Scientific Research Publishing

Journal of Computer and Communications, 2024, 12, 123-140
https://www.scirp.org/journal/jcc
ISSN Online: 2327-5227
ISSN Print: 2327-5219

Quality of Service and Security on Cisco Network Devices, Coupled with the Development of a Mobile Application Prototype Software for Server Room Temperature Monitoring

Desire Mudenda, Charles Smart Lubobya

Department of Electrical and Electronic Engineering, UNZA, Lusaka, Zambia
Email: desiremudenda@gmail.com, charles.lubobya@gmail.com

How to cite this paper: Mudenda, D. and Lubobya, C.S. (2024) Quality of Service and Security on Cisco Network Devices, Coupled with the Development of a Mobile Application Prototype Software for Server Room Temperature Monitoring. *Journal of Computer and Communications*, **12**, 123-140.
https://doi.org/10.4236/jcc.2024.128008

Received: June 7, 2024
Accepted: August 23, 2024
Published: August 26, 2024

Abstract

In an era where digital technology is paramount, higher education institutions like the University of Zambia (UNZA) are employing advanced computer networks to enhance their operational capacity and offer cutting-edge services to their academic fraternity. Spanning across the Great East Road campus, UNZA has established one of the most extensive computer networks in Zambia, serving a burgeoning community of over 20,000 active users through a Metropolitan Area Network (MAN). However, as the digital landscape continues to evolve, it is besieged with burgeoning challenges that threaten the very fabric of network integrity—cyber security threats and the imperatives of maintaining high Quality of Service (QoS). In an effort to mitigate these threats and ensure network efficiency, the development of a mobile application to monitor temperatures in the server room was imperative. According to L. Wei, X. Zeng, and T. Shen, the use of wireless sensory networks to monitor the temperature of train switchgear contact points represents a cost-effective solution. The system is based on wireless communication technology and is detailed in their paper, "A wireless solution for train switchgear contact temperature monitoring and alarming system based on wireless communication technology", published in the *International Journal of Communications, Network and System Sciences*, vol. 8, no. 4, pp. 79-87, 2015 [1]. Therefore, in this study, a mobile application technology was explored for monitoring of temperatures in the server room in order to aid Cisco device performance. Additionally, this paper also explores the hardening of Cisco device security and QoS which are the cornerstones of this study.

Keywords

Quality of Service (QoS), Network Security, Temperature Monitoring, Mobile Application, Cisco Devices

1. Introduction

This expository study aims to dissect the extent to which hardening of Network QoS and Security have permeated the landscape of the University of Zambia. In line with this imperative, UNZA has embarked on a journey of digital reinvention, ensuring that its core infrastructure not only supports the existing requirements but is also scalable to accommodate future expansions. UNZA embarks on improving performance on the array of Cisco core and distribution devices, which serve as the backbone for the university's network architecture. Through these components, a robust and adaptable network is established, capable of handling the intense data traffic and the sophisticated services demanded by the academic consortium.

Acknowledging the Digital Revolution: A Persistent Growth

The digital revolution's impact on education cannot be overstated. A plethora of scholarly sources confirm the profound transformation that has been prompted by the progressive integration of technology within the educational domain [2]. The resurgence and continuous growth of the digital innovations has revolutionized "Teaching and Learning", "Research and Innovation" and collaboration in universities. Higher Education learning institutions are the hubs of cognitive research and explorations which have to be shared on multi-layered computer networks which have become the nerve conduits of communication, collaboration and administrative platforms for academic institutions [3].

2. Literature Review

2.1. Quality of Service and Security

Quality of Service illuminates key network metrics, including but not limited to bandwidth, delay, jitter, and error prevalence, underpinning the foundations of robust and proficient network operations.

Simultaneously, the domain of network security is of paramount priority, comprising stringent protocols to shield data and system integrity against unauthorized intrusions and cyber threats. Techniques like encryption, firewall deployment, and the incorporation of intrusion detection mechanisms collectively fortify the digital fortress that is the network's architecture, thereby securing the critical and confidential data coursing through its channels. In this regard, [4] [5] recommends that it is imperative that techniques are employed to control traffic especially when there is network congestion as this causes slow packet transmission, data loss, or even dropping of traffic [6]. Therefore, utilizing QoS techniques assures management of traffic in an event that the infrastructure reaches limits in how they can process data—especially on networks such as campus networks, the demand for internet applications has increased rapidly, thereby making QoS provisioning more challenging than ever before. Some of the challenges include traffic control, shaping, bandwidth allocation and management and general accessibility of network resources by network users. A major component that is basically challenging compared to other areas of interest within the QoS, is the transport layer interrupted traffic flow.

2.2. Security (Internet Security)

The campus network's services are heavily internet reliant hence the focus on internet security and its protocols such as the Internet Protocol Security (IPsec) which was developed by IETF in the year 1998. This is a set of protocols that was developed to accommodate VPNs. IPSec is an official interoperable security standard developed to provide data integrity, encryption services and basic authentication in order to provide privacy and to protect data from being modified or tampered with. IPSec comprises Authentication Header (AH), Encapsulating Security Payload (ESP) and Internet Key Exchange (IKE) protocols. Essentially IPSec is very useful for the purposes of proving VPN environment across a public network [7]. Additionally, IPSec operates at the network layer (Layer 3), providing security directly to IP packets, which allows it to secure all types of network traffic passing through the IP layer [8]. On the other hand, OpenVPN is one other protocol that operates at the transport layer (Layer 4). Contrary to IPSec, OpenVPN is generally preferred for the purposes of remote access due to its ease of use and flexibility—especially in a case that staff are accessing campus network services away from campus [9] while IPSec is preferred to providing security directly to IP packets, to secure all types of network traffic passing through the IP layer.

2.3. Enhancing Security through Access Control Lists

Access Control Lists (ACLs) are important in safeguarding and hardening both security and QoS on a network. The implementation of ACLs allows for a solid mesh of policy enforcement of predefined and implemented configurations that allow and deny permissions to respective services, files, directories and sensitive data. [10] emphasized that by implementing ACLs, it curtails potential security breaches. In addition, [11] highlighted that ACLs allow fortification to the network in that user and group access rights are enforced.

Table 1. An example of an ACL as demonstrated by [12].

Rule	SIP	DIP	DPort	Proto	Act
1	192.168.*.*	1.2.3.*	[4000, 5000]	TCP	Discard
2	192.168.*.*	1.2.3.*	[0, 3999]	TCP	Accept
3	192.168.*.*	1.2.3.*	[5001, 65,535]	TCP	Accept
4	*	*	*	*	Discard

In the example above (Table 1), the ACL controls TCP traffic from a specified local network (192.168.*. *) to a specific external subnet (1.2.3.*). This entails that the rules selectively block or allow traffic based on destination ports. In this regard below are the rules applied; Ports 4000 to 5000 are blocked while the ports below 4000 and above 5000, up to 65,535, are allowed. Implicitly, any traffic not explicitly matching the first three rules (including all other protocols, source IPs, or destination IPs) will be blocked by Rule 4, ensuring a default security stance of denying access unless explicitly permitted. In a campus setting, traffic can be controlled in this sense e.g., highly on demand applications such as TikTok, Snapchat can be dealt with by applying ACLs as shown above.

Access Control Lists can also be used in controlling permissions to a computer system or server and not only computer networks [13]. Highlighted that ACLs can be a list of statements used to filter traffic in and out of a specific device. Distribution and core devices such as firewalls, routers, and any border technical access device are dependent upon ACLs in order to properly function. ACLs are a list of objects entries that describe the subjects that may access a respective object. Therefore, access to a particular object without matching entries will be denied unless the entries match [14].

2.4. Summary of Existing Research

According to [6] QoS is a suite of technologies utilized to manage bandwidth usage as data crosses computer networks. [6] Its most common use is for the protection of real-time and high-priority data applications in converged networks. To achieve QoS, [15] suggests that many networks especially WANs and MANs will require a deployment model for advanced security services, such as intrusion detection and prevention systems, firewalls, content filters and optimization mechanisms. These security devices will be placed at a few choke points such as core routers and distributed switches, and depend on layer 3 for routing to the intended destinations so that traffic inspection is enforced too. suggested an architecture that included a load balancer to distribute the computational load across the platform resources on the network to further monitor the traffic on the platform [16]. Also emphasized on the incorporation of the SLA and suggested that if deployments deviated from the SLA, it meant that there would have occurred a violation. The platform would be reconfigured dynamically in order to incorporate additional resources from the cloud.

However, on implementing QoS on chock points such as distributed switches and core devices, proposed a maximum weight matching (MWM) scheduling which included additional weight calculation algorithms. The performance of the output-queued packet switches could not be attained. However, a controlled service class separation could be achieved by using the weighted fair queuing (WFQ) weight calculation algorithm. In this regard, fair bandwidth distribution could be achieved.

Meanwhile, [15] decided to calculate weights by using the weighted fair queuing (WFQ) and virtual clock (VC). The foregoing algorithms are very well-known output-queuing scheduling algorithms [17].

3. Network Architecture for Enhanced QoS and Security

3.1. Introduction

Network architecture within academic institutions significantly contributes to the efficacy of information flow and resource accessibility. At the heart of this architecture lies the need for both robust Quality of Service (QoS) and rigorous security. The University of Zambia (UNZA) is no exception to the challenges that arise when attempting to balance these two critical factors. This section explores the optimisation of network architecture, specifically focusing on enhancements in QoS and security within UNZA's Cisco core and distribution devices, alongside the innovative implementation of a mobile application to monitor environmental conditions, such as temperature, within server rooms.

3.2. A Comprehensive Strategy for Optimising Network Architecture for Enhanced Quality of Service and Security: A Case Study of the University of Zambia

To ensure that services remained resilient and available, the undertook UNZA's network architecture so that not only the QoS was implemented but also the security protocols. Cisco Systems, leading providers of networking equipment, offer tools and methodologies that can significantly aid in QoS optimization [18]. By implementing advanced QoS features on Cisco devices, such as prioritisation of bandwidth for essential services and traffic shaping, the study ensured that critical applications received priority so that network congestion was adequately managed. These processes contribute to maintaining service levels that support the academic and administrative functions of the university.

3.2.1. Networking Tools and Hardware

To distillate and implement an optimized network architecture, an array of specialist tools and hardware were into play. At the vanguard of network simulation resides Packet Tracer, a potent tool that conjures a virtual theatre for complex network environments. By leveraging this tool, each twist and turn in the maze of connectivity was scrutinized, laying bare any lacunae in the data packets. The dialogue between Packet Tracer, Wireshark, and SolarWinds, interlaced with the robustness of Cisco devices, formed a network ecosystem.

To achieve optimal QoS and security, the following tools and hardware were employed:

Packet Tracer: Used for simulating complex network environments, enabling detailed visualization and testing of network configurations.

Wireshark: Facilitated in-depth traffic analysis, identifying potential security vulnerabilities and performance bottlenecks.

SolarWinds: Managed network performance, providing real-time monitoring and management to ensure high QoS standards.

The network infrastructure included:

Cisco Network Devices: Routers, switches, and firewalls were selected for their reliability and robust security features.

High-Performance Servers: Equipped with advanced network interfaces to handle high data throughput and ensure low latency.

3.2.2. Components for the Temperature Monitoring System

The server room temperature monitoring system was meticulously designed to ensure accurate environmental monitoring and robust performance:

Raspberry Pi 3: Acts as the central processing unit for the temperature monitoring system.

DHT22 Temperature Sensor: Known for its precision and range, providing accurate readings of temperature and humidity.

LED Indicators: Three LEDs (green, amber, and red) simulate different temperature states, offering a visual representation of the server room's conditions.

Technology Stack

To ensure seamless integration and functionality, the following technology stack was employed:

Java for Android: Powers the mobile application's user interface (UI), ensuring a robust and user-friendly experience.

Python: Handles server-side processing, managing hardware interactions and data processing efficiently.

Nginx Web Server: Deployed on the Raspberry Pi to handle HTTP requests, facilitating real-time monitoring.

PHP: Acts as an intermediary, processing requests between the mobile device and the Raspberry Pi, ensuring smooth communication.

Adafruit-DHT Library: Provides reliable interfacing with the DHT22 sensor, ensuring accurate temperature readings.

3.2.3. System Architecture and Implementation for the Mobile Application

The architecture of the temperature monitoring system includes:

Raspberry Pi Setup: Connected with a breadboard, jumper wires, and 10KΩ resistors to support the DHT22 sensor.

Redundancy and Fault Tolerance: Multiple sensors and Raspberry Pi units are distributed across server racks to ensure data accuracy and system reliability.

Enhanced Security Measures: Encrypted communication and secure authentication protocols are integrated to protect data integrity.

System Implementation and Design

The design of the temperature monitoring tool encompassed:

Real-Time Monitoring: Ensures continuous monitoring with features like data logging and trend analysis.

Real-Time Alerts: Provides immediate notifications for threshold breaches, enhancing proactive management.

Energy Efficiency: Optimizes power consumption to maintain system efficiency.

User Interface: The mobile application's UI not only displays current data but also offers historical trend analysis and alerts, ensuring comprehensive monitoring.

4. Network Design Prototype

The convergence of strategies aimed at optimizing network architecture for improved QoS and security represents a cohesive front in the ever-present battle for network performance and integrity (Figure 1). Through the discerning application of EIGRP, which ensures efficient route selection and swift convergence [19], combined with the meticulous deployment of QoS policies that manage bandwidth and prioritize traffic [6] [20] [21], networks are becoming increasingly adept at handling the variegated demands placed upon them. The stronghold of interface security configurations, reinforced by resilient password protections and ACLs, merge to create a bulwark against the myriad of security threats that abound. Together, these measures coalesce into a formidable force, one that secures the network's foundation while facilitating a service landscape wherein reliability, speed, and security are not mutually exclusive but are, in fact, intrinsically linked tenets of modern network administration. The imperative now is not just to implement these strategies but to continue evolving them in tandem with the threats and demands that define our digital epoch [22].

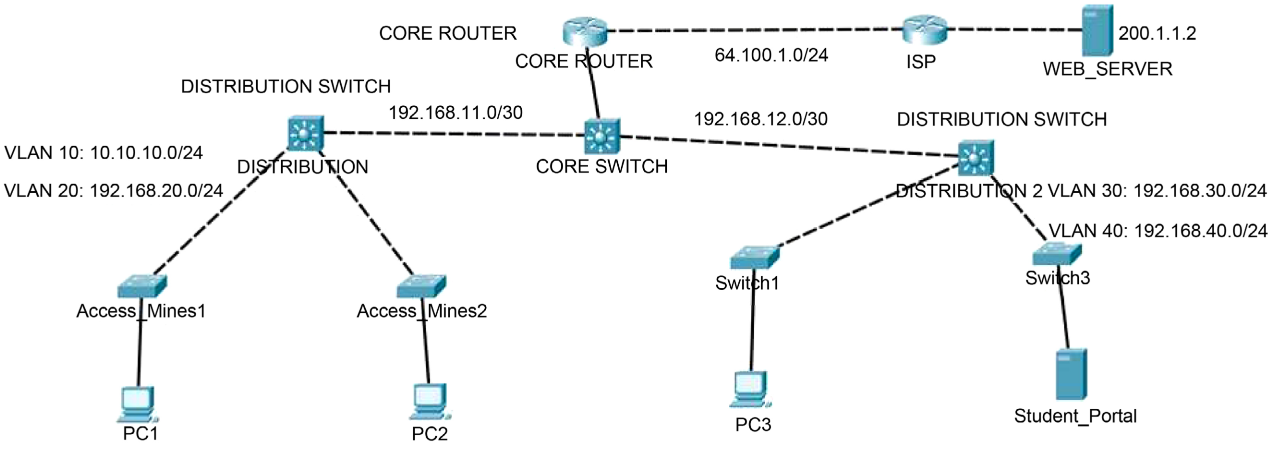

Figure 1. Network design prototype in packet tracer.

4.1. Design and Implementation of the Temperature Monitoring Tool for the Server Room

During the initial design phase of the server room temperature monitoring system, Zigbee sensors were considered for their networking capabilities and suitability in creating a mesh of sensors across the server room. However, after a detailed analysis of the project requirements and a review of the available resources, the decision was made to employ DHT sensors, specifically the DHT22 model, due to several key factors.

To monitor the temperature in the server room of the University of Zambia a monitoring prototype was designed. This included, a Raspberry Pi 3, a DHT22 Temperature sensor, 3 LED Lights and an Android device as shown in Figure 2 below.

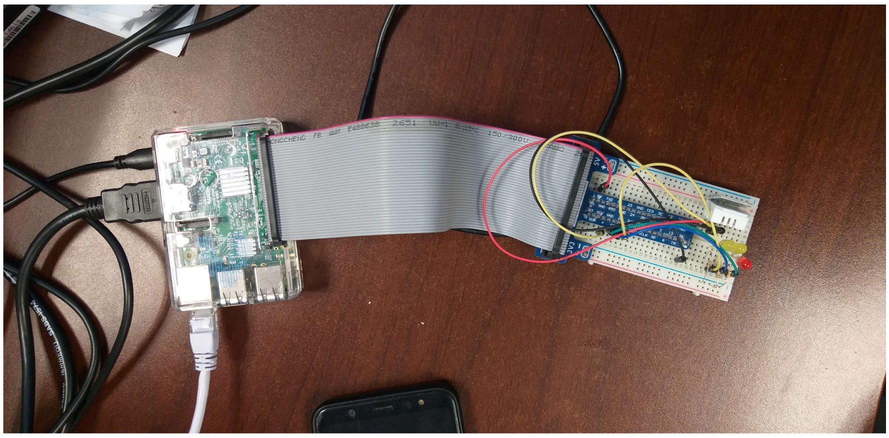

Figure 2. Hardware system setup.

In as much as the study was focused on hardening security and QoS of the Cisco devices, the performance of the network is hampered by the temperatures and the design and implementation of the server room [23], hence the implementation of a temperature monitoring tool to monitor temperatures of the server room in order to not only safeguard the Cisco Core and Distribution devices but to also improve their Quality of Service. Figure 2 furnishes a schematic representation of the interaction between the user and the temperature control system. This interaction commences with the user initiating the application and signing in, a gateway to the intricate dance of requests and responses that characterize modern-day application programming interfaces (APIs) and microcontroller functionality [24]. The authentication protocol enshrined in the login screen is but the prologue to a narrative where API calls facilitate communication between the user's mobile device and a remote Raspberry Pi, a credit card-sized computer that has transcended its educational roots to find myriad applications in hobbyist projects and professional IoT solutions alike [25].

The Raspberry Pi processes requests sent by the API. The Raspberry Pi is armed with peripherals such as LED lights and temperature sensors. When the user engages the temperature readings or adjustments, the Raspberry Pi interprets these API requests as commands, executing actions such as toggling LED lights based on the parameters received. This tangible change, executed by the LEDs, becomes a physical manifestation of the virtual interaction, a feature highlighted as crucial to the user experience by [3] in their discourse on interaction design.

To be able to read the temperatures, python language and the DHT Library were used to fulfill the whole process. To be able to read the temperatures, python language and the DHT Library were used to fulfill the whole process (Figure 3).

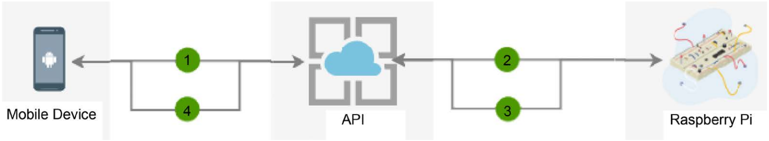

Figure 3. Illustrates the user interaction with the system.

4.2. Units

Underscoring the very fabric of the research is the affinity towards the International System of Units (SI), extensively recognized as the modern form of the metric system. Predominantly, the SI unit system, also known as the MKS (Meter, Kilogram, Second) system, serves as the cornerstone for all measurements and computations detailed in the study.

Each type of delay, from propagation to queuing, is meticulously quantified in milliseconds, a derivative of the base unit second within the SI framework. The meticulous detailing of such units ensures that each measured aspect of delay, Propagation Delay, Transmission Delay, Processing Delay, and Queuing Delay—resides within a unified temporal spectrum, favouring milliseconds (ms) as the unit of choice. By adhering to this convention, the study harmonizes its findings and facilitates a seamless comparison among the various delay metrics.

In summary, we primarily incorporate SI (MKS) units such as meters (m), seconds (s), and bits per second (bps). For instance:

Propagation Delay: Measured in milliseconds (ms).

Transmission Delay: Expressed in milliseconds (ms).

Processing Delay: Recorded in milliseconds (ms).

Queuing Delay: Noted in milliseconds (ms).

Total Delay (Latency): Summed up in milliseconds (ms).

4.3. Performance Comparison: Peak vs. Off-Peak Hours Calculations/Equations

4.3.1. Off-Peak Hours

During off-peak hours, the network performance was observed to be stable with minimal packet loss and consistent TTL values. The following metrics were recorded:

During off peak hours, the network produces the following results:

$$\text{Propagation Delay} = \text{Distance} \div \text{Propagation Speed}$$

At the speed of light in fiber optics (Approximately 2×10^8 m/s and an average distance of 1 km (1000 meters) for the internal network communication:

$$\text{Propagation Delay} = \frac{1000 \text{ meters}}{2 \times 10^8 \text{ m/s}} = 5 \times 10^{-6} \text{ seonds} = 0.005 \text{ ms} \quad (1)$$

Transmission Delay

Transmission delay is the time it takes to push all the packet's bits into the wire was calculated as follows:

$$\text{Transmission} = \frac{12000 \text{ bits}}{100 \times 10^6 \text{ bps}} = 0.12 \text{ ms} \quad (2)$$

Packet Loss Calculation

At the time of testing the network transmitted 10 packets out of 500 are not received at the destination, the packet loss was calculated as follows:

Packet Loss = Number of Lost Packets Total Number of Packets Sent × 100 = 10500 × 100 = 2% Packet Loss = Total Number of Packets Sent Number of Lost Packets × 100 = 50010 × 100 = 2% (3)

4.3.2. Peak Hours

During peak hours, network performance showed noticeable degradation with higher packet loss and inconsistent TTL values:

Packet Loss Calculation:

At peak times, the network experienced approximately 8% packet loss.

For comparative purposes, assume the network transmitted the same number of packets (500). Hence, the packet loss calculation would be:

$$\text{Number of Lost Packets} = 8\% \times 500 = 40 \text{ packets}$$

$$\text{Packet Loss} = 40/500 \times 100 = 8\%$$

This indicates 40 packets lost out of 500, conforming the 8% packet loss during peak hours.

5. Discussion and Results

Given that server performance and longevity are closely linked to environmental conditions, notably temperature and humidity, the necessity for continuous monitoring cannot be overstated. Addressing this need, the mobile application developed through the described project performs the dual task of collecting and disseminating temperature data while offering a user-friendly interface to facilitate data access and interaction.

5.1. Development and Functionality of the Mobile Application

```
from flask import Flask, jsonify
import Adafruit_DHT
import logging
app = Flask(__name__)
# Configure logging
logging.basicConfig(filename='error.log', level=logging.ERROR)
@app.route("/temperature")
def read_temperature():
    try:
        humidity, temperature = Adafruit_DHT.read_retry(Adafruit_DHT.AM2302,
4)
        if humidity is not None and temperature is not None:
            return jsonify({"response": True, "temperature": temperature, "humidity":
humidity})
        else:
            return jsonify({"response": False, "error": "Sensor reading failed."})
    except Exception as e:
        logging.error(f"Error reading sensor: {e}")
        return jsonify({"response": False, "error": str(e)})

if __name__ == "__main__":
    app.run(host='0.0.0.0', port=8080)
```

5.1.1. Temperature Reading

Figure 4 displays the reading of temperature in the server room. Central to the mobile application's utility is its ability to provide real-time temperature and humidity readings. By integrating an Adafruit DHT sensor, the project showcases how an unobtrusive piece of hardware can become a data nexus when coupled with a suitable software environment. Flask—a micro web framework written in Python—serves as the conduit through which sensory data is read and conveyed to the end-user. This data's polling frequency—every 10 seconds—is carefully chosen to balance the need for timely information against the bandwidth constraints and potential for sensor data "noise".

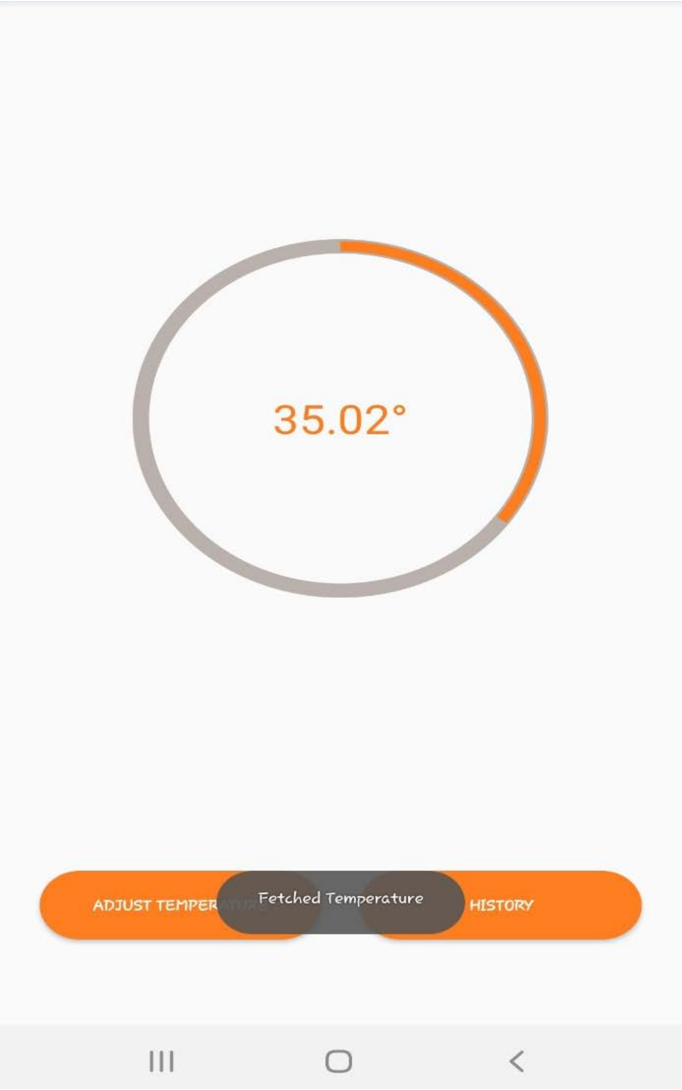

Figure 4. Temperature reading.

5.1.2. User Authentication

Figure 5 displays user system authentication. Security cannot be an afterthought in a system that serves critical data. The user authentication functionality is a testament to the understanding that the sanctity of server room metrics extends beyond physical protection to digital defense. By necessitating user authentication before providing access to temperature readings, the developed prototype ensures that sensitive information falls solely into trusted hands. User credentials are the keys to the digital gateway, maintaining a barrier between the data and potential unauthorized access.

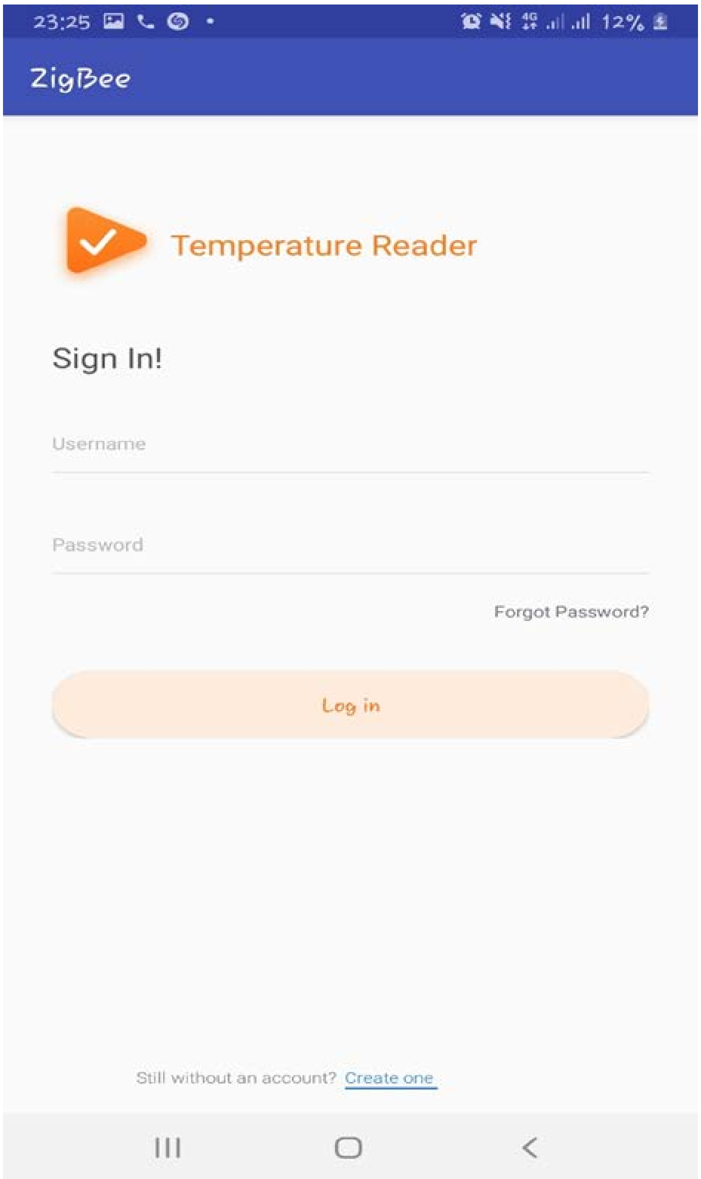

Figure 5. User system authentication.

5.2. Error Handling

Invariably, technology is susceptible to anomalies and errors—sensor malfunctions, network interruptions, or even software glitches. Robust error handling mechanisms are hence incorporated within the application, illustrating a proactive stance in anomaly identification and resolution. Logging, an essential error management strategy, notifies administrators of irregularities, triggering a cascade of responses to mitigate the identified issues without user intervention.

5.3. Network Management and Resource Allocation Analysis

Network management—specifically in terms of resource allocation—is pivotal to the smooth operation of any IT-related endeavor. DHCP (Dynamic Host Configuration Protocol) lease data analysis, an aspect of this project, provides an in-depth look at network usage patterns. Analyzing this data across varied network pools, the project unveils diverse lease utilization trends, affording crucial insight into how network resources are allocated and consumed.

Table 2. DHCP lease allocation.

Network Pools	Pool 1	Pool 2	Pool 3
Leased IP Addresses	202	51	137
Excluded/Reserved	46	13	117
Max IPs	254	254	254

Table 2 extrapolates a varied DHCP lease. Data examination revealed a spectrum of utilization patterns, potentially indicative of differing demand dynamics. This research showed a varied IP address usage across three network pools examined. Pool 1's high lease count (202) suggests heavy network traffic, necessitating stringent Quality of Service (QoS) measures to ensure efficient bandwidth distribution. Pool 2, with fewer leases (51), may require less intensive QoS management and security threats. Pool 3's high number of reserved IPs (117) indicates prioritized devices or critical services, emphasizing security needs to protect these assets. Ensuring robust security protocols and monitoring are crucial for safeguarding reserved IPs and maintaining network integrity across all pools.

Such data informs computer networking staff about network resource allocation that respects both efficiency and sufficiency. Optimal allocation facilitates an organized approach to resource distribution, preventing bottlenecks and maximizing server room operations under varying load conditions.

Fortifying Policies for Critical Traffic Prioritization and Congestion Control

1) Fortifying QoS Policies for Traffic Prioritization

Quality of Service policies are critical for ensuring that essential traffic is given priority over less important data. Establishing more robust QoS mechanisms involves implementing frameworks that can distinguish between types of traffic so that essential services receive precedence.

2) Network Upgrades to Optimize Bandwidth

The continuous increase in network traffic necessitates the expansion of bandwidth to support the growing demand. Upgrading network infrastructure includes deploying additional resources and optimizing existing frameworks to manage peak loads efficiently [26].

3) Load Balancing for Traffic Distribution

Load balancing is the strategic distribution of traffic across available network paths to prevent any single resource from being overwhelmed. This is particularly vital during times of peak load, where loads are dispersed over multiple servers or network links [27].

4) Access Control Lists

Implementation of Access Control Lists on the Cisco devices both in implementing QoS and in securing Network resources [28].

6. Conclusions and Implications

The confluence of mobile application development with network management for server room temperature control narrates a tale of modern-day necessity meeting innovation. The project not only contributes to the pragmatic aspect of server room temperature monitoring but also offers a blueprint for similar applications in other domains of IT infrastructure oversight. The implications of this integration are far-reaching—user authentication and error handling underscore a commitment to security and reliability, while DHCP lease data analysis heralds a methodical approach to network management.

Future advancements may see the integration of more complex artificial intelligence algorithms to predict anomalies before they occur or even extend the functionalities to encompass predictive maintenance of server hardware. The upshot is clear: as organizations continue to digitalize, the role of application-induced simplification, coupled with data-driven network management, will become increasingly central not just to the upkeep of hardware but to the resilience and efficiency of business operations at large. The project thus reflects a microcosm of the larger narrative of technological evolution: the ceaseless drive to harmonize the potential of software with the demands of real-world application. This exploratory essay encapsulates the importance of understanding and implementing precise monitoring and management solutions tailored to the needs of server rooms—the crucial data centers of the contemporary digital world.

7. Figures and Tables

Table 3 elucidates data that was collected based on applications that are mostly used on the university network as follows and that may impact on QoS and Security, especially on the Cisco Core and Distribution devices.

Based on **Table 3**, we observe that internet application usage is considerably varied, with TikTok dominating at 31.3%, outpacing traditional heavyweights like Facebook at 17.5%. Other applications collectively occupy a predominant usage of 28.5%, significant but less than the individual leaders. YouTube and QUIC follow with 5.7% and 4.9% respectively. Security protocols, namely TLS, represent 4.6%, while Cloudflare accounts for 4.2%, suggesting robust online security concerns. WhatsApp involves a share of 3.3%.

Table 3. Network application usage.

Common Internet Application Platforms	Usage (%)
Facebook	17.5
Other applications	28.5
YouTube	5.7
QUIC	4.9
TLS	4.6
Cloudflare	4.2
WhatsApp	3.3
TikTok	31.3
Total Application Usage	**100**

a. Network application usage.

7.1. Distribution of Internet Performance Ratings by the User Community

Figure 6 displays the distribution of internet performance ratings as evaluated by the UNZA community. Each bar represents the proportion of ratings across different performance levels, providing a visual summary of user experiences and feedback on internet performance.

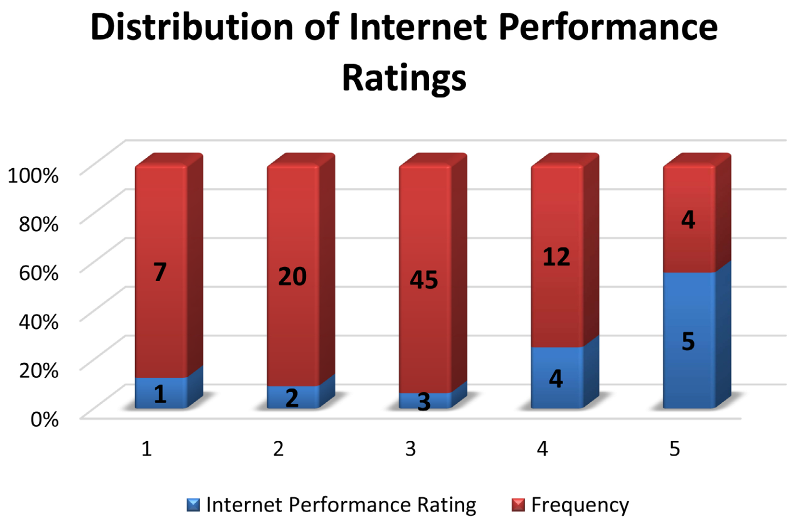

Figure 6. Stacked bar chart of internet performance ratings distribution by user community.

Figure 6 Labels: A recent survey conducted within the campus setting employed a Stacked Bar Chart to visually represent user satisfaction in relation to the dependability of the campus's internet connection. The analysis revealed distinct trends: periods with stable internet connectivity coincided with heightened end-user satisfaction. Conversely, periods characterized by significant network fluctuations or outages saw a stark depreciation in satisfaction levels.

7.2. Analysis of Results

The chart shows that the most common rating for Internet performance is relatively "Good" (Rating 3), with 45% of responses. The second most common rating is "Fair" (Rating 2) with 20% of responses. Ratings of "Poor" (Rating 1) and "Excellent" (Rating 5) are the least frequent, with 7% and 4% of responses respectively. This distribution indicates a general satisfaction with the Internet service, but also highlights areas for potential improvement, particularly for those who rated the performance poorly. Overall, the data demonstrates a central tendency towards average ratings, with fewer respondents at the extremes.

In view of the survey, the study highlighted the benefits of implementing real-time network monitoring and optimization of QoS. This proactive measure could detect and potentially preempt disruptions before they negatively impact network users.

Above all, this could extend to an array of improvements like increased bandwidth, more access points, and advanced hardware capable of handling higher data demands and above in relation to the existing infrastructure, to harden configurations of the core and distribution cisco devices.

Acknowledgements

We extend our sincere gratitude to the University of Zambia for their invaluable support of our research.

Conflicts of Interest

The authors declare no conflicts of interest regarding the publication of this paper.

References

[1] Wei, L., Zeng, X. and Shen, T. (2015) A Wireless Solution for Train Switchgear Contact Temperature Monitoring and Alarming System Based on Wireless Communication Technology. *International Journal of Communications*, *Network and System Sciences*, **8**, 79-84. https://doi.org/10.4236/ijcns.2015.84010

[2] Selwyn, N., Nemorin, S. and Johnson, N. (2016) High-Tech, Hard Work: An Investigation of Teachers' Work in the Digital Age. *Learning*, *Media and Technology*, **42**, 390-405. https://doi.org/10.1080/17439884.2016.1252770

[3] Choi, B., Song, S., Koffler, G. and Medhi, D. (2007). Outage Analysis of a University Campus Network. 2007 16*th International Conference on Computer Communications and Networks*, Honolulu, 13-16 August 2007, 675-680. https://doi.org/10.1109/icccn.2007.4317895

[4] Srivastava, B., Krithikaivasan, S., Beard, C., Medhi, D., Alanqar, W. and Nagarajan, A. (2002) Benefits of Traffic Engineering Using QoS Routing Schemes and Network Controls. *Computer Communication*, **27**, 271-275.

[5] Al-Shammari, B.K.J., Al-Aboody, N. and Al-Raweshidy, H.S. (2018) Iot Traffic Management and Integration in the Qos Supported Network. *IEEE Internet of Things Journal*, **5**, 352-370. https://doi.org/10.1109/jiot.2017.2785219

[6] LiveAction, Inc. (2020) Cisco QoS Handbook. 2nd Edition, LiveAction, Inc.

https://www.liveaction.com/wp-content/uploads/2021/04/Cisco-QoS-Handbook-92620.pdf

[7] Chawla, B., Gupta, O.P. and Sawhney, B.K. (2014) A Review on IPsec and SSL VPN. *International Journal of Scientific Engineering and Research*, **5**, 21-24.

[8] Frankel, S., Kent, K., Lewkowski, R., Orebaugh, A.D., Ritchey, R.W. and Sharma, S.R. (2005) Guide to IPsec VPNs.

[9] Qbal, M. (2019) Analysis of Security Virtual Private Network (VPN) Using Openvpn. *International Journal of Cyber-Security and Digital Forensics*, **8**, 58-65. https://doi.org/10.17781/p002557

[10] Jones, B. and Bejtlich, R. (2019) The Practice of Network Security Monitoring: Understanding Incident Detection and Response. No Starch Press.

[11] Hernandez, J.M. (2018) Implementing Cisco IP Switched Networks (SWITCH) Foundation Learning Guide: (CCNP SWITCH 300-115). Cisco Press.

[12] Liu, A.X., Torng, E. and Meiners, C.R. (2011) Compressing Network Access Control Lists. *IEEE Transactions on Parallel and Distributed Systems*, **22**, 1969-1977. https://doi.org/10.1109/tpds.2011.114

[13] Lutkevich, B. (2023) What is Access Control List (ACL)? https://www.techtarget.com/searchnetworking/definition/access-control-list-ACL

[14] Knipp, E., Browne, B., Weaver, W., Baumrucker, C.T., Chaffin, L., Caesar, J., *et al*. (2002) Network Security Management. In: Knipp, E., *et al*., Eds., *Managing Cisco Network Security*, Elsevier, 593-648. https://doi.org/10.1016/b978-193183656-2/50018-0

[15] Homan, P. and Bester, J. (2001) Enabling Quality of Service in Input-Queued Packet Switches. *EUROCON'*2001. *International Conference on Trends in Communications*, *Technical Program*, *Proceedings* (Cat. No.01EX439), Bratislava, 04-07 July 2001, 496-499. https://doi.org/10.1109/eurcon.2001.938170

[16] Rastogi, R., Breitbart, Y., Garofalakis, M. and Kumar, A. (2002) Optimal Configuration of OSPF Aggregates. *Proceedings of Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies*, New York, 23-27 June 2002, 874-882. https://doi.org/10.1109/infcom.2002.1019334

[17] Mellouk, A., Hoceini, S. and Zeadally, S. (2011) A Bio-Inspired Quality of Service (QoS) Routing Algorithm. *IEEE Communications Letters*, **15**, 1016-1018. https://doi.org/10.1109/lcomm.2011.071211.110741

[18] Vinod Chandra, S.S. and Anand Hareendran, S. (2024) Modified Smell Detection Algorithm for Optimal Paths Engineering in Hybrid SDN. *Journal of Parallel and Distributed Computing*, **187**, Article 104834. https://doi.org/10.1016/j.jpdc.2023.104834

[19] John, R. and Ying, S. (2015) A Comparison of OSPFv3 and EIGRPv6 in a Small IPv6 Enterprise Network. *International Journal of Advanced Computer Science and Applications*, **6**, 162-167. https://doi.org/10.14569/ijacsa.2015.060123

[20] Jourjon, G., Lochin, E. and Sénac, P. (2008) Design, Implementation and Evaluation of a QoS-Aware Transport Protocol. *Computer Communications*, **31**, 1713-1722. https://doi.org/10.1016/j.comcom.2007.11.015

[21] Ferretti, S., Ghini, V., Turrini, E., Pellegrini, M. and Panzieri, F. (2010) "QoS" in 2013 IEEE Sixth International Conference on Cloud Computing, 321-328. http://doi.ieeecomputersociety.org/10.1109/CLOUD.2010.17

[22] Dangwal, K. and Kumar, V. (2014) Comparative Study of Eigrp and Rip Using Cisco Packet Tracer. *International Journal of Engineering Sciences and Emerging Tech-*

nologies, **6**, 475-480.

[23] Sizemore, B., Snook, T., and Neumeister, W. (2010) Cisco Data Center Energy Efficiency. Bachelor's Thesis, Senior Project, California Polytechnic State University, 1-23. https://digitalcommons.calpoly.edu/do/search/?q=B.%20Sizemore%2C%20T.%20Snook%2C%20and%20W.%20Neumeister&start=0&context=374206&facet=

[24] Pavithra, D. and Balakrishnan, R. (2015) IoT Based Monitoring and Control System for Home Automation. 2015 *Global Conference on Communication Technologies*, Thuckalay, 23-24 April 2015, 169-173. https://doi.org/10.1109/gcct.2015.7342646

[25] Louis, L. (2016) Working Principle of Arduino and Using It as a Tool for Study and Research. *International Journal of Control*, *Automation*, *Communication and Systems*, **1**, 21-29. https://doi.org/10.5121/ijcacs.2016.1203

[26] Wairisal, M. and Surantha, N. (2018) Design and Evaluation of Efficient Bandwidth Management for a Corporate Network. 2018 *International Conference on Information Management and Technology*, Jakarta, 3-5 September 2018, 98-102. https://doi.org/10.1109/icimtech.2018.8528162

[27] Sim, K.M. and Sun, W.H. (2003) Ant Colony Optimization for Routing and Load-Balancing: Survey and New Directions. *IEEE Transactions on Systems*, *Man*, *and Cybernetics—Part A*: *Systems and Humans*, **33**, 560-572. https://doi.org/10.1109/tsmca.2003.817391

[28] Sedayao, J. (2001) Cisco IOS Access Lists. O'Reilly Media.