

OPENWRT/RASPAP INTEGRATED RASPBERRY PI ROUTER

A PROJECT REPORT

Submitted by

- 1. Gouri Nandana A - 21BCY10029**
- 2. Ben Tom Abey - 21BCY10035**
- 3. Bennet Binu Varghese - 21BCY10085**

*in partial fulfillment for the award of the degree
of*

BACHELOR OF TECHNOLOGY
in
COMPUTER SCIENCE AND ENGINEERING
(Cyber Security and Digital Forensics)



SCHOOL OF COMPUTING SCIENCE AND ENGINEERING
VIT BHOPAL UNIVERSITY
KOTHRI KALAN, SEHORE
MADHYA PRADESH – 466114

APRIL 2024

**VIT BHOPAL UNIVERSITY, KOTHRI KALAN, SEHORE
MADHYA PRADESH – 466114**

BONAFIDE CERTIFICATE

Certified that this project report titled **“OPENWRT/RASPAP INTEGRATED RASPBERRY PI ROUTER”** is the Bonafide work of **Gouri Nandana A (21BCY10029), Ben Tom Abey (21BCY10035), Bennet Binu Varghese (21BCY10085)** who carried out the project work under my supervision. Certified further that to the best of my knowledge the work reported at this time does not form part of any other project/research work based on which a degree or award was conferred on an earlier occasion on this or any other candidate.

PROGRAM CHAIR,
Dr. D. Saravanan,
Assistant Professor Sr
School of Computing Science
Engineering and Artificial
Intelligence (SCAI) VIT
Bhopal University.

PROJECT SUPERVISOR,
Dr. Rizwan Ur Rahman,
Assistant Professor
School of Computing Science
Engineering and Artificial
Intelligence
(SCAI) VIT
Bhopal
University.

The Capstone Project Examination is held on _____

ACKNOWLEDGEMENT

First and foremost, I would like to thank the Lord Almighty for His presence and immense blessings throughout the project work.

I wish to express my heartfelt gratitude to **Dr D. Saravanan**, Program Chair, Cyber Security and Digital Forensics for much of his valuable support and encouragement in carrying out this work.

I would like to thank my internal guide **Dr. Rizwan Ur Rahman**, for continually guiding and actively participating in my project, giving valuable suggestions to complete the project work.

I would like to thank all the technical and teaching staff of the School Computing Science and Engineering, who extended directly or indirectly all support.

Last, but not least, I am deeply indebted to my parents who have been the greatest support while I worked day and night for the project to make it a success.

LIST OF FIGURES AND GRAPHS

FIGURE NO.	TITLE	PAGE NO.
1	Raspberry Pi 4	12
2	USB Wifi Adapter	13
3	Micro SD Card	13
4	System Architecture Overview	16
5	Flow of network traffic through the pi	19
6	Manually assigning IP address in same subnet	22
7	Logging into OPENWRT/RASPAP/RASPAP via SSH	24
8	Env.sh Bash Script To Handle OPENWRT/RASPAP Configuration Files	28

ABSTRACT

Privacy is a fundamental individual right, even in the digital realm. However, awareness of the importance of online privacy is often underestimated by internet users. This research addresses this issue by implementing the Raspberry Pi 4 Model B as a portable VPN router. The research methodology involves observation and literature review. The results demonstrate that using the OpenWRT operating system, a portable VPN router based on Raspberry Pi can encrypt internet access, safeguarding privacy and providing better performance than the OpenVPN Connect application. This portable VPN router offers a practical solution to enhance user data security and network privacy in the digital world, especially when connected to public Wi-Fi networks. This research provides a better understanding of the significance of online privacy and presents an effective alternative to protect it. Using a Raspberry Pi-based portable VPN router, users can access the internet securely and maintain their privacy in the digital realm. The findings of this research contribute to the fields of information security and network technology and offer practical guidance for individuals to enhance their online privacy and security.

Keywords: Data Security, Portable VPN Router, Privacy, OpenWRT/RASPAP, OpenVPN, Raspberry Pi

TABLE OF CONTENTS

CHAPTER NO.	TITLE	PAGE NO.
	List of Figures and Graphs	1
	Abstract	2
1	CHAPTER-1: PROJECT DESCRIPTION AND OUTLINE 1.1 Introduction 1.2 Motivation for the work 1.3 Introduction to the project 1.4 Problem Statement 1.5 Objective of the work 1.6 Summary	5
2	CHAPTER-2: RELATED WORK INVESTIGATION 2.1 Introduction 2.2 Existing Work 2.3 Proposed Work	9
3	CHAPTER-3: REQUIREMENT ARTIFACTS 3.1 Hardware Components 3.2 Software Components	12
4	CHAPTER-4: DESIGN METHODOLOGY AND ITS NOVELTY 4.1 System Design / Architecture 4.2 Hardware Selection 4.2.1 Operating System Installation 4.2.2 Connectivity Configuration 4.2.3 VPN Integration 4.2.4 Network Wide AD-Blocking	16

	4.3 The Novelty of the Design	
5	CHAPTER-5: TECHNICAL IMPLEMENTATION & ANALYSIS 5.1 Implementation 5.2 Implementation Procedure	21
6	CHAPTER-6: PROJECT OUTCOME AND APPLICABILITY 6.1 Project Outcomes 6.2 Adaptability of the Project 6.3 Project applicability on Real-world applications 6.4 Inference	32
7	CHAPTER-7: CONCLUSIONS AND RECOMMENDATION 7.1 Conclusion of the Project	35
8	References	36

CHAPTER 1

PROJECT DESCRIPTION AND OUTLINE

1.1) INTRODUCTION

In today's digital age, the internet has become an essential part of daily life, offering convenience and accessibility across various domains (Wahyuni & Yel, 2022). However, the growing dependence on the internet brings significant challenges related to privacy and data security. Users connecting through public networks or free Wi-Fi are particularly vulnerable to risks such as eavesdropping and data theft (Lubis & Tarigan, 2017; Batubara et al., 2021; Lubis, Hariyanto, et al., 2022). Addressing these concerns requires robust solutions to safeguard personal information and ensure secure communications.

One effective method to enhance online security is the use of Virtual Private Networks (VPNs). VPNs enable users to transmit data over public networks as if operating on a private, secure network, providing protection against potential threats (Zakaria et al., 2022; Milsa Pratama et al., 2023). Beyond security, users also seek flexibility and mobility in their internet connectivity, ensuring secure access regardless of their location—be it at home, in the workplace, or on the go (Dutkowska-Zuk et al., 2022).

Open WRT, a Linux-based operating system tailored for routers and network devices, offers extensive customization and support for advanced networking features, including VPN integration (Rivera-Dourado et al., 2023). Paired with the Raspberry Pi, a compact, affordable, and versatile single-board computer, OPENWRT/RASPAP serves as a powerful platform for creating custom router solutions. This combination provides a cost-effective way to build a portable VPN router with robust security features, tailored to individual or organizational needs (Lubis, Septian, et al., 2022).

This project aims to design and implement an Open WRT-integrated Raspberry Pi router that focuses on flexibility, security, and ease of use. By configuring OPENWRT/RASPAP and integrating VPN services, the router provides users with greater control over their network and data. This study further evaluates the performance and stability of the proposed solution, highlighting its potential as a customizable and cost-efficient alternative for secure networking in various environments (Jadhav & Malode, 2019; Gentile et al., 2022; Osswald et al., n.d.).

1.2) MOTIVATION OF THE WORK

The primary motive of this project is to develop a customizable, cost-effective, and secure networking solution using OPENWRT/RASPAP and Raspberry Pi. As internet usage grows, so do concerns about privacy, data security, and the need for advanced networking features that are not always available in traditional commercial routers. This project aims to address these gaps by offering an affordable and flexible alternative that empowers users with greater control over their network.

Key objectives include:

1. **Enhanced Security:** To provide a secure networking environment by integrating advanced security features like VPNs and customizable firewalls. This ensures the protection of user data, particularly for those connecting through public or unsecured networks.
2. **Cost Efficiency:** To design a router that is significantly more affordable than commercial options while maintaining high performance and functionality.
3. **Customization and Flexibility:** To enable users to tailor their router settings to specific needs, such as ad-blocking, Quality of Service (QoS), and network monitoring, which are essential for home networks, small businesses, and specialized applications like IoT.
4. **Portability and Scalability:** To develop a portable solution that can be easily deployed in various environments, from home networks to small offices, and scaled to meet growing demands.
5. **User Empowerment:** To promote the use of open-source technology, giving users the freedom to experiment, modify, and optimize their network infrastructure according to their preferences.

This project not only aims to create a functional router but also seeks to demonstrate the potential of open-source tools in solving real-world networking challenges effectively and affordably.

1.3) ABOUT INTRODUCTION TO THE PROJECT

An Open WRT-integrated Raspberry Pi router is a powerful, customizable, and cost-effective solution for managing your network. Open WRT is an open-source firmware that turns your Raspberry Pi into a fully functional router with advanced features like firewall management, VPN support, and network monitoring.

This setup is ideal for home networks, small offices, or specialized applications such as IoT or ad-blocking. With the flexibility of Open WRT and the affordability of the Raspberry Pi, you can build a compact router tailored to your specific needs, offering robust performance and extensive customization options.

1.4) PROBLEM STATEMENT

In the current digital era, reliable and secure internet access is a fundamental requirement for individuals and businesses. However, commercial routers often come with limitations such as high costs, restricted customization options, and inadequate security features. Users relying on public networks or free Wi-Fi are particularly vulnerable to data breaches and cyber threats, posing significant risks to their privacy and sensitive information.

Moreover, the lack of flexibility in traditional routers makes it challenging to implement advanced networking features like VPNs, ad-blocking, and detailed network monitoring, which are increasingly necessary for modern network management. There is a pressing need for an affordable, customizable, and secure networking solution that can cater to the unique requirements of different users, from home environments to small offices and specialized applications.

This project addresses these challenges by leveraging Open WRT's open-source firmware and the Raspberry Pi's versatility to create a cost-effective and highly customizable router. The goal is to provide a solution that not only enhances security and performance but also offers the flexibility to meet diverse networking needs.

1.5) OBJECTIVE OF THE WORK

The objective of this project is to design and implement a customizable, cost-effective, and secure router solution using OPENWRT/RASPAP and Raspberry Pi. The primary goals include:

1. **Enhance Network Security:** To integrate advanced security features such as Virtual Private Network (VPN) support and customizable firewalls to protect user data from cyber threats, especially on public or unsecured networks.
2. **Provide Cost-Effective Solutions:** To develop an affordable alternative to commercial routers without compromising on performance or functionality.

3. **Enable Advanced Customization:** To offer users the ability to configure and optimize their network settings, including features like ad-blocking, Quality of Service (QoS), and network traffic monitoring, tailored to their specific needs.
4. **Improve Portability and Scalability:** To create a portable router solution that can be deployed in various environments, from home networks to small businesses, and easily scaled to accommodate growing demands.
5. **Promote Open-Source Technology:** To encourage the adoption of open-source tools, providing users with greater control, flexibility, and the ability to contribute to and benefit from a global community of developers.

1.6) SUMMARY

This project seeks to develop a secure, flexible, and cost-efficient router by integrating Open WRT with Raspberry Pi. Traditional routers often lack advanced features and customization, which exposes users to privacy and security risks, especially on public networks. By harnessing the open-source power of OPENWRT/RASPAP and the affordability of Raspberry Pi, the project delivers a portable router packed with features such as VPN support, customizable firewalls, and network monitoring. This solution addresses the diverse needs of home users and small businesses, providing a robust, economical alternative that enhances security and control.

CHAPTER 2

RELATED WORK INVESTIGATION

2.1) INTRODUCTION

The growing demand for secure and customizable network solutions has driven extensive research into open-source platforms like OPENWRT/RASPAP. Open WRT, known for its flexibility and robust feature set, has been widely used to enhance the functionality of standard networking devices. Its adaptability allows users to implement advanced features such as VPNs, traffic monitoring, and firewall customization, addressing the limitations of traditional routers.

In parallel, the Raspberry Pi has emerged as a cost-effective and versatile hardware platform capable of supporting various networking applications. Combining Open WRT with Raspberry Pi has enabled the development of highly customizable, secure, and portable routers. Previous studies have demonstrated the effectiveness of this integration in improving network security, performance, and user control, particularly for home users and small businesses.

This investigation reviews existing work in the field, focusing on the implementation of Open WRT-based solutions on Raspberry Pi. It aims to identify the strengths, limitations, and potential areas for improvement in current approaches, providing a foundation for further innovation in this domain.

2.2) EXISTING WORK

The integration of OPENWRT/RASPAP with Raspberry Pi has been explored in various studies and projects, highlighting its potential to provide cost-effective and customizable network solutions. Researchers and developers have successfully utilized Open WRT to transform Raspberry Pi into a fully functional router, enabling advanced features like VPN support, dynamic routing, traffic shaping, and ad-blocking.

One significant area of focus in existing work is enhancing network security. Projects have demonstrated the effective use of VPNs on OPENWRT/RASPAP-powered Raspberry Pi devices to secure data

transmission over public networks. Studies have also explored the implementation of robust firewall configurations to protect against unauthorized access and cyber threats.

Another key aspect of previous work is improving network performance and flexibility. Configurations such as Quality of Service (QoS) for bandwidth management and detailed network monitoring tools have been implemented to optimize network usage. Additionally, the modular nature of Open WRT has enabled developers to integrate custom packages, providing tailored solutions for specific applications, including IoT device management and content filtering.

While these projects have showcased the feasibility and advantages of using OPENWRT/RASPAP on Raspberry Pi, challenges remain in areas such as scalability, performance under high network loads, and ease of setup for non-technical users. This project builds on these existing efforts, aiming to address these limitations and further enhance the functionality and usability of OPENWRT/RASPAP-integrated Raspberry Pi routers.

2.3) PROPOSED WORK

The proposed work seeks to develop an advanced, cost-effective, and secure router solution by integrating OPENWRT/RASPAP with a Raspberry Pi. Building on existing research, this project aims to enhance the functionality of Raspberry Pi-based routers while addressing existing limitations.

Key objectives of the proposed work include:

1. **Improved Network Security:** The project will focus on strengthening security by integrating features such as Virtual Private Network (VPN) support, custom firewall configurations, and secure data transmission protocols to protect users from cyber threats, particularly in public or unsecured networks.
2. **Optimized Performance and Efficiency:** Performance will be optimized by configuring network traffic prioritization (QoS), resource allocation, and load balancing to ensure efficient handling of high network traffic and prevent bottlenecks, especially in environments with multiple connected devices.
3. **Customizable and Scalable Architecture:** The router will be designed with scalability in mind, allowing for easy upgrades and the integration of additional services (e.g., ad-blocking,

network monitoring, or IoT device management) based on user requirements. The modular architecture will allow users to tailor the system to their needs.

4. **Simplified User Management Interface:** A streamlined, user-friendly interface will be developed to simplify router configuration, monitoring, and troubleshooting. The interface will be intuitive for both technical and non-technical users, ensuring broad accessibility.
5. **Portable and Compact Design:** The portable nature of Raspberry Pi will be leveraged to ensure that the router can be deployed in a variety of environments, from home networks to small office settings, without the need for bulky hardware.

The proposed work aims to deliver a versatile, user-centric, and secure networking solution that not only meets the needs of today's digital users but also anticipates future requirements in an ever-evolving tech landscape.

CHAPTER 3

REQUIRED ARTIFACTS

3.1) HARDWARE COMPONENTS

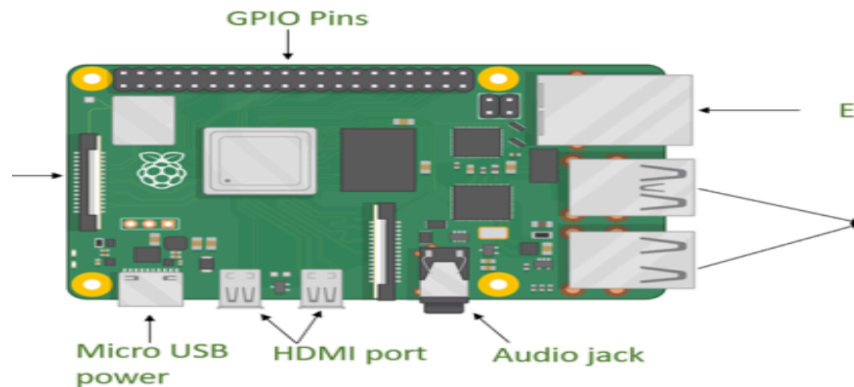


Figure 1 - Raspberry Pi 4

3.1.1) RASPBERRY PI 4: The Raspberry Pi 4 is a powerful, low-cost single-board computer that has become popular for a wide range of applications, including education, DIY projects, and professional development. It features a quad-core ARM Cortex-A72 processor, up to 8GB of RAM, multiple USB ports, HDMI output, and support for dual monitors. With its improved performance compared to previous models, the Raspberry Pi 4 is ideal for resource-intensive tasks, such as running servers, building IoT devices, and powering custom networking solutions. Its compact size, affordability, and versatility make it a perfect choice for projects like creating a secure and customizable router.



Figure 2 - USB Wifi Adapter

3.1.2) USB WI-FI ADAPTER: A USB wireless adapter is essentially a wireless network card that is used to access a network through a USB port on your computer. Wireless adapters allow your computer to connect to a wireless network for the internet in the absence of an internal network card. It typically plugs into a USB port and provides Wi-Fi connectivity by enabling the device to communicate with wireless routers or access points. USB Wi-Fi adapters are commonly used when the built-in wireless capabilities of a device are insufficient, or when users need to upgrade to a faster or more reliable Wi-Fi standard (e.g., from 802.11n to 802.11ac or 802.11ax).



Figure 3 - Micro SD Card

3.1.3) MICRO SD CARD: A microSD card is a small, portable storage device commonly used in devices like smartphones, cameras, and single-board computers such as the Raspberry Pi. The microSD card serves as the primary storage medium where the operating system (like Raspberry Pi OS or OPENWRT/RASPAP) and any required software are stored. It also provides the necessary space for saving configurations, scripts, and other files crucial to the functioning of the Raspberry Pi. The performance of the microSD card, such as its read and write speeds, can significantly impact the

overall performance of the Raspberry Pi, especially in tasks requiring high data throughput or frequent file access.

3.2) SOFTWARE COMPONENTS

3.2.1) OPEN WRT: Open WRT is an open-source, Linux-based operating system designed primarily for routers and embedded devices. It offers a highly customizable platform with advanced networking features, including firewall management, VPN support, and quality of service (QoS) control. OPENWRT/RASPAP is known for its flexibility, allowing users to install additional software packages to extend functionality, making it an ideal choice for creating custom networking solutions. It supports a wide range of devices, from consumer routers to single-board computers like the Raspberry Pi, offering enhanced control and security for network management.

3.2.2) ADGUARD HOME: AdGuard Home is an open-source network-wide ad blocker and privacy protection tool that functions as a DNS server. It helps block ads, trackers, and malicious websites by intercepting DNS requests and filtering out unwanted content before it reaches the user's device. AdGuard Home can be deployed on various devices, including routers, Raspberry Pi, or servers, providing centralized control over network traffic and filtering.

By setting up AdGuard Home on a network, users can prevent ads from appearing on websites, apps, and even streaming services, while also improving privacy by blocking tracking domains and phishing sites.

3.2.3) OPEN VPN: OpenVPN is an open-source virtual private network (VPN) solution that enables secure point-to-point or site-to-site connections in routed or bridged configurations and remote access facilities. It uses robust encryption standards such as AES for data protection and supports various authentication methods, including certificates, username/password, or two-factor authentication. It supports both IPv4 and IPv6, is highly configurable for different use cases (e.g., secure internet browsing, remote work, or accessing private networks), and works well in environments with restricted network conditions, including NAT (Network Address Translation) traversal and firewall bypass.

When deployed on devices like Raspberry Pi or integrated into OPENWRT/RASPAP-based routers, OpenVPN allows users to create secure, encrypted tunnels over the internet, ensuring privacy and data integrity for their online activities.

3.2.4) PYTHON AND GPIO LIBRARIES: Python, a versatile and easy-to-learn programming language, is widely used in conjunction with the Raspberry Pi for various hardware-related projects. One of the key libraries used for interacting with the GPIO (General Purpose Input/Output) pins on the Raspberry Pi is the RPi.GPIO library. This library allows users to control the pins for various tasks, such as reading digital inputs from sensors, controlling LEDs, or interfacing with other electronic components like motors and relays.

RPi.GPIO provides simple functions to configure the pins as input or output, read signals from sensors (e.g., temperature or motion sensors), and send signals to actuators (e.g., turning on an LED or triggering a relay). This makes it essential for Raspberry Pi projects that involve hardware control and automation.

3.2.5) PARTITION MANAGEMENT SOFTWARE: Partition management software is used to organize and optimize storage devices by creating, resizing, deleting, or merging disk partitions. Popular tools like GParted, EaseUS Partition Master, and MiniTool Partition Wizard offer user-friendly interfaces and a variety of features to manage partitions on different operating systems. These tools help users efficiently allocate disk space, ensure smooth data management, and maintain system performance, especially for tasks like multi-boot setups or managing storage on devices like Raspberry Pi.

CHAPTER 4

DESIGN METHODOLOGY AND ITS NOVELTY

4.1) SYSTEM DESIGN / ARCHITECTURE

The design methodology for the Open WRT-integrated Raspberry Pi router emphasizes the importance of merging easily obtainable hardware with flexible open-source software. This combination is aimed at creating a networking solution that is not only secure and private but also highly efficient. This new solution enables users to convert the Raspberry Pi, a small and inexpensive computer, into a multipurpose router. When connected with OPENWRT/RASPAP, the Raspberry Pi provides features such as VPN tunneling, which encrypts users' internet connections to help protect their online activity. It also allows you to set up a Local Area Network (LAN), which makes it easier to communicate and share files across connected devices. Another useful feature is network-wide ad blocking, which enhances the browsing experience by reducing bothersome adverts. Overall, this design process gives users control over their networking needs, resulting in a solution that is both versatile and user-friendly.

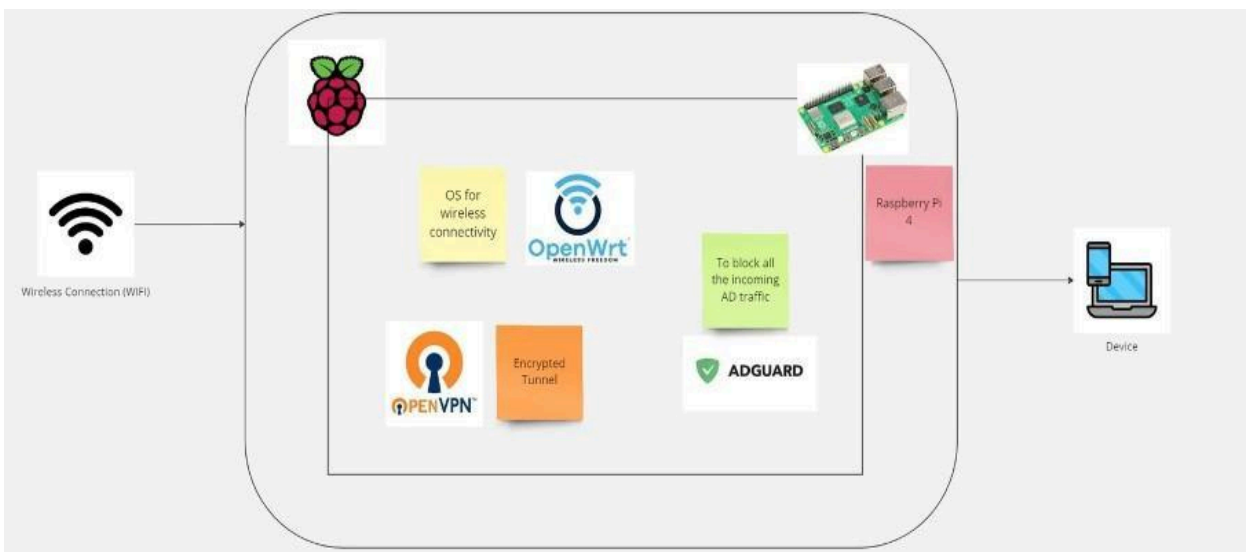


Figure 4 - System Architecture Overview

4.2) HARDWARE SELECTION

For this project, the Raspberry Pi 3B or 4 has been chosen as the main hardware component. These models are well-regarded for their ability to handle various tasks efficiently, providing enough processing power to run applications smoothly. Additionally, they come with multiple connectivity options, such as USB ports, HDMI, and Wi-Fi, which allow for easy connections to other devices and networks. This versatility is essential for enhancing the project's capabilities.

4.2.1) OPERATING SYSTEM INSTALLATION

To store the operating system and essential files, an SD card is used as the primary storage medium. If adjustments are required or if new features need to be added, swapping out the SD card or modifying its contents can be done quickly and efficiently.

Open WRT is a lightweight and highly flexible operating system created primarily for networking devices. This operating system is very beneficial for developing and managing complex networking capabilities. The first step in setting up OPENWRT/RASPAP is to install it on an SD card. This method entails utilizing specialist tools such as Balena Etcher or Raspberry Pi Imager to facilitate the installation and guarantee that Open WRT is properly set on the card. Once the installation is complete, plug the SD card into the Raspberry Pi to create a powerful and dependable base for completing complex routing tasks and improving overall network

4.2.2) CONNECTIVITY CONFIGURATION

The Raspberry Pi operates on two main components that facilitate internet access and secure networking for connected devices. The first component is the Raspberry Pi's built-in Wi-Fi module. This module allows the device to connect to public Wi-Fi networks that are available in various locations, such as coffee shops, libraries, or other places that provide free internet access. When the Raspberry Pi connects to these external Wi-Fi networks, it serves as the primary source of internet access for the device itself.

In addition to the Wi-Fi connection, there is an important second component involved in establishing a secure network. An external network adapter can be connected to the Raspberry Pi through a USB

port. This external adapter functions as a Local Area Network (LAN) gateway. By doing so, it creates a dedicated and private network that allows other devices to connect securely. This setup ensures that while the Raspberry Pi accesses the public internet through Wi-Fi, it can also maintain a safe and controlled environment for any devices that join the private network created by the external adapter. This combination of the built-in Wi-Fi and the external network adapter effectively balances public connectivity with private security.

4.2.3) VPN INTEGRATION

This involves configuring the external network adapter to manage internet connections securely. The setup ensures that all data sent and received over the internet is directed through OpenVPN, which is a popular and reliable virtual private network service. By using OpenVPN, the network adapter creates a secure and encrypted channel for communication, which guards against various potential online threats. These threats can include hackers trying to access personal information, as well as various forms of surveillance that could compromise user privacy. The encryption process helps in encoding the data, making it difficult for unwanted parties to decipher or misuse the information being transmitted. Ultimately, this integration is essential for maintaining a high level of security and privacy for users while they browse the internet.

4.2.4) NETWORK WIDE AD-BLOCKING

AdGuard Home is set up on a Raspberry Pi to provide complete ad blocking for all devices connected to the local area network. By installing this software on the Raspberry Pi, users can effectively filter out unwanted advertisements and trackers, enhancing their online experience. Every device that connects to the home network, including smartphones, tablets, computers, and smart TVs, benefits from a cleaner and faster browsing experience. The use of AdGuard Home not only improves the speed of internet browsing by reducing the amount of data loaded but also offers an added layer of privacy. Users can enjoy a more streamlined online experience without the distractions and potential risks associated with intrusive ads.

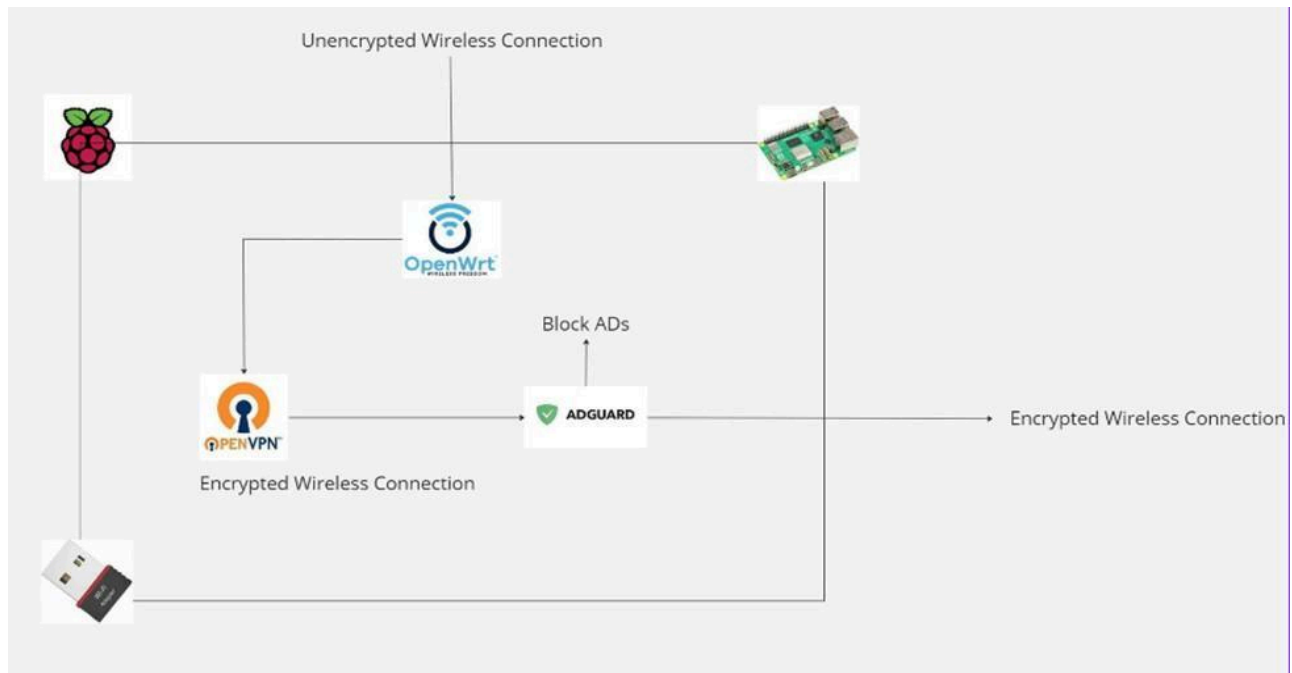


Figure 5 – Flow of Network Traffic through the Pi

4.3) THE NOVELTY OF THE DESIGN

This design uses an innovative combination of existing technologies to improve functionality:

- 1) **Cost-Effective Security system:** By combining affordable hardware and open-source software, this system provides a low-cost yet powerful alternative to commercial routers with similar functionality.
- 2) **Comprehensive Privacy:** The integration of OpenVPN ensures that all traffic is encrypted, providing users with improved privacy and security, particularly in public Wi-Fi environments.
- 3) **Customizable Functionality:** Using OPENWRT/RASPAP and AdGuard Home, customers can tailor the router's behavior to their unique requirements, such as ad filtering and sophisticated network configurations.
- 4) **Compact and Portable Design:** The compact nature of the Raspberry Pi makes this router highly portable, ideal for use at home or on the go.

The novelty of this OPENWRT/RASPAP-integrated Raspberry Pi router project lies in its ability to merge affordability, portability, and advanced networking features into a compact and versatile solution.

By leveraging the computational capabilities of a Raspberry Pi, coupled with the power of open-source software such as OPENWRT/RASPAP/RASPAP, AdGuard Home, and OpenVPN, this design offers functionalities typically reserved for high-end commercial routers. The project prioritizes user privacy through VPN tunneling, ensuring secure and encrypted internet access even when connecting to public Wi-Fi networks.

CHAPTER 5

TECHNICAL IMPLEMENTATIONS AND ANALYSIS

5.1) IMPLEMENTATION

A standard SanDisk 64gb high endurance SD card can be used to image the Open WRT OS. However, the root partition on the SD card used for OPENWRT/RASPAP is set to a default size of approximately 100 MB. This size is generally not enough when trying to install additional software or carry out various operations that require more space. To solve this issue, it is essential to expand the partition so that it can make use of the full storage capacity available on the SD card. Using GParted on a Linux system is one of the best approaches, as it provides an easy-to-use interface for managing disk partitions as follows:

- 1) Using an SD card reader, insert the SD card into the Linux computer. Then, open the graphical disk partitioning utility GParted.
- 2) Determine the partitions on the SD card. Generally speaking, it will appear as /dev/mmcblk0. We must find the partition that matches the root partition, which is /dev/mmcblk0p2.
- 3) To unmount the root partition (/dev/mmcblk0p2), right-click on it and choose Unmount. This makes the partition unusable and allows for safe modification after it is unmounted.
- 4) Once more, right-click on /dev/mmcblk0p and choose Resize/Move. To assign the remaining unpartitioned space, drag the slider on the right to the very end of the drive.
- 5) To validate the modifications, click Resize/Move and click on the toolbar's green checkmark icon.
- 6) Now we can insert the SD card into the Pi and connect the Raspberry Pi to power ethernet. Any standard ethernet cable such as a cat5e or cat6 and higher can be connected directly to our laptop/computer.

After you have completed the setup process for the Raspberry Pi router and it is powered on, you will be able to access the router using its default IP address, which is 192.168.1.1. This address serves as the gateway to the router's configuration interface. Users have a couple of different options

for how they can configure the router to meet their specific needs or preferences. One method may involve using a web browser to enter the IP address, allowing you to interact with a graphical interface where settings can be modified. Another method might include command-line tools or configuration files that can be edited directly, offering a more hands-on approach to changing the router's settings. In case, the IP of the local device (laptop) is in a different IP address range, then go to the control panel and manually change the IP address.

5.2) IMPLEMENTATION PROCEDURE

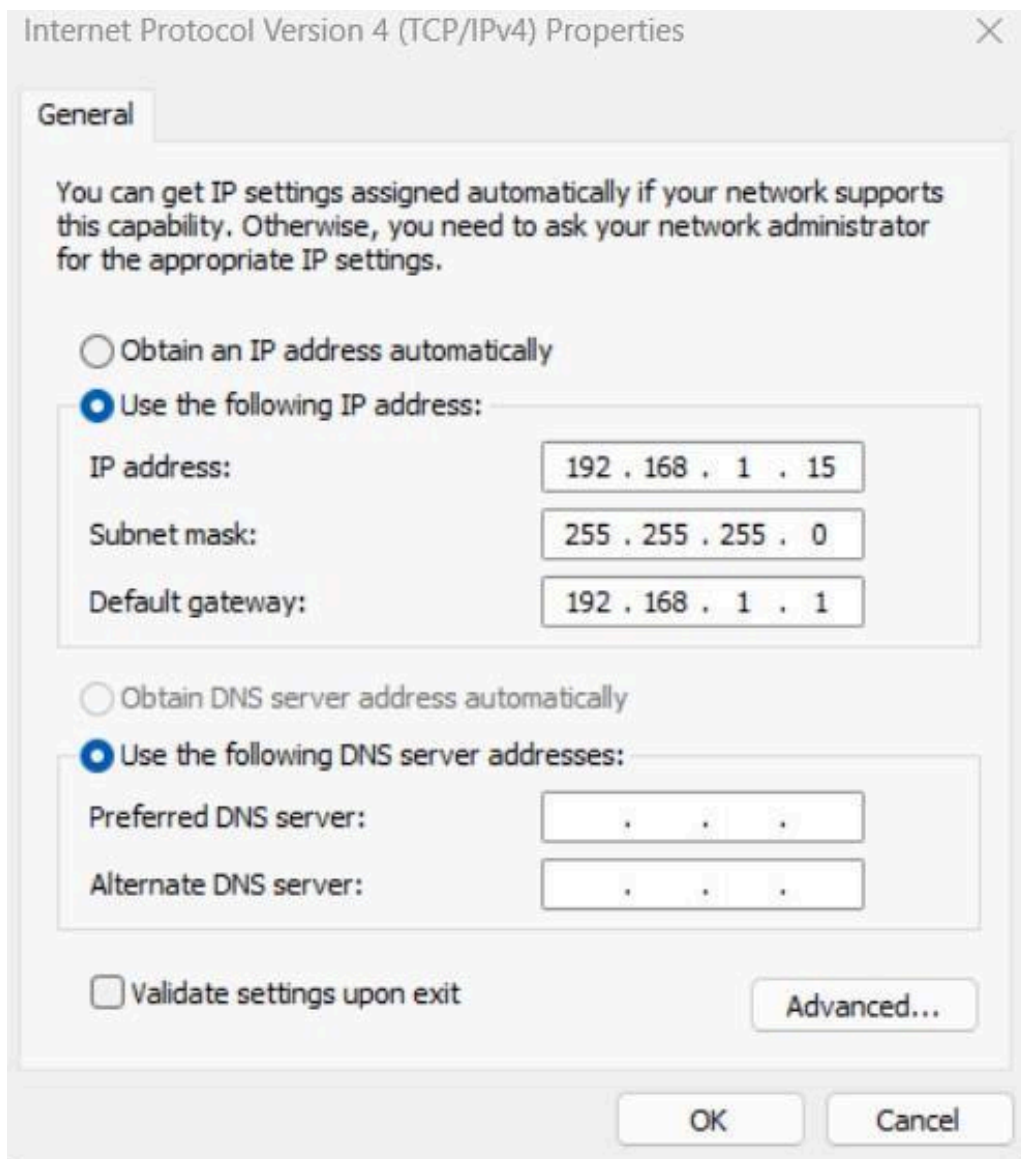


Figure 5 – Manually Assigning IP Address in same Subnet

1. Access via SSH:

- Connect your computer to the Raspberry Pi's LAN network.
- Open a terminal on your computer.
- Use the following command to initiate an SSH session:
ssh root@192.168.1.1
- When prompted, enter the default password: OPENWRT/RASPAP.

2. Access via Web Interface:

- Connect your computer or device to the router's LAN network.
- Open your preferred web browser.
- Enter the URL: **http://192.168.1.1**.
- Log in with the following credentials:
 - Username: root
 - Password: OPENWRT/RASPAP

3. Change Default Credentials: For security, immediately change the default password using either the command line or the web interface.

Via SSH: Use the passwd command.


```
root@localhost:/home/bharath
root@localhost:/home/bharath x bharath@localhost:~ x
#!/bin/bash
#Custom script created to set up the environment and subnet
#22112024-12:i02
#192.168.1.1 to 10.71.71.1

file1='network'
file2='firewall'

#check file existence and backup to roll back if error happens
if [ -f "/etc/config/$file1" ]
then
    cp "/etc/config/$file1" "/etc/config/${file1}.backup"
fi
if [ -f "/etc/config/$file2" ]
then
    cp "/etc/config/$file2" "/etc/config/${file2}.backup"
fi

for each in `ls -l /etc/config`; do
    if [ $each == $file1 ]
    then
        echo 'default network file found.'
        #remove generic network file and move
        #custom config to /etc/config directory

        rm /etc/config/$file1
    end
    else if [ $each == 'firewall' ]
    then
        echo 'default firewall file found.'
        #remove generic firewall file and move
        #custom firewall config to /etc/config

        rm /etc/config/$file2
    fi
fi

1,1 Top
```

Figure 6 – ENV.SH BASH SCRIPT TO HANDLE OPENWRT/RASPAP CONFIGURATION FILES

```
root@localhost:/home/bharath
root@localhost:/home/bharath x bharath@localhost:~ x
    cp "/etc/config/$file2" "/etc/config/${file2}.backup"
fi

for each in `ls -l /etc/config`; do
    if [ $each == $file1 ]
    then
        echo 'default network file found.'
        #remove generic network file and move
        #custom config to /etc/config directory

        rm /etc/config/$file1
    end
    else if [ $each == 'firewall' ]
    then
        echo 'default firewall file found.'
        #remove generic firewall file and move
        #custom firewall config to /etc/config

        rm /etc/config/$file2
    fi
fi

#move custom configurations to /etc/config directory
if [ -f "/home/bharath/$file1" ] && -f "/home/bharath/$file2" ]
then
    mv /home/bharath/network /etc/config
    mv /home/bharath/firewall /etc/config
else
    echo "error.. file not presnet"
    exit 1
fi

#maintain log of changes and updates
LOGFILE='/var/log/routerConfig.log'
echo "$(date) - updated $file1 and $file2" >> $LOGFILE

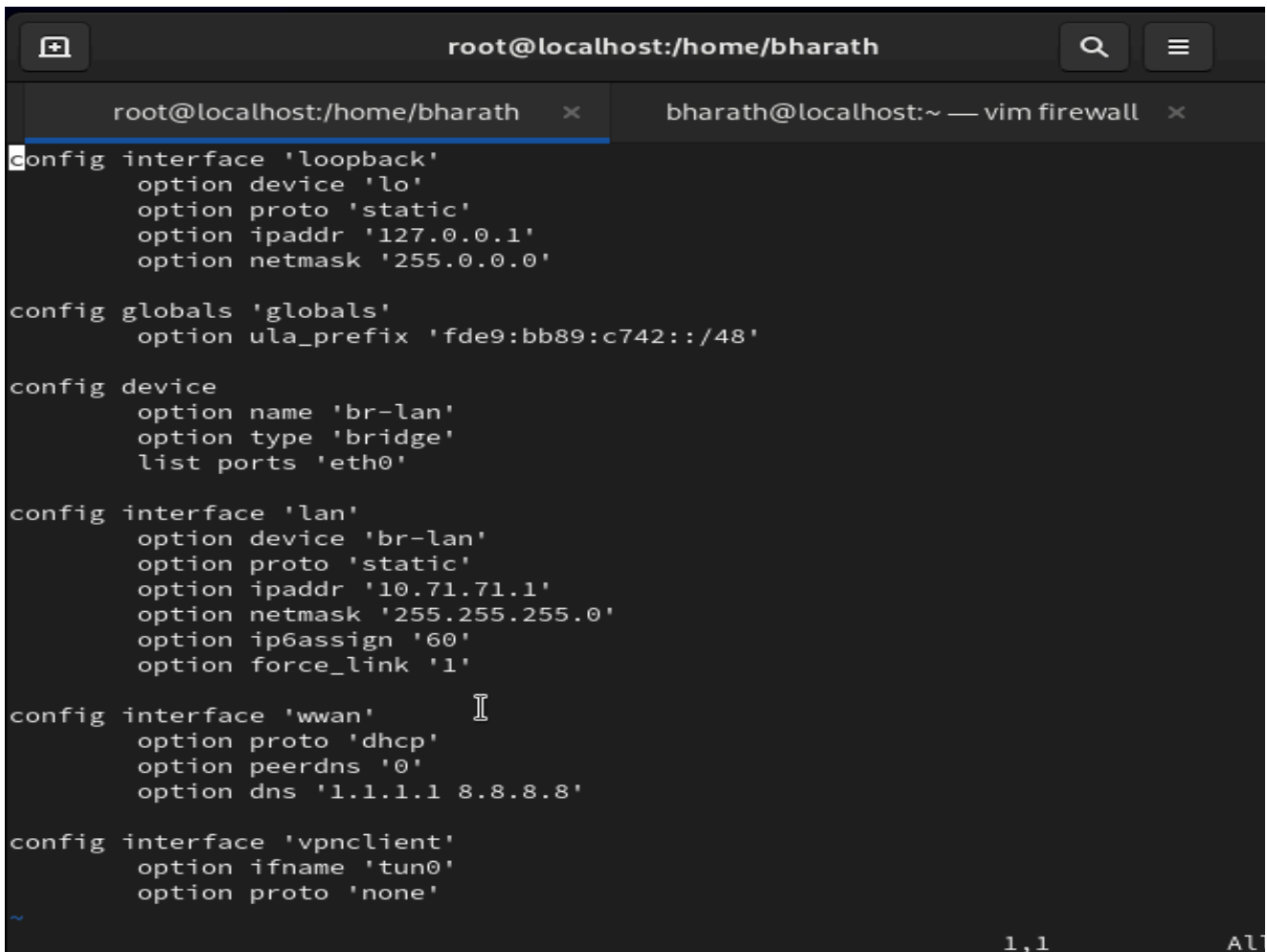
#reboot router to reflect changes and revoke ssh connection
reboot

54,1 Bot
```

Figure 7 - ENV.SH SCRIPT TO HANDLE OPENWRT/RASPAP CONFIGURATION FILES (continuation)

5. Create your custom network configuration file with the following settings:

- config interface 'loopback': The section configures the router's loopback interface for internal communication. It specifies device name (lo), static IP Address (127.0.0.1) which is the standard local address, and subnet mask, allowing access to 127.x.x.x addresses.
- config globals 'globals': Specifies the Unique Local Address (ULA) IPv6 prefix. This prefix is used for private IPv6 addressing within the local network.
- config device: The router's physical or virtual network device is identified by the 'br-lan' option, which specifies a bridge for the local area network, and the ethernet interface(eth0) is added to the bridge.
- config interface 'lan': The section configures the LAN interface, connecting internal devices to the router. It specifies device (br-lan), static IP address (10.71.71.1), and netmask (/24). The LAN interface uses the router's IP, and the prefix size is set to 60 bits.
- config interface 'wwan': The section configures the WAN interface for internet connection, setting options to get ip assigned dynamically by the DHCP and use the cloudflare dns (1.1.1.1) or the google dns (8.8.8.8).
- config interface 'vpn client': The section configures a VPN client interface, specifying tun0 as the interface name for VPN tunnels and protocol as 'none', which doesn't require an IP protocol configuration.



```
root@localhost:/home/bharath
bharath@localhost:~ — vim firewall
config interface 'loopback'
    option device 'lo'
    option proto 'static'
    option ipaddr '127.0.0.1'
    option netmask '255.0.0.0'

config globals 'globals'
    option ula_prefix 'fde9:bb89:c742::/48'

config device
    option name 'br-lan'
    option type 'bridge'
    list ports 'eth0'

config interface 'lan'
    option device 'br-lan'
    option proto 'static'
    option ipaddr '10.71.71.1'
    option netmask '255.255.255.0'
    option ip6assign '60'
    option force_link '1'

config interface 'wwan'
    option proto 'dhcp'
    option peerdns '0'
    option dns '1.1.1.1 8.8.8.8'

config interface 'vpnclient'
    option ifname 'tun0'
    option proto 'none'

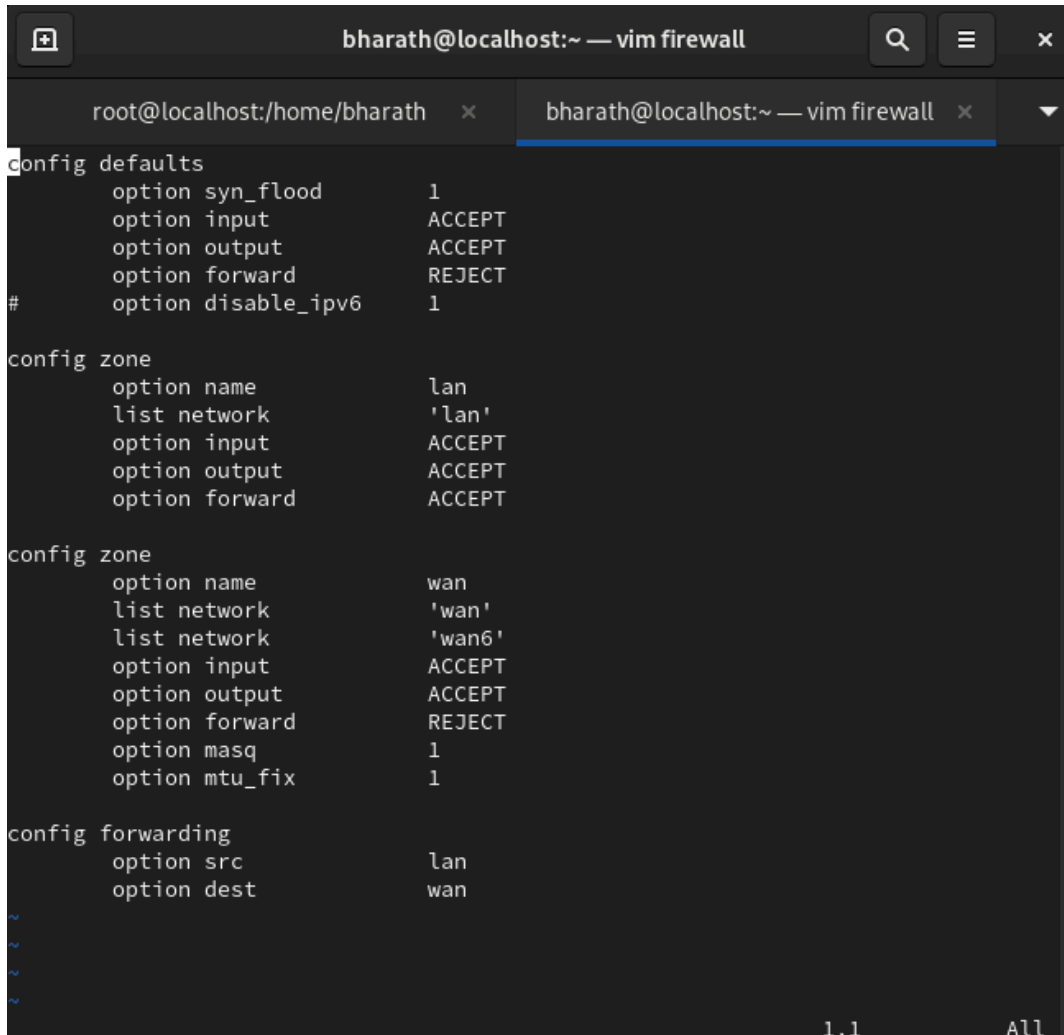
~
1,1 ALT
```

Figure 8 – Network Configuration File

6. Create your custom firewall configuration file with the following settings:

- Defaults Section: Configures the default behaviour of the firewall. Incoming and outgoing traffic is allowed, network traffic forwarding is rejected. DOS attacks like SYN FLOOD attacks are mitigated (syn_flood 1).
- LAN Zone Configuration: The LAN zone is identified by its name, associated with a network interface, and allows incoming, outgoing, and forwarding traffic.
- WAN Zone Configuration: The WAN zone is identified by its name, associated with the WAN and WAN6 network interface. Input and output allow incoming and outgoing traffic, respectively. Network Address Translation (mask 1) and packet fragmentation correction (mtu_fix) are enabled.

- Forwarding Configuration: This rule specifies the source zone for traffic forwarding (LAN) and the destination zone for forwarded traffic (WAN), enabling LAN devices to communicate with the internet.



```
config defaults
    option syn_flood      1
    option input          ACCEPT
    option output         ACCEPT
    option forward        REJECT
#    option disable_ipv6  1

config zone
    option name           lan
    list network          'lan'
    option input          ACCEPT
    option output         ACCEPT
    option forward        ACCEPT

config zone
    option name           wan
    list network          'wan'
    list network          'wan6'
    option input          ACCEPT
    option output         ACCEPT
    option forward        REJECT
    option masq           1
    option mtu_fix        1

config forwarding
    option src            lan
    option dest           wan

~
~
~
~
```

1,1 All

Figure 9 – Firewall Configuration File

7. Create your custom wireless interface configuration file with the following settings:

Wireless Device Configuration (radio0): The configuration file for the wireless hardware device interface defines the device's configuration options, including the wireless channel 7 (operates in the 2.4GHz band), driver type (mac80211), hwmode set to 11g (operates using the 802.11g standard), hardware path, and Short Guard Interval for 40MHz channels.

Wireless Interface Configuration (default_radio0): The config wifi-iface 'default_radio0' defines a wireless interface associated with radio0, with options for device, network, mode, SSID, and encryption. It is part of the LAN network and what the user will connect to, so that we get private secure internet access.

```

config wifi-device 'radio0'
    option type 'max80211'
    option channel '7'
    option hwmode '11g'
    option path 'platform/soc/fe300000.mnncnr/mmc_host'
    option disabled '0'
    option short_gi_40 '0'

config wifi-iface 'default_radio0'
    option device 'radio0'
    option network 'lan'
    option mode 'ap'
    option ssid 'OpenWrt'
    option encryption 'none'

```

"wireless" 14L, 339B

Figure 10 – Wireless Interface Configuration File

When we run the `env.sh` script again, it will cause us to lose our connection to the Raspberry Pi. This disconnection occurs because the router is restarting to apply the changes made using the custom scripts we set up earlier. After the router has finished rebooting, the IP address assigned to it should now be updated to the new address we specified in our network configuration file, which is

10.71.71.1. To confirm that the changes have taken effect, we can either open a web browser, enter the new IP address or use SSH to connect directly to 10.71.71.1. At this stage, the Raspberry Pi should show that it has an active WAN connection through its internal Wi-Fi adapter, as well as a LAN connection via its LAN port.



Figure 11 – All the materials used for the project

OPENWRT/RASPAP

```
C:\Users\agent>ssh root@192.168.1.1
Subnet Mask . . . . . : 255.255.255.0
Default Gateway . . . . . : fe80::8ec7:c3ff:fec7:c67b%19
                        192.168.1.1

C:\Users\agent>ssh root@192.168.1.1
The authenticity of host '192.168.1.1 (192.168.1.1)' can't be established.
ED25519 key fingerprint is SHA256:pJSGdeZUKFHSDSzW3rICSiMfGEtVnylEWxIJnXdFbAg.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.1.1' (ED25519) to the list of known hosts.
```

BusyBox v1.36.1 (2025-02-03 23:09:37 UTC) built-in shell (ash)

```
- - - W I R E L E S S F R E E D O M
```

OpenWrt 24.10.0, r28427-gdf0e3d02a

==== WARNING! =====
There is no root password defined on this device!
Use the "passwd" command to set up a new password
in order to prevent unauthorized SSH logins.
=====

root@OpenWrt:~# client_loop: send disconnect: Connection reset

```
C:\Users\agent>
```

Figure 12 – OPENWRT/RASPAP ssh command

```

root@OpenWrt: /etc/config #
|-----| | |-----| |-----| |-----| |-----|
|_|_| W I R E L E S S   F R E E D O M
-----
OpenWrt 24.10.0, r28427-gdf0e3d02a
-----

root@OpenWrt:~# cd /etc/config
root@OpenWrt:/etc/config# ls
dhcp      dropbear  firewall  luci      network  rpcd      system  uhttpd  wireless

root@OpenWrt:/etc/config# ls -la
-rash: la: not found
root@OpenWrt:/etc/config# ls -la
drwxr-xr-x 1 root root      1024 Feb 3 23:09 .
drwxr-xr-x 1 root root      1024 Feb 3 23:14 ..
-rw-r--r-- 1 root root      1029 Feb 3 23:09 dhcp
-rw-r--r-- 1 root root       221 Feb 3 23:09 dropbear
-rw-r--r-- 1 root root     4066 Feb 3 23:09 firewall
-rw-r--r-- 1 root root       968 Feb 3 23:09 luci
-rw-r--r-- 1 root root      428 Feb 3 23:09 network
-rw-r--r-- 1 root root      167 Feb 3 23:09 rpcd
-rw-r--r-- 1 root root      358 Feb 3 23:09 system
-rw-r--r-- 1 root root      783 Feb 3 23:09 uhttpd
-rw-r--r-- 1 root root      365 Feb 3 23:09 wireless

root@OpenWrt:/etc/config# cp firewall firewall.bk
root@OpenWrt:/etc/config# cp network network.bk
root@OpenWrt:/etc/config# cp wireless wireless.bk
root@OpenWrt:/etc/config# ls -la
drwxr-xr-x 1 root root      1024 Feb 3 23:16 .
drwxr-xr-x 1 root root      1024 Feb 3 23:14 ..
-rw-r--r-- 1 root root      1029 Feb 3 23:09 dhcp
-rw-r--r-- 1 root root       221 Feb 3 23:09 dropbear
-rw-r--r-- 1 root root     4066 Feb 3 23:09 firewall
-rw-r--r-- 1 root root     4066 Feb 3 23:16 firewall.bk
-rw-r--r-- 1 root root       968 Feb 3 23:09 luci
-rw-r--r-- 1 root root      428 Feb 3 23:09 network
-rw-r--r-- 1 root root      428 Feb 3 23:16 network.bk
-rw-r--r-- 1 root root      167 Feb 3 23:09 rpcd
-rw-r--r-- 1 root root      358 Feb 3 23:09 system
-rw-r--r-- 1 root root      783 Feb 3 23:09 uhttpd
-rw-r--r-- 1 root root      365 Feb 3 23:09 wireless
-rw-r--r-- 1 root root      365 Feb 3 23:16 wireless.bk

root@OpenWrt:/etc/config#

```

Figure 13- OPENWRT/RASPAP Listing Files

```
root@OpenWrt: /etc/config
config interface 'loopback'
    option device 'lo'
    option proto 'static'
    option ipaddr '127.0.0.1'
    option netmask '255.0.0.0'

config globals 'globals'
    option ula_prefix 'fd12:31ab:7691::/48'

config device
    option name 'br-lan'
    option type 'bridge'
    list ports 'eth0'

config interface 'lan'
    option device 'br-lan'
    option proto 'static'
    option ipaddr '10.10.3.14'
    option netmask '255.255.255.0'
    option ip6assign '60'
    option force_link '1'

config interface 'wwan'
    option proto 'dhcp'
    option peerdns '0'
    option dns '1.1.1.1 8.8.8.8'

config interface 'vpnclient'
    option ifname 'tun0'
    option proto 'none'

~
~
~
~
~
~
~
- network 1/32 3%
```

Figure 14- Network Interface

```
root@OpenWrt: /etc/config
config defaults
    option syn_flood 1
    option input REJECT
    option output ACCEPT
    option forward REJECT
# Uncomment this line to disable ipv6 rules
# option disable_ipv6 1

config zone
    option name lan
    list network 'lan'
    option input ACCEPT
    option output ACCEPT
    option forward ACCEPT

config zone
    option name wan
    list network 'wan'
    list network 'wan6'
    option input ACCEPT
    option output REJECT
    option forward REJECT
    option masq 1
    option mtu_fix 1

config forwarding
    option src lan
    option dest wan

# We need to accept udp packets on port 68,
# see https://dev.openwrt.org/ticket/4108
config rule
    option name Allow-DHCP-Renew
    option src wan
    option proto udp
    option dest_port 68
    option target ACCEPT
    option family ipv4

# Allow IPv4 ping
- firewall 1/189 0%
```

Figure 15-Network Configuration

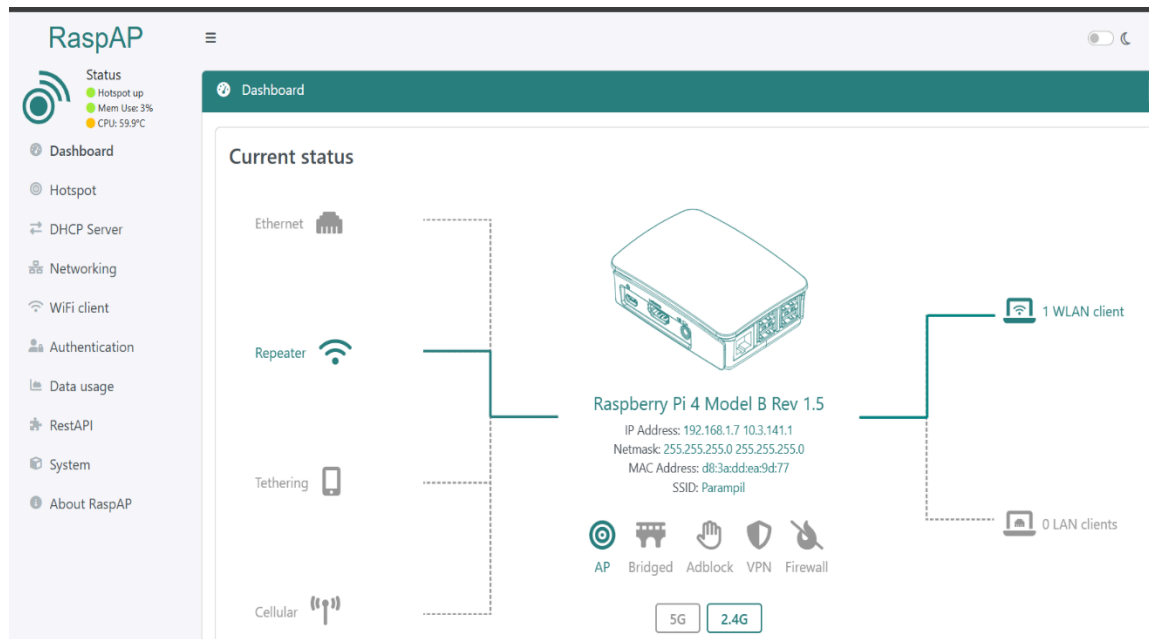


Figure 18- RaspAP Web Interface

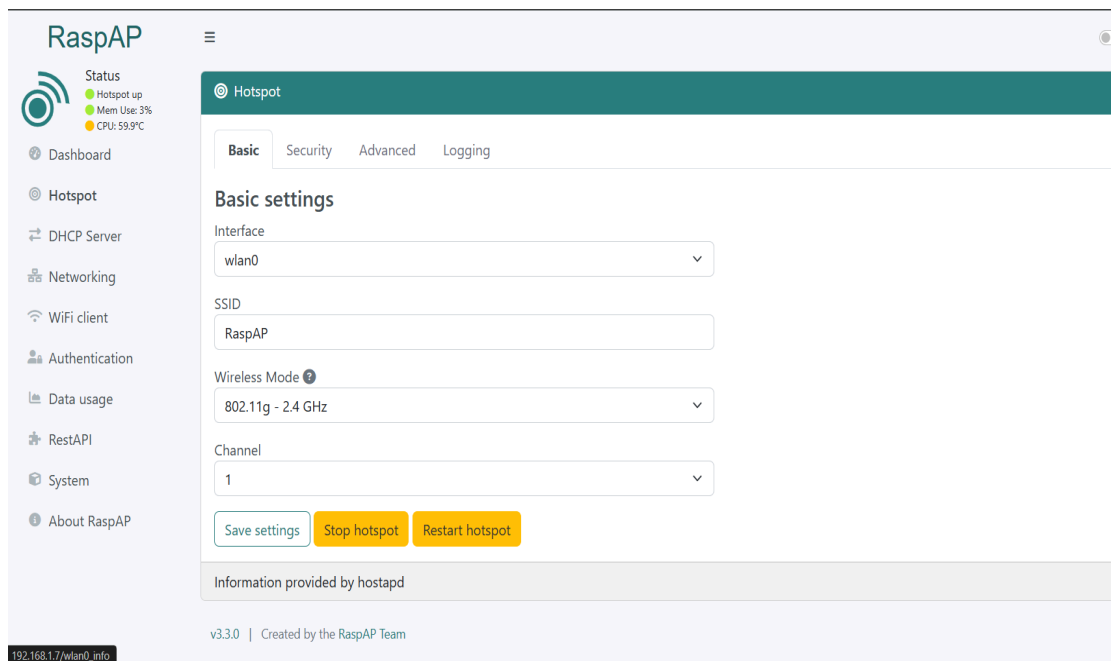


Figure 19- Hotspot Settings

RaspAP

Status

Hotspot up

Mem Use: 3%

CPU: 59.9°C

Dashboard

Hotspot

DHCP Server

Networking

WiFi client

Authentication

Data usage

RestAPI

System

About RaspAP

192.168.1.7/dhcpd_conf#server-settings

DHCP Server

Server settings

Advanced

Static Leases

Client list

Logging

DHCP server settings

Interface

wlan0

Adapter IP Address Settings

DHCP

Static IP

Enable fallback to static option

Enable this option to configure a static profile and fall back to it when DHCP lease fails.

Static IP options

IP Address

10.3.141.1

Subnet Mask

255.255.255.0

Default gateway

10.3.141.1

Figure 20- DHCP server Settings

RaspAP

Status

Hotspot up

Mem Use: 3%

CPU: 58.9°C

Dashboard

Hotspot

DHCP Server

Networking

WiFi client

Authentication

Data usage

RestAPI

System

About RaspAP

Current settings

eth0

2: eth0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc mq state DOWN group default qlen 1000
link/ether d8:3a:dd:ea:9d:76 brd ff:ff:ff:ff:ff:ff

wlan0

3: wlan0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq state UP group default qlen 1000
link/ether d8:3a:dd:ea:9d:77 brd ff:ff:ff:ff:ff:ff
inet 192.168.1.7/24 brd 192.168.1.255 scope global dynamic noprefixroute wlan0
valid_lft 86151sec preferred_lft 86151sec
inet 10.3.141.1/24 brd 10.3.141.255 scope global noprefixroute wlan0
valid_lft forever preferred_lft forever
inet6 fe80::938d:6c75:4619:ec9b/64 scope link noprefixroute
valid_lft forever preferred_lft forever

wlan1

4: wlan1: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc noqueue state DOWN group default qlen 1000
link/ether b0:19:21:b8:f3:b6 brd ff:ff:ff:ff:ff:ff

Figure 21- Ip Address

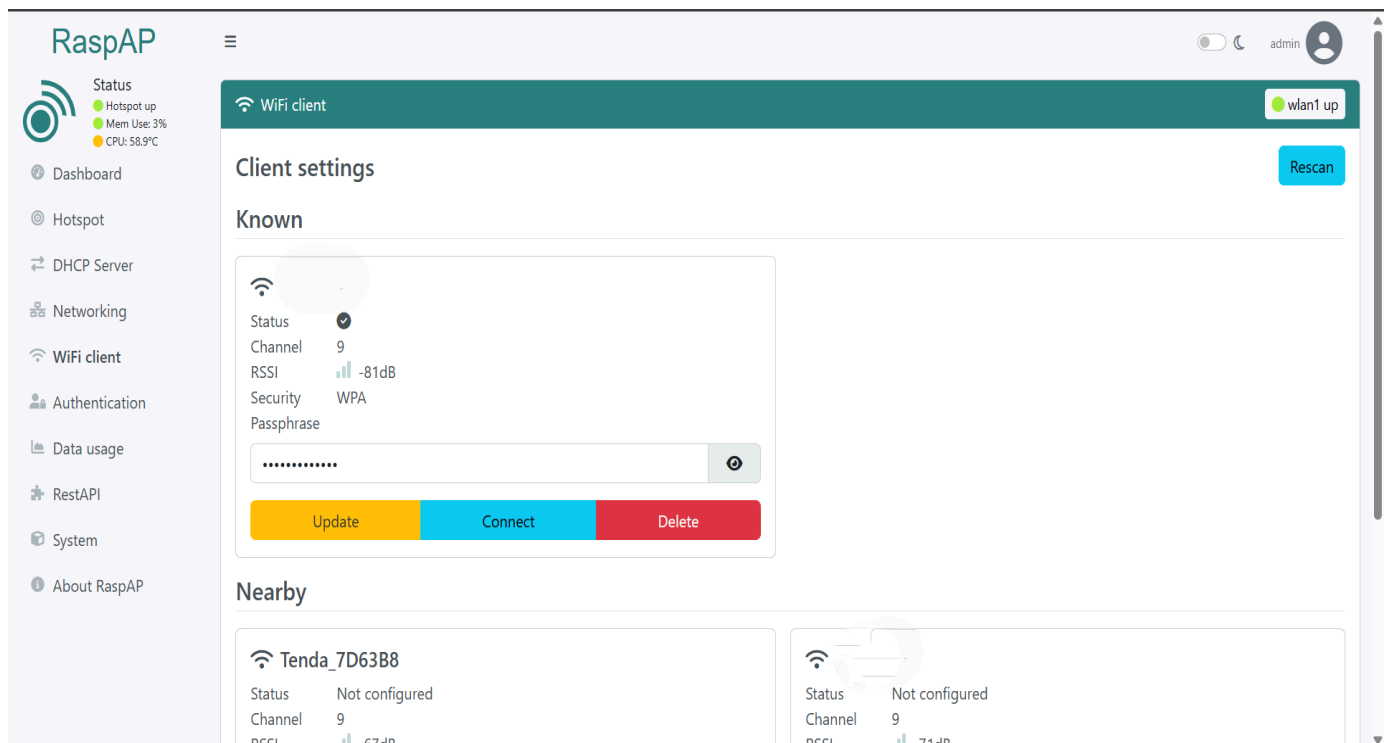


Figure 22- Wifi Client

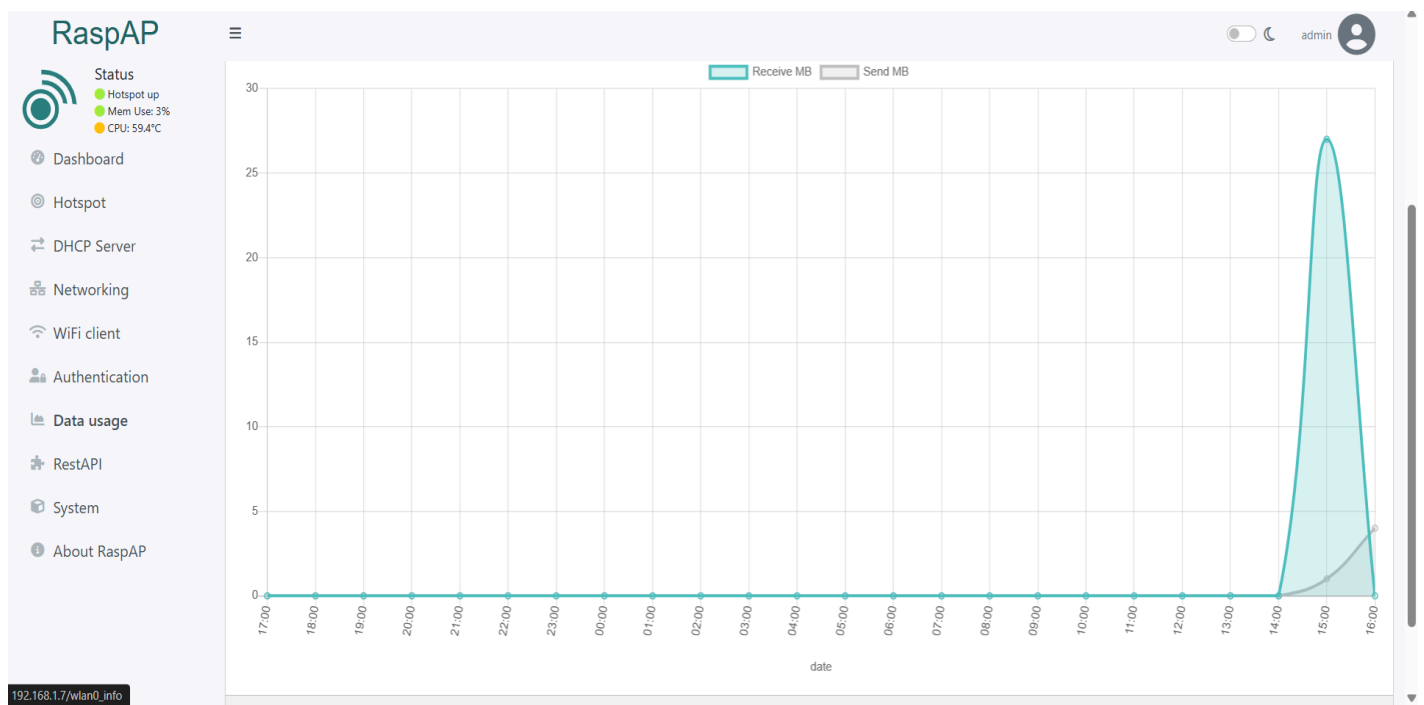


Figure 23- Data Usage

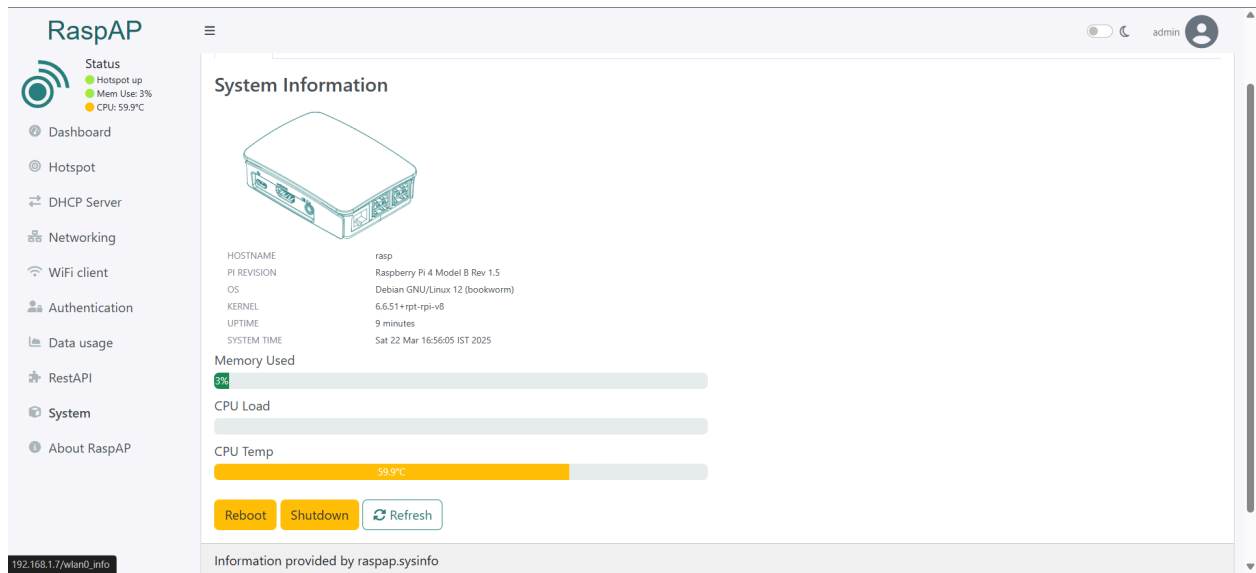


Figure 24- System Info

- **Securing public Wi-Fi:** Encrypt your connection and protect your data.
- **Creating a personal hotspot:** Share a single Ethernet or Wi-Fi connection with multiple devices.
- **Bypassing captive portals:** Some hotels or public Wi-Fi networks require you to log in on each device. A travel router allows you to log in once.
- **VPN integration:** Route all your traffic through a VPN for enhanced privacy.

2. Hardware Components:

- **Raspberry Pi:**
 - Raspberry Pi 4 Model B or Raspberry Pi Zero W/2 W (The Pi 4 offers better performance for multiple devices).
- **Power Supply:**
 - A reliable power supply suitable for your Raspberry Pi model.
 - Power bank for portable usage.
- **Storage:**
 - MicroSD card (at least 32GB recommended).
- **Networking:**
 - Ethernet cable (if using a wired connection).
 - USB Wi-Fi adapter (if the onboard Wi-Fi is insufficient or if you need to create a second access point).

- **Optional:**

- Case for the Raspberry Pi.
- Portable power bank.
- Small screen for displaying connection info.

3. Software Setup:

- **Operating System:**

- Raspberry Pi OS (formerly Raspbian) Lite is recommended for a minimal footprint.
- Download the latest Raspberry Pi OS Lite image from the official Raspberry Pi website.
- Use Raspberry Pi Imager or a similar tool to flash the image onto your microSD card.

- **Initial Configuration:**

- Insert the microSD card into your Raspberry Pi and boot it up.
- Connect to your Raspberry Pi via SSH (if you are using the Lite OS) or through a connected monitor and keyboard.
- Update the system:

Bash

```
sudo apt update
sudo apt full-upgrade -y
```

- Configure Wi-Fi (if needed):
 - Edit the `wpa_supplicant.conf` file:

Bash

```
sudo nano /etc/wpa_supplicant/wpa_supplicant.conf
```

- Add your Wi-Fi network details:

```
network={
    ssid="YOUR_WIFI_SSID"
    psk="YOUR_WIFI_PASSWORD"
}
```

- **Install Required Software:**
 - Install **hostapd** and **dnsmasq**:
Bash

```
sudo apt install hostapd dnsmasq -y
```

4. Configuration:

- **hostapd Configuration:**
 - Create the **hostapd** configuration file:
Bash

```
sudo nano /etc/hostapd/hostapd.conf
```

- Add the following configuration (adjust as needed):

```
interface=wlan0
driver=nl80211
ssid=MyTravelRouter
hw_mode=g
channel=6
wpa=2
wpa_passphrase=YourRouterPassword
wpa_key_mgmt=WPA-PSK
wpa_pairwise=TKIP
rsn_pairwise=CCMP
```

- Configure **hostapd** to use the configuration file:
Bash

```
sudo nano /etc/default/hostapd
```

- Uncomment or add the following line:

```
DAEMON_CONF="/etc/hostapd/hostapd.conf"
```

- **dnsmasq Configuration:**
 - Create the **dnsmasq** configuration file:
Bash

```
sudo nano /etc/dnsmasq.conf
```

- Add the following configuration:

```
interface=wlan0
dhcp-range=192.168.4.2,192.168.4.20,255.255.255.0,24h
address=/router.local/192.168.4.1
```

- **IP Forwarding and NAT:**

- Enable IP forwarding:
Bash

```
sudo nano /etc/sysctl.conf
```

- Uncomment or add the following line:

```
net.ipv4.ip_forward=1
```

- Apply the changes:
Bash

```
sudo sysctl -p
```

- Configure NAT (Network Address Translation) using **iptables**:
Bash

```
sudo iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE #If
using ethernet for internet.
sudo iptables -t nat -A POSTROUTING -o wlan1 -j MASQUERADE #If
using a second wifi adapter for the internet.
sudo sh -c "iptables-save > /etc/iptables.ipv4.nat"
```

- Restore **iptables** rules on boot:
 - Edit **/etc/rc.local** and add the following line before **exit 0**:
Bash

```
iptables-restore < /etc/iptables.ipv4.nat
```

- **Restart Services:**
Bash

```
sudo systemctl unmask hostapd
sudo systemctl enable hostapd
sudo systemctl start hostapd
sudo systemctl enable dnsmasq
sudo systemctl restart dnsmasq
```

5. Potential Enhancements:

- **VPN Integration:**
 - Install and configure a VPN client (e.g., OpenVPN, WireGuard) to route all traffic through a VPN.
- **Ad Blocking:**
 - Install Pi-hole to block ads and trackers on your network.
- **Captive Portal Bypass Automation:**
 - Write a script to automate the login process for captive portals.
- **Web Interface:**
 - Install a web-based management interface (e.g., Webmin) for easier configuration.
- **Traffic Shaping/QoS:**
 - Configure Quality of Service to prioritize certain traffic.
- **Dual Band wifi:**
 - Use a dual band USB wifi adapter to offer 2.4Ghz and 5Ghz networks.
- **Portable Power:**
 - Use a high capacity power bank to allow for extended use when there is not a wall outlet available.

6. Security Considerations:

- Use a strong password for your Wi-Fi network.
- Keep your Raspberry Pi OS and software updated.
- Consider using a VPN for added security.
- Change the default Raspberry Pi password.

1. Wi-Fi Access Point Creation (hostapd):

- **Purpose:** `hostapd` (Host Access Point Daemon) turns your Raspberry Pi's Wi-Fi adapter (wlan0) into a Wi-Fi access point.

- **How it Works:**

- It creates a wireless network with a specific SSID (network name) and password.
- It handles the authentication process when devices connect to this network.
- It broadcasts the Wi-Fi signal, allowing devices to discover and connect to your Raspberry Pi.
- The configuration file `/etc/hostapd/hostapd.conf` defines the network's parameters, such as the SSID, security protocol (WPA2), and password.

2. DHCP and DNS (dnsmasq):

- **Purpose:** `dnsmasq` provides DHCP (Dynamic Host Configuration Protocol) and DNS (Domain Name System) services to devices connected to your Wi-Fi network.

- **How it Works:**

- **DHCP:** When a device connects to your Raspberry Pi's Wi-Fi, `dnsmasq` assigns it an IP address from a specified range (e.g., 192.168.4.2 to 192.168.4.20). This allows devices to communicate with each other and with the Raspberry Pi. It also provides the devices with the gateway address, and the DNS server address.
- **DNS:** `dnsmasq` acts as a local DNS server, translating domain names (e.g., [google.com](https://www.google.com)) into IP addresses. This enables devices to access websites and online services.
- The configuration file `/etc/dnsmasq.conf` defines the IP address range, DNS settings, and other related parameters.

3. Network Address Translation (NAT) and IP Forwarding (iptables):

- **Purpose:** These are the core elements that allow devices connected to your Raspberry Pi's Wi-Fi to access the internet through another network connection (e.g., Ethernet or a second Wi-Fi adapter).
- **How it Works:**
 - **IP Forwarding:** This enables the Raspberry Pi to route network traffic between its different network interfaces (e.g., from wlan0 to eth0). This is controlled by the kernel parameter `net.ipv4.ip_forward`, which we enable in `/etc/sysctl.conf`.
 - **NAT:** `iptables` with NAT (Network Address Translation) is the mechanism that translates the private IP addresses of devices on your Wi-Fi network (e.g., 192.168.4.x) to the public IP address of your internet connection.
 - When a device on your Wi-Fi network sends a request to a website, `iptables` changes the source IP address of the request to the Raspberry Pi's public IP address.

- When the website sends a response, `iptables` translates the destination IP address back to the private IP address of the device that made the request.
- This allows multiple devices to share a single internet connection. The MASQUERADE command, within iptables, is what allows this dynamic changing of IP addresses.

4. Network Flow:

- A device connects to the raspberry pi's wifi network.
- The device requests an IP address from the raspberry pi.
- The raspberry pi, using dnsmasq, assigns an Ip address to the device.
- The device then requests a website.
- The raspberry pi, using dnsmasq, translates the website name into an IP address.
- The raspberry pi, using iptables, and IP forwarding, sends that request out of its internet connected port.
- The website server responds to the raspberry pi.
- The raspberry pi, using iptables, and IP forwarding, sends that response to the correct device.

CHAPTER 6

PROJECT OUTCOME AND APPLICABILITY

6.1) PROJECT OUTCOMES

This Raspberry Pi x OPENWRT/RASPAP Travel Router project brings about a new small portable router improved in networking activities. The router is entirely in a bespoke, 3D printed case. It comes with functionalities like network-wide ad-blocking, VPN routing, and physical portability. Some of the most important results include:

Portable Secure Networking: A travel router built on a Raspberry Pi that employs a secure internet VPN (WireGuard/OpenVPN) for traffic.

Ad-Blocking Capability: The use of AdGuard Home across the network will block ads to greatly enrich the browsing experience.

Better Connectivity: A very versatile access point using a USB Wi-Fi adaptor that connects a variety of devices.

Simplicity: The intuitive tactile shutdown button provides safe power management.

Durability and Custom Aesthetics: It is stored in a specially made 3D-printed casing for increased durability and portability.

Optimization in Performance: Increased throughput owing to proper selection of USB Wi-Fi.

6.2) ADAPTABILITY OF THE PROJECT

The Raspberry Pi x OPENWRT/RASPAP Travel Router project has resulted in a new compact portable router with greater networking capabilities. The full-fledged router is housed in a custom 3D printed enclosure. It has features like physical mobility, VPN routing, and network-wide ad-blocking. Among the most important results are:

Portable secure networking, which is a travel router based on raspberry pi with a secure internet VPN (WireGuard/OpenVPN) traffic.

Block ad: AdGuard Home's will implement this for the entire network and hence block ads, thus improving a lot of aspects of browsing.

Improved Connectivity: A highly flexible access point that supports a multitude of devices through a USB Wi-Fi adapter.

Easy: Users have a comfortable tactile shutdown button for safe energy management.

Sturdiness & Custom Style: It is placed within a specifically designed 3D-printed casing for greater strength and ease in portability.

Performance Optimization: High throughput due to selective use of USB Wi-Fi.

6.3) PROJECT APPLICABILITY IN REAL WORLD APPLICATIONS

The integration of OPENWRT/RASPAP with Raspberry Pi to create a customizable router has numerous real-world applications. It provides a cost-effective solution for home networks and small businesses, offering features such as advanced firewall protection, VPN support for secure remote access, and network monitoring to ensure optimal performance. This setup can also be tailored for specialized use cases, such as IoT device management, ad-blocking, or parental controls. Additionally, the portable and scalable nature of this solution makes it ideal for temporary network setups at events, remote work environments, or educational purposes. Its flexibility and security make it a practical choice in today's increasingly connected world.

6.4) INFERENCE

The integration of Open WRT with Raspberry Pi demonstrates a highly effective and practical approach to building customizable, secure, and cost-efficient networking solutions. This project has shown that with minimal investment, it is possible to achieve advanced features such as VPN support, firewall management, and network monitoring, typically found in high-end commercial routers.

The flexibility and scalability of this solution make it suitable for a wide range of real-world applications, including home networks, small offices, and specialized use cases like IoT management and ad-blocking. Furthermore, the project highlights the growing importance of open-source tools and affordable hardware in providing accessible and robust networking solutions, addressing the evolving demands of privacy, security, and performance in today's connected world.

CHAPTER 7

CONCLUSION AND RECOMMENDATIONS

7.1) CONCLUSION OF THE PROJECT

The Raspberry Pi x OPENWRT/RASPAP Travel Router project demonstrates that open-source technology and cost-effective hardware can be harnessed to deliver an effective networking solution that's safe, portable, and feature-rich. Network-wide ad-blocking, VPN routing, and efficient hardware utilization satisfy both the needs of tech savvy and privacy-conscious travelers alike.

The addition of a physical shutdown button, a bespoke 3D-printed enclosure, and enhanced performance through thoughtful hardware and software selections demonstrate innovation and usefulness. This project serves as a useful learning tool for networking, Linux systems, and hardware design in addition to demonstrating the versatility of the Raspberry Pi and OPENWRT/RASPAP.

As mobility, security, and privacy keep gaining importance in our life, the travel router is a handy device for everyone and professional use, making it an excellent base for further enhancements while keeping it relevant in an evolutionary landscape.

REFERENCES

- [1] Batubara, S., Wahyuni, S., Hariyanto, E., & Lubis, A. (2021). Webinar Menangkal Cyberporn Pada Internet dan Android Memanfaatkan Add Ons dan Aplikasi Antipornografi Parental Control Di SMA Panca Budi. In Jurnal Pengabdian Kepada Masyarakat (Vol. 4, Issue 1). <http://ejournal.bsi.ac.id/ejurnal/index.php/abdimas>
- [2] Dutkowska-Zuk, A., Hounsel, A., Morrill, A., Xiong, A., Chetty, M., & Feamster, N. (2022). How and why people use virtual private networks. 31st USENIX Security Symposium (USENIX Security 22), 3451â€“3465.
- [3] Gentile, A. F., Macr`i, D., De Rango, F., Tropea, M., & Greco, E. (2022). A VPN performance analysis of constrained hardware open source infrastructure deploy in IoT environment. Future Internet, 14(9), 264.
- [4] Jadhav, A. P., & Malode, V. B. (2019). Raspberry Pi based offline media server. Proceedings of the 3rd International Conference on Computing Methodologies and Communication, ICCMC 2019, Iccmc, 531â€“533. <https://doi.org/10.1109/ICCMC.2019.8819718>
- [5] Lubis, A., Hariyanto, E., Harahap, M. I., Pembangunan, U., & Budi, P. (2022). Wireless controller menggunakan capsman di jaringan laboratorium komputer perguruan panca budi medan wireless controller using capsman in computer laboratory network of panca budi education medan. Journal of Information Technology and Computer Science (INTECOMS), 5, 97â€“103.
- [6] Lubis, A., Septian, R., Sain, F., Pembangunan, U., & Budi, P. (2022). Pengembangan Aplikasi Troubleshooting Jaringan Melalui Sistem Notifikasi dengan Integrasi Cacti dan Telegram. 4(1), 104â€“109.
- [7] Lubis, A., & Tarigan, A. (2017). Security Assessment of Web Application Through Penetration System Techniques. International Journal of Recent Trends in Engineering & Research (IJRTER, 03(01).
- [8] Milsa Pratama, R., Wahyuni, S., & Lubis, A. (2023). RANCANG BANGUN KEAMANAN KONEKSI PRIBADI MELALUI OPEN VPN BERBASIS CLOUD. Journal of Information Technology and Computer Science (INTECOMS), 6(1).
- [9] Osswald, L., Haeberle, M., & Menth, M. (n.d.). Performance Comparison of VPN Solutions. <https://www.wireguard.com/>.
- [10] raspberrypi.org. (2021). What is a Raspberry Pi? Introduction to the Raspberry Pi. <https://www.raspberrypi.org/help/what-is-a-raspberry-pi/>
- [11] Rivera-Dourado, M., Gestal, M., Pazos, A., & VÃ¡zquez-Naya, J. (2023). Adapting a Captive Portal for Phishing-Resistant Network Authentication Using Security Keys. 2023 JNIC Cybersecurity Conference (JNIC), 1â€“8.

- [12] Skendzic, A., & Kovacic, B. (2017). Open source system OpenVPN in a function of Virtual Private Network. IOP Conference Series: Materials Science and Engineering, 200(1). <https://doi.org/10.1088/1757-899X/200/1/012065>
- [13] Wahyuni, S., & Betty Yel, M. (2022). Aplikasi Bank Sampah Berbasis Website Dalam Mewujudkan Desa Bebas Sampah. Prosiding Seminar Nasional Riset Dan Information Science (SENARIS), 4, 242â€“250.
- [14] Zakaria, M. I., Norizan, M. N., Isa, M. M., Jamlos, M. F., & Mustapa, M. (2022). Comparative analysis on virtual private network in the internet of things gateways. Indones. J. Electr. Eng. Comput. Sci, 28(1), 488â€“497.
- [15] M. Ariman, G. Seçinti, M. Erel and B. Canberk, "Software defined wireless network testbed using Raspberry Pi of switches with routing add-on," 2015 IEEE Conference on Network Function Virtualization and Software Defined Network (NFV-SDN), San Francisco, CA, USA, 2015, pp. 20-21, doi: 10.1109/NFV-SDN.2015.7387397. keywords: {Routing;Hardware;Software;Control systems;Wireless communication;Network topology;Ports (Computers);SDWN;Raspberry Pi;OpenFlow Switches;Routing},
- [16] "Developer's guide - basic approach to OPENWRT/RASPAP/RASPAP", *OPENWRT/RASPAP/RASPAP - Wireless Freedom Tech. Rep.*, 2012.
- [17] Josef Hammer, Dragi Kimovski, Narges Mehran, Radu Prodan, Hermann Hellwagner, "C3-Edge – An Automated Mininet-Compatible SDN Testbed on Raspberry Pis and Nvidia Jetsons", *NOMS 2023-2023 IEEE/IFIP Network Operations and Management Symposium*, pp.1-5, 2023.
- [18] https://OPENWRT/RASPAP/RASPAP.org/toh/raspberry_pi_foundation/raspberry_pi
- [19] <https://tristam.ie/2023/582/>
- [20] <https://www.raspberrypi.com/products/raspberry-pi-4-model-b/specifications/>
- [21] S. Samonas and D. Coss, "The CIA strikes back: Redefining confidentiality integrity and availability in security", *Journal of Information System Security*, vol. 10, no. 3, 2014.
- [22] Y. Li and Q. Liu, "A comprehensive review study of cyber-attacks and cyber security; Emerging trends and recent developments", *Energy Reports*, vol. 7, pp. 8176-8186, 2021.
- [23] B. Cashell, W.D. Jackson, M. Jickling and B. Webel, "The economic impact of cyber-attacks", *Congressional research service documents CRS RL32331 (Washington DC)*, pp. 2, 2004.
- [24] M. Liyanage, P. Kumar, M. Ylianttila and A. Gurtov, "Novel secure VPN architectures for LTE backhaul networks", *Security and Communication Networks*, vol. 9, no. 10, pp. 1198-1215, 2016.
- [25] A. Karaymeh, M. Ababneh, M. Qasaimeh and M. Al-Fayoumi, "Enhancing data protection provided by VPN connections over open WiFi networks", *2019 2nd International Conference on new Trends in Computing Sciences (ICTCS)*, pp. 1-6, October 2019.

- [26] S.K.H. Dziaudin and H.M. Salleh, "Alternative Vpn Solution Using Raspberry Pi As Router", *Journal of Computing Technologies and Creative Content (JTec)*, vol. 5, no. 2, pp. 18-21, 2020.
- [27] S.I.N.M. Yusoff and S.A. Baharudin, "Virtual Private Network Server and Adblock Server using Raspberry Pi with Parental Control", *Journal of Computing Technologies and Creative Content (JTec)*, vol. 5, no. 2, pp. 88-92, 2020.
- [28] J.R. Raj and S. Srinivasulu, "Design of IoT based VPN gateway for home network", *2022 International Conference on Electronics and Renewable Systems (ICEARS)*, pp. 561-564, March 2022.
- [29] A. Karaymeh, M. Ababneh, M. Qasaimeh and M. Al-Fayoumi, "Enhancing data protection provided by VPN connections over open WiFi networks", *2019 2nd International Conference on new Trends in Computing Sciences (ICTCS)*, pp. 1-6, October 2019.
- [30] B. Yang and T. Gao, "Building a secure and reliable network via multi-homed vpn", *Session IT*, pp. 303-088, 2006.