Network Design Proposal for Internet Cafe

K. Gnana Shanmukha
AM.EN.U4ECE22125
Department of Electronics and Communication Engineering
Amrita Vishwa Vidyapeetham, Amritapuri, Kerala
Email: am.en.u4ece22125@am.students.amrita.edu

M. L. Ravindra
AM.EN.U4ECE22130
Department of Electronics and Communication Engineering
Amrita Vishwa Vidyapeetham, Amritapuri, Kerala
Email: am.en.u4ece22130@am.students.amrita.edu

P. V. Shiva Sai
AM.EN.U4ECE22134
Department of Electronics and Communication Engineering
Amrita Vishwa Vidyapeetham, Amritapuri, Kerala
Email: am.en.u4ece22134@am.students.amrita.edu

M. Varshith Reddy
AM.EN.U4ECE22155
Department of Electronics and Communication Engineering
Amrita Vishwa Vidyapeetham, Amritapuri, Kerala
Email: am.en.u4ece22155@am.students.amrita.edu

***Abstract*—This network design proposal aims to address the requirements of an Internet cafe that requires a network infrastructure to support 30 users sharing one ADSL internet connection. The proposed solution includes web filtering and billing software to ensure safe browsing and manage customer usage. This paper outlines the requirement analysis, network architecture, implementation, Python-based billing system, and expected results. To ensure reliable and secure connectivity, the proposed network design includes a requirement analysis that takes into account factors such as the cafe's location, the number of users, the required bandwidth, and the need for redundancy and backup. Based on this analysis, a network diagram and IP address design are presented to illustrate the proposed network architecture and address allocation plan.**

I. Introduction

The objective of this project is to propose a network design solution for an Internet cafe with 30 users sharing one ADSL internet connection. The cafe also requires web filtering and billing software for secure and managed access. This paper presents the network's requirement analysis, architecture, and implementation for efficient and secure operations. The proposed network design solution takes into consideration the cafe's connectivity and management requirements, as well as its need for secure and reliable internet access. The network design solution includes a requirement analysis, a network diagram, an IP address design, and a list of recommended hardware and software products.

II. Requirements

- **Connectivity:** Support for 30 users with at least 20 Mbps bandwidth.
- **Web Filtering:** Block access to inappropriate websites.
- **Billing Software:** Track user usage and compute charges.
- **Security:** Ensure protected access and prevent misuse.
- **Redundancy:** Provide backup connectivity.
- **Scalability:** Room for future expansion.
- **Ease of Management:** Simplified maintenance and upgrades.
- **Compatibility:** Support future software and hardware.
- **Cost-Effective:** Meet needs within budget constraints.

III. Design Requirement Analysis

A detailed requirement analysis identifies the necessary components to ensure the internet cafe functions securely and efficiently:

- **Network Topology:** A star topology is implemented due to its ease of troubleshooting, scalability, and centralized control.
- **Core Devices:** The setup includes a high-speed ADSL router, a managed Layer 2 switch, web filtering appliance/software, a billing server, and client PCs.
- **IP Planning:** Logical subnetting segregates LAN (192.168.1.0/24) from the DMZ zone (192.168.2.0/24) to enhance security and simplify traffic control.
- **Firewall and Access Control:** Firewall rules are defined to allow only necessary services, and access control lists (ACLs) ensure policy enforcement.
- **Monitoring and Logging:** Tools like SNMP or third-party software provide real-time network analytics and ensure SLA monitoring.

IV. Architecture and Design

The following diagram illustrates the operational sequence between the user, admin, and server which governs the interaction for internet access, billing approval, and web filtering. The sequence of operations is as follows:

1) **User:** Initiates the sequence by requesting internet access from the server.
2) **Admin:** Receives the billing request from the server and reviews it for accuracy.
3) **Server:** Sends the billing request to the admin for approval.
4) **Admin:** Approves the request if accurate.
5) **Server:** Sends billing details to the user and grants internet access.
6) **User:** Confirms the payment after receiving billing details.
7) **Server:** Grants access upon payment confirmation.

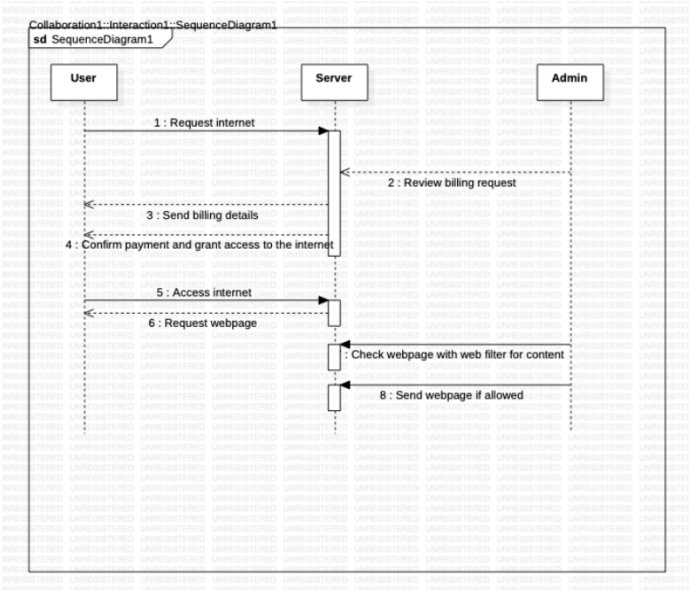

Fig. 1. Sequence Diagram

8) **User:** Sends a webpage request.
9) **Server:** Checks the request against the web filter.
10) **Admin:** Also validates the request against prohibited content.
11) **Server:** Delivers the page to the user if allowed.
12) **User:** Accesses the approved content.

This structured process ensures that Internet access is both secure and monitored, with billing and filtering enforced at each stage.

V. Implementation

The physical implementation of the Internet cafe network is demonstrated in the image below, which includes the actual layout and connections of client systems, router, switch, and server.

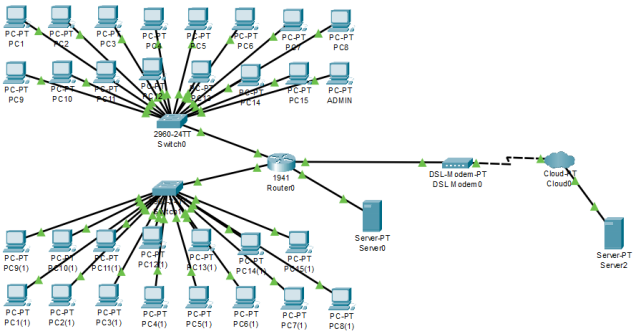

Fig. 2. Network Layout and Device Interconnection

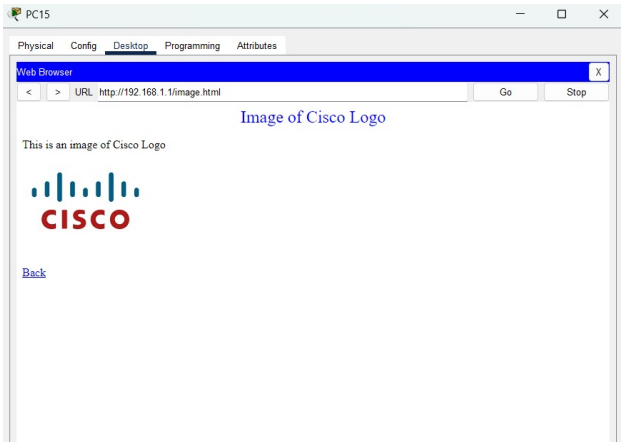

Fig. 3. Runtime Demo

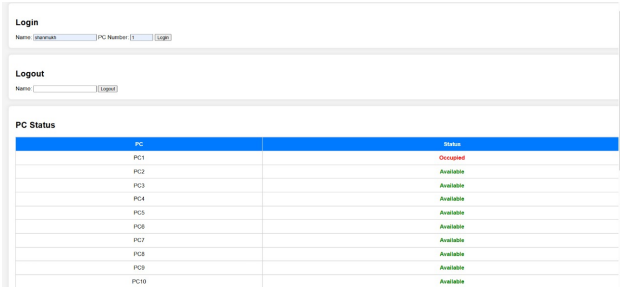

Fig. 4. Login Details

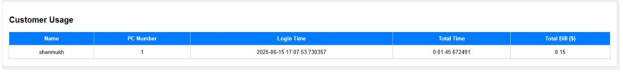

Fig. 5. Billing Details

VI. CODE:USER FRIENDLY INTERFACE

Listing 1. Web-Based Billing Software Using Flask

```
from flask import Flask, render_template, request,
    redirect, url_for
from datetime import datetime, timedelta

app = Flask(__name__)

hourly_rate = 5
customers = {}
pc_status = {f'PC{pc_num}': 'available' for pc_num
    in range(1, 31)}

@app.route('/')
def index():
    return render_template('index.html', customers=
        customers, pc_status=pc_status)

@app.route('/login', methods=['POST'])
def login():
    name = request.form['name']
    pc_num = request.form['pc_num']
    if name in customers:
        return redirect(url_for('index'))
    login_time = datetime.now()
    customers[name] = {
        'pc_num': pc_num,
        'login_time': login_time,
        'total_time': timedelta(0),
        'total_bill': 0
    }
    pc_status[f'PC{pc_num}'] = 'occupied'
    return redirect(url_for('index'))

@app.route('/logout', methods=['POST'])
def logout():
    name = request.form['name']
    if name not in customers:
        return redirect(url_for('index'))
    logout_time = datetime.now()
    login_time = customers[name]['login_time']
    total_time = logout_time - login_time
    total_bill = round(hourly_rate * total_time.
        total_seconds() / 3600, 2)
    customers[name]['total_time'] = total_time
    customers[name]['total_bill'] = total_bill
    pc_num = customers[name]['pc_num']
    pc_status[f'PC{pc_num}'] = 'available'
    return redirect(url_for('index'))

if __name__ == '__main__':
    app.run(host='0.0.0.0', port=5000)
```

VII. RESULTS AND OUTPUTS

Sample output:

- User: Mukesh
- Login Time: 2023-05-01 03:48:13
- Logout Time: 60 Minutes later
- Total Bill: 100 INR

VIII. CONCLUSION

The network design successfully meets the requirements of a 30-user Internet cafe by offering secure, manageable, and scalable internet access. The integration of billing software automates the management of user sessions and cost calculation, while web filtering ensures safe browsing. Cisco Packet Tracer simulation confirmed the feasibility of the proposed architecture. Future improvements may include remote administration, real-time monitoring dashboards, load balancing, and support for wireless clients using dual-band access points. The network design solution also includes a comprehensive requirement analysis, network diagram, IP address design, and recommendations for hardware and software products. These elements work together to provide a scalable, secure, and cost-effective network infrastructure that meets the cafe's current and future needs.

REFERENCES

[1] Geeks for Geeks, "DHCP Server Configuration in Cisco Packet Tracer," Available: https://www.geeksforgeeks.org/dhcp-server-configuration-in-cisco/

[2] Cisco Systems, Inc., "Cisco Packet Tracer," Available: https://www.netacad.com/courses/packet-tracer