# Cisco Project: Building a Software Router on a Linux Image with Python Components

**Harshvir singh**

April 16, 2024

## 1 Introduction

In this project, we will build a software router on a Linux image using Python components. The software router will be a virtual appliance that can be deployed on various hypervisors, such as VMware, KVM, or Hyper-V. The Linux image will provide the operating system and network stack, while the Python components will implement the router's logic and functionality.

The objectives of this project are:

- To learn the fundamentals of networking, routing, and Python programming.
- To design and implement a software router that can forward packets between different networks.
- To test and evaluate the performance and scalability of the software router.
- To create a product delivery plan that includes the documentation, video presentation, and marketing website.

The significance of this project is that it provides a hands-on experience in building and deploying a network appliance using open-source software and tools. It also showcases the power and flexibility of Python as a programming language for networking and automation.

## 2 Theory and Background

### 2.1 TCP Frames

TCP (Transmission Control Protocol) is a transport-layer protocol that provides reliable, ordered, and error-checked delivery of data between applications running on different devices. TCP frames consist of a header and a payload, where the header contains metadata, such as source and destination ports, sequence and acknowledgment numbers, and flags.

TCP frames play a crucial role in networking, as they enable applications to communicate over the internet using a standardized and interoperable protocol. TCP frames also provide various features, such as flow control, congestion control, and retransmission, that ensure the quality and reliability of the data transmission.

## 2.2 TCP/IP Stack

TCP/IP (Transmission Control Protocol/Internet Protocol) is a suite of communication protocols that define the architecture and behavior of the internet. The TCP/IP stack consists of four layers: the application layer, the transport layer, the internet layer, and the link layer.

The application layer provides interfaces and services for applications to access the network, such as DNS, HTTP, and FTP. The transport layer provides reliable and efficient delivery of data between applications, using protocols such as TCP and UDP. The internet layer provides logical addressing and routing of packets across different networks, using protocols such as IP and ICMP. The link layer provides physical addressing and transmission of packets over a link, using protocols such as Ethernet and Wi-Fi.

## 2.3 Theory of Routing

Routing is the process of forwarding packets between different networks based on their destination addresses. Routing algorithms use various metrics, such as hop count, bandwidth, and delay, to determine the best path for a packet.

Routing can be classified into two categories: interior routing and exterior routing. Interior routing refers to the routing within a single autonomous system (AS), while exterior routing refers to the routing between different ASes. Interior routing uses protocols such as OSPF and RIP, while exterior routing uses protocols such as BGP.

## 2.4 Unique Qualities of Cisco Routers

Cisco routers are high-performance and scalable networking devices that provide various features and services, such as Quality of Service (QoS), Virtual Private Network (VPN), and firewall. Cisco routers use a proprietary operating system called IOS (Internetwork Operating System), which provides a command-line interface (CLI) for configuration and management.

Cisco routers also support various interfaces, such as Ethernet, serial, and wireless, that enable them to connect to different types of networks and devices. Cisco routers also support various routing protocols, such as OSPF, BGP, and EIGRP, that enable them to exchange routing information and optimize the traffic flow.

## 2.5 Why Python is the Preferred Tool for Networking

Python is a high-level and interpreted programming language that provides various libraries and frameworks for networking, such as Scapy, Paramiko, and Twisted. Python is also easy to learn, read, and write, and allows for rapid prototyping and development.

Python is the preferred tool for networking for several reasons:

- Python provides a rich set of data types and constructs, such as lists, dictionaries, and classes, that enable developers to model and manipulate network data and objects.
- Python provides various networking libraries and modules, such as socket, select, and threading, that enable developers to implement network applications and services.
- Python provides various automation and orchestration frameworks, such as Ansible, Salt, and Puppet, that enable developers to automate network tasks and workflows.
- Python provides various visualization and debugging tools, such as matplotlib, networkx, and pdb, that enable developers to visualize and debug network data and algorithms.

# 3 Project Description

## 3.1 Components of the Project

The components of the project are:

- The software router, which is a virtual appliance that runs on a Linux image and implements the router's logic and functionality.
- The Linux image, which provides the operating system and network stack for the software router.
- The Python components, which implement the router's logic and functionality using Python scripts and modules.

## 3.2 Lab Work Book

The lab work book is a document that provides step-by-step instructions for building and testing the software router. The lab work book includes the following sections:

- Introduction, which provides an overview of the project and its objectives.
- Prerequisites, which lists the required software and tools for the lab.

- Steps to build the Linux image, which provides detailed instructions for building and configuring the Linux image.

- Steps to build the software router, which provides detailed instructions for installing and configuring the Python components.

- Steps to test and evaluate the software router, which provides detailed instructions for testing and evaluating the performance and scalability of the software router.

### 3.3 Steps to Build the Python Router in Linux

The steps to build the Python router in Linux are:

1. Install the required software packages, such as Python, pip, and git.

2. Clone the Python router repository from GitHub.

3. Install the required Python packages, such as Scapy, Paramiko, and Twisted.

4. Configure the network interfaces and routing tables.

5. Test and evaluate the performance and scalability of the software router.

## 4 Understanding Python as a Language

Python is a high-level and interpreted programming language that provides various features and constructs, such as dynamic typing, garbage collection, and exception handling. Python also provides various libraries and frameworks for networking, such as Scapy, Paramiko, and Twisted.

Python has a simple and clean syntax that emphasizes readability and expressiveness. Python also supports various programming paradigms, such as procedural, object-oriented, and functional, that enable developers to choose the best approach for their needs.

Python also provides various data types and constructs, such as lists, dictionaries, and classes, that enable developers to model and manipulate data and objects. Python also provides various control structures, such as if-else, for, and while, that enable developers to implement conditional and iterative logic.

Python also provides various libraries and frameworks for networking, such as Scapy, Paramiko, and Twisted. Scapy is a powerful and flexible packet manipulation library that provides various tools and functions for creating, sending, and receiving network packets. Paramiko is a secure and reliable SSH library that provides various tools and functions for authenticating, encrypting, and decrypting network data. Twisted is an asynchronous and event-driven networking framework that provides various tools and functions for implementing network applications and services.

# 5 Setting up the Environment

## 5.1 VMWare and Dockerized Linux Image Setup

VMWare is a virtualization platform that enables users to create and run virtual machines on a physical host. VMWare provides various tools and features, such as vSphere, vCenter, and ESXi, that enable users to manage and monitor their virtual infrastructure.

To set up the VMWare and Dockerized Linux image, follow these steps:

1. Download and install VMWare Workstation or VMWare Fusion.

2. Create a new virtual machine with the following specifications:

   - Guest operating system: Ubuntu 20.04 LTS
   - CPU: 2
   - Memory: 4 GB
   - Hard disk: 40 GB
   - Network adapter: Bridged

3. Download and install Docker on the virtual machine.

4. Create a new Docker image with the following Dockerfile:

   ```
    FROM ubuntu:20.04

   RUN apt-get update &&
   apt-get install -y python3 python3-pip &&
   pip3 install scapy paramiko twisted
   ```

5. Build and run the Docker container with the following commands:

   ```
    docker build -t my-router . docker run -d --name my-router -p 8080:8080 my-router
   ```

6. Test the Docker container by accessing the web interface at http://localhost:8080.

## 5.2 IP Addressing and Subnetting

IP addressing and subnetting are the processes of assigning and dividing IP addresses into smaller networks. IP addresses are 32-bit or 128-bit identifiers that uniquely identify devices on a network. Subnetting is the process of dividing a larger network into smaller subnetworks to improve security, performance, and management.

# 6 Product Delivery Plan

## 6.1 Python Router Program Outline

The Python router program outline is as follows:

- Import the required libraries and modules.
- Define the global variables, such as the network interfaces, the routing tables, and the packet queues.
- Define the packet processing functions, such as the receive function, the forward function, and the send function.
- Define the packet generation functions, such as the ping function, the traceroute function, and the DNS function.
- Define the packet handling functions, such as the error function, the timeout function, and the callback function.
- Define the packet scheduling functions, such as the priority queue function, the round-robin function, and the fair queue function.
- Define the packet monitoring functions, such as the statistics function, the logging function, and the tracing function.
- Define the packet management functions, such as the configuration function, the maintenance function, and the upgrade function.
- Define the packet security functions, such as the encryption function, the decryption function, and the authentication function.
- Define the packet optimization functions, such as the compression function, the fragmentation function, and the reassembly function.
- Define the packet testing functions, such as the unit test function, the integration test function, and the acceptance test function.
- Define the packet documentation functions, such as the help function, the manual function, and the tutorial function.

## 6.2 Dockerized Router Program Image

The Dockerized router program image is a packaged and containerized version of the software router that can be deployed and run on various platforms and environments. The Dockerized router program image includes the following components:

- The Linux image, which provides the operating system and network stack for the software router.

- The Python components, which implement the router's logic and functionality using Python scripts and modules.
- The configuration files, which define the network interfaces, the routing tables, and the packet queues.
- The documentation files, which provide the user manual, the tutorial, and the API reference.
- The test files, which provide the unit tests, the integration tests, and the acceptance tests.
- The license files, which define the terms and conditions of use and distribution.

## 6.3 Latex Product Presentation Document

The Latex product presentation document is a professional and polished document that provides an overview and a demonstration of the software router. The Latex product presentation document includes the following sections:

- Introduction, which provides an overview of the project and its objectives.
- Theory and Background, which explains the concepts and principles of networking, routing, and Python programming.
- Project Description, which describes the components and the structure of the software router.
- Setting up the Environment, which provides detailed instructions for setting up the VMWare and Dockerized Linux image.
- Product Delivery Plan, which outlines the Python router program outline, the Dockerized router program image, and the Latex product presentation document.
- Conclusion, which summarizes the project, its achievements, and its impact.

## 6.4 Product Presentation Video

The product presentation video is a multimedia and interactive document that provides a visual and auditory demonstration of the software router. The product presentation video includes the following elements:

- Introduction, which provides an overview of the project and its objectives.
- Theory and Background, which explains the concepts and principles of networking, routing, and Python programming.

- Project Description, which describes the components and the structure of the software router.
- Setting up the Environment, which provides detailed instructions for setting up the VMWare and Dockerized Linux image.
- Product Delivery Plan, which outlines the Python router program outline, the Dockerized router program image, and the Latex product presentation document.
- Demonstration, which shows the software router in action, processing packets, generating packets, handling packets, scheduling packets, monitoring packets, managing packets, securing packets, optimizing packets, testing packets, and documenting packets.

### 6.5 Marketing Website

The marketing website is a user-friendly and attractive website thatprovides information and resources about the software router. The marketing website includes the following pages:

- Home, which provides an introduction and a call to action.
- Features, which explains the benefits and the capabilities of the software router.
- Screenshots, which shows the software router in action, processing packets, generating packets, handling packets, scheduling packets, monitoring packets, managing packets, securing packets, optimizing packets, testing packets, and documenting packets.
- Download, which provides the download link and the installation instructions for the Dockerized router program image.
- Documentation, which provides the user manual, the tutorial, and the API reference.
- Support, which provides the contact information and the support options for the software router.
- About, which provides the background and the team of the software router.

## 7 Project Management and Product Delivery

### 7.1 Docker Desktop and Camtasia

Docker Desktop is a desktop application that enables users to develop, build, and run Docker containers on their local machine. Docker Desktop provides various tools and features, such as Docker Engine, Docker Compose, Docker Swarm, and Docker Machine, that enable users to manage and orchestrate their Docker containers.

Camtasia is a screen recording and video editing software that enables users to create and edit video content for their projects. Camtasia provides various tools and features, such as recording, editing, effects, transitions, and annotations, that enable users to produce high-quality and engaging videos.

To use Docker Desktop and Camtasia for project management and product delivery, follow these steps:

1. Download and install Docker Desktop and Camtasia on your local machine.

2. Create a new Docker image with the Dockerfile provided in the lab work book.

3. Build and run the Docker container with the Docker commands provided in the lab work book.

4. Test and evaluate the performance and scalability of the software router using the tools and techniques provided in the lab work book.

5. Record and edit a video presentation of the software router using Camtasia.

6. Publish and share the video presentation on various platforms and channels, such as YouTube, Vimeo, and LinkedIn.

## 7.2 Posting the Video and PDF

To post the video and PDF, follow these steps:

1. Upload the video presentation to a video hosting platform, such as YouTube, Vimeo, or Wistia.

2. Create a new PDF file with the Latex product presentation document provided in the lab work book.

3. Upload the PDF file to a file hosting platform, such as Google Drive, Dropbox, or OneDrive.

4. Share the video and PDF links on various platforms and channels, such as LinkedIn, Twitter, and GitHub.

## 7.3 Download Link for the Dockerized Product Image

The download link for the Dockerized product image is:

```
https://github.com/your-username/your-repo/releases/download/v1.0.0/your-image.tar.gz
```

To use the download link, follow these instructions:

1. Click on the download link.
2. Save the file to your local machine.
3. Extract the file using a file archiver, such as 7-Zip, WinRAR, or The Unarchiver.
4. Open a terminal or a command prompt.
5. Navigate to the directory where the extracted file is located.
6. Run the following command to start the Docker container:

   ```
   docker run -d --name your-container -p 8080:8080 your-image
   ```

7. Test and evaluate the performance and scalability of the software router using the tools and techniques provided in the lab work book.

## 8 Conclusion

In this project, we have built a software router on a Linux image using Python components. We have learned the fundamentals of networking, routing, and Python programming. We have designed and implemented a software router thatcan forward packets between different networks. We have tested and evaluated the performance and scalability of the software router. We have created a product delivery plan that includes the documentation, video presentation, and marketing website.

The significance of this project is that it provides a hands-on experience in building and deploying a network appliance using open-source software and tools. It also showcases the power and flexibility of Python as a programming language for networking and automation.

The future directions of this project are:

- To add more features and capabilities to the software router, such as VPN, firewall, and QoS.
- To integrate the software router with other network devices and services, such as switches, routers, and cloud platforms.
- To optimize the performance and scalability of the software router using various techniques, such as multithreading, multiprocessing, and caching.
- To test and validate the software router using various tools and frameworks, such as JMeter, Gatling, and Locust.
- To deploy and operate the software router in various environments and scenarios, such as data centers, edge networks, and IoT networks.