MY Smart Bridge Router: Secure and Adaptive Connectivity Anywhere

Senior Project

by

Moayad K. Salloum – 22230296,

Yousef R. Younis - 22230294

Submitted to the School of Engineering of the

Lebanese International University

Bekaa, Lebanon

in partial fulfillment of the requirements for the degree of

BACHELOR OF SCIENCE IN COMPUTER ENGINEERING

Spring 2024-2025

Approved by:

Supervisor

Dr. Abdel-Mehsen Ahmad

Committee Member

Dr. Ibrahim El Bitar

DEDICATION

We dedicate this project to every passionate and hard-working engineer who chooses to keep going when things get frustrating, who enjoys untangling messy code, and who finds beauty in solving problems one bug at a time.

To the ones who stay up late configuring routers, testing DNS rules, fixing broken UI layouts, and rewriting scripts until they finally work—you inspire us. Whether your passion lies in networks, backend logic, user interfaces, or systems design, this is for you. Your determination and love for the craft are what push innovation forward.

We also dedicate this work to each other. Throughout the challenges and breakthroughs, we stood side by side—sharing ideas, picking up the slack when needed, and learning from one another. This project is a result of our shared vision, effort, and commitment to seeing it through to the end.

And finally, we dedicate this project to every student who's ever wondered if they were capable of building something impactful. The answer is: yes, you are.

Moayad Salloum and Yousef Younis

ACKNOWLEDGMENT

We would like to sincerely thank our project supervisor, Dr. Abdel Mehsen Ahmad, whose thoughtful guidance, honest feedback, and deep technical insight played a central role in shaping this project from start to finish. His support helped us maintain both technical clarity and academic integrity throughout the process.

We also thank the Department of Computer and Communication Engineering and all of its instructors for the knowledge, encouragement, and principles they instilled in us throughout our undergraduate years. The foundation we built in those classrooms prepared us to take on this project with confidence.

To our families and friends, thank you for your support: emotional, mental, and logistical. From testing features and reviewing ideas, to simply being there during late-night stress sessions, your presence made this journey easier and more meaningful.

And of course, we thank each other. This project wouldn’t have been the same without mutual trust, balanced effort, and shared ambition. We challenged each other, respected each other’s strengths, and worked as a team in the truest sense.

This work represents not only our academic experience, but also the relationships, resilience, and growth that came with it.

ABSTRACT

With the rapid expansion of internet usage in both personal and professional settings, users increasingly demand greater control, security, and flexibility in managing their networks. Traditional routers often fall short, offering limited customization, weak security features, and inadequate real-time monitoring, leaving users vulnerable to cyber threats, inefficient bandwidth management, and privacy breaches. Additionally, the reliance on public networks, such as those in hotels, cafes, and airports, exposes individuals to significant security risks, including data interception and tracking. The need for a portable, secure, and customizable networking solution has never been more critical. This project presents " MY Smart Travel Router: Secure and Adaptive Connectivity Anywhere", an advanced Raspberry Pi-based smart router that leverages OpenWrt firmware and a web-based management interface. The system integrates essential features such as VPN support, content filtering via AdGuard, smart queue management, device monitoring, firewall controls, and voice command capabilities. By combining open-source technologies with intelligent networking solutions, this project offers users a cost-effective, flexible, and secure alternative to commercial routers, enhancing network control, privacy, and ease of use.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS

ADGUARD: AdGuard (Name of an ad blocking service and DNS resolver)

AI: Artificial Intelligence

API: Application Programming Interface

BASH: Bourne Again SHell

BSD: Berkeley Software Distribution (Often refers to the BSD License)

BW: Bandwidth

CGI: Common Gateway Interface

CGI-BIN: Common Gateway Interface - Binary (Common directory name for CGI scripts)

CHMOD: Change Mode (Unix/Linux command)

CLI: Command Line Interface

CPU: Central Processing Unit

CSS: Cascading Style Sheets

D3.JS: Data-Driven Documents (JavaScript library)

DDR1/DDR2: Double Data Rate (Generation 1/2 Synchronous Dynamic Random-Access Memory)

DHCP: Dynamic Host Configuration Protocol

DIY: Do-It-Yourself

DNS: Domain Name System

E-WASTE: Electronic Waste

GB: Gigabyte

GHz: Gigahertz

GL.INET: GL.iNet (Company Name)

GNU: GNU's Not Unix (Recursive acronym)

GPL: General Public License (e.g., GNU GPL)

HTML: HyperText Markup Language

HTTP: HyperText Transfer Protocol

IDE: Integrated Development Environment

IDS: Intrusion Detection System

IEC:1 International Electrotechnical Commission

IEEE: Institute of Electrical and Electronics Engineers

IETF: Internet Engineering Task Force2

IOT: Internet of Things

IP: Internet Protocol

IPSEC:3 Internet Protocol Security

ISMS: Information Security Management System

ISO: International Organization for Standardization4

ISP: Internet Service Provider5

JS: JavaScript

LAN: Local Area Network

LUCI: Lua Configuration Interface (OpenWrt)

MAC: Media Access Control

MB: Megabyte

MBPS: Megabits per second

MHz: Megahertz

MICROSD: Micro Secure Digital (card)

MIT: Massachusetts Institute of Technology (Often refers to the MIT License)

MUMIMO: Multi-User - Multi-Input, Multi-Output

NFC: Non-Functional Test Case

NIC: Network Interface Card

NIST: National Institute of Standards and Technology

OFDMA: Orthogonal Frequency-Division Multiple Access

OPNSENSE: OPNsense (Firewall software name, derived from "Open source makes sense")

OPENWRT: Open Wireless Router

OS: Operating System

OSI: Open Systems Interconnection (model)

PC: Personal Computer

PCI: Payment Card Industry (often PCI DSS - Data Security Standard) or Peripheral Component Interconnect

PFSENSE: pfSense (Firewall software name, derived from Packet Filter - pf)

PUTTY: PuTTY (SSH client software name, no official meaning)

QOS: Quality of Service

RAM: Random Access Memory

RESTFUL: Representational State Transfer

RFC: Request for Comments (IETF standards documents)

ROI: Return on Investment

SBC: Single-Board Computer

SP: Special Publication (Often refers to NIST SP documents)

SQM: Smart Queue Management

SSH: Secure Shell

SSID: Service Set Identifier

STA: Station (Wi-Fi client mode)

TC: Test Case

TCP: Transmission Control Protocol

TLS: Transport Layer Security

TP LINK: TP-Link (Company name, derived from Twisted-Pair Link)

UCI: Unified Configuration Interface (OpenWrt)

UDP: User Datagram Protocol

UHTTPD: micro-HTTP Daemon (Web server)

UI: User Interface

UML: Unified Modeling Language

USA: United States of America

USB: Universal Serial Bus

VLAN: Virtual Local Area Network

VOIP: Voice over Internet Protocol

VPN: Virtual Private Network

VS CODE: Visual Studio Code (Code editor software name)

W3C: World Wide Web Consortium

WAI-ARIA: Web Accessibility Initiative – Accessible Rich Internet Applications

WAN: Wide Area Network

WIFI: Wireless Fidelity

WTFAST: WTFast (Gaming VPN service name)

WWW: World Wide Web

CHAPTER 1
INTRODUCTION

1.1 Background

The modern world relies heavily on networking infrastructure for communication, data exchange, and internet access. Routers play a crucial role in directing data traffic between networks, ensuring seamless connectivity. Traditional consumer routers prioritize ease of use over advanced networking features, making them unsuitable for users who require deeper customization, enhanced security, and improved performance.

One of the key limitations of commercial routers is their restricted configurability. Most models provide basic firewall protection, limited VPN support, and minimal traffic management capabilities. Moreover, they are often subject to manufacturer-imposed constraints, preventing users from optimizing their network settings to meet specific needs. These shortcomings have driven the rise of open-source firmware solutions such as OpenWrt, which allow users to implement advanced features like enhanced firewall configurations, bandwidth prioritization, VPN integration, and ad-blocking.

1.1.1 Networking Components and Security

1. Routers and Their Role

Routers are networking devices responsible for forwarding data packets between computer networks. They function as the backbone of internet communication, ensuring that data reaches its intended destination (See Figure 1-1). A router assigns IP (Internet Protocol) addresses to connected devices and manages network traffic through a process called routing. [1]

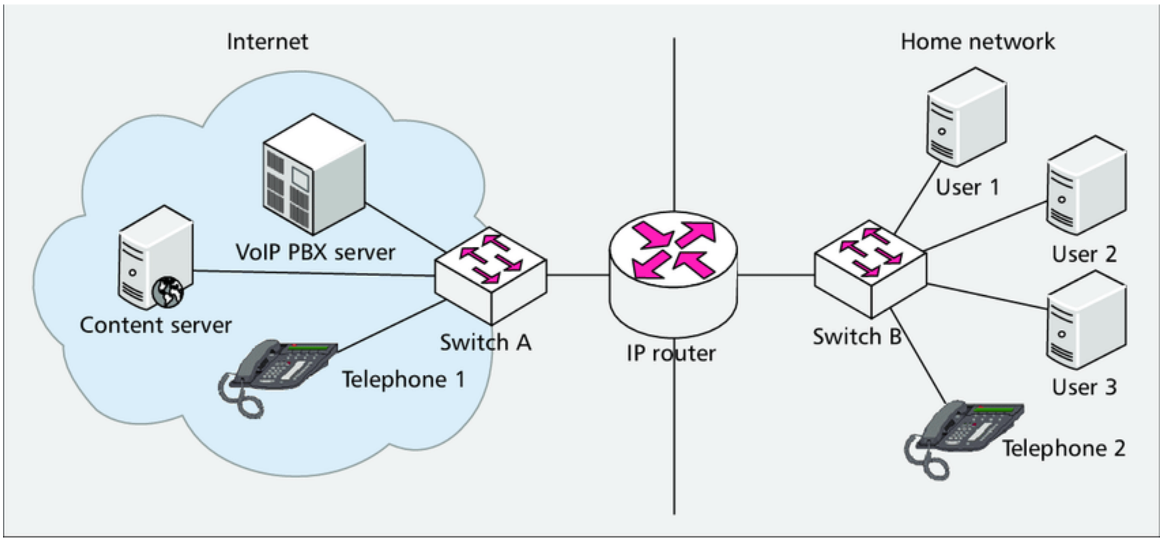


Figure 1-1: Network Topology Diagram

2. IP Addresses

IP addresses are unique identifiers assigned to devices on a network. They come in two versions: IPv4 and IPv6. IPv4 uses a 32-bit address scheme, while IPv6 employs a 128-bit system to accommodate the growing number of internet-connected devices.

3. MAC Addresses

MAC (Media Access Control) addresses are hardware-based identifiers assigned to network interfaces. Unlike IP addresses, which can change based on network configurations, MAC addresses are permanent and are used for device identification within local networks.

4. Firewalls

A firewall is a network security system that monitors and controls incoming and outgoing traffic based on predetermined security rules. Firewalls can be software- or hardware-based and serve as the first line of defense against unauthorized access and cyber threats. Open-source firmware solutions like OpenWrt enable users to configure advanced firewall rules, enhancing network security.

5. Virtual Private Networks (VPNs)

VPNs create secure, encrypted connections between a user's device and a remote network (See Figure 1-2). This technology is essential for maintaining privacy, bypassing geographic restrictions, and securing data transmissions over public networks. Many commercial routers provide limited VPN support, whereas OpenWrt allows seamless integration of VPN services such as WireGuard and OpenVPN for enhanced security and performance. [2]

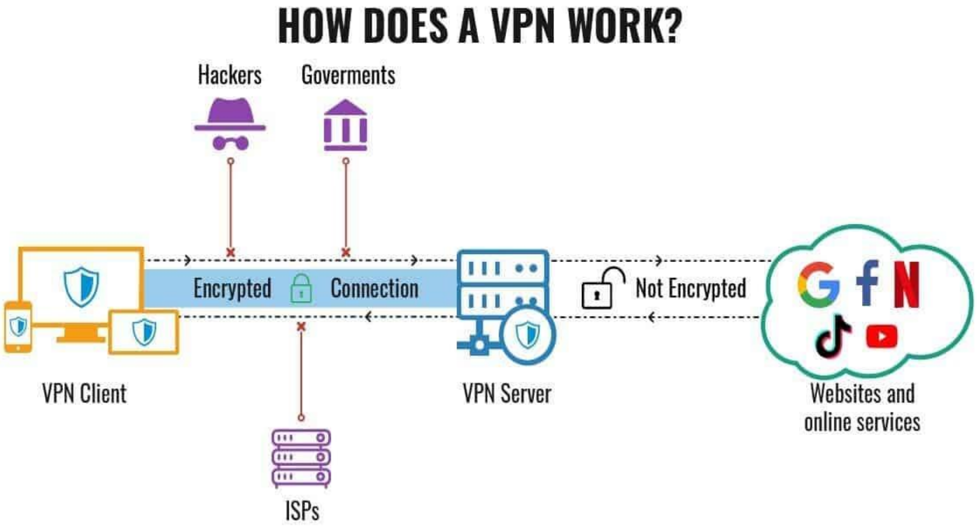


Figure 1-2: VPN Tunneling Diagram

6. Network Segmentation

Network segmentation is the practice of dividing a network into separate sections, or segments, to improve security, performance, and management. Instead of having all devices connected to a single, shared network, segmentation creates isolated networks that limit unnecessary communication between them. This approach helps contain cyber threats, reduces congestion, and improves overall network efficiency.

For example, a network can be split into different segments such as a Guest WiFi, a General Network, a Payment Card Industry (PCI) Network, and an IoT Network, as shown in Figure 1-3. Each segment has its own security policies, ensuring that sensitive systems are not exposed to less secure devices. Open-source firmware like OpenWrt allows users to configure advanced segmentation rules, enhancing network control and protection.

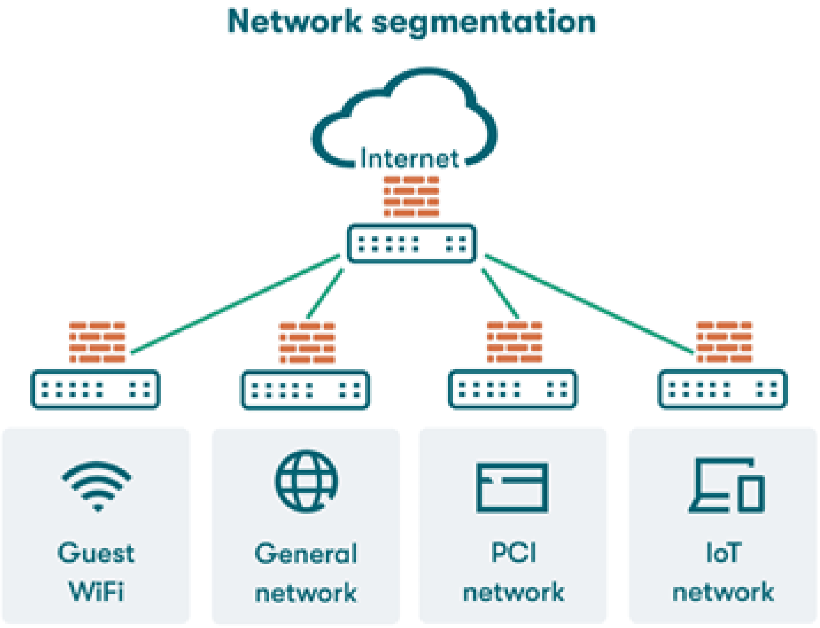


Figure 1-3: Network Segmentation

7. <u>AdGuard Home DNS</u>

AdGuard Home DNS is a powerful network-wide ad blocker and security tool that filters out unwanted content, including ads, trackers, and malicious websites. It works at the DNS level, meaning it blocks harmful domains before they even load on any device connected to the network.

When a browser or app requests access to a domain, AdGuard DNS checks whether the domain is safe. As shown in Figure 1-4, if the request is to a known harmful site, such as an ad network, a tracking service, or a malicious page, the DNS server redirects it to a "sinkhole" instead of the actual site. This prevents the browser from loading the unwanted content, enhancing privacy and security across all devices on the network. [3]

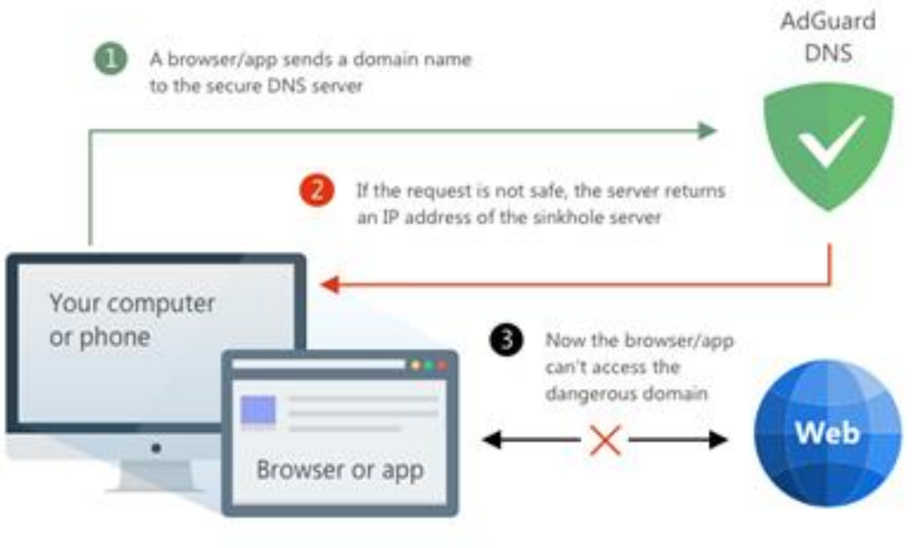

AdGuard DNS
1 A browser/app sends a domain name to the secure DNS server
2 If the request is not safe, the server returns an IP address of the sinkhole server
3 Now the browser/app can't access the dangerous domain
Your computer or phone
Browser or app
Web

Figure 1-4: AdGuard DNS workflow diagran

8. <u>Traffic Management and Bandwidth Prioritization</u>

Network traffic management involves optimizing data flow to ensure efficient utilization of bandwidth. Bandwidth prioritization enables users to allocate resources based on activity type, ensuring that high-priority tasks such as video calls and online gaming receive more bandwidth than background downloads, as represented in Figure 1-5.

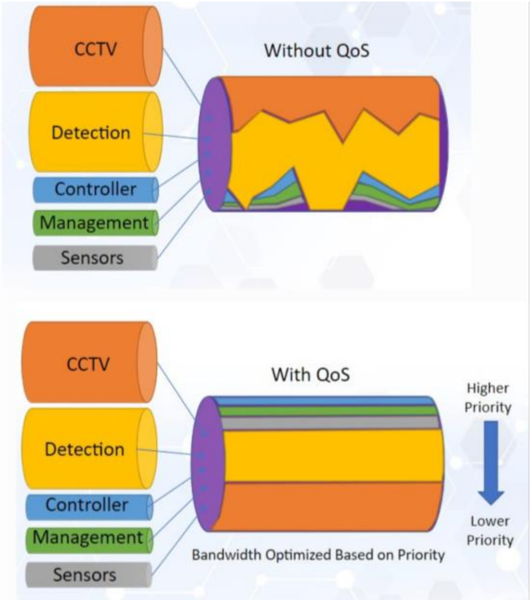


Figure 1-5: QoS (Traffic Prioritization) Graph

1.1.2 Networking Challenges Worldwide

The increasing reliance on digital connectivity has highlighted significant challenges in networking infrastructure worldwide. Many regions face issues related to network security, limited customizability, and user accessibility. Consumer-grade routers often prioritize ease of use over advanced features, leaving users with minimal control over security configurations, traffic management, and performance optimization. Additionally, the rise in cyber threats has made it crucial for users to implement stronger security measures, yet many commercial routers offer only basic firewall protection and limited VPN support.

To address these challenges, open-source firmware solutions provide greater flexibility, allowing users to enhance network security, customize firewall rules, and prioritize bandwidth efficiently. A smart, user-configurable router equipped with advanced security features and simplified management can empower individuals and businesses to take control of their networks, ensuring both protection and performance in an increasingly connected world.

1.2 Problem Statement

In today’s digital world, internet connectivity is a fundamental necessity for individuals, businesses, and travelers. However, traditional consumer routers fail to provide the necessary flexibility, security, and control required by advanced users. These routers come with limited features, making it difficult for users to customize their network settings, implement enhanced security measures, or optimize their network performance.

One major concern is the vulnerability of public and shared networks, such as those found in hotels, airports, and coffee shops. Users are frequently exposed to cyber risks like data interception, tracking, and unauthorized access. Many standard routers lack essential security features such as built-in VPN support, advanced firewall configurations, and real-time traffic monitoring, leaving users at risk of cyber threats.

Furthermore, conventional networking solutions often lack portability and adaptability. While mobile hotspots and portable routers exist, they usually do not offer comprehensive customization, network prioritization, or security enhancements. Users requiring network optimization tools, device monitoring, and bandwidth management are forced to rely on costly enterprise-grade solutions that are not always accessible or affordable.

This project addresses these challenges by developing an advanced, Raspberry Pi-based smart router with OpenWrt firmware and a user-friendly web interface. Key features include VPN support, AdGuard filtering, smart queue management (SQM), firewall controls, device monitoring, and real-time analytics. The system bridges the gap between consumer and enterprise-grade networking solutions, providing an affordable, customizable, and secure alternative for individuals and businesses seeking enhanced control over their networks.

1.3 General overview of the project

This project creates a versatile travel router using the Raspberry Pi 4 Model B with OpenWrt operating system, which is displayed in Figure 1-6. The device functions as both a travel and home router, making it perfect for users who need reliable networking in multiple environments.

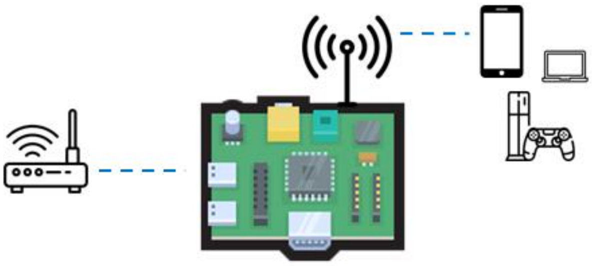


Figure 1-6: Diagram of Raspberry Pi 4 Model B setup as a router

At its core, this router prioritizes user-friendliness without sacrificing advanced features. The intuitive web interface makes complex networking operations accessible to non-technical users. This eliminates the need to call in experts for common network adjustments.

Key security features include:

- AdGuard implementation for network-wide ad-blocking and tracker prevention
- OpenVPN integration for secure connections on public networks
- Multiple network segmentation (main, guest, IoT) to isolate potentially vulnerable devices (example presented in Figure 1-3)
- Comprehensive parental controls to restrict access to inappropriate content

Performance monitoring and optimization is handled through:

- Smart Queue Management (SQM) to reduce buffer bloat and optimize bandwidth
- Real-time monitoring of CPU temperature and system load

- Network performance metrics tracking
- Ping monitoring and periodic reboot capabilities for enhanced stability

Unlike commercial routers that often limit customization or require technical expertise, this solution strikes a balance between accessibility and advanced functionality. The project addresses a clear gap in the market by combining the portability of travel routers, the robust features of high-end home routers, and the flexibility of DIY solutions.

The end result is a powerful networking tool that empowers users to take control of their internet experience without needing specialized knowledge in networking technologies.

1.4 Thesis Outline

Chapter 1: Introduces the project's motivation, background, and problem statement. It provides a general overview of the smart travel router and outlines the thesis structure.

Chapter 2: Reviews existing networking solutions, comparing commercial routers, VPN services, and network monitoring tools with the proposed system. The advantages of OpenWrt customization and enhanced security features are highlighted.

Chapter 3: Discusses the system's design, including hardware selection, software architecture, and networking configurations, ensuring a clear understanding of its functionality and security mechanisms.

Chapter 4: Details the implementation and testing phases, covering the setup of the Raspberry Pi as a router, the integration of OpenWrt and networking tools, and system performance evaluations.

Chapter 5: Concludes the study, summarizing key contributions and challenges while suggesting future improvements such as expanded protocol compatibility and AI-driven automation enhancements.

CHAPTER 2
SURVEY OF EXISTING METHODS AND SIMILAR SYSTEMS

2.1 Introduction

The networking landscape has evolved significantly over the past decade, with routers transitioning from simple internet gateways to sophisticated devices that provide security, content filtering, quality of service management, and various other advanced features. This chapter examines existing router solutions in three major categories: commercial travel routers, standard advanced home routers, and DIY/open-source router projects. Each category offers different approaches to solving common networking challenges related to portability, security, content filtering, and user experience. Understanding the capabilities and limitations of existing solutions provides critical context for the development of a Raspberry Pi 4 based travel router with enhanced features and improved usability.

2.2 System 1: Commercial Travel Routers

Commercial travel routers are compact, portable networking devices designed for users on the move. These devices typically prioritize small form factors, battery operation, and simplified setup procedures over advanced features.

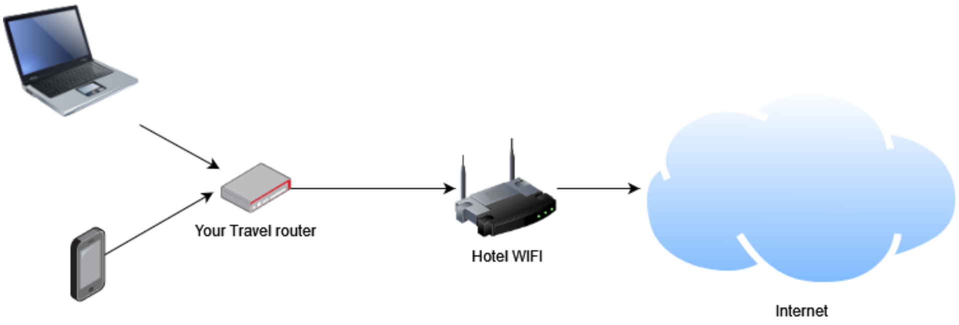


Figure 2-1: Travel Router Diagram

Figure 2-1 shows a diagram that represents how travel routers work by acting as a midway access point that connects to the main hotel Wi-Fi, which then accesses the internet.

2.2.1 GL.iNet Devices

GL.iNet is a prominent manufacturer in this space, offering models such as the GL-MT300N-V2 (Mango) and GL-AR750S (Slate). The GL-MT300N-V2 features a 580MHz MTK 7620NN CPU, 64MB DDR1 RAM, and 16MB Nor Flash storage. It supports 2.4GHz Wi-Fi with transmission rates up to 300Mbps. The device includes one WAN and one LAN port, a USB 2.0 port, and is powered via micro USB. [1]

The GL-AR750S (Slate) is equipped with a QCA9563 SoC @775MHz, 128MB DDR2 RAM, and dual flash storage (16MB Nor + 128MB Nand). It offers dual-band Wi-Fi (2.4GHz and 5GHz) and includes three WAN/LAN ports, a USB 2.0 port, and microSD support. [2]

Figure 2-2: GLi.Net travel router

2.2.2 TP-Link Devices

TP-Link's TL-WR902AC is a compact travel router offering dual-band connectivity with a combined wireless data transfer rate of up to 733Mbps, making it suitable for various applications simultaneously, with a simple web interface (Figure 2-3). [3]

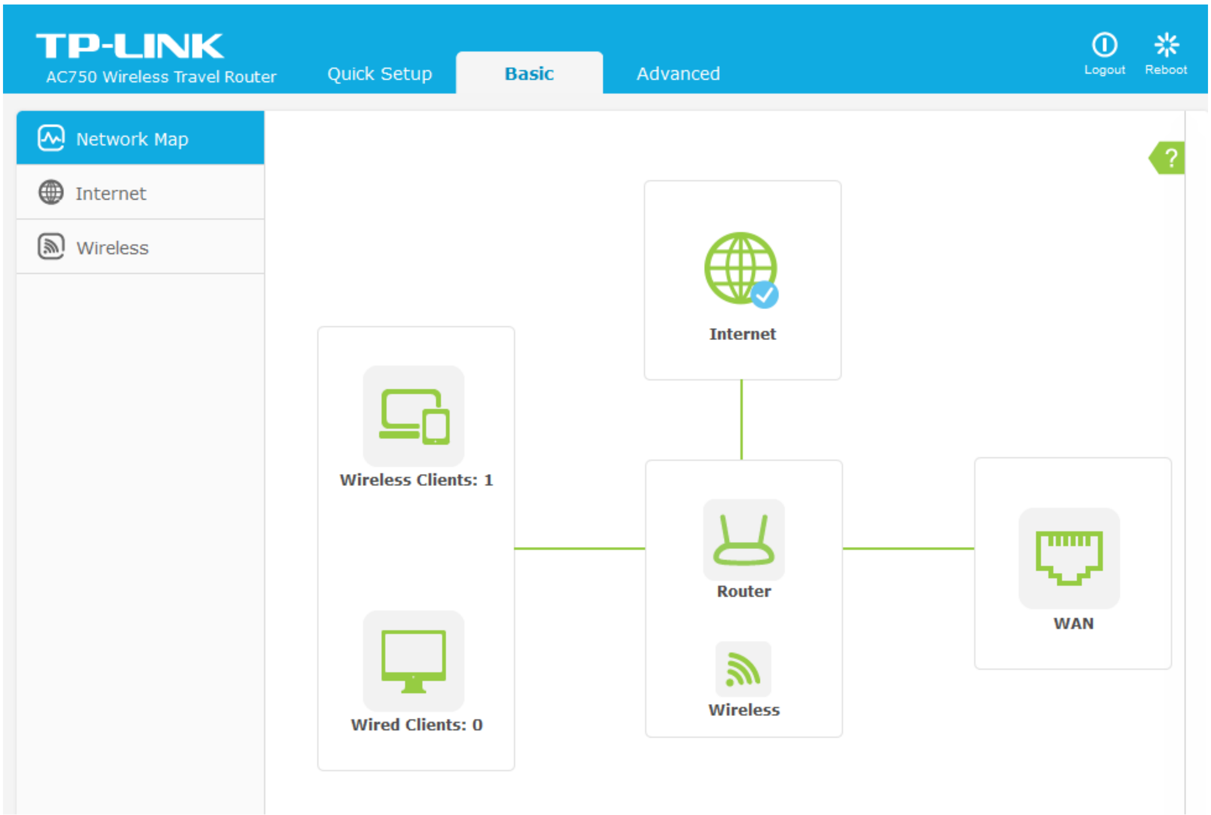


Figure 2-3: TP-Link travel router user interface

2.2.3 Limitations

While these devices offer convenience through their size and portability, they often have limited processing capabilities, restricting their ability to run resource-intensive services such as comprehensive ad-blocking or deep packet inspection. For instance, upgrading the TL-WR902AC with OpenWrt enables WireGuard VPN functionality; however, the device achieves approximately 15 Mbps download and 16 Mbps upload speeds through the VPN, indicating limitations in handling higher bandwidth applications. [4]

Additionally, user experiences highlight potential reliability issues. For example, some users have reported frequent Wi-Fi disconnects and missed pings with the GL-AR750S, whereas the TP-Link TL-WR902AC demonstrated more stable performance in similar conditions. [5]

2.2.4 Advancements

Recent advancements in travel routers include the integration of Wi-Fi 6 (802.11ax) technology, which offers improved efficiency and higher data rates. Devices like the TP-Link TL-WR1502X (AX1500) Wi-Fi 6 Travel Router provide enhanced performance, supporting features such as MU-MIMO and OFDMA for better connectivity. [6]

Similarly, ASUS has introduced the RT-AX57 Go, a portable AX3000 Wi-Fi 6 router featuring tri-mode connectivity and comprehensive VPN support, catering to users requiring secure and high-speed connections on the go. [7]

To add, since these devices are very compact and lack advanced features, they have very minimal and user-friendly web interface for minor monitoring and basic setups, which serves as a great selling point to consumers for their portability and easy setup when traveling or changing locations often (Figure 2-3).

In summary, while commercial travel routers offer portability, minimalistic user interface, and basic networking features, their limited processing power and simplified interfaces may not meet the needs of users requiring advanced functionalities and high performance. Recent models incorporating Wi-Fi 6 technology present improvements; however, considerations regarding processing capabilities and feature sets remain pertinent when selecting a suitable device.

2.3 System 2: Advanced Home Routers

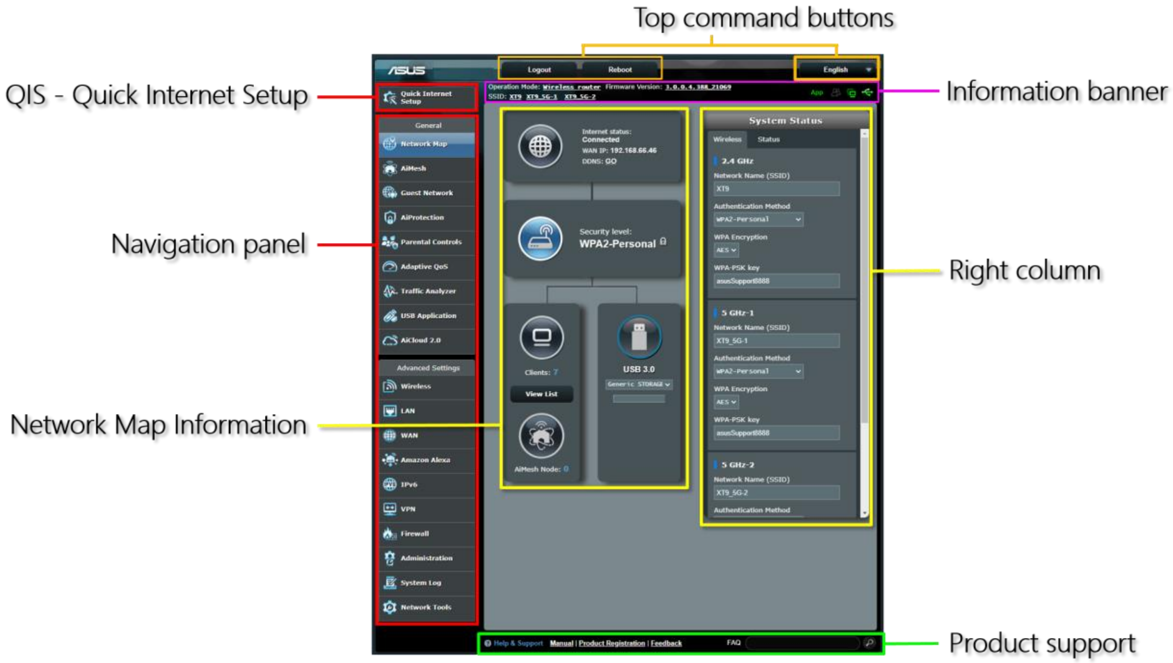
Standard advanced home routers from manufacturers like ASUS, Netgear, and Linksys offer significantly more powerful hardware than travel routers, enabling better performance and more sophisticated features. These devices are designed for stationary use in homes or small offices.

2.3.1 ASUS Routers:

ASUS offers advanced home routers like the RT-AC86U and RT-AX88U, designed to provide high performance and robust features for home networks.

- **ASUS RT-AC86U**: This router delivers Wi-Fi speeds up to 2,167 Mbps, utilizing MU-MIMO technology to enhance simultaneous data streaming. It includes game-boosting features such as the WTFast® game accelerator and Adaptive QoS to optimize gaming experiences. The RT-AC86U also supports AiProtection, providing network security features, parental controls, and some ad-blocking capabilities (Figure 2-4)
- **ASUS RT-AX88U**: Equipped with a 1.8 GHz quad-core processor, 1 GB RAM, and supporting Wi-Fi 6 (802.11ax), the RT-AX88U offers speeds up to 6,000 Mbps. It features OFDMA and 160 MHz bandwidth for improved efficiency and throughput. The router also provides advanced security features through AiProtection, powered by Trend Micro™, offering network security and parental controls. [11]

Figure 2-4: ASUS RT-AC86U Home Router



Operation Mode: Wireless router Firmware Version: 3.0.0.4.388_21069
SSID: XT9 XT9_5G-1 XT9_5G-2

Internet status: Connected
WAN IP: 192.168.66.46
DDNS: GO

Security level: WPA2-Personal

Clients: 7

Generic STORAGE

AiMesh Node: 0

2.4 GHz
Network Name (SSID): XT9
Authentication Method: WPA2-Personal
WPA Encryption: AES
WPA-PSK key: asusSupport8888

5 GHz-1
Network Name (SSID): XT9_5G-1
Authentication Method: WPA2-Personal
WPA Encryption: AES
WPA-PSK key: asusSupport8888

5 GHz-2
Network Name (SSID): XT9_5G-2
Authentication Method

Figure 2-5: ASUS router user interface

2.3.2 Netgear Nighthawk Series:

Netgear's Nighthawk series, including models like the R7800 and R8000, offers high-performance routers with advanced features suitable for home and small office environments.

- **Netgear Nighthawk R7800 (X4S)**: This router features a 1.7 GHz dual-core processor and supports AC2600 Wi-Fi speeds (up to 2,533 Mbps). It includes MU-MIMO and Quad Stream technology for simultaneous streaming to multiple devices. The R7800 also offers Dynamic QoS for bandwidth prioritization and ReadySHARE® USB access for network storage. [12]
- **Netgear Nighthawk R8000 (X6)**: The R8000 is a tri-band router with combined Wi-Fi speeds up to 3,200 Mbps. It features a 1.8 GHz dual-core processor, MU-MIMO technology, and supports NETGEAR Armor™ for cybersecurity protection. The router also offers Smart Parental Controls to manage internet usage. [13]

2.3.3 Limitations

While these advanced home routers provide enhanced performance and features compared to travel routers, they have certain limitations:

- **Complex User Interfaces**: The comprehensive interfaces can be intimidating for non-technical users, making advanced configurations challenging.
- **Subscription Services**: Some advanced features, such as enhanced security or parental controls, may require additional subscription services, increasing the total cost of ownership.
- **Portability**: Designed for stationary use, these routers have larger form factors and lack battery operation, making them unsuitable for portable applications.

In summary, while advanced home routers from manufacturers like ASUS and Netgear offer powerful hardware and sophisticated features, they are primarily designed for stationary use in homes or small offices. Their complexity and lack of portability may limit their suitability for users seeking mobile networking solutions.

2.4 System 3: DIY / Open-source Routers

DIY and open-source router projects enable users to repurpose general-purpose computing hardware into customized networking solutions, offering flexibility and advanced features beyond those of standard consumer routers. These projects typically utilize operating systems such as OpenWrt, pfSense, or OPNsense, each providing unique capabilities.

2.4.1 OpenWrt

OpenWrt is a Linux-based operating system specifically designed for embedded devices, including routers. It supports a wide array of hardware platforms, from commercial routers to single-board computers like the Raspberry Pi. OpenWrt offers a customizable environment with a robust package management system, enabling users to install features such as ad-blocking (via AdGuard Home or Pi-hole), VPN services, traffic shaping, and monitoring tools. [14]

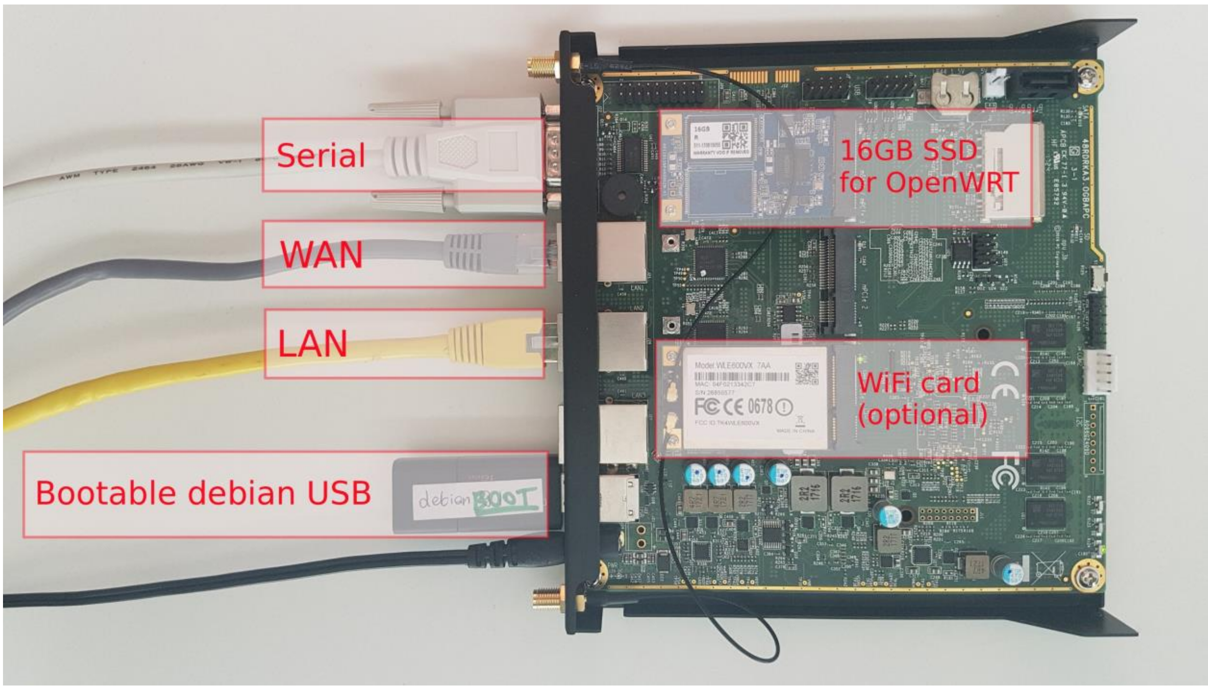


Figure 2-6: DIY OpenWrt Router

2.4.2 pfSense and OPNsense

pfSense and OPNsense are open-source firewall and routing platforms based on FreeBSD. They are typically deployed on x86 hardware, including repurposed PCs or dedicated network appliances. Both platforms offer comprehensive features like stateful packet filtering, VPN support (including IPsec and OpenVPN), intrusion detection and prevention systems, and detailed traffic monitoring. Their web-based interfaces facilitate configuration and management, catering to users with varying levels of technical expertise.

2.4.3 Hardware Platforms

DIY router projects often utilize single-board computers (SBCs) or mini PCs as hardware platforms:

- **Raspberry Pi**: Widely used due to its affordability and community support, the Raspberry Pi can function as a router when equipped with additional network interfaces via USB adapters. Projects like Jeff Geerling's "Raspberry Pi Router" demonstrate its viability in networking applications.

- **Mini PCs**: Compact devices with multiple network interfaces, such as those based on Intel's NUC platform, serve as capable router hardware. These devices offer sufficient processing power to handle advanced networking tasks and can run operating systems like OPNsense. [15]

2.4.4 Advantages

- **Customization**: Users can tailor the router's functionality to specific needs, installing only the desired features and services.
- **Advanced Features**: DIY solutions often support enterprise-grade features, including advanced security configurations, detailed traffic analysis, and extensive VPN options.
- **Community Support**: Active communities around these projects provide documentation, plugins, and forums for assistance and development.

2.4.5 Limitations

- **Technical Expertise**: Setting up and maintaining DIY routers requires a higher level of technical knowledge compared to consumer-grade routers.
- **User Interface**: While powerful, interfaces like LuCI for OpenWrt are designed for technical users and may present a learning curve for casual users.

- **Hardware Compatibility**: Not all hardware is compatible with these operating systems, necessitating careful selection and potential troubleshooting.

In summary, DIY and open-source router projects offer a high degree of customization and advanced features, appealing to users with specific networking requirements and the technical proficiency to implement them. However, the complexity and maintenance demands may not suit all users, particularly those seeking straightforward, plug-and-play solutions.

2.5 Systems Comparison

This section compares the three categories of router solutions based on three criteria relevant to users seeking a balance between portability, performance, features, and usability.

Table 2-1 illustrates compares the three main routing systems in terms of graphical interfaces (GUIs), and how easy and responsive it is.

Table 2-1: Comparison Table Based on Graphical Interfaces

Criterion 1	**Commercial Travel Routers**	**Advanced Home Routers**	**DIY/Open-Source Solutions**
Graphical Interface			
Good user interface	Medium (simplified but limited)	Medium (feature-rich but complex)	Low (technical and intimidating)
Easy and effective navigation	High	Medium	Low
Simple and professional Design	High	Medium	Low
Responsive	Medium	High	Medium

Table 2-2 compares the three aforementioned systems based on their content and how well they function.

Table 2-2: Comparison Table Based on Content and Functionality

Criterion 2	**Commercial Travel Routers**	**Advanced Home Routers**	**DIY/Open-Source Solutions**
Content and Functionality			
Quality content structure	Medium	High	Medium
Usability	High	Medium	Low
Dynamic content	Low	Medium	High
Content control	Low	Medium	High

Table 2-3 compares the commercial travel routers and home routers versus the open-source routers based on the advanced features that each of them holds.

Table 2-3: Comparison Table Based on Features

Criterion 3	**Commercial Travel Routers**	**Advanced Home Routers**	**DIY/Open-Source Solutions**
Features			
Security measures	Medium	High	Very High (customizable)
Ad-blocking capabilities	Low	Medium	High
VPN support	Medium	High	Very High
Parental controls	Low	High	Medium
Multiple networks	Low	Medium	High
System monitoring	Low	Medium	Very High
Portability	Very High	Very Low	Medium
Customization	Low	Medium	Very High
Cost effectiveness	Medium	High	High
Technical expertise required	Low	Medium	High

2.6 Conclusion and Motivation

After reviewing the existing router solutions across commercial travel routers, advanced home routers, and DIY/open-source projects, several limitations become apparent. Commercial travel routers offer excellent portability but lack advanced features like comprehensive ad-blocking, detailed system monitoring, and effective parental controls. Advanced home routers provide robust feature sets but are expensive, not portable, and often have complex interfaces that overwhelm non-technical users. DIY solutions offer maximum flexibility and customization but require significant technical expertise to set up and maintain, with interfaces designed primarily for technical users.

A clear gap exists in the market for a solution that combines the portability of travel routers, the advanced features of home routers, and the customization of DIY solutions while maintaining a user-friendly interface accessible to non-technical users. This gap is particularly evident when considering users who need enhanced security features, ad-blocking capabilities, and parental controls without the complexity typically associated with these advanced features.

The proposed Raspberry Pi 4-based travel router with OpenWrt aims to address these limitations by providing a balanced solution that is both portable and feature-rich. By leveraging the hardware capabilities of the Raspberry Pi 4, the project can support resource-intensive features like AdGuard for ad-blocking, OpenVPN for secure connections, SQM for traffic management, and comprehensive system monitoring. The customized web interface will make these advanced features accessible to users without technical backgrounds, allowing them to manage their network settings without requiring expert assistance.

Additionally, the proposed solution addresses specific use cases that existing solutions handle poorly, such as providing secure networking for travelers, offering robust parental controls for families, and enabling non-technical users to access advanced networking features. The potential integration of an AI assistant for the web interface further enhances usability by providing guided configuration and troubleshooting.

This project is motivated by the need to democratize access to advanced networking features, making them available to users regardless of their technical expertise. By combining the best aspects of existing solutions while addressing their limitations, the proposed Raspberry Pi 4-based travel router aims to provide a more inclusive, versatile, and user-friendly networking solution.

CHAPTER 3
SYSTEM DESIGN

3.1 Introduction

This chapter outlines the design process of the system, covering key aspects such as requirements analysis, specifications, and functional design. It provides a structured overview of the system's architecture and the necessary components to achieve the desired functionality.

3.2 Requirements and Specification Analysis

The project was implemented to create a router management system that provides users with seamless control over their network, ensuring optimized connectivity, security, and device management. The system simplifies network administration by allowing users to monitor network activity, manage devices, and configure security settings through an intuitive interface.

To design an efficient router management system, it is crucial to have a clear understanding of its functionalities and how it will be used by the main stakeholder (the network administrator). This is where use case diagrams, sequence diagrams, activity diagrams, and class diagrams play a vital role in defining the system’s structure and behavior.

Together, these diagrams provide a comprehensive overview of the router management system, detailing its functionality, data flow, and interactions between different components. They serve as essential tools for designers and developers to ensure the system is well-structured, user-friendly, and optimized for performance, security, and scalability.

3.2.1 Functional Requirements

The system must support a range of operations and activities to ensure efficient network management and user control. Below are the key functionalities and benefits provided by the system:

- Facilitates seamless interaction between users and their network by providing real-time monitoring and management of connected devices.
- Enables network customization by allowing users to configure settings such as SSID, passwords, channel, bandwidth limits and security options to fit their specific needs.
- Ensures network security and access control by offering features like device blocking, firewall management, and VPN integration.
- Enhances user experience with intuitive controls, allowing users to easily toggle services like AdGuard, VPN, and parental controls for optimized browsing.
- Provides real-time insights into network performance, including internet speed, bandwidth usage, and latency, helping users make informed decisions about their network usage.
- Improves accessibility by offering a user-friendly mobile or web interface, making it easy for users to manage their network from any device.

These functionalities contribute to an efficient, secure, and user-centric router management system, ensuring users have full control over their network with ease and flexibility.

3.2.2 Use Case Diagrams

A use case diagram for a learning system is a graphical representation of the system's use cases, actors, and their interactions. It provides a high-level view of the system's functionality and the actors that interact with it. The use case diagram depicts the dynamic behavior of the learning system and helps to identify the system's requirements and the actors' roles (See Figure 3-1).

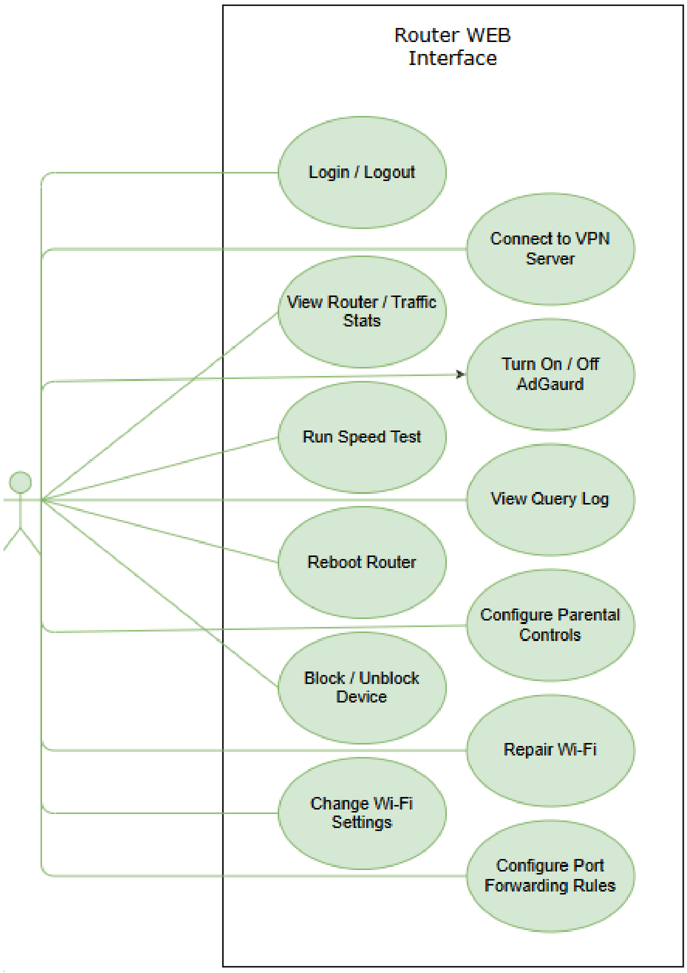


Figure 3-1: Use Case Diagram

In this project, the use cases include monitoring network performance, managing connected devices, configuring router settings, enabling/disabling VPN and AdGuard, and applying parental controls. The actors interacting with the system include network administrators and regular users, each with specific access privileges and responsibilities.

The use case diagram defines the relationships between these actors and system functionalities, outlining who can perform specific actions and how they interact with the router management system. For instance, a network administrator may have the ability to block devices, modify SSID settings, and manage firewall rules, while a regular user may be limited to checking internet speed and toggling VPN on/off.

By mapping out these interactions, the use case diagram helps to clarify system requirements, streamline development, and ensure a well-structured user experience that meets the needs of all stakeholders.

3.3 System Architecture

The system architecture of our project is structured into three main layers:

3.3.1 Hardware and Operating System Layer

At the foundation of our system is the Raspberry Pi 4 Model B, running OpenWrt, a Linux-based operating system optimized for networking applications. The Raspberry Pi is equipped with a USB Wi-Fi adapter antenna for improved wireless performance and a 32GB microSD card for storage. This layer handles low-level networking operations, managing internet traffic, and executing core router functionalities.

3.3.2 Feature Layer

Built on top of the hardware and OS, the feature layer provides essential network services, including:

- **VPN Integration (OpenVPN)** – Encrypts internet traffic to ensure privacy and security.
- **AdGuard** – Implements DNS-based ad blocking and filtering.
- **Firewall Management** – Controls network traffic and enforces security policies.
- **Port Forwarding** – Allows selective access to internal services from external networks.

- **Guest Wi-Fi Management** – Provides temporary internet access with customizable settings.
- **Speed Test & Traffic Stats** – Monitors network performance and usage in real time.

These features enhance network security, control, and optimization, making the router smarter and more efficient.

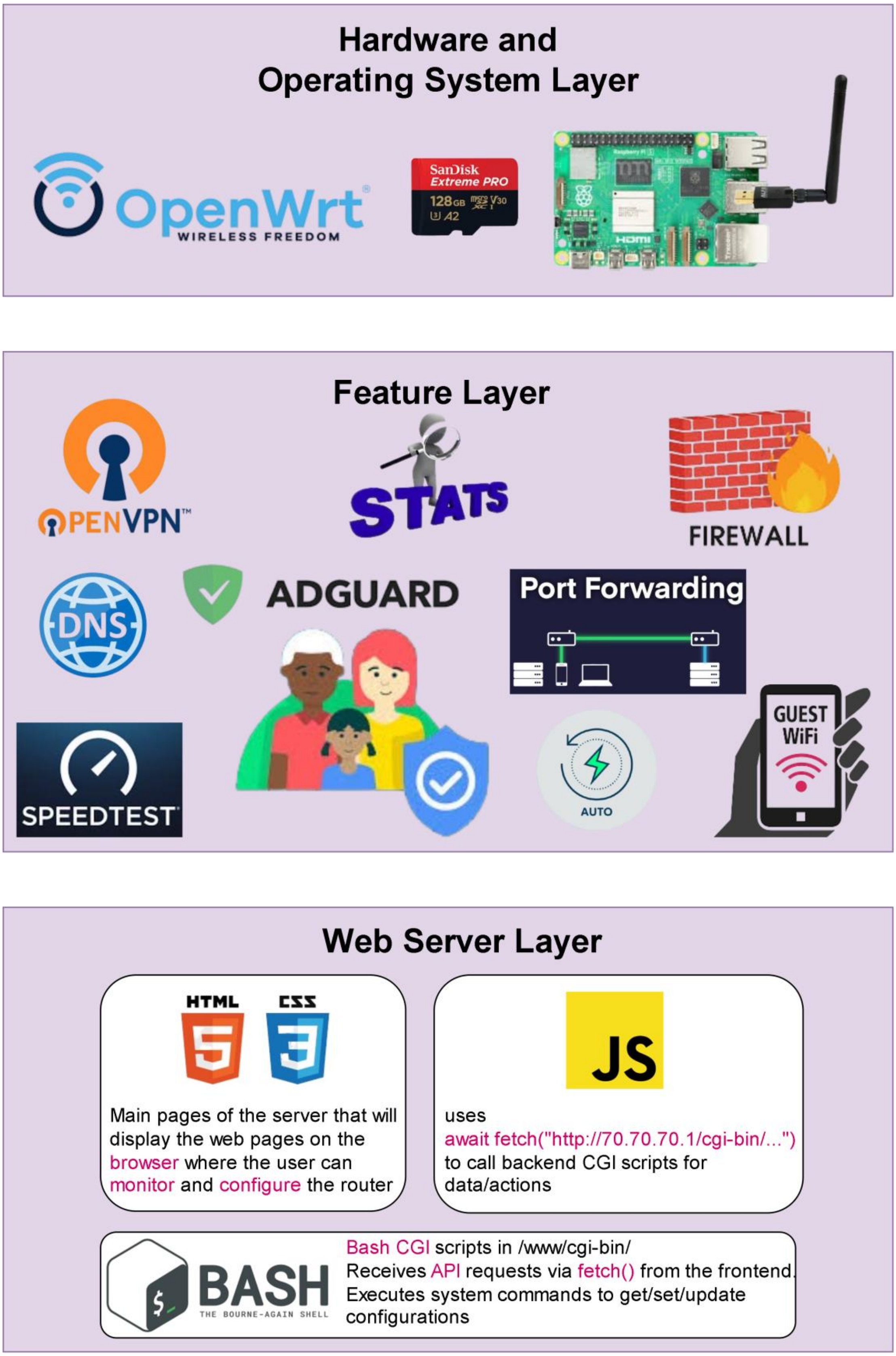
3.3.3 Web Server Layer

The user interacts with the system through a web-based interface hosted on the Raspberry Pi. This interface consists of multiple HTML pages, each serving a specific function:

- **main.html** – Handles user authentication.
- **dashboard.html** – Displays router statistics, speed test results, and traffic data, and allows the blocking and unblocking of devcices.
- **vpn.html** – Allows admin to toggle VPN settings and switch servers.
- **adguard.html** – Controls DNS filtering, parental controls and displays real-time statistics.
- **guest.html** – Manages guest Wi-Fi settings, including SSID, bandwidth limits, and QR code generation.
- **settings.html** – Configures Wi-Fi, LAN, and port forwarding settings.

The web interface communicates with Bash CGI scripts via JavaScript fetch() API calls, allowing users to execute network commands and update configurations dynamically. These scripts handle system-level commands such as retrieving device information, modifying firewall rules, and managing VPN connections.

This architecture ensures a modular, scalable, and user-friendly system, enabling seamless interaction between the user and the router's core functionalities. (Figure 3-2)



Main pages of the server that will display the web pages on the browser where the user can monitor and configure the router

uses
await fetch("http://70.70.70.1/cgi-bin/...")
to call backend CGI scripts for data/actions

Bash CGI scripts in /www/cgi-bin/
Receives API requests via fetch() from the frontend.
Executes system commands to get/set/update configurations

Figure 3-2: System Architecture

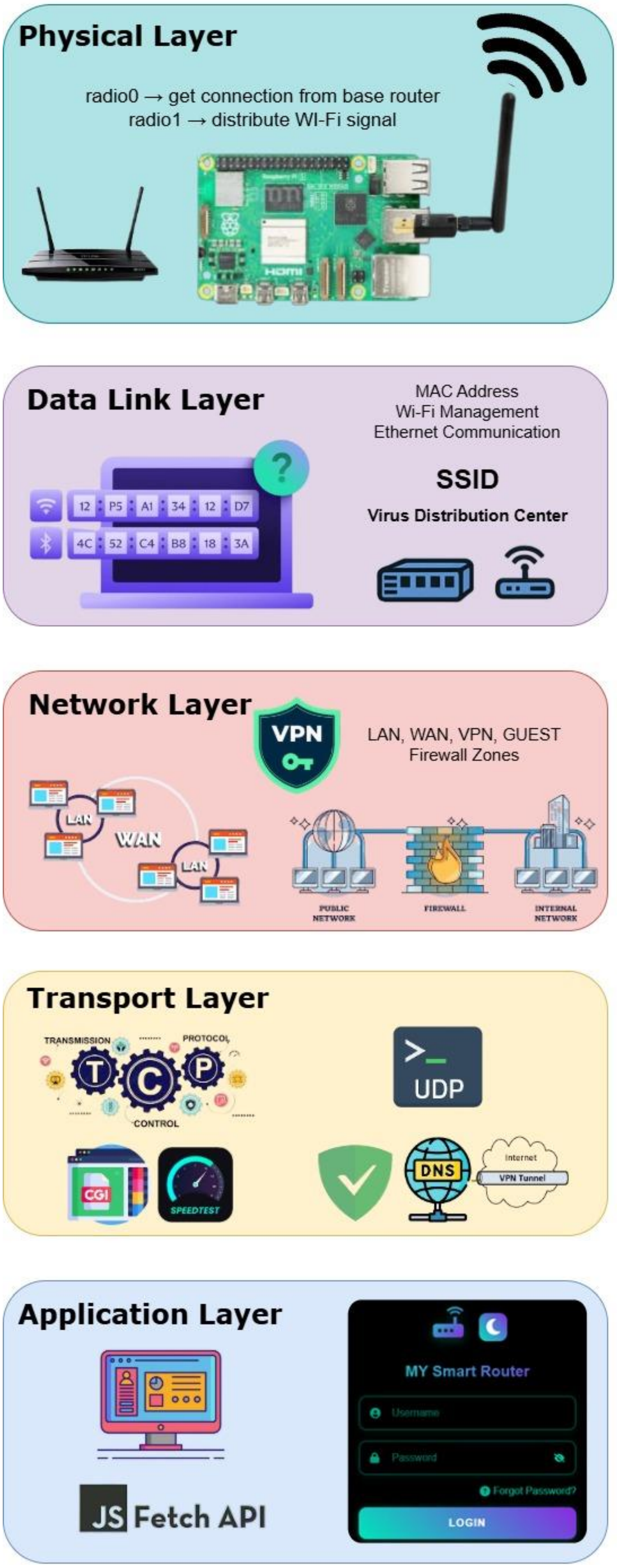


Figure 3-3: Network Protocol Stack Visualization

Figure 3-3 illustrates the OSI model's layered architecture and its practical implementation in a router-based setup. Each layer represents a critical function in the data communication process:

1. Physical Layer: Depicts hardware elements like the home / public router and the Raspberry-pi router and its radios (NIC and USB Wi-Fi adapter) responsible for signal transmission and Wi-Fi distribution.
2. Data Link Layer: Manages MAC addresses, SSID configurations, and Ethernet communication.
3. Network Layer: Highlights IP-based routing between LAN, WAN, and VPN networks, emphasizing firewall zones for security.
4. Transport Layer: Demonstrates reliable (TCP) and fast (UDP) transmission protocols, along with DNS resolution and VPN tunneling for secure connections.
5. Application Layer: Showcases user-facing components like the web interface and JavaScript Fetch API used for interacting with router configurations.

This layered breakdown reflects how the smart router handles data from the physical connection up to user interactions, emphasizing both network management and security. An example is provided in Figure 3-4.

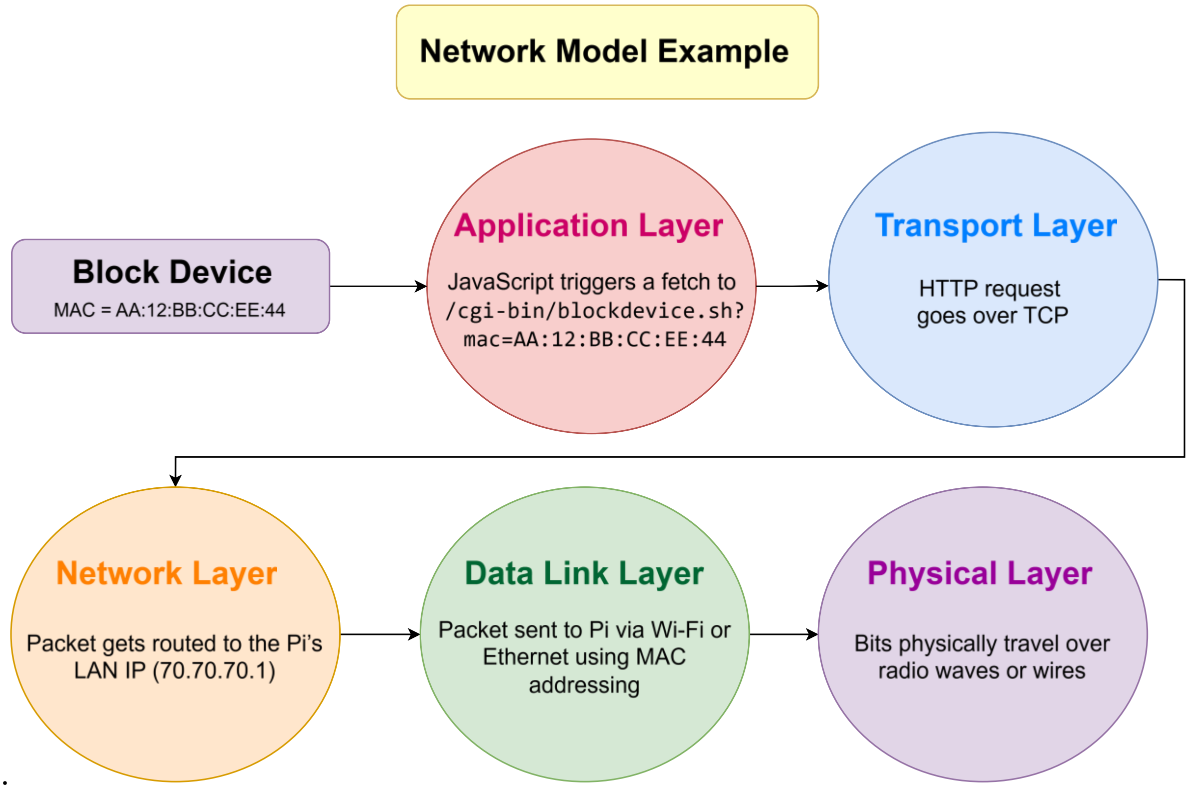

Network Model Example
Block Device
MAC = AA:12:BB:CC:EE:44
Application Layer
JavaScript triggers a fetch to /cgi-bin/blockdevice.sh?mac=AA:12:BB:CC:EE:44
Transport Layer
HTTP request goes over TCP
Network Layer
Packet gets routed to the Pi's LAN IP (70.70.70.1)
Data Link Layer
Packet sent to Pi via Wi-Fi or Ethernet using MAC addressing
Physical Layer
Bits physically travel over radio waves or wires

Figure 3-4: Network Model Example

3.4 Class Diagrams

A class diagram is a UML representation that outlines the structural blueprint of a software system by defining its classes, attributes, methods, and relationships. It provides a static view of the system's architecture, helping developers understand how different components interact.

In our router management system, the class diagram (Figure 3-5) represents key system components and their associations. The Router class serves as the central entity, linking various functionalities, including VPN management, AdGuard filtering, parental controls, guest network settings, and speed testing. Each of these modules has its own attributes and methods, defining their role within the system.

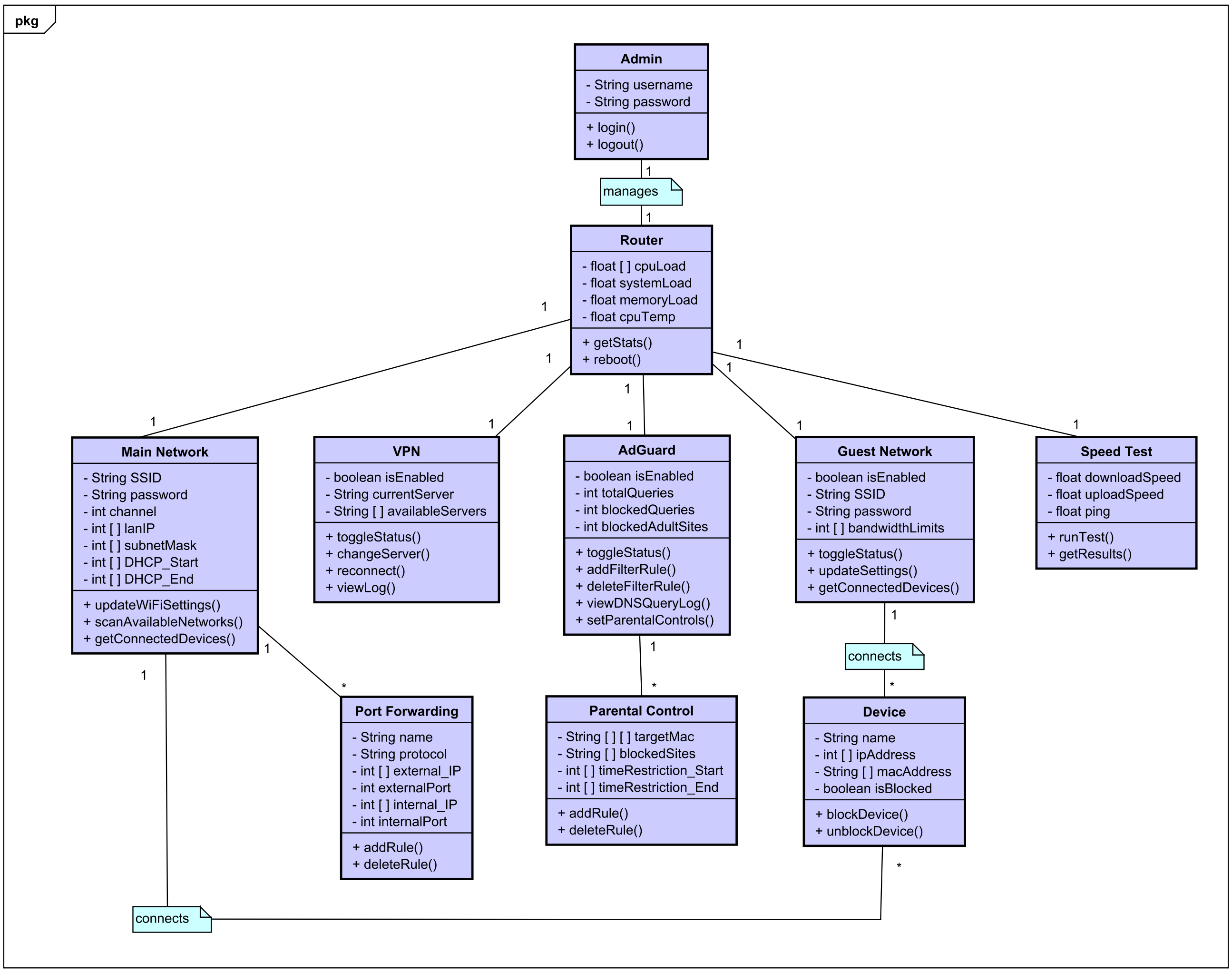


Figure 3-5: Class Diagram

For instance, the Admin class manages authentication with attributes such as username and password, while the Main Network and Guest Network classes handle SSID configurations. Other critical classes include Parental Control, which manages blocked devices based on time-based restrictions, and Device, which tracks connected devices and their status.

By organizing the system into well-defined classes and relationships, the class diagram ensures a structured, scalable, and maintainable implementation. It serves as a valuable reference during design, development, and debugging, ensuring that the system's functionality is both efficient and logically structured.

3.5 Sequence Diagrams

Sequence diagrams are a type of UML interaction diagram that visually represent the sequence of operations within a system. They emphasize the order of interactions over time, using a vertical axis to indicate the chronological flow of messages between system components. These diagrams help in understanding how different objects and actors collaborate to achieve specific functionalities.

For our router management system, the sequence diagram (Figure 3-6) illustrates the interactions that happen in the Dashboard page between the user, the router, and various system components such as the firewall and device manager. It details processes like user authentication, viewing network statistics, running a speed test, managing connected devices, blocking unauthorized devices, and rebooting the router.

By modeling these interactions, the sequence diagram provides a clear depiction of the system’s behavior and response to user actions. It aids in identifying potential inefficiencies, ensuring smooth operation, and optimizing communication between system modules. Ultimately, the diagram serves as a valuable tool for refining the system's design and improving its usability.

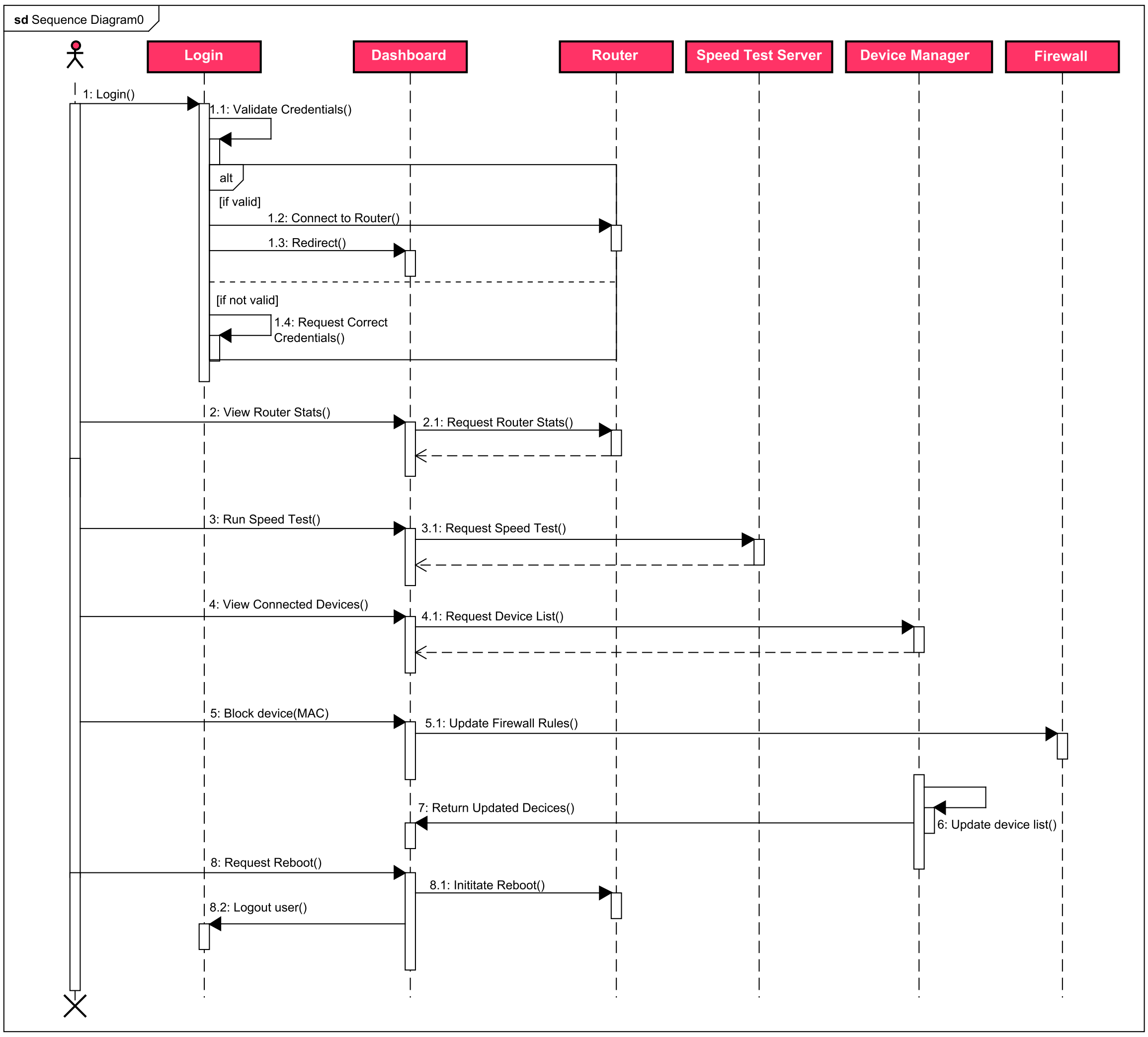


Figure 3-6: Dashboard Sequence Diagram

3.6 Activity Diagrams

Activity diagrams are UML representations that depict the flow of processes and interactions within a system. They help visualize how tasks are coordinated to provide a service, particularly in scenarios where multiple operations need synchronization or when events within a use case overlap and require structured coordination.

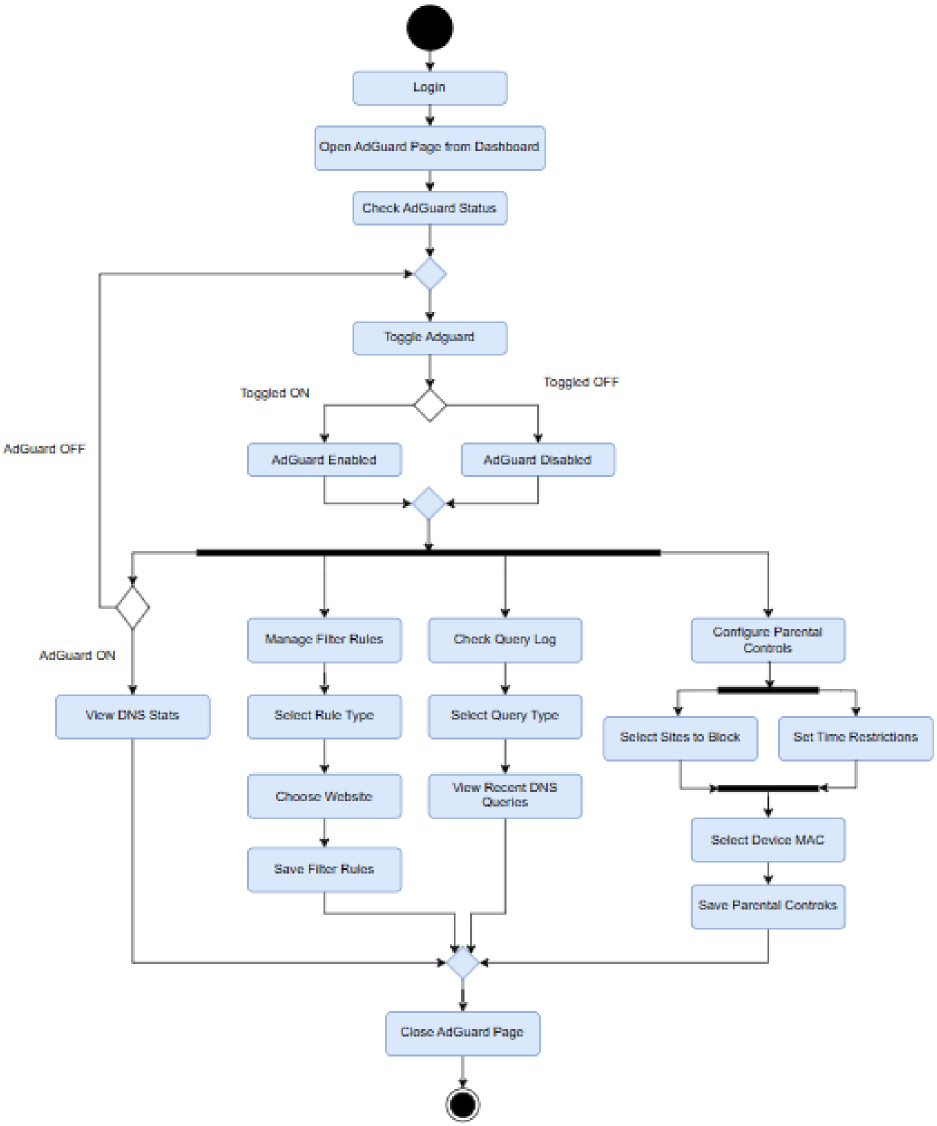


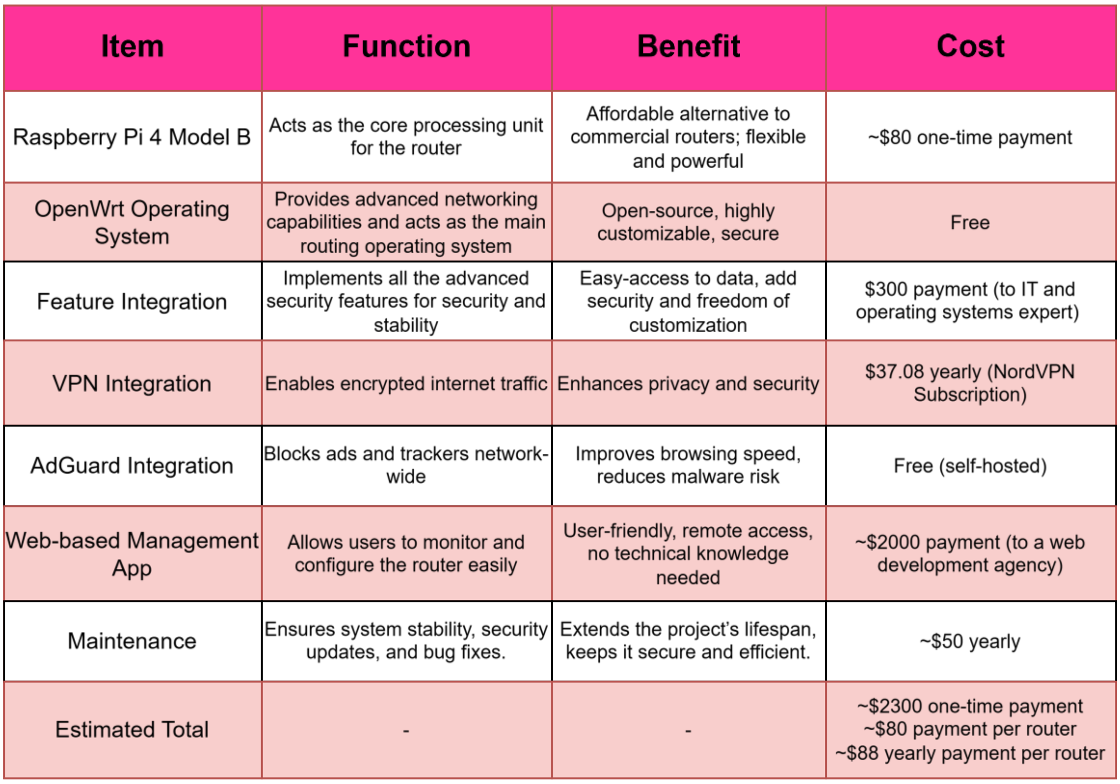
Figure 3-7: AdGuard Activity Diagram

In the context of our router management system, activity diagrams illustrate the step-by-step execution of key functionalities in the AdGuard page, such as enabling AdGuard, managing custom filtering rules, viewing DNS statistics, checking query logs, and configuring parental controls (Figure 3-7). These diagrams highlight decision points, such as toggling AdGuard on or off, selecting blocked or allowed queries, and setting up filtering criteria.

By providing a clear, graphical representation of system processes, activity diagrams help in refining system logic, optimizing user interactions, and ensuring efficient workflow execution. They serve as a crucial tool in the system's design phase, enabling better development, troubleshooting, and user experience analysis.

3.7 Non-Technical Aspects

3.7.1 Financial Viability



Item	Function	Benefit	Cost
Raspberry Pi 4 Model B	Acts as the core processing unit for the router	Affordable alternative to commercial routers; flexible and powerful	~$80 one-time payment
OpenWrt Operating System	Provides advanced networking capabilities and acts as the main routing operating system	Open-source, highly customizable, secure	Free
Feature Integration	Implements all the advanced security features for security and stability	Easy-access to data, add security and freedom of customization	$300 payment (to IT and operating systems expert)
VPN Integration	Enables encrypted internet traffic	Enhances privacy and security	$37.08 yearly (NordVPN Subscription)
AdGuard Integration	Blocks ads and trackers network-wide	Improves browsing speed, reduces malware risk	Free (self-hosted)
Web-based Management App	Allows users to monitor and configure the router easily	User-friendly, remote access, no technical knowledge needed	~$2000 payment (to a web development agency)
Maintenance	Ensures system stability, security updates, and bug fixes.	Extends the project's lifespan, keeps it secure and efficient.	~$50 yearly
Estimated Total	-	-	~$2300 one-time payment ~$80 payment per router ~$88 yearly payment per router

Figure 3-8: Cost - Benefit Analysis Diagram

The financial viability of our Raspberry Pi-based router solution is evident when compared to traditional commercial routers. With a one-time hardware investment of approximately $80, plus optional features like VPN integration ($37.08 / year), our system delivers enterprise-level security, network management, and ad-blocking at a fraction of the cost of high-end commercial routers.

Most advanced commercial routers cost $250 or more upfront, with additional subscription fees for features such as VPN services, parental controls, and ad-blocking. In contrast, our self-hosted, open-source approach eliminates these recurring costs, allowing users to achieve the same functionality without hidden fees or limitations.

Additionally, our custom web-based management app ensures ease of use without requiring external services or expensive third-party software. This reduces operational costs and increases accessibility, making it a viable solution for both individual users and small businesses.

From a long-term perspective, our project offers a high return on investment (ROI) by reducing the need for frequent hardware replacements, licensing fees, and premium service subscriptions. The scalability and flexibility of our open-source design also mean that users can upgrade and modify the system rather than purchasing entirely new hardware, further increasing cost efficiency.

By combining affordability, sustainability, and high-performance networking, our Raspberry Pi-based solution presents a cost-effective and financially sustainable alternative to traditional routers, proving its viability as a market-ready product.

3.7.2 Stakeholders

1. Primary Beneficiaries (Who Will Benefit?)

- Home Users & Network Administrators: Individuals who require a customizable, cost-effective router solution with enhanced control over their home network will benefit the most. The project provides a way to optimize network performance, improve security, and have more transparency over their internet usage.
- Developers & Open-Source Community: The project contributes to the broader OpenWrt ecosystem, allowing developers to modify, improve, and extend its functionality. It serves as a learning tool and a platform for innovation.

- Educational Institutions & Students: Networking, security, and embedded systems students can use the project as a hands-on learning experience, enhancing their understanding of real-world networking concepts.
- Privacy-Conscious Users: People who want an alternative to proprietary routers with unknown firmware behaviors and data collection policies can benefit from an open-source, transparent solution that they can audit and customize.

2. Potentially Affected Parties (Who May Be Harmed?)

- Internet Service Providers (ISPs): Some ISPs impose restrictions on routers or enforce specific settings. A fully customizable OpenWrt-based router might allow users to bypass certain limitations, which could create conflicts with ISP policies.
- Inexperienced Users: Users unfamiliar with networking concepts might misconfigure the router, leading to security vulnerabilities, network failures, or performance issues. A lack of understanding could result in unintended exposure to cyber threats.
- Manufacturers of Proprietary Routers: The availability of a low-cost, open-source router solution could reduce demand for commercial routers, especially those that rely on proprietary firmware with limited customizability.

3. Key Decision-Makers (Who Should Have a Say?)

- Project Developers & Maintainers: The individuals responsible for building and maintaining the OpenWrt router system must ensure its security, stability, and usability while keeping it open for further development.
- Users & Testers: End users play a critical role in providing feedback on usability, performance, and feature requests, which helps refine the project.

- Security Experts & Ethical Hackers: Professionals in the cybersecurity field should be involved in auditing the system to ensure it does not introduce security flaws that could be exploited.
- Regulatory Bodies: Governments and cybersecurity organizations may impose regulations regarding wireless communication, encryption standards, and user data protection. The project must comply with these regulations to avoid legal concerns.

3.7.3 Scope

I. In-Scope (What Will Be Done in the Project)

The project focuses on developing a custom OpenWrt-based router using a Raspberry Pi 4 Model B with a wireless dongle, aiming to provide an advanced yet user-friendly networking solution. The core functionalities include:

1. Router & Network Management:
   - The Raspberry Pi will serve as a fully functional router, connecting to an existing WiFi network and broadcasting another via a wireless dongle.
   - Support for multiple networks, including a guest network for better security and bandwidth separation.
2. User-Friendly Web Interface:
   - A modern, intuitive web UI with light and dark themes, offering an improved experience over traditional router interfaces.
   - The interface will be fully responsive, ensuring optimal performance across all devices, including desktops, tablets, and smartphones.
   - Essential features needed by regular users will be available in an easy-to-navigate dashboard.
3. Security & Privacy Enhancements:

- VPN support for encrypted connections.
- AdGuard integration for blocking ads, adult content, malware, and phishing attempts.
- Parental control features, allowing users to block WiFi or specific sites on children’s devices for custom durations.

4. Performance Optimization & Monitoring:
    - QoS SQM with bandwidth control for optimizing network performance.
    - Scheduled reboots to maintain optimal performance.
    - WiFi repair tool to automatically resolve common issues without ISP intervention.
    - Speed test functionality for measuring ping, download, and upload speeds.
    - Raspberry Pi system stats display (CPU temperature, system load, cpu cores’ load).
    - Device usage statistics and overall network usage will be tracked, displayed on the web interface to give users insights into data consumption per device and bandwidth usage over time.
5. Advanced Network Controls:
    - Port forwarding with built-in predefined rules for common applications.
    - Query log monitoring to track DNS queries, including blocked ones.
    - Device management, allowing users to block/unblock any device from the network.
    - Ability to connect wirelessly to any WiFi network, with signal strength indicators for better selection.

II. Out-of-Scope (What Will NOT Be Done in the Project)

To ensure feasibility and focus, the following aspects are not included:

1. No Hardware Modifications:
    - The project will strictly use a Raspberry Pi 4 Model B with a USB WiFi dongle, without any custom hardware modifications.
2. No Enterprise-Level Features:
    - It is designed for home and small network use, not large-scale corporate environments. (could be solved with a better Wi-Fi dongle)
    - No advanced VLAN configurations, enterprise-level security (e.g., IPS/IDS), or complex multi-WAN setups.
3. No ISP-Dependent Features:
    - The router will function as a secondary networking layer and will not replace an ISP-provided modem.
    - No deep integration with ISP services beyond standard networking features.
4. No AI-Based Network Analysis:
    - The system will log and monitor network activity but will not include AI-powered traffic analysis or automated security threat detection, since it requires machine learning.
5. No Dedicated Mobile App:
    - The focus is on a responsive web-based interface, optimized for desktop and mobile browsers.

By clearly defining the project scope, we ensure that the most essential and impactful features are implemented while allowing room for future enhancements.

3.7.4 Risks

Several factors may hinder the project from achieving its intended goals. The key risks include:

1. Hardware Limitations

- The Raspberry Pi lacks dedicated hardware for networking—its Gigabit Ethernet and USB controller share the same bus, which can create a bottleneck when handling high-speed data transfers.
- Since WiFi is provided via a USB WiFi dongle, performance will be slower and less reliable than routers with built-in, optimized WiFi chipsets.
- USB WiFi dongle limitations may lead to weaker signal strength, reduced range, or driver compatibility issues in OpenWrt.

2. Software & Compatibility Issues

- WiFi dongle support in OpenWrt can be inconsistent, potentially requiring additional drivers, firmware patches, or alternative hardware.
- The integration of AdGuard, VPN, and QoS SQM must be tested carefully to avoid conflicts or performance bottlenecks.
- Ensuring seamless wireless connectivity to any WiFi network while maintaining stability could require additional scripting or custom configurations.

3. Security Risks

- Vulnerabilities in OpenWrt, the web interface, or third-party packages (AdGuard, VPN, etc.) could expose the router to cyber threats.
- Weak firewall rules, open ports, or misconfigured settings could allow unauthorized access.

- Parental controls and device blocking features must be implemented securely to prevent bypassing.

4. Performance & Reliability Concerns

- The USB-based networking on the Raspberry Pi may introduce potential bottlenecks, affecting throughput and latency.
- Scheduled reboots, while improving stability, could disrupt active connections if not properly managed.
- Real-time device usage statistics and network monitoring may add processing overhead, requiring optimization.

5. User Experience Challenges

- The web interface must be modern, intuitive, and responsive on all devices (desktop, tablet, mobile). Ensuring a smooth user experience may require significant testing and iteration.
- Balancing simplicity and advanced controls can be challenging—technical users may want deeper customization, while regular users need an easy-to-use interface.

6. Implementation & Time Constraints

- Fine-tuning QoS SQM, parental controls, VPN, and AdGuard settings requires rigorous testing to ensure they work without unintended conflicts.
- Developing a feature-rich web interface with real-time stats, light/dark modes, and intuitive navigation within the project timeline may be demanding.

7. ISP-Dependent Limitations

- The router functions as a secondary networking layer, meaning it cannot override ISP-imposed restrictions, such as speed throttling or deep packet inspection (DPI).

- Users may have varying ISP configurations, affecting performance and requiring additional troubleshooting.

8. Scalability & Maintainability

- As network demands increase, future upgrades or performance optimizations may be necessary.
- Long-term maintenance, including security patches, bug fixes, and feature enhancements, will be required to keep the system secure and up to date.

9. Mitigation Strategies

To minimize these risks, the project will:

- Test multiple WiFi dongles for stability and OpenWrt compatibility.
- Implement efficient system resource management to optimize Raspberry Pi performance.
- Apply strict security measures, including firewall rules, encrypted connections, and access controls.
- Design the web interface to be fully responsive and user-friendly, with extensive testing across different devices.
- Structure development milestones to ensure timely implementation and feature completeness.

3.7.5 Schedule and Milestones

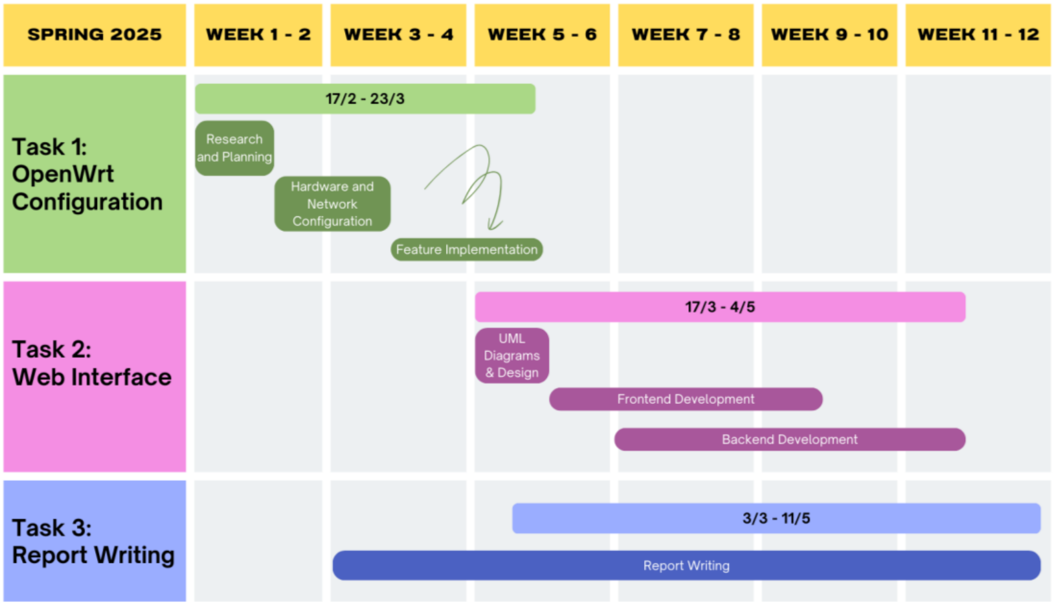


Figure 3-9: Scheduling Tasks and Milestones

Figure 3-9 summarizes the time plan to complete this project with its task division and duration of each task.

3.7.6 Ethical and Social Considerations

The development of an OpenWrt-based router using a Raspberry Pi 4 Model B involves several ethical and social considerations related to privacy, security, accessibility, responsible network management, and sustainability. These factors must be addressed to ensure ethical and responsible implementation.

1. User Privacy and Data Protection

Since the router processes network traffic, DNS queries, and device connections, user privacy must be a priority. Key considerations include:

- Minimal Data Logging: The system should avoid unnecessary logging of user activity beyond essential diagnostics.

- Secure Communications: Administrative access should be secured through SSH keys, HTTPS, and strong authentication to prevent unauthorized access.
- Transparency in Filtering: Features such as AdGuard and parental controls must allow users to understand and control filtering settings.

2. Security and Responsible Network Use

- Preventing Unauthorized Access: Security measures should be in place to prevent misuse, such as unauthorized device blocking or excessive content filtering.
- Open-Source Compliance: The use of OpenWrt and related software must adhere to open-source licensing regulations (e.g., GPL, MIT) to ensure ethical software distribution.
- Resilience Against Misuse: While the router supports privacy-enhancing tools such as VPN and DNS encryption, it must not encourage bypassing legal restrictions or engaging in illicit activities.

3. Accessibility and Digital Inclusion

- User-Friendly Design: The web interface should be intuitive and accessible to users with varying technical expertise.
- Support for Disabilities: The system should consider screen reader compatibility, high-contrast themes, and keyboard navigation to enhance accessibility.
- Affordable and Open Networking: By utilizing a low-cost Raspberry Pi, this project contributes to digital inclusion, offering an affordable alternative to proprietary networking solutions.

4. Ethical Considerations in Content Filtering and Parental Controls

- Balancing Safety and Privacy: Parental controls must allow customization and transparency to prevent excessive surveillance or unjust censorship.
- Avoiding Filtering Bias: Ad-blocking and content filtering rely on predefined blocklists, which may introduce bias. Users should have the ability to review and modify these lists as needed.

5. Environmental Impact and Sustainability

- Energy Efficiency: While the Raspberry Pi is more power-efficient than traditional routers, the implementation of power-saving mechanisms can further optimize energy consumption.
- Reducing E-Waste: The modular design of the system allows for long-term software updates, reducing the need for frequent hardware replacements.

6. Legal and Regulatory Compliance

- Network Security Regulations: The project must comply with local laws regarding data protection, content filtering, and VPN usage.
- ISP Policy Compliance: The router operates as a secondary networking layer and must not promote unauthorized bypassing of ISP-imposed restrictions.

3.7.7 Environmental and Sustainability Considerations

While this project does not have a significant environmental impact, certain aspects of its design contribute to energy efficiency, electronic waste reduction, and sustainability.

1. Energy Efficiency

- The Raspberry Pi 4 Model B consumes significantly less power than traditional commercial routers, making it a more energy-efficient alternative for home networking.

- Power consumption can be further optimized through scheduled reboots, service management, and power-saving configurations.

2. Reduction of Electronic Waste (E-Waste)

- By using open-source software (OpenWrt) and modular hardware, the system allows for continuous updates and repurposing, reducing the need for frequent hardware replacements.
- Unlike proprietary routers that may become obsolete due to manufacturer-imposed software limitations, this project extends the usable lifespan of the Raspberry Pi and compatible WiFi dongles.

3. Sustainable and Cost-Effective Networking

- The project promotes the reuse of existing Raspberry Pi hardware, which reduces the demand for new electronic components.
- It provides an affordable and sustainable alternative to purchasing new, high-cost routers, making it more accessible for users while minimizing resource consumption.

3.7.8 Relevant Standards

The design and implementation of the OpenWrt-based router using a Raspberry Pi 4 Model B must adhere to several technical and non-technical standards to ensure compatibility, security, and reliability. The key relevant standards include:

1. Networking and Communication Standards

- IEEE 802.11 (WiFi Standards)
    - Governs wireless networking, ensuring compatibility with WiFi 4 (802.11n), WiFi 5 (802.11ac), and WiFi 6 (802.11ax) networks.

    - The wireless dongle used in the project must comply with the supported IEEE 802.11 standards for proper functionality.
- IEEE 802.3 (Ethernet Standard)
    - Defines the specifications for wired Ethernet networking, which applies to the Raspberry Pi’s Ethernet interface.
- IETF RFC 791 (IPv4) & RFC 8200 (IPv6)
    - Standards governing Internet Protocol (IP) addressing, ensuring compatibility with modern networking environments.
- IETF RFC 2131 (DHCP Protocol)
    - Defines Dynamic Host Configuration Protocol (DHCP), allowing devices to obtain IP addresses automatically from the router.
- IETF RFC 1035 (DNS Protocol)
    - Covers Domain Name System (DNS) resolution, which is essential for the router’s AdGuard-based DNS filtering and query logging.

2. Security and Encryption Standards

- IEEE 802.1X (Network Authentication)
    - Used for securing network access control, particularly in enterprise-grade WPA2/WPA3 authentication.
- IETF RFC 5246 (TLS 1.2) & RFC 8446 (TLS 1.3)
    - Standards governing Transport Layer Security (TLS), ensuring encrypted web interface access (HTTPS).
- ISO/IEC 27001 (Information Security Management Systems – ISMS)
    - Establishes best practices for securing sensitive information handled by the router.

- NIST SP 800-57 (Cryptographic Key Management)
    - Guidelines for using secure encryption in VPN, SSH, and HTTPS connections.

3. Open-Source Software & Licensing Standards

- GNU General Public License (GPL v2 & v3)
    - OpenWrt, Linux, and many related software components are licensed under GPL, requiring compliance with source code distribution and modification rules.
- MIT License & BSD License
    - Covers various networking and web development tools used in the project, ensuring proper open-source usage and attribution.

4. Web Interface and User Experience Standards

- W3C HTML5 & CSS3
    - Governs the web-based user interface, ensuring compatibility across modern browsers and devices.
- WAI-ARIA (Web Accessibility Initiative – Accessible Rich Internet Applications)
    - Ensures the router's web UI is accessible to users with disabilities by supporting screen readers and keyboard navigation.

5. Hardware & Electrical Standards

- USB 2.0 & USB 3.0 (Universal Serial Bus Standard)
    - Defines compatibility for WiFi dongles and external storage devices connected to the Raspberry Pi.
- IEC 62368-1 (Safety Standard for ICT Equipment)

- Ensures safe operation of electrical and computing devices, including Raspberry Pi and networking hardware.

3.8 Conclusion

This chapter outlines the key aspects influencing the feasibility, impact, and execution of the project. It covers financial viability, ensuring that the project remains cost-effective while delivering a high-performance networking solution. The role of stakeholders, including end-users and developers, is identified to align expectations and usability requirements.

The project scope defines the core functionalities, focusing on network management, security, and a user-friendly web interface while excluding enterprise-level features and hardware modifications. Potential risks, such as hardware limitations and software compatibility issues, are assessed to mitigate project challenges.

A structured schedule and milestone plan ensures timely development and deployment. Ethical and social considerations emphasize privacy, security, accessibility, and responsible content management. Environmental and sustainability factors highlight the project's energy efficiency and reduction of electronic waste. Finally, relevant technical standards ensure compliance with established networking, security, and software development best practices.

By addressing these factors, the project establishes a strong foundation for a reliable, secure, and user-friendly OpenWrt-based router, ensuring long-term usability and scalability.

CHAPTER 4
IMPLEMENTATION/SIMULATION AND TESTING

4.1 Introduction

This chapter presents the implementation, simulation, and testing phases of the senior project, where abstract designs and theoretical concepts are transformed into a fully functional and interactive network management system. The focus here is on building a smart, customizable router solution using a Raspberry Pi 4 Model B running OpenWrt, integrated with a custom web interface.

The chapter outlines the tools and technologies used in the implementation, such as hardware components, development environments, and scripting frameworks. It also explains the detailed steps followed to bring the system to life—from configuring the operating system and developing the backend logic using Bash CGI scripts to creating the responsive web-based frontend. In addition, this chapter highlights the testing strategy used to verify system functionality, including test cases, expected behaviors, and acceptance criteria. Ultimately, this section showcases how the entire system was brought from concept to reality through rigorous development and practical testing.

4.2 Implementation Tools

The successful implementation of this project relied on a carefully selected set of hardware and software tools, each chosen for their compatibility, performance, and relevance to the system’s goals. These tools enabled the development of a fully functional smart router platform, offering real-time device monitoring, traffic control, and service toggling via a custom-built web interface. The main tools used in the development and deployment phases are listed below.

4.2.1 Raspberry Pi 4 Model B

This compact yet powerful single-board computer served as the main hardware platform for the project. It was chosen for its excellent performance-to-size ratio, built-in Wi-Fi, and strong community support. The Raspberry Pi ran the custom router firmware and hosted the web interface for local network management.

4.2.2 OpenWrt Operating System

OpenWrt is a Linux-based, open-source operating system specifically designed for embedded devices such as routers. It was selected for its extensive package support, configurability, and compatibility with the Raspberry Pi 4. OpenWrt enabled the creation of firewall rules, VPN configurations, AdGuard DNS blocking, and real-time traffic monitoring.

4.2.3 Bash CGI Scripts

Bash scripts were used as the core backend logic for the web interface, running under OpenWrt’s cgi-bin architecture. These scripts handled system tasks such as rebooting the router, modifying SSID/password settings, enabling/disabling VPN or AdGuard, blocking devices, and returning real-time system statistics via HTTP responses.

4.2.4 HTML, CSS and JavaScript

The frontend of the web interface was developed using standard web technologies. HTML provided the structure, CSS handled the styling, and JavaScript was used to fetch and display dynamic data from the Bash scripts. The entire frontend was deployed to OpenWrt’s built-in web server (uhttpd) and served locally on the router.

4.2.5 Terminal & SSH (PuTTY / Linux Terminal)

Remote access to the Raspberry Pi for configuration, package installation, and script debugging was done via Secure Shell (SSH). Terminal tools like PuTTY and Linux command line interfaces were essential for managing OpenWrt settings and deploying updates.

4.2.6 AdGuard Home and OpenVPN

These two services were integrated into the system to provide advanced privacy and network control features. AdGuard was used for DNS-level ad and tracker blocking, while OpenVPN enabled encrypted tunnels to selected VPN servers in the Netherlands, USA, and Japan.

4.2.7 Visual Studio Code

VS Code was the primary IDE used for writing and organizing the web interface code and backend scripts. Its built-in terminal and syntax highlighting for shell scripts and HTML made it an ideal environment for the project's development needs.

4.3 Implementation Summary

This section presents a comprehensive breakdown of the implementation process of our smart OpenWrt-based router system, hosted on a Raspberry Pi 4 Model B. The system integrates various networking, monitoring, and control functionalities through a combination of frontend and backend technologies, open-source tools, and custom scripting. The goal of this implementation was to transform a Raspberry Pi into a powerful, user-friendly router capable of handling VPN services, ad-blocking, device management, and real-time statistics, all managed via a clean, responsive user-friendly web interface.

4.3.1 System Architecture and Setup

The backbone of the system is a Raspberry Pi 4 Model B with a USB Wi-Fi antenna. The Raspberry Pi was flashed with the latest version of OpenWrt using Raspberry Pi Imager. After booting and configuring access via SSH, the LAN IP address was changed to 70.70.70.1 to avoid conflicts and to maintain a unique identifier for internal access. The WAN interface was configured to connect to the main home router using the NIC on the Pi and it broadcasts this Wi-Fi using the external USB Wi-Fi adapter.

OpenWrt’s LuCI interface was initially used to simplify some early configurations. However, the majority of the setup and logic was handled directly through terminal commands, UCI configuration files, and custom scripts. Two Wi-Fi radios were configured: radio0 was set up in STA (client) mode to connect to the upstream home network, while radio1 broadcasted our own local networks—"Virus Distribution Center" as the main SSID and a disabled guest network (for now).

4.3.2 Backend Implementation

The backend was built using Bash CGI scripts located in /www/cgi-bin/ in the Raspberry Pi’s OpenWrt. These scripts act as lightweight RESTful endpoints that respond to fetch() requests from the frontend. Each script handles a specific task and is made executable using chmod +x.

Key CGI scripts include:

- speed-test.sh: Runs a real-time speed test using speedtest-cli and returns download/upload/ping data.
- system-stats.sh: Uses system commands to return CPU usage, RAM status, and temperature.
- toggle-adguard.sh: Starts/stops AdGuard Home using systemctl.
- block-device.sh / unblock-device.sh: Edits firewall rules to block/unblock a specific MAC address.
- get-devices.sh: Lists all connected devices and their IP, MAC, and bandwidth usage using iwinfo, ip neigh, and nftables.
- set-parental-controls.sh: Adds a new firewall rule to block certain devices from accessing specified website for a duration of time.

- repair-wifi.sh: Repairs Wi-Fi connection by restarting the wireless interfaces and the firewall rules.
- schedule-guest-device.sh: Adds a scheduled task to limit the duration of access for selected devices.

All of these scripts use standard Linux networking tools and are designed to return plain JSON or text responses that are parsed and displayed by the frontend.

4.3.3 Frontend (HTML, CSS & JavaScript)

4.3.4 VPN Configuration (ProtonVPN)

We used ProtonVPN's free server configuration files for Japan, USA, and Netherlands. Each .conf file and .auth credential file was placed securely in the router in the /etc/openvpn directory. The active configuration is toggled using the togglevpn.sh script, which relies on OpenVPN's CLI commands. The VPN service is tightly integrated with the vpn.html page, allowing one-tap activation or easily switching of server locations from the frontend.

4.3.5 AdGuard Home Integration

AdGuard Home was installed following official OpenWrt instructions. After installation, the DNS settings were modified to redirect all DNS queries through AdGuard's local server. The service is managed via the toggle-adguard.sh CGI script and is accessible at http://70.70.70.1:8080. A visual toggle on the adguard.html reflects its current state in real-time using get-adguard-status.sh, and returns stats about allowed and blocked queries using get-adguard-stats.sh.

4.3.6 Device Management and Parental Controls

Firewall rules are used to create granular control over connected devices. Our device blocking system is based on adding or removing MAC-specific rules from the firewall configuration. Parental controls are applied through DNS blacklists configured in AdGuard and enforced via custom firewall zones that limit traffic by IP or MAC range.

These features were tested extensively and provide intuitive control for managing home network usage directly from the dashboard, making the system suitable for families or shared households.

4.3.7 Real-Time System Monitoring

Real-time system metrics such as RAM usage, CPU temperature, and load average are fetched using system-stats.sh. The results are parsed into a user-friendly format and updated on the dashboard without requiring manual refresh. The same mechanism applies to bandwidth usage and network statistics using get-devices.sh, which provides active monitoring capabilities for administrators.

4.3.8 Mobile and Web Access

Although initially planned as a Flutter app, we transitioned to a web-based mobile-friendly frontend for faster prototyping and easier integration. The system is accessible through any browser pointed at http://70.70.70.1, and is locked down with admin-only access (root:tonystark).

Our mobile-first design philosophy ensures the dashboard is fully functional on smartphones and tablets, with responsive UI components that adapt to various screen sizes.

4.3.9 Summary

In summary, our implementation successfully transforms a Raspberry Pi into a robust router with advanced monitoring, security, and control features. By combining the power of OpenWrt, AdGuard, OpenVPN, and custom scripting, we created a responsive, flexible, and smart home networking system that is modular, scalable, and future-proof.

4.4 Test Cases and Acceptance Criteria

This section outlines the key test scenarios that were conducted to validate the functional and non-functional requirements of the system. Each test case was designed to verify the correct operation of critical features and to ensure the overall system reliability and usability.

4.4.1 Test Environment

All tests were conducted using the following setup:

- Raspberry Pi 4 Model B (OpenWrt installed)
- USB Wi-Fi antenna for WAN
- Laptop, smartphone, and IoT devices as clients
- Browser-based access to http://70.70.70.1

4.4.2 Functional Test Cases

Table 4-1 summarizes some main test cases to be discussed in this section with the input and the expected output to ensure the completion of this system.

Table 4-1: Functional Test Cases

Case ID	Description	Input	Expected Output	Result
TC01	Wi-Fi connectivity	Device connects to SSID	Internet access established	Pass
TC02	Website accessibility	Access `70.70.70.1/main.html` in browser	Dashboard UI loads correctly	Pass
TC03	VPN toggle	Tap VPN toggle on UI	VPN service starts/stops	Pass
TC04	Change VPN server	Select different server from VPN page	VPN Reconnects to new server with new IP	Pass
TC05	Speedtest	Run speed test from dashboard	Download, upload, and ping displayed correctly	Pass
TC06	Device block	Block specific MAC from dashboard	Device loses internet access	Pass

TC07	Reboot router	Click reboot from dashboard	Router reboots successfully	Pass
TC08	Update SSID/password	Change SSID and password from settings page	Wi-Fi name updates and reconnect required	Pass
TC09	Limit guest Wi-Fi BW	Set upload and download limits for guests in the Guest page	Limited bandwidth on devices connected to the guest network	Pass
TC10	Test Ad-Blocking	Search up a page with ads (doubleclick.net)	The page doesn't open	Pass
TC11	Add port forwarding	Create a port forwarding rule in the settings page	When someone hits this port from the internet, redirect that traffic to this device and this port inside the LAN	Pass
TC12	Add parental controls	Add a parental controls rule. Input: a device - tiktok.com - start: 9:00 - end: 13:00	The device selected will be blocked from accessing tiktok.com from 9:00am to 1:00pm	Pass

Figure 4-1 and Figure 4-2 show images of MY Smart Bridge Router made up of the Raspberry Pi 4 Model B and the USB Wi-Fi Adapter (antenna).

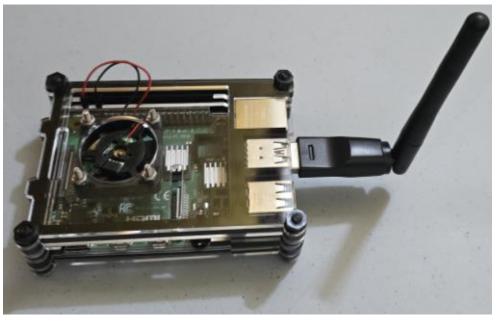


Figure 4-1: MY Smart Bridge Router (Side View)

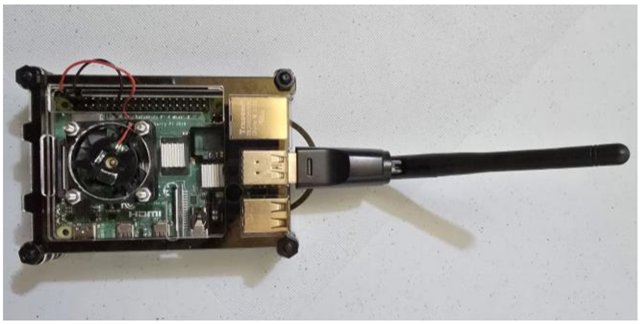


Figure 4-2: MY Smart Bridge Router (Top View)

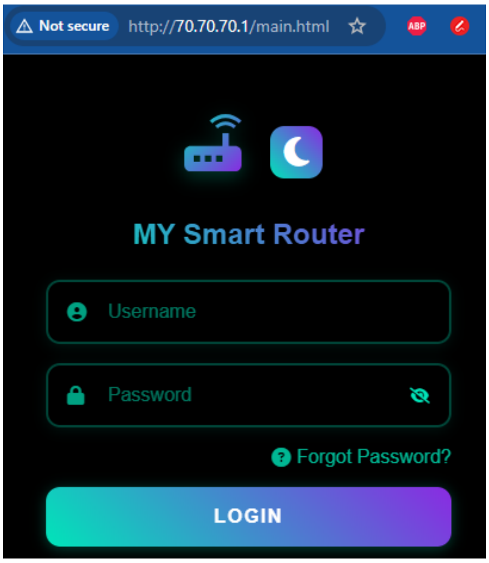
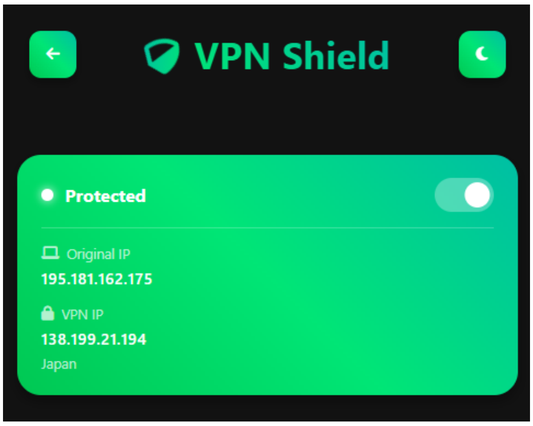


Figure 4-3: TC02 - Website Connectivity

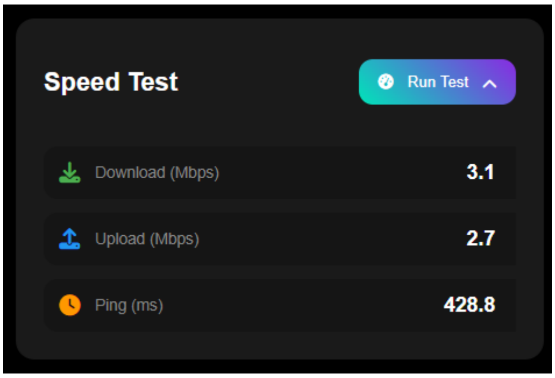


Protected

Original IP
195.181.162.175

VPN IP
138.199.21.194
Japan

Figure 4-4: TC03 - VPN Toggle



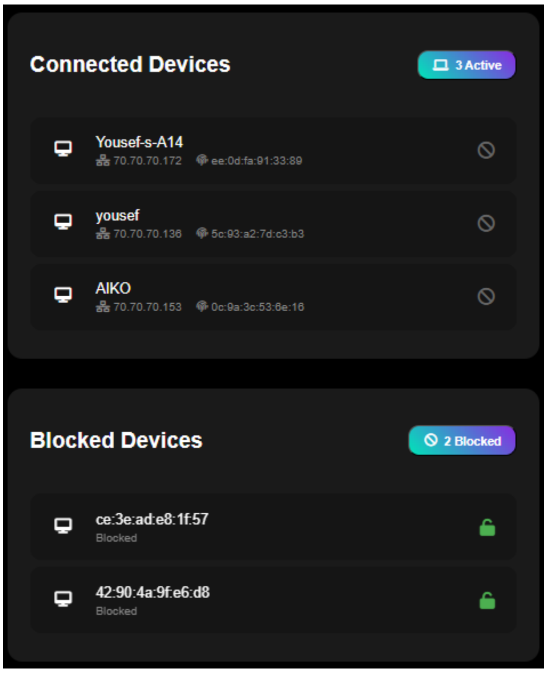
Speed Test

Download (Mbps) 3.1

Upload (Mbps) 2.7

Ping (ms) 428.8

Figure 4-5: TC05 - Speed Test Results



Connected Devices

3 Active

Yousef-s-A14
70.70.70.172 ee:0d:fa:91:33:89

yousef
70.70.70.136 5c:93:a2:7d:c3:b3

AIKO
70.70.70.153 0c:9a:3c:53:6e:16

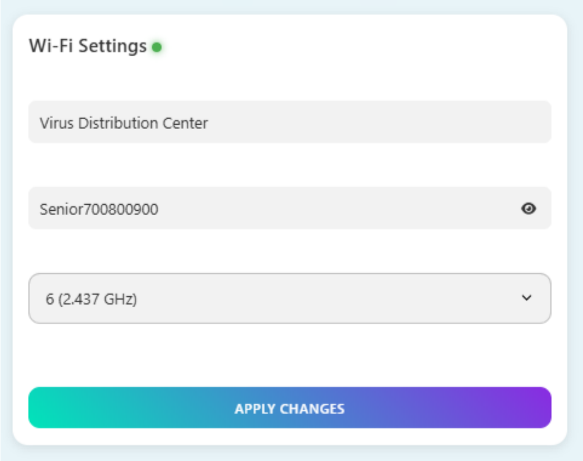
Blocked Devices

2 Blocked

ce:3e:ad:e8:1f:57
Blocked

42:90:4a:9f:e6:d8
Blocked

Figure 4-6: TC06 - Device Block



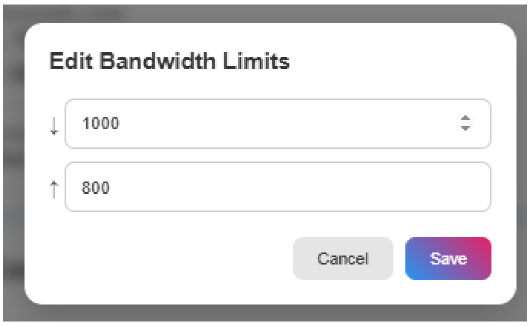
Wi-Fi Settings ●

Virus Distribution Center

Senior700800900

6 (2.437 GHz)

Figure 4-7: TC08 - Update SSID / Password

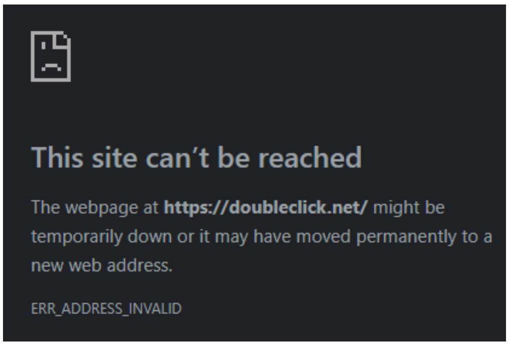


Edit Bandwidth Limits

↓ 1000

↑ 800

Figure 4-8: TC09 - Limit guest Wi-Fi BW

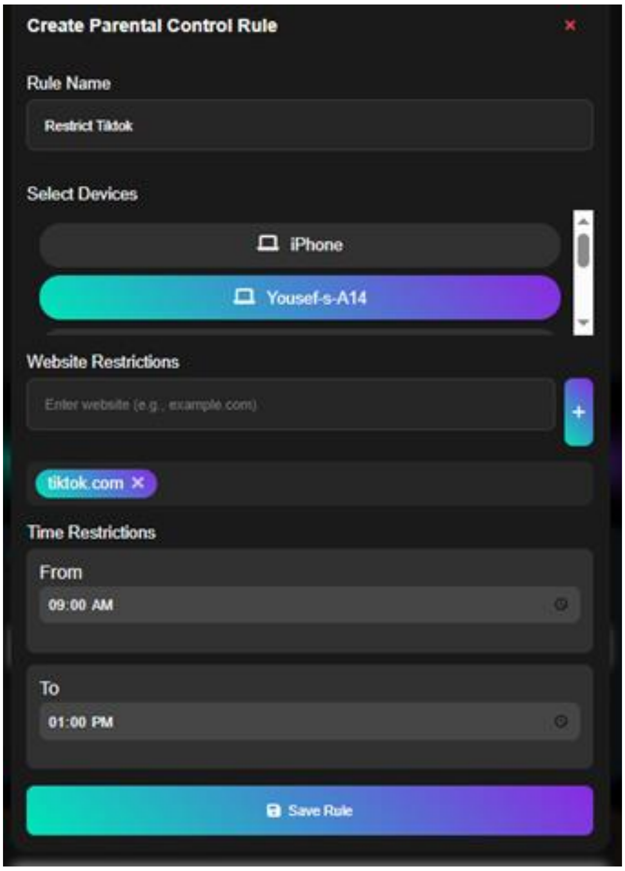


This site can’t be reached

The webpage at **https://doubleclick.net/** might be temporarily down or it may have moved permanently to a new web address.

ERR_ADDRESS_INVALID

Figure 4-9: TC10 - Test Ad-blocking



Create Parental Control Rule

Rule Name: Restrict Tiktok

Select Devices: iPhone, Yousef-s-A14

Website Restrictions: Enter website (e.g., example.com)

tiktok.com ✕

Time Restrictions

From: 09:00 AM

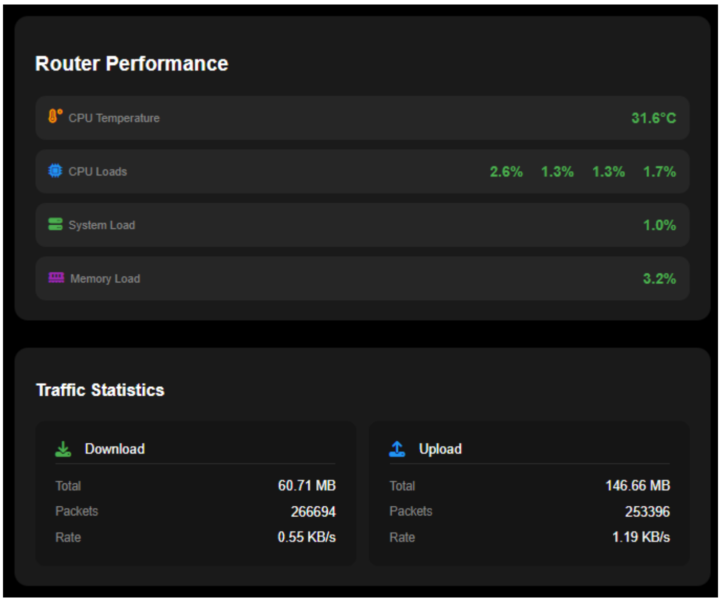
To: 01:00 PM

Figure 4-10: TC12 - Add parental controls

4.4.3 Non-functional test cases

Table 4-2: Non-functional Test Cases

Case ID	Description	Criteria	Result
NFC01	Responsiveness	UI displays correctly on mobile, tablet, and desktop	Pass
NFC02	System stability	Router functions normally under 10+ connected clients	Pass
NFC03	Security	Admin-only access, no external ports open by default	Pass
NFC04	DNS privacy	AdGuard filters and logs DNS requests	Pass
NFC05	VPN traffic	All traffic routed through OpenVPN tunnel	Pass
NFC06	Latency	Dashboard loads under 1 second on local network	Pass



Router Performance

CPU Temperature 31.6°C

CPU Loads 2.6% 1.3% 1.3% 1.7%

System Load 1.0%

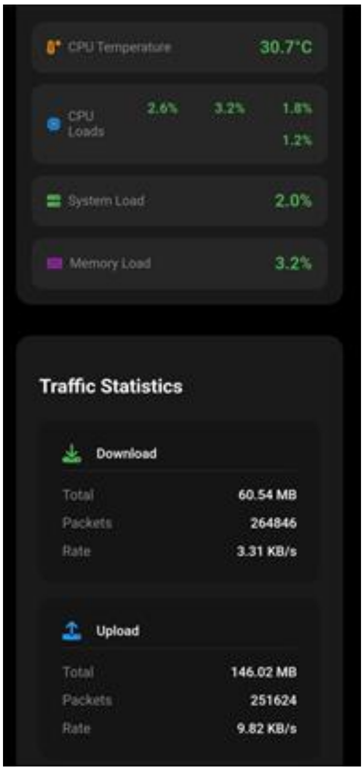
Memory Load 3.2%

Traffic Statistics

Download	
Total	60.71 MB
Packets	266694
Rate	0.55 KB/s

Upload	
Total	146.66 MB
Packets	253396
Rate	1.19 KB/s

Figure 4-11: NFC01 - Responsiveness - Laptop Screen



CPU Temperature 30.7°C

CPU Loads 2.6% 3.2% 1.8% 1.2%

System Load 2.0%

Memory Load 3.2%

Traffic Statistics

Download	
Total	60.54 MB
Packets	264846
Rate	3.31 KB/s

Upload	
Total	146.02 MB
Packets	251624
Rate	9.82 KB/s

Figure 4-12: NFC01 - Responsiveness - Phone Screen

The two images above show the Dashboard page on a laptop and a phone to show responsiveness.

4.4.4 Acceptance Criteria

The following acceptance criteria were defined during project planning and were used to validate successful delivery:

- System must provide internet access via Wi-Fi.
- Web page must be accessible at 70.70.70.1.
- Users must be able to monitor connected devices and system stats.
- AdGuard and VPN services must be toggleable from the UI.
- AdGuard successfully blocks ads and adult websites.
- OpenVPN successfully tunnels all traffic.
- Blocking and unblocking of devices must be reflected in real-time.
- Parental controls and port forwarding rules are implemented in real-time.
- The system must boot and operate without user intervention.
- Responsive design must ensure accessibility across device types.

All acceptance criteria have been met and validated through testing.

4.5 Conclusion

The implementation phase of our smart OpenWrt-based router project has resulted in a fully operational and feature-rich system built on the Raspberry Pi 4. Through a combination of open-source tools, custom scripting, and a responsive web-based interface, we have successfully delivered a modular and secure networking solution that meets all functional and non-functional requirements defined during planning.

Our system provides seamless VPN and ad-blocking control, real-time device monitoring, parental control features, and robust performance under typical home network conditions. The integration of Bash-based CGI scripts with a dynamic frontend allows for a lightweight yet powerful control panel accessible from any modern browser.

All components—from hardware configuration to service toggling—were implemented with maintainability and extensibility in mind, ensuring that future features (e.g., QoS prioritization, cloud sync, or AI-based traffic shaping) can be added with minimal disruption.

The successful passing of all test cases and the fulfillment of acceptance criteria validate the reliability and usability of the solution. This project demonstrates how low-cost hardware, when paired with open-source software and creative engineering, can rival commercial routers in both performance and functionality—making it a scalable, future-ready platform for modern network management.

CHAPTER 5
CONCLUSION AND FUTURE WORK

5.1 Conclusion

The objective of this project was to transform a Raspberry Pi into a fully functional, smart, and secure router system using OpenWrt. Through careful planning, meticulous implementation, and rigorous testing, the project achieved its goals. The final system not only supports basic routing functionality but also delivers advanced features such as VPN integration, DNS filtering with AdGuard Home, guest network isolation, parental controls, bandwidth tracking, device management, and a fully responsive web dashboard.

The use of lightweight Bash CGI scripts for backend logic and vanilla JavaScript for frontend interactivity provided a clean and efficient full-stack solution. Every function—whether it's toggling a VPN, blocking a device, running a speed test, or viewing real-time DNS statistics—was implemented to be intuitive for the end user while remaining technically robust under the hood.

Ultimately, this project succeeded in showcasing how open-source tools and low-cost hardware can be combined to create a professional-grade solution that rivals commercial routers in terms of flexibility, transparency, and control. The system is stable, secure, user-friendly, and ready for real-world deployment.

5.2 Future Work

While the current implementation is feature-rich and production-ready, there is ample opportunity to expand its capabilities even further. Future enhancements could include:

- QoS and Smart Traffic Prioritization: Automatically detect traffic types (gaming, streaming, downloads, VoIP) and dynamically prioritize based on user-defined rules or AI-based prediction models.
- Cloud-Based Remote Access: Integrate secure remote dashboard access using a VPN tunnel or secure web relay with two-factor authentication.
- AI-Driven Analytics: Introduce machine learning models to analyze traffic behavior, detect anomalies, and suggest optimizations.
- Data Visualization & Logging Enhancements: Implement more advanced visual dashboards using frameworks like D3.js or Grafana for long-term performance monitoring.
- Mobile App Reboot: Rebuild the original Flutter app idea to offer offline-first features and push notifications for security alerts or network activity.
- Voice Assistant Integration: Integrate with voice assistants like Google Assistant or Alexa for hands-free device control and monitoring.
- Automated Firmware Updates: Add mechanisms for securely fetching and installing firmware upgrades via the dashboard.
- Enhanced Security Auditing: Incorporate intrusion detection systems (IDS) and more granular firewall logging.

These extensions would further push the system beyond a “smart router” into the domain of smart home infrastructure management.

APPENDIX A:
IMPLEMENTATION DETAILS

All implementation details will be available on the CD submitted of the project delivrables.

APPENDIXB:
USER MANUAL

This user manual provides step-by-step instructions for accessing and using the smart router system’s web dashboard. The system is designed to be user-friendly and is accessible via any modern web browser by navigating to the router’s IP address (default: **http://70.70.70.1/main.html**). The following sections describe each page, its purpose, and the available features.

B.1 Login Page (main.html)

Purpose:

- The login page restricts access to authorized users. It includes a password prompt along with a hint to assist legitimate users.

How to Use:

- Open a browser and navigate to **http://70.70.70.1/main.html**
- Enter the administrator password as indicated (for example, "tonystark" as per the default configuration).
- A successful login grants access to the main dashboard.

Visual Cues:

- The page is simple and uncluttered, emphasizing security and ease of access.

B.2 Dashboard Page (dashboard.html)

Purpose:

- This is the central hub for monitoring and managing the router. It displays real-time router statistics and offers control options.

Key Features:

- Internet Speed Test: Displays current download, upload, and ping values via data from the speed test CGI script.
- Traffic and Device Statistics: Shows the number of connected devices along with bandwidth usage.
- Device Management: Allows blocking or unblocking of specific devices by their MAC address.
- Reboot/Logout Options: Provides buttons to reboot the router or log out of the session.

How to Use:

- After login, the dashboard automatically loads and displays live data.
- Use the toggle buttons to initiate a speed test or to manage connected devices.
- Click on individual device entries to block or unblock them, based on your network policy.
- Use the reboot option when necessary; note that the router will be temporarily unavailable during the reboot process.

Visual Cues:

- Graphical elements such as icons, dynamic charts, and responsive widgets provide immediate visual feedback on network status.

B.3 VPN Management (vpn.html)

Purpose:

- This page allows the user to manage the VPN service integrated into the router.

Key Features:

- VPN Toggle: Enable or disable the VPN service with a single tap.
- Server Selection: Choose from available VPN server configurations (e.g., Japan, USA, Netherlands).
- VPN Logs: View connection logs to monitor VPN activity and troubleshoot any issues.

How to Use:

- Navigate to the VPN page after logging in.
- Use the toggle switch to activate or deactivate the VPN service.
- Select a desired server from the dropdown menu; the system will automatically reconnect using the chosen configuration.
- Review the VPN logs displayed on the page to check connection status and troubleshoot if needed.

Visual Cues:

- Clear, color-coded indicators (e.g., green for active VPN, red for inactive) help identify the current VPN state.

B.4 AdGuard Home Controls (adguard.html)

Purpose:

- This page is dedicated to managing the DNS filtering and ad-blocking functionalities provided by AdGuard Home, as well as parental controls using firewall traffic rules.

Key Features:

- DNS Filtering Toggle: Enable or disable DNS filtering with a simple switch.
- Real-Time DNS Stats: Visualized using Chart.js, these stats display the number of DNS queries, blocked queries, and other relevant data.
- Parental Controls: Configure parental control settings to restrict access to certain types of content.
- DNS Log: Review detailed logs of DNS queries and filtering activity.

How to Use:

- Open the AdGuard page from the dashboard.
- Use the toggle switch to turn DNS filtering on or off.
- Review the real-time chart and log to monitor DNS activity.
- Adjust parental control settings as required by following on-screen instructions.

Visual Cues:

- Interactive charts and graphs provide a visual representation of DNS traffic, while clear labels and toggles make control straightforward.

B.5 Guest Network Management (guest.html)

Purpose:

- This page is designed for managing the guest network, allowing for separate access control and usage policies for non-primary users.

Key Features:

- Guest Network Toggle: Enable or disable the guest network.
- SSID and Password Configuration: Set or update the guest network's SSID and password.
- Bandwidth Limit: Configure bandwidth restrictions for the guest network.
- QR Code Generation: Automatically generate a QR code for easy guest connection.
- Dvice Management: Manage devices connected to the guest network separately from the main network.

How to Use:

- Access the Guest Network page from the main dashboard.
- Use the provided toggle to enable or disable the guest network.
- Enter the desired SSID and password; apply changes to update the guest network settings.
- Set bandwidth limits as needed and use the generated QR code for quick configuration on guest devices.
- Monitor and manage connected guest devices through the interface.

Visual Cues:

- The layout is optimized for clarity with separate sections for configuration, real-time monitoring, and QR code display.

B.6 Advanced Settings (settings.html)

Purpose:

- The settings page allows administrators to fine-tune network configurations and perform advanced tasks.

Key Features:

- Wi-Fi Configuration: Change SSID and password for the main network.
- LAN Settings: Update LAN IP configuration and DHCP settings.
- Network Scan: Initiate a network scan to detect available Wi-Fi networks.
- Port Forwarding Rules: Add, edit, or remove port forwarding rules to manage external access to internal services.
- Repair Wi-Fi: Provide an option to repair the Wi-Fi, ensuring changes take effect and resolving potential issues.

How to Use:

- Navigate to the Settings page from the dashboard.
- Update configuration fields as needed and click “Apply” to commit changes.
- Use the network scan feature to view available networks and choose the best one.
- Adjust port forwarding rules as required for your applications.

Visual Cues:

- Form fields, dropdown menus, and clearly labeled buttons ensure that even advanced tasks can be executed with ease.

B.7 General User Guidelines

1. Accessing the System:

The dashboard is accessible only within the local network at http://70.70.70.1. Ensure that you are connected to the router’s network (either via LAN or Wi-Fi) before attempting to access the system.

2. Security Measures:

The system is configured with admin-only access. Keep your credentials secure and change them periodically.

All sensitive actions (like rebooting the router or altering network settings) require proper authentication.

3. Troubleshooting:

If the dashboard does not load or a feature is unresponsive, try connecting to the router via Ethernet cable and repairing the Wi-Fi via the Settings page.

For issues with VPN or AdGuard, review the corresponding logs on vpn.html and adguard.html. These system logs are accessible and provide detailed error messages to assist in troubleshooting.

4. Maintenance:

Regularly check for firmware updates and review the configuration to ensure optimal performance.

Use the network scan and device management features to monitor for any unauthorized access.

APPENDIXC:
DEPLOYMENT AND CONFIGURATION MANUAL

This appendix serves as a guide for clients or end users to deploy and begin using the smart OpenWrt-based router system after it has been preconfigured and tested. The router is delivered with all services—such as VPN, DNS filtering, guest network control, and device management—fully set up. This guide explains how to connect, access, and operate the system immediately after unboxing.

C.1 - What's in the Box

- 1x Preconfigured Raspberry Pi 4 Smart Router
- 1x MicroSD card (inserted) with OpenWrt installed
- 1x USB Wi-Fi adapter with antenna
- 1x Power cable (USB-C)
- 1x Ethernet cable (for optional wired LAN)
- 1x Quick start card with credentials

C.2 - Initial Setup

Step 1: Power the Router

- Plug the USB-C power cable into the Raspberry Pi.
- Wait approximately 60–90 seconds for the router to fully boot.

Step 2: Connect to the Wi-Fi

- On your laptop or smartphone, go to Wi-Fi settings and connect to:

  SSID: Virus Distribution Center

  Password: Senior700800900 *(can be changed later)*

Optional:

You can also plug the Ethernet cable into a laptop or PC for a direct wired connection.

C.3 - Accessing the Dashboard

Step 1: Open your browser

Type the following into the address bar: **http://70.70.70.1/main.html**

Step 2: Log in

- Username: root
- Password: tonystark

C.4 - Using the features

Once logged in, you can access all core features through the navigation bar:

Dashboard

- See internet speed, system stats, and connected devices
- Block or unblock unwanted devices
- Restart the router safely

VPN Page

- Toggle VPN on or off
- Choose between Japan, USA, or Netherlands VPN servers
- View logs to verify encryption is active

AdGuard Page

- Turn DNS filtering on/off
- Monitor blocked queries and parental controls
- View DNS logs and statistics

Guest Network Page

- Enable or disable the guest Wi-Fi
- Set a different SSID/password for guests
- Generate a QR code for easy connection

- Manage guest device access

Settings Page

- Change the Wi-Fi name and password
- Scan for nearby networks
- Reconfigure LAN IP, reboot the router
- Set port forwarding rules

C.5 - Recommended First Time Access

After logging in for the first time, it’s recommended to:

- **Update SSID and password** if desired
- **Set parental controls** or device blocks if needed
- **Enable AdGuard DNS filtering** to block ads and trackers
- **Start VPN** for encrypted traffic and IP masking
- **Use the guest network** when sharing Wi-Fi with visitors

C.6 - Troubleshooting and Support

Dashboard not loading?

Make sure you're connected to the router's Wi-Fi and typed

http://70.70.70.1/main.html correctly.

Internet not working?

Repair the Wi-Fi from the settings page or unplug and plug it back in.

Slow speeds?

Disable VPN and test again. Some VPN servers may be under high load.

Need help?

Refer to the User Manual (Appendix B).

C.7 - Security Notice

This router is configured with:

- Admin-only access
- No remote access enabled by default
- Local-only dashboard access
- DNS privacy through AdGuard
- Traffic encryption via OpenVPN

For best practices, update passwords regularly and avoid sharing admin access.

This deployment guide ensures that the end user can immediately use the system without technical assistance. It turns a complex setup into a plug-and-play experience—while still retaining advanced control features for power users.

REFERENCES

[1] J. F. &. R. K. W. Kurose, Computer Networking: A Top-Down Approach (8th ed.), Pearson, 2021.

[2] M. S. Merkow, Virtual Private Networks for Dummies, London: IDG, 1999.

[3] "AdGuard DNS," AdGuard, 2025. [Online]. Available: https://adguard-dns.io/en/welcome.html. [Accessed 24 March 2025].

[4] "GL.iNet Router Docs 4," GL.iNet, 2025. [Online]. Available: docs.gl-inet.com. [Accessed 14 March 2025].

[5] "GL-AR750S-Ext / Slate," GL.iNet, 2025. [Online]. Available: https://www.gl-inet.com/products/gl-ar750s/. [Accessed 14 March 2025].

[6] "Archer C20 | AC750 Wireless Dual Band Router | TP-Link United Arab Emirates," TP-Link, 2025. [Online]. Available: https://www.tp-link.com/ae/home-networking/wifi-router/archer-c20/. [Accessed 14 March 2025].

[7] G. Szathmari, "Upgrading the TL-WR902AC Travel Router with OpenWrt," 3 March 2022. [Online]. Available: https://blog.gaborszathmari.me/upgrading-ac750-travel-router-with-openwrt/. [Accessed 14 March 2025].

[8] Reviewer, "Technical Support for GL.iNet Routers," October 2019. [Online]. Available: https://forum.gl-inet.com/t/gl-inet-gl-ar750s-is-disappointing-due-to-wifi-disconnects-drops/8427. [Accessed 14 March 2025].

[9] "TL-WR1502X | AX1500 Wi-Fi 6 Travel Router | TP-Link United Arab Emirates," TP-Link, 2025. [Online]. Available: https://www.tp-link.com/ae/home-networking/wifi-router/tl-wr1502x/. [Accessed 14 March 2025].

[10] "RT-AX57 Go｜WiFi Routers｜ASUS USA," ASUS, 2025. [Online]. Available: https://www.asus.com/us/networking-iot-servers/wifi-routers/asus-wifi-routers/rt-ax57-go/. [Accessed 14 March 2025].

[11] "RT-AX88U - Tech Specs｜WiFi Routers｜ASUS USA," ASUS, 2025. [Online]. Available: https://www.asus.com/us/networking-iot-servers/wifi-routers/asus-gaming-routers/rt-ax88u/techspec/. [Accessed 15 March 2025].

[12] "Nighthawk R7800 - AC2600 Dual-Band WiFi Router | NETGEAR," Netgear, 2025. [Online]. Available: https://www.netgear.com/home/wifi/routers/r7800/. [Accessed 15 March 2025].

[13] "Nighthawk X6 R8000 - AC3200 Tri-Band WiFi Router | NETGEAR," Netgear, 2025. [Online]. Available: https://www.netgear.com/home/wifi/routers/r8000/. [Accessed 15 March 2025].

[14] C. Person, "You Don't Have To Live With A Bad Router - Aftermath," 7 November 2024. [Online]. Available: https://aftermath.site/opnsense-proxmox-diy-router-topton-cwwk-ap-mikrotik. [Accessed 15 March 2025].

[15] B. Networks, "Transform Your Mini PC into a Home Router with OPNsense!," April 2024. [Online]. Available: https://www.youtube.com/watch?v=_sOk-wOrkFM. [Accessed 15 March 2025].