'NAT PAT' and contains two sections: 'STATIC NAT' and 'DYNAMIC NAT'. The 'STATIC NAT' section has fields for 'Inside' and 'Outside' interfaces, and 'Private' and 'Public' IP addresses, with an 'Execute' button below. The 'DYNAMIC NAT' section has fields for 'Inside' and 'Outside' interfaces, a 'Public IP Addresses' section with 'Pool name', 'From', 'TO', and 'Prefix Length i.e 24' fields, and an 'Apply Dynamic NAT' section with an 'Access-list number' field, also featuring an 'Execute' button.

Fig 4.23 Static NAT

i) Inside & Outside Interfaces:

This feature allows users to configure static NAT mappings for devices located on the inside (local) network, translating their private IP addresses to public IP addresses when communicating with the outside network.

ii) Public & Private IP Addresses:

Administrators can define the mapping between the private IP address of an internal device and the corresponding public IP address for external communication.

b) Dynamic NAT:

Fig 4.24 Dynamic NAT

i) Inside & Outside Interfaces:

This feature involves the dynamic translation of private IP addresses to a pool of public IP addresses when devices from the inside network communicate with the outside network.

ii) Public & Private IP Addresses:

Users can define a range of private IP addresses that will be translated to a pool of public IP addresses for external communication.

iii) Apply Access List:

Administrators can apply an access list to dynamic NAT, allowing the translation of only specific devices based on the access list criteria.

c) PAT (Port Address Translation):

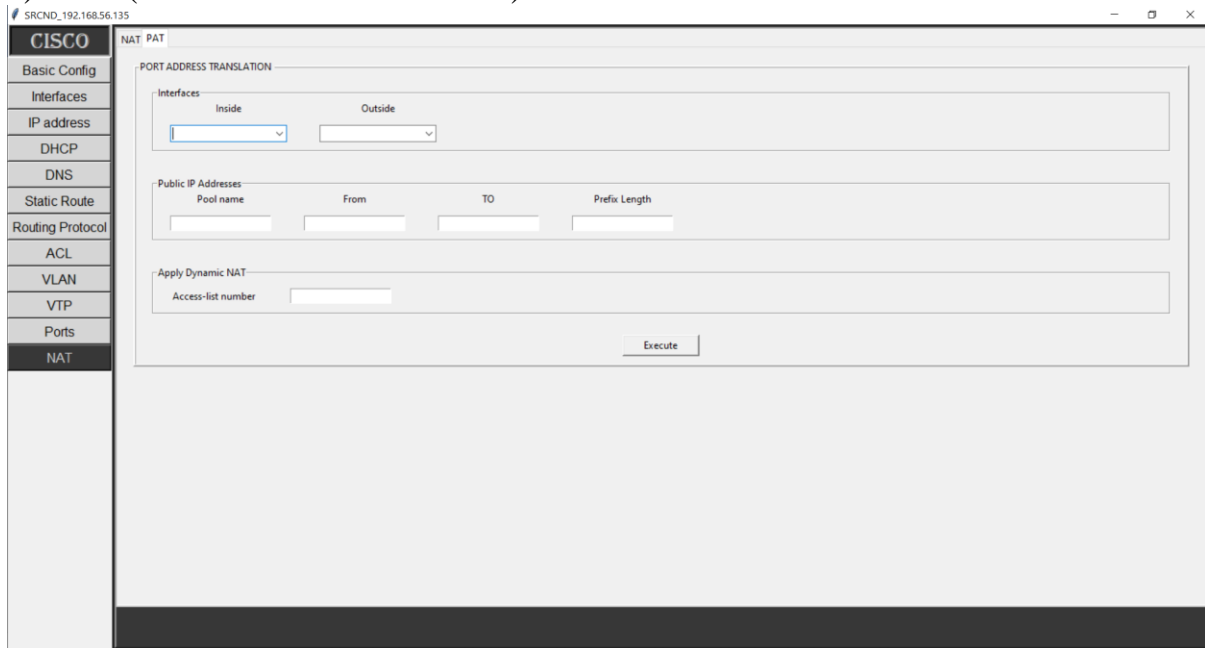


Fig 4.25 PAT

i) Inside & Outside Interfaces:

In this mode, PAT translates private IP addresses to a single public IP address, using different port numbers to distinguish between devices.

ii) Public & Private IP Addresses:

Administrators can specify the mapping between private IP addresses and the single public IP address used for PAT.

iii) Apply Access List:

Users can apply an access list to PAT, ensuring that only certain devices are eligible for translation.

Benefits:

The "NAT" module offers several advantages for network administrators and engineers:

1. **IPv4 Address Conservation:** NAT allows multiple devices within a private network to share a single public IP address, conserving the limited pool of public IP addresses.
2. **Enhanced Security:** NAT hides internal IP addresses from external networks, adding an additional layer of security to the network.
3. **Network Isolation:** NAT isolates internal devices from external networks, preventing direct communication between them.
4. **Scalability:** Dynamic NAT allows a pool of public IP addresses to be used for translation, accommodating the needs of growing networks.

5. Load Balancing: PAT assigns different port numbers to devices, enabling multiple devices to share a single public IP address.

By incorporating the "NAT" module, your software enhances network connectivity and security by managing the translation of IP addresses, facilitating communication between devices in private and external networks. This module optimizes address usage, provides security, and supports efficient communication within the network infrastructure.

4.5) Supported Vendor & Devices of our Software SRCND:

The "Software for Remote Configuration of Network Devices" is designed to provide seamless compatibility with specific vendors and their network devices. This compatibility ensures that users can effectively manage and configure their network infrastructure without facing device-related limitations. The software currently supports the following vendors and devices:

4.5.1) Supported Vendor: Cisco

Cisco is a renowned leader in networking solutions, and the software is tailored to work harmoniously with Cisco network devices, offering a comprehensive suite of configuration options.

4.5.2) Supported Devices:

1. Routers: The software supports the configuration of Cisco routers, empowering users to manage routing protocols, security settings, and other crucial parameters.
2. Switches: Cisco switches, both layer 2 and layer 3, are fully supported by the software. Users can configure VLANs, spanning tree protocols, switchports, and more.
3. Multi-Layer Switches (MLS): Multi-layer switches are a key component in modern networks. The software enables users to configure advanced features like Inter-VLAN routing and Layer 3 switching.
4. L3 Switches: Layer 3 switches, commonly used for efficient routing within local networks, are fully compatible with the software's capabilities.

By focusing on Cisco devices, the software ensures that users can confidently manage their network infrastructure, knowing that their specific devices are well-supported. This compatibility aligns with the project's aim to provide a comprehensive solution for remote configuration and management, optimizing network efficiency and security.

CHAPTER 5

Conclusion and Future Work

Conclusion and Future work

5.1) Features of the "SRCND"

The "Software for Remote Configuration of Network Devices" is equipped with a diverse array of features designed to simplify and enhance the process of configuring network devices. These features are aimed at providing network administrators, engineers, and users with a comprehensive tool to efficiently manage their network infrastructure. Here are the key features of the software:

i. **Intuitive GUI for Configuration:**

The software offers a user-friendly graphical interface that eliminates the need to memorize complex command-line instructions. Users can easily configure network devices by selecting options from the intuitive GUI.

ii. **Remote Configuration:**

Users can configure network devices remotely, reducing the need for physical access. This remote configuration capability saves time, effort, and resources, especially in large-scale network environments.

iii. **Secure Communication with SSH:**

The software ensures secure communication with network devices through the implementation of the Secure Shell (SSH) protocol. This guarantees encrypted and authenticated connections, enhancing overall security.

iv. **Support for Telnet:**

In addition to SSH, the software supports Telnet for communication with network devices. This flexibility accommodates varying security requirements and user preferences.

v. **Advanced Configuration Options:**

The software facilitates advanced network configurations, including Routing Information Protocol (RIP), Enhanced Interior Gateway Routing Protocol (EIGRP), and Open Shortest Path First (OSPF), empowering users with intricate network setups.

vi. **Device Backup and Restoration:**

Users can create backups of device configurations, ensuring the ability to revert to a known state if necessary. This feature safeguards against accidental configuration changes.

vii. **Time-Efficient Configuration:**

With the software's point-and-click interface, users can quickly configure devices by selecting settings from a list. This saves time compared to manually entering command-line instructions.

viii. **DNS Configuration:**

The software supports Domain Name System (DNS) configuration, allowing users to manage DNS server settings and ensure proper name resolution across the network.

ix. **VLAN Management:**

Users can create, modify, and manage Virtual Local Area Networks (VLANs) within the network infrastructure, enabling efficient network segmentation and management.

x. **DHCP Setup and Management:**

The software provides DHCP configuration capabilities, enabling the setup and management of Dynamic Host Configuration Protocol (DHCP) servers and clients.

xi. **Access Control Lists (ACLs):**

Users can define and apply Access Control Lists (ACLs) globally and on specific interfaces, enhancing network security by controlling traffic flow.

xii. **Interactive Help and Documentation:**

The software offers interactive help and documentation, guiding users through the configuration process and providing explanations for different settings.

5.2) Limitations of the Software

While the "Software for Remote Configuration of Network Devices" offers a wide array of features and benefits, it's important to acknowledge its limitations. Understanding these limitations helps users set realistic expectations and make informed decisions about its use.

Here are some limitations of the software:

5.2.1) Vendor Specificity:

The software is currently tailored for Cisco devices, limiting its compatibility to a specific vendor. Users with devices from other vendors might not fully benefit from its features.

5.2.2) Lack of Real-Time Monitoring:

The software primarily focuses on configuration tasks. Real-time monitoring, network analysis, and performance evaluation are beyond its scope.

5.2.3) Network Complexity:

The software might be better suited for smaller to medium-sized networks. Managing very large, complex networks might require additional functionalities and scalability.

5.3) Conclusion:

In the fast-paced world of networking, where efficiency and accuracy are paramount, the "Software for Remote Configuration of Network Devices" emerges as a groundbreaking solution. Through the journey of conceptualization, design, development, and implementation, this project has successfully culminated in a tool that addresses critical challenges in network device configuration.

The primary objective of this project was to create a user-friendly, secure, and efficient means of configuring network devices, particularly Cisco routers and switches. With the proliferation of networking technologies, the demand for simplified configuration methods has never been greater. The software's graphical user interface (GUI) has overcome the barriers presented by the traditional Command Line Interface (CLI), offering an intuitive platform accessible to both novice users and experienced professionals.

Through the realization of this project, several notable achievements have been realized

1. **Ease of Configuration:** The software's GUI has transformed the complexity of command-based configuration into an intuitive point-and-click process. This empowerment transcends the need for intricate command memorization.
2. **Remote Accessibility:** The ability to configure network devices remotely revolutionizes network management. Administrators can seamlessly make changes, minimizing downtime and reducing on-site requirements.
3. **Security Enhancement:** Incorporation of Secure Shell (SSH) and Telnet protocols ensures encrypted and secure communication, maintaining the confidentiality of critical configurations.

4. **Vendor Compatibility:** While currently centered on Cisco devices, the software's architecture is poised for expansion to support devices from other vendors, reflecting its potential for growth.
5. **Advanced Configuration Capabilities:** Beyond basic configurations, the software caters to advanced users by enabling the setup of dynamic routing, VLANs, NAT, and more.

5.4) Future work:

The "Software for Remote Configuration of Network Devices" has set a solid foundation, yet its journey is far from complete. As technology advances and networking landscapes evolve, there are exciting opportunities for the software to push boundaries and address emerging challenges. The following avenues represent the software's potential future work:

1. **Vendor Diversification:** While currently tailored for Cisco devices, the software's horizon can expand to encompass a wider range of vendors like Huawei, Juniper, and others. This inclusivity will make it a go-to solution for multi-vendor environments.
2. **Device Diversity:** Extending support beyond routers and switches to include devices like firewalls, load balancers, and more will offer users a holistic network management experience.
3. **BGP Configurations:** The software's potential can be elevated by integrating Border Gateway Protocol (BGP) configuration options, catering to complex network setups that demand advanced routing protocols.
4. **OSPF Multi-Area Configuration:** As networks scale, multi-area Open Shortest Path First (OSPF) configurations become essential. Implementing this feature will enable seamless management of OSPF areas.
5. **Simplified VPN Setup:** The software's intuitive interface can be harnessed to offer effortless Virtual Private Network (VPN) configuration, streamlining secure network connections.
6. **IPv6 Integration:** With the transition to IPv6, adding configuration capabilities for IPv6 addresses and protocols will ensure the software remains relevant in the evolving IP landscape.

REFERENCES:

<https://www.gns3.com/>

<https://www.vmware.com/products/workstation-player/workstation-player-evaluation.html>

<https://gns3.com/marketplace/appliances/cisco-iou-l3>

<https://www.vandyke.com/download/securecrt/download.html>

<https://www.jetbrains.com/pycharm/>

<https://www.jetbrains.com/pycharm/>

<https://docs.python.org/3/library/tkinter.html>

<https://docs.python.org/3/library/telnetlib.html>

<https://pypi.org/project/netmiko/>

<https://docs.python.org/3/library/re.html>

https://en.wikipedia.org/wiki/IP_address

https://en.wikipedia.org/wiki/Dynamic_Host_Configuration_Protocol

https://en.wikipedia.org/wiki/Domain_Name_System

<https://en.wikipedia.org/wiki/Routing>

<https://www.cisco.com/c/en/us/products/routers/what-is-routing.html#:~:text=Routing%20is%20the%20process%20of,Explore%20routers>

<https://www.techtarget.com/searchnetworking/answer/Static-and-dynamic-routing#:~:text=What%20is%20static%20routing%3F,one%20entry%20for%20each%20destination.>

<https://www.ccnablog.com/static-routing/>

<https://www.geeksforgeeks.org/difference-between-static-and-dynamic-routing/>

<https://www.geeksforgeeks.org/what-is-dynamic-routing-in-computer-network/>

https://www.cisco.com/assets/sol/sb/isa500_emulator/help/guide/ad1213063.html

<https://www.geeksforgeeks.org/routing-information-protocol-rip/>
<https://www.computernetworkingnotes.com/ccna-study-guide/rip-protocol-configuration-guide-with-examples.html>
<https://docs.sophos.com/nsg/sophos-utm/utm/9.708/help/en-us/Content/utm/utmAdminGuide/RoutingDynamic.htm>
<https://www.catchpoint.com/dynamic-routing-protocols/eigrp-vs-ospf>
<https://ipcisco.com/lesson/dynamic-routing-protocols-ccnp/>
<https://www.networkstraining.com/configuring-ospf-on-cisco-routers/>
<https://www.geeksforgeeks.org/eigrp-configuration/#:~:text=Enhanced%20Interior%20Gateway%20Routing%20Protocol,and%20many%20other%20useful%20features.>
https://www.cisco.com/c/en/us/td/docs/ios-xml/ios/iproute_eigrp/configuration/15-mt/ire-15-mt-book/ire-enhanced-igrp.html
<https://study-ccna.com/what-is-a-vlan/>
<https://ipcisco.com/lesson/vlans-virtual-local-area-networks/>
<https://www.geeksforgeeks.org/difference-between-network-address-translation-nat-and-port-address-translation-pat/>
<https://www.w3schools.com/python/>
https://www.w3schools.com/python/python_regex.asp