CISCO Packet Tracer Enterprise Level Architecture using the Concept of SDN

Agha Muhammad Yar Khan[1], Saima Shaheen [1], Abdul Samad Danish[2], Umul Warah[3], Onib -UR- Rehman[4], Muhammad Faizan Hassan[2]

[1]Department of Software Engineering, HITEC University
[2]Department of Computer Science, HITEC University
[3]School of Computer Science and Electrical Engineering, NUST Islamabad
[4]Department of Electrical Engineering, HITEC University

Abstract- This study investigates the utilization of Cisco Packet Tracer to implement Software-Defined Networking (SDN) principles in the enterprise-level architecture of HITEC University. Although the study does not entirely migrate to SDNs, it illustrates how the implementation of SDN principles improves network control and flexibility for administrators and users alike. By utilizing centralized control and API-driven automation, it is possible to effectively distribute configurations across the entire network topology. This facilitates prompt adaptations to changing business demands. The controller facilitates the seamless implementation of Quality of Service (QoS) policies, resulting in enhanced network operations. Virtualization for realistic network performance simulation and improved architectural efficiency may be incorporated into future optimization. Furthermore, the paper explores the incorporation of backup tools as a precautionary measure to safeguard data in virtualization or server-level scenarios. In general, the incorporation of SDN principles into the enterprise architecture of HITEC University demonstrates enhanced network management agility, efficiency, and resilience.

Index Terms- Software-Defined Networking, Quality of Service, enhanced network management

I. Introduction

The entertainment industry and smart device proliferation have increased, the internet has become an indispensable component of daily life. This development has facilitated progress in both wired and wireless communication paradigms. The origins of ARPANET can be identified in the 1960s, at which point the initiative was launched by the Advanced Research Project Agency (ARPA) [1]. In 1969, two computers located at the Stanford Research Institute, University of California, exchanged a message. This occurrence is fundamentally considered to be the inception of the internet. A multitude of advancements transpired subsequent to this inception, which has subsequently become an indispensable juncture in history. This, among other things, increased the demand for network devices capable of managing fluctuating network traffic, an extensive pool of IP addresses, broadband transmission links, and sophisticated network applications, and prompted an ever-increasing need for an efficient architecture of a communication system. The communication networks comprise a diverse range of interconnected devices that facilitate the exchange of information and the sharing of resources. These devices may include hubs, modems, WiFi routers, servers, computers, switches, routers, and multi-layer switches. There are three distinct types of networks: LAN, WAN, and MAN. In regard to both its magnitude and function, each is distinct. A Local Area Network (LAN) comprises a collection of interconnected network devices that function as a local area network. In essence, it represents a scaled-down version of the internet that caters to a restricted user base. In contrast, WAN stands for Wide Area Network and, as its name suggests, connects various LANs located in different countries or cities across a larger geographical area. The fundamental architecture of a WAN (Wide Area Network) and LAN (Local Area Network) is illustrated in Fig. 1. This illustrates why WANs are frequently used to connect disparate LANs. In terms of geographic coverage, MAN (Metropolitan Area Network) falls between LAN and WAN. It encompasses a greater extent than a LAN but a lesser extent than a WAN.

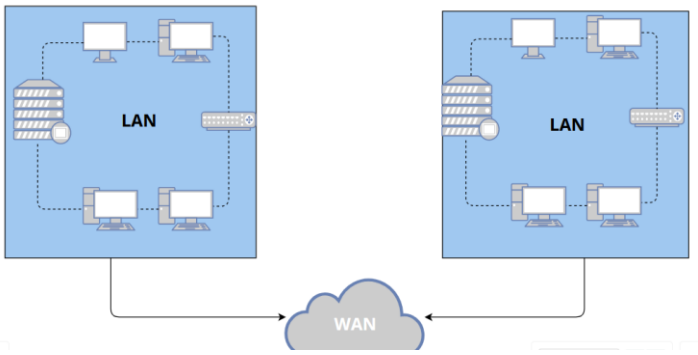

Figure 1 Basic Local Area Network Architecture

According to the most recent figures, there will be an average of 5.30 billion people connected to the internet by the time October 2023 rolls around [2]. This ever-increasing demand for internet services necessitates the Quality of Experience (QoE) in terms of an upgraded architecture, increased security, and higher link speed which in terms can enhance the experience for the user i.e. a student in this case. By enhancing the experience dose not just means to increase the speed but also enhance the response time significantly.

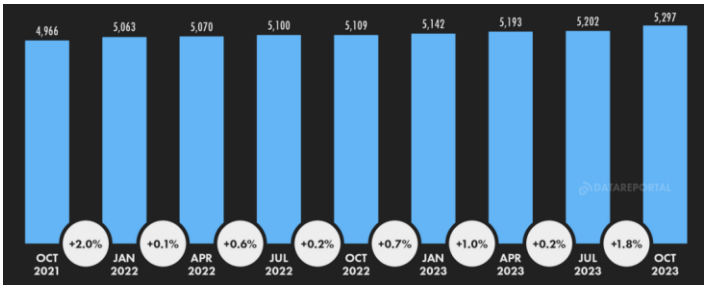

Figure 2 Statistics of Network Users

The interconnection between local and global development is evident in the necessity to enhance the efficiency of wide area networks while simultaneously optimizing and increasing the effectiveness of LANs. Therefore, when network administrators design a network architecture, they do not utilize physical devices directly; rather, they utilize a simulation tool to design and simulate a fundamental architecture. Network simulation is a critical element of contemporary network engineering because it provides specialists with a controlled environment in which to analyze, generate, and resolve problems associated with intricate network infrastructures. The evolutionary trajectory of network simulation tools has been marked by pivotal moments in their development and progression. In the initial stages (1970s-1980s), simulation tools were quite basic, primarily employing custom programs and rudimentary simulation languages to execute simulations at the packet level. In the late 1980s, Network Simulator (NS) represented a significant turning point; it was subsequently enhanced into NS-2, a widely employed discrete-event simulator among academics. OPNET (Optimized Network Engineering Tools), a commercial tool introduced in the 1990s, boasted a comprehensive feature set that included a graphical user interface and a vast repository of components. Prominent in academic and research circles throughout the 2000s, OMNeT++ offered a modular and extensible framework that facilitated the simulation of discrete events and networks. Simultaneously with its release, GNS3 (Graphical Network Simulator-3) revolutionized the field of network simulation by permitting the emulation of tangible network devices within a simulated environment. This development brought about substantial enhancements in the area of certification preparation and training. Developed during the 2000s, Cisco Packet Tracer gained significant traction as an educational tool due to its intuitive interface that facilitated network design, configuration, and troubleshooting. NS-3, which replaced NS-2 in the 2010s, introduced a contemporary and expandable simulation platform. Concurrently, Mininet rose to prominence as a lightweight virtualization-based simulator for simulating large-scale software-defined networks (SDNs). During the 2010s-2020s, network simulations were increasingly concerned with fidelity and realism; tools such as EVE-NG and VIRL improved the emulation of actual network environments, enabling the testing and training of complex scenarios. The aforementioned historical summary highlights the ongoing evolution and variety of network simulation tools in order to meet the changing demands of scholars and practitioners in the field of networks. Network simulation is commonly conducted using Cisco technologies owing to their historical importance, industry congruence, and pedagogical suitability. Cisco Packet Tracer is among a multitude of products that exemplify the organization's commitment to the advancement of network technology and provision of comprehensive educational resources. Its connection with Cisco's certification programs, user-friendly interface, and extensive device compatibility make it an excellent choice for educational environments. Furthermore, the incorporation of the Cisco IOS simulation into the Cisco Packet Tracer simplifies the transition from theoretical comprehension to practical implementation, thereby enhancing the educational experience for aspiring network professionals. The tool's widespread adoption in academic and training institutions is partially attributable to its capability of multi-user collaboration and frequent updates. The instrument of choice is Cisco Packet Tracer owing to its extensive feature set, industry compatibility, and capacity to facilitate the development of networking skills. When devising the network architecture, optimization should be regarded as a primary consideration. The process of optimizing the network architecture entails refining the configuration and design in order to enhance efficiency, productivity, and resource utilization. The primary objectives are to enhance overall throughput, reduce latency, and streamline operations within the network architecture. Software-Defined Networking (SDN) has significantly transformed network administration with the aim of enhancing network performance. By separating the control plane from the data plane, SDN empowers centralized software controllers to deliver dynamic control. Control and data plane separation, an SDN controller, the Open Flow protocol, programmability, and centralized intelligence are all defining characteristics of SDN. It is difficult to bridge the divide between SDN and legacy-based network equipment due to the static nature of legacy networks. Utilizing network principles is essential for optimizing and securing the internet. Layered networks and progressive migration processes enable SDN integration to occur seamlessly, without causing any disruptions to the existing infrastructure. Protocol converters and gateways facilitate interoperability, which ensures compatibility with pre-existing systems. The application of network policies can be standardized and defined across both SDN and legacy components through the implementation of policy-based techniques. The intelligent integration described herein establishes a secure and optimized network environment through the harmonization of SDN's intelligence and adaptability with the dependable nature of legacy infrastructure. This article explores the application of network principles to facilitate the integration of legacy-based network apparatus with software-defined networks, with the ultimate goal of enhancing internet security and optimization. Utilizing some attributes of legacy-based network architecture in conjunction with novel concepts from SDN, a pioneering innovation, can result in an optimized network architecture, as the subject of this article. We designed an optimized architecture in this article in consideration of the case study described in greater detail at the conclusion of the section on background knowledge and motivation. An introductory section of this article establishes the groundwork by delineating the requirements of the internet and its exponential growth; this, in turn, necessitates optimized network availability, as we subsequently elaborate. In the literature review section, we examined the existing research in the field of network architecture optimization, including literature pertaining to

software-defined networking (SDN) and Cisco packet tracers. Following that, in the third section of this article, which describes our strategy for designing the network architecture, we detailed the methodology. The results of the implementation have been described in the implementation segment. We concluded the article in the fifth section by presenting the results and outlining potential avenues for future enhancements to this implementation. The impetus for composing this research article stemmed from an extensive examination and ensuing enhancement of the university campus network infrastructure. This endeavor was motivated by the imperative to address the growing demands of a modern educational environment, where redundancy, scalability, and continuous connectivity are critical. The resulting case study serves as a foundational inquiry, offering valuable perspectives on the research objectives and methodologies elaborated in the forthcoming research article. Principal Objectives of the Case Study: The case study, meticulously implemented in accordance with the university's specific requirements, aimed to develop a network architecture that not only satisfied immediate needs but also anticipated and facilitated future growth. The main goals were to optimize network performance, ensure scalability to support expanding user bases and devices, and implement redundancy measures to bolster network reliability. Significance to the Research Article: The case study serves as the fundamental premise for our research article, offering a pragmatic context in which to evaluate our proposed methodologies and conclusions. This exemplifies the principles and strategies that we advocate for within the broader context of network optimization in academic institutions. Field Contributions: The implications of the case study's findings are far-reaching and add to the current body of knowledge regarding network architecture design. Based on the findings of the case study, our research article presents recommendations for the establishment of network infrastructures that are resilient, scalable, and adaptable in similar educational settings. Abstract of the article on procedural research: Building upon the knowledge acquired from the case study, the prospective scholarly article conducts a comprehensive analysis of cutting-edge methodologies, technologies, and ideal approaches pertaining to the architecture of university networks. The main goal is to provide a sophisticated framework that enables the creation of dependable, expandable, and resilient network infrastructures, with a particular focus on accommodating the evolving needs of academic institutions. As their name suggests, legacy networks consist of devices and configurations that were established with less advanced technologies and information from the past. Therefore, the discourse may commence by defining the term "legacy," which simply denotes the obsolescence of a contemporary network architecture. In the contemporary era, network devices have undergone substantial advancements in both hardware and software applications [3]. Figure 3 illustrates the legacy-based network architecture that is highly interconnected. In legacy-based network topologies, the network administrator or user has reduced autonomy. Network architectures that are based on legacy systems often restrict the independence of network administrators in regards to the management and improvement of the network infrastructure. The existing designs' scalability may prove insufficient to accommodate the growing demands of modern businesses, potentially requiring time-consuming and arduous configuration procedures throughout network expansion. Typically, the limitations of automation capabilities lead to augmented operational expenses and delayed response durations. Furthermore, the implementation of vendor-supplied proprietary technologies could hinder administrators' capacity to choose the most suitable solutions for their needs, thus creating a vendor lock-in circumstance. Security is hindered by obsolete hardware, software, and protocols; consequently, administrators are unable to implement the most recent security measures. Insufficient support for virtualization and dynamic resource allocation presents an obstacle to meeting the changing demands of the business in an efficient manner. Furthermore, the absence of advanced monitoring tools in legacy networks may potentially reduce visibility, thus hindering the proactive detection and resolution of issues by administrators. Further impeding the seamless integration of emerging technologies and the prompt establishment of novel services are the complexity of change management procedures. By transitioning to more modern architectures, these limitations can be alleviated, providing network administrators with increased independence, scalability, automation, and adaptability in managing their networks. Nonetheless, these network architectures are unsuitable for the vast majority of current network utilization for a variety of reasons. Legacy network architectures are not favored for the majority of contemporary network applications for a variety of valid reasons. To begin with, the scalability challenges of these designs impede their ability to accommodate the expanding requirements of modern enterprises. The advancement of legacy networks in tandem with an organization can present challenges, potentially resulting in operational inefficiencies and bottlenecks. Moreover, the inherent manual configuration processes of these systems impede the ability to promptly adapt to evolving business demands, thereby diminishing agility. An additional drawback is the lack of advanced automation functionalities, which prolongs the duration of routine tasks and hinders administrators from responding promptly to network incidents. Moreover, the reliance on vendor-specific technologies gives rise to interoperability challenges, impeding the incorporation of innovative and diverse solutions. The susceptibility of legacy architectures to evolving cyber threats stems from security vulnerabilities introduced by obsolete hardware and protocols. Such vulnerabilities can significantly compromise the integrity and confidentiality of data. The absence of assistance for dynamic resource allocation and virtualization further hinders cost-effectiveness by impeding optimal resource utilization. When all things are considered, these disadvantages demonstrate that legacy network architectures are insufficient to meet the demands of the contemporary, constantly evolving network consumption environment. Due to the intimate connection between the control plane and data plane in legacy-based network architectures. Local markets are typically the best option for acquiring the devices utilized in legacy-based network architectures, as doing so allows for cost savings and eliminates the need to deal with obsolete documentation. Vendors do not always dispatch these devices directly to consumers. One of the drawbacks of legacy-based designs is that they are vendor-specific and the majority of devices depend on particular hardware components. However, it is not secure to rely on legacy

network architectures to fulfill client expectations. Due to a multitude of factors, legacy network architectures were deemed insufficiently secure to meet the demands and expectations of customers. The inherent security vulnerabilities present in obsolete hardware, software, and protocols are a significant cause for concern. The likelihood that ransomware, malware, and other malicious attacks will target legacy systems increases as technology advances. These vulnerabilities may facilitate access to, compromise of, or breach of sensitive consumer data, which could potentially lead to the loss of data. An additional significant challenge pertains to the insufficient assistance that legacy architectures offer in accommodating modern security solutions. Emerging technologies often incorporate sophisticated security measures that provide superior protection against evolving threats. The absence of these functionalities in legacy systems may expose clients to the risk of cybercrime and complicate the implementation of robust security measures by network administrators. The inability to promptly detect and address security issues may also be compromised by the time-consuming configuration processes and delayed response times of antiquated networks. In the current dynamic threat environment, clients desire networks to be equipped with adaptable and automated security protocols to effectively manage emerging threats. It is possible that legacy designs fail to satisfy the performance, scalability, and dependability expectations of clients. As organizations grow and demand more advanced capabilities from their networks, the deficiencies of obsolete systems become increasingly apparent. The overall contentment of customers could potentially be affected by network outages, compromised performance, and challenges in adapting to evolving business requirements. Additionally, clients frequently desire interoperable and adaptable network solutions. Due to their dependence on vendor-specific technologies and inability to support virtualization, legacy architectures might encounter challenges in effectively integrating with newer technologies and meeting the diverse expectations of clients within a rapidly evolving digital environment. Conversely, an enhanced, expedited, and more regulated internet infrastructure is required.

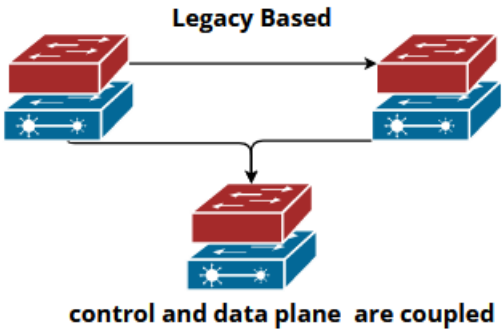

Figure 3 Control and Data Planes coupled

The term "SDN" lacks a universally accepted definition. SDN is presently being executed in the domain of networking via three discrete methods: Open Flow, the SDN API, and virtual machines (VMs), including their utilization in the construction of VXLAN tunnels. "The control plane and data plane are separated in the SDN architecture, network intelligence and state are conceptually centralized, and the applications are abstracted from the underlying network infrastructure."[4].. SDN is a multidimensional concept that defies reduction to a singular entity on account of the fact that its definition differs between individuals. Through the separation of the data plane and control plane, SDNs have increased the networking industry's adaptability. An analysis of prior research concerning legacy-based network design, Software-Defined Networking (SDN), and Cisco Packet Tracer constitutes the literature review section of this paper. The method utilized to migrate from a prior network architecture to the SDN architecture has been described in the methodology section. As the implementation of the enterprise network architecture has progressed, the strategies employed in its development have become more transparent. Prior to this, the paper concludes with a discourse on future undertakings and suggestions arising from the undertaken research.

II. Literature Review

This section contains scholarly works pertaining to the progressions of SDN. The local area network's optimization, the progress made thus far, and the implementation of a Cisco packet tracer. Cisco Packet Tracer is a network simulation application that is one of the best for learning networking through simulation and is compatible with a wide range of Cisco networking devices. WAN architectures, residential networks, smart home networks, enterprise-level architectures, and numerous other types of networks can be created utilizing Cisco Packet Tracer. To date, numerous initiatives have been successfully concluded utilizing Cisco Packet Tracer. The intelligent home concept was conceived and implemented through the use of the Cisco packet tracer application to demonstrate that users can monitor and control the devices connected to their smartphones [5]. Additionally, CPT permits realistic virtualization [6], [7]. The execution of various fundamental principles, such as DHCP, EMAIL, VLAN, and DNS, in addition to the configuration of local area networks that incorporate wireless and tethered devices [8]. Additionally, CPT is a well-known simulation application that is highly functional and user-friendly. Prior to the twenty-first century, computer networks remained unchanged in response to the internet's increasing optimization demands. Due to its applications and implementations, software defined networking is a completely new domain, era, and concept; the term "software defined networking" was first coined in the 2000s, coinciding with its emergence. The fundamental notion embodied by this nomenclature was the segregation of the control and data planes, thereby enabling enhanced programmability and adaptability in the realms of network administration and virtualization. Martin Cos-ado is widely recognized as a trailblazer in the development of the Software Defined Networking paradigm [9]. The concept of partitioning the infrastructure supporting network devices by separating the control and data planes was introduced in the Ethane paper, to elucidate a few points. This groundbreaking development in network architecture significantly influenced the expansion of software-defined networks. Martin continued his efforts by establishing Nicira, a company that specialized in network virtualization, in collaboration with his fellow members. Nicira was dedicated to the innovative concept of virtualizing networks. This organization was subsequently acquired by VMware.In addition to Martin Cos-ado's contributions to the field of

Software-Defined Networking (SDN), scholars have been engaged in the development of novel concepts and the utilization of this interface freedom to introduce a programmable and adaptable era of computer networks. The novel separation of the data and control planes in the SDN architecture has increased the programmability and adaptability of the network. The recent modification has led to a simplification of the network architecture through the implementation of a centralized controller. A software defined architecture consists of three layers, as per one definition of software defined networking (SDN), which posits the separation of the control and data planes: application, control, and data. Moreover, it is equipped with three APIs, namely south-bound, east-west, and north-bound [10].The Open Network Foundation defines SDN as "an architecture in which the control plane and data plane are decoupled, network intelligence and state are logically centralized, and the foundational network infrastructure is abstracted from the applications" [4].

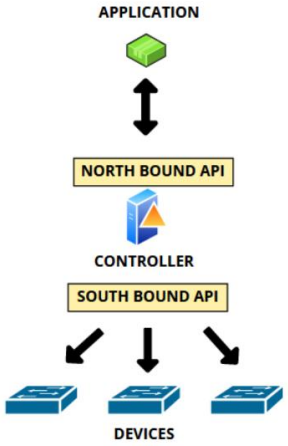

Figure 4 Workflow of Open SDN

To facilitate rapid application development, open SDN is predicated on the configuration of networking devices with an abstraction layer and an open interface. Through an open flow protocol, the abstraction layer/controller communicates with the networking devices. Open Flow furnishes networking devices with an accessible interface. The underlying principle is that the flow table of any networking device is controlled by an open interface, irrespective of the manufacturer or operating system of the device. The SouthBound API is responsible for this. The northBound API provides access through two interfaces: one that utilizes the Java API and the other that employs the Restful API. An application can modify the flow table and flow entries in network devices without establishing direct communication with those devices by utilizing these two APIs. Given that application developers do not engage with hardware components, it is unnecessary for them to possess knowledge regarding the intricacies and specifications of the physical attributes of networking devices. The complex operations of modifying flow tables on switches are managed by the controller, with the application developer establishing communication through a high-level application programming interface (API), which is a Java API in this particular instance. The processing unit of the networking devices has been moved to the center. These controllers transmit the modifications or configurations to the devices. Additionally, it enables an application to regulate the movement of traffic. Following the application's creation, the controller is notified via the REST API of a command that utilizes open flow to update the switches. This directive specifies the route that network switches should take in regard to traffic bearing MAC address A that is en route to MAC address B. This results in the principal centralized controller housing the cognitive resources, as opposed to the devices within an Open Flow SDN that implements open flow. On the switches, the flow table and flow entries are now present. The controller functions fundamentally as an abstraction layer, facilitating communication between application developers and the flow table via the North Bound interface via the Java API. The configuration is subsequently pushed to the switches by the controller. The controller exercises control over these via South Bound APIs. The fundamental progression from the application to the controller to the devices is illustrated in Figure 4. Devices include load balancers, firewalls, switches, and routers, among others. SDN is notified of a packet-in message from the switch (device). Each device within the flood light (SDN controller) is linked to the control channel of the controller via a TCP connection. The primary module of the controller scans continuously for new connections. Applications are able to subscribe to request particular events and provide appropriate responses. Consider, for instance, an application that applies a mandatory rule automatically when a new switch is added. Consequently, we shall presume that this transpires automatically. Therefore, when our application receives the switch added event, it can direct the switch to insert a particular entry that satisfies specified criteria, including action priority and expiration, into the flow table of that particular switch. In situations where a packet is being transmitted to the SDN controller with the explicit intention of affecting it, or when the flow table does not contain a corresponding entry, the SDN controller receives the message as a packet-in message. The primary flood light, the SDN controller module, will be the recipient of the message packet. Notified will be any applications that have an interest in these packet-in events. These applications will subsequently use flow modification to implement or remove rules based on the received packet. Topology discovery is a module within Flood Light. This module requires the exchange of particular control packets between SDN switches in order to operate. The topology discovery module will be notified by a switch that it has obtained one of these discovery events during the exchange of these packets across a link with another switch. Following that, this event will be forwarded to any applications that may exhibit interest in it. Initial connection to the SDN controller occurs during startup via a switch; any applications that are interested in this connection are directed to the SDN controller's central module. Applications may initiate a flow modification command on a switch in reaction to this occurrence, directing it to install a specific entry that specifies criteria for actions, priority, hard, inactive timeout, and match across Ethernet, IP, TCP, and UDP header fields. Furthermore, a link discovery module can be implemented to direct the payload towards an application. Then, flow modifications to add or remove entries from the specified flow table can be issued by this application. Finally, the action to be taken is determined by

comparing the incoming transmission to the entries that are already in the flow table. This operation may involve modifying the header fields of the packet, dropping it, transmitting it to the controller, or opening a specific port. Upon the arrival of a packet that does not correspond to any entry in the flow table, a packet-in event is generated for our application. Flow modification events enable our application to add or remove constraints. By accumulating and analyzing data, generating strategies, and executing those strategies, the control layer is capable of performing a closed loop of intelligent control made possible by the implementation of SDN and artificial intelligence technology. With the assistance of artificial intelligence, the controller possesses the ability to mitigate security threats and vulnerabilities that affect 90% of the network [11]. By isolating the data plane, which regulates traffic routing decisions, from the control plane, SDN implements centralized control and policy enforcement. The data plane is where actual data flows. By enabling policy enforcement and centralized configuration across the entire network, this functionality eliminates the need for configuration specific to individual devices. Through the establishment of network-wide policies that regulate load balancing, security, and traffic prioritization, administrators can effectively maximize resource utilization and improve application performance [12]. Dynamic Traffic Management: The capability to analyze and manipulate traffic in real time is made possible by the programmable nature of SDN. Network controllers possess the capability to dynamically modify routing paths in response to congestion levels, application priorities, and traffic demands. Through the implementation of this particular strategy, bandwidth is allocated in an efficient manner, latency is diminished, and critical applications are guaranteed quality of service (QoS) [13]. Network Automation and Agility: SDN facilitates the automation of network duties by streamlining the configuration, provisioning, and troubleshooting processes through the use of APIs and scripting. By minimizing human error and reducing manual intervention, this approach facilitates swift network adaptations to evolving demands, ultimately resulting in enhanced agility and responsiveness [14]. SDN controllers provide an all-encompassing view of the network, including resource utilization, traffic patterns, and device conditions, thereby enhancing visibility and analytics. The increased visibility provides administrators with the ability to detect bottlenecks, diagnose problems, and make well-informed decisions regarding additional optimization. Furthermore, the gathered data can be utilized to conduct sophisticated network analytics, which can yield valuable insights regarding utilization patterns and facilitate proactive capacity planning [15]. One notable feature of SDN is its ability to seamlessly integrate with virtualization and cloud technologies. This integration empowers the dynamic provisioning and administration of resources within virtual networks. This feature enables the deployment and expansion of networks in response to changing duties and distributed applications, thereby enhancing the efficiency of resource utilization and reducing costs [16].This time-honored technique is effective when dealing with problems that are precisely defined and have linear constraints. The efficiency of network flow optimization is notably high, as demonstrated by the Ford-Fulkerson algorithms [17], which determine the most optimal data transit routes. However, it struggles to effectively handle complex, extensive networks and non-linearity. Heuristics and metaheuristics are methods that guide optimization through repeated development, albeit often lacking precise guarantees. Methods that aim to identify optimal solutions in complex environments employ techniques inspired by thermodynamics, such as Simulated Annealing [18], which traverse the search space by evading local optima. However, they may require modification to address specific concerns, and they lack any theoretical guarantees. The burgeoning domain of artificial intelligence and machine learning offers algorithms, including reinforcement learning and Q-learning, that emulate human decision-making in order to adapt to dynamic network environments [19]. They are exceptionally adept at learning and adjusting in real time for congestion and load balancing using network data. Training these models, nevertheless, requires a substantial quantity of data and processing capacity. Graph optimization techniques involve the application of graph theory principles, including minimal spanning trees and shortest pathways, to enhance network architecture, bandwidth allocation, and resource utilization [20]. Its strength derives from the fact that it efficiently resolves issues involving graph representations that are unique. Conversely, networks that serve intricate functions or possess dynamic topologies may encounter challenges when attempting to implement. Game theory is applied to network resource allocation in order to generate solutions such as Nash equilibrium, which effectively reconciles the interests of the group and the individual. This is achieved by conceptualizing various network actors as strategic participants competing for resources [21]. While extensive modeling is necessary and not all network behaviors may be captured, this approach proves beneficial in the areas of congestion control, spectrum share, and resource allocation. Network Function Virtualization (NFV) and Optimization: By separating network functions from hardware, NFV enables dynamic service provisioning and flexible resource allocation. The existing body of literature explores optimization strategies that aim to efficiently coordinate virtual network functions (VNFs) across the network, while simultaneously considering cost, performance, and reliability [22]. Notwithstanding its strengths, a meticulous evaluation of the compromises between flexibility and efficiency is required. Multi-objective optimization: Networks often strive to achieve conflicting objectives, such as transmission expansion and latency reduction. This field of study seeks solutions that achieve a "good balance" among multiple objectives. In the literature [23], approaches such as weighted sums and Pareto optimization are examined as means to effectively handle these trade-offs. However, it can be challenging to find the proper weights and balance objectives. Distributed Optimization: In large, decentralized networks, centralized control may not be practicable or scalable. Distributed optimization approaches are agnostic in nature; they establish solutions in which network elements collaborate harmoniously to achieve a global objective [24]. The literature investigates protocols and algorithms that facilitate efficient information exchange and decision-making in such contexts. While this feature offers scalability, it complicates the process of decision-making within a distributed system. optimization Inspired by Nature: This rapidly expanding field draws inspiration from natural processes, including genetic algorithms

and ant colony optimization, in order to resolve complex network problems. The literature [25] examines the ways in which they can be applied to resource allocation, routing, and aggregation issues. While these methods exhibit flexibility and adaptability, they may lack robust theoretical guarantees and require careful parameter adjustment.

III. METHODOLOGY

. In this paper, we primarily employed the Cisco Packet Tracer to simulate our concept of a bridge connecting the past of legacy-based network architectures with software-defined networks. A university use case was initially developed so that the design of the local area network architecture could be delineated. In this particular case, the university serves as an illustrative enterprise, boasting a user base of 5000, which is average for a business organization. A comprehensive examination of the use case resulted in the collection of several requirements, encompassing aspects such as network monitoring, remote access, engineering quality, traffic management across multiple university sites, effectiveness, and bandwidth control. Following the collection of essential data, we initiated the process of devising the network architecture, wherein the principal objective was to optimize resource utilization in order to attain superior results and cost efficiency. As previously discussed, the network is thus intimately connected due to the coexistence of the control and data planes. In software-defined networks and within a single device, there are two distinct domains. All of the devices reside in both the data plane of the devices and the control plane of the controller. At this time, we aimed to maintain a position that was distinct from both categories. Due to challenges associated with network availability and device management from the office, we decided to discontinue the utilization of the outdated network architecture. Furthermore, the current state of affairs makes it difficult for software-defined networks to be widely adopted because of the costs that are connected with them and the inadequate technical skills of the general population. We built a network architecture that makes use of the Cisco Network Controller, which is not specific to any particular manufacturer, in order to make remote device management and monitoring easier, as well as to exercise a larger degree of control over the Internet from our end. Not only can the controller be used for routing settings, quality of service rules, access controls, and user edits and deletions, but it can also be applied to all applications. This gives the user with a streamlined technique to control and configure all network devices using a single graphical user interface. This makes the process simple and straightforward. As a result of the changes made to the topology of the network, the network controller has taken on an increasingly important function. Because of this, virtual local area networks (vlans) and intervals were put into place to guarantee the data's authenticity. The router was outfitted with access control lists that were generated and put into operation for both incoming and outgoing traffic. A further accomplishment was the completion of the translation of the network address. As a result of the cost-effectiveness of this strategy, a firewall was incorporated into the router. Through the implementation of the firewall, the security services that were made available to these interfaces were made available in order to safeguard the server farm. In order to foil the malware server's effort to mimic a DHCP server by employing the same interfaces, DHCP snooping was installed on the dynamic host configuration protocol server. This was done in order to prevent the malware server from succeeding. The DHCP debugging feature was also activated so that faults that occurred on the DHCP server could be investigated and fixed. Through the utilization of static routing, the circulation of traffic from the private network to the public network was directed in its entirety. Following the completion of all configurations, the Cisco Network Controller was utilized to carry out procedures for monitoring and controlling the network.

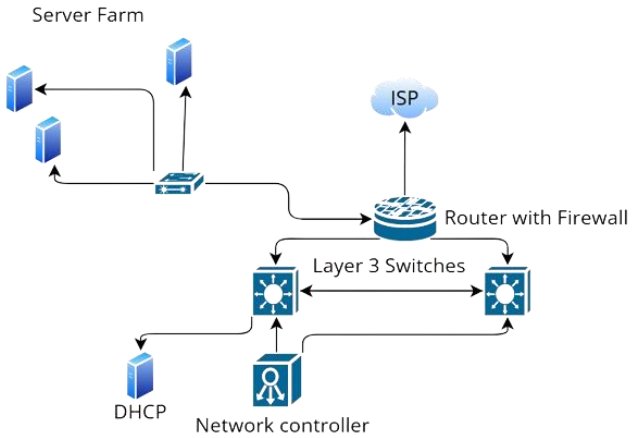

Figure 5 Network Architecture

IV. IMPLEMENTATION

Thus, the architectural design and organization process commenced initially. Figures 5 and 6 illustrate the fundamental level design of our network architecture. After designing, we decided to implement Vlans. Listed below are the vlans that we designed for our network architecture.Vlans, or virtual area networks, facilitate secure communication. Currently, this secure communication operates by impeding the ability of a device or computer belonging to vlan 10 to establish a connection with vlan 20, and vice versa, when vlan 10 and vlan 20 are the two vlans involved. This is the result of two vlans being incapable of exchanging information. The inter-vlan protocol can be employed to establish communication between two vlans. Why is this required, considering that there are three virtual lanes and we occasionally require one of them to connect to another virtual lane? It should be prohibited for the third virtual lane to establish communication with the first two virtual lanes. Thus, the vlan segment not only provides security but also conceptually partitions the network into numerous expansive domains, thereby augmenting network performance, security, and resource efficiency. Inter-Vlan enables regulated communication between two VLANs or, if required, between two VLAN segments that are isolated from one another.

1. • Vlan 10 for VC office
2. • Vlan 20 for Registrar office
3. • Vlan 30 for Accounts
4. • Vlan 40 for Exams

5. • Vlan 50 for DSA and QAE
6. VLAN 390 FOR SEC APS
7. • Vlan 60 for civil and math department
8. • Vlan 70 370 for the whole faculty of HITEC university
9. • Vlan 80 for Sir Syed block
10. • Vlan 90,100,110 for hostel
11. • Vlan 120,130,140,150 for library
12. • Vlan 160,170,180,190,200,210,220 for computer science department
13. • Vlan 230,240,250,260 ,270 ,280 for electrical engineering department
14. • Vlan 290,300,310,320,330, for mechanical engineering department
15. • Vlan 340,350,360 for Rumi block(café and bba department)

Following the establishment of VLANs, the procedure for allocating IP addresses to the various VLANs commences, considering appropriate subnetting and scaling allowances. At present, the external network is utilizing the publicly accessible IP addresses 175.10.0.1-2; these are the sole IP addresses employed for objectives related to the internet. The address range between 172.16.0.0 and 172.16.22.0 is utilized on our private network. The IP address of our incoming interface is 192.168.2.1, while the IP address of our outgoing interface is 175.10.0.2. A distinct VLAN is allocated to each department, and each VLAN is correlated with a dedicated subnet that comprises elements those with growth potential. Scalability is crucial while designing a network architecture. Figures 6, 7, 8, and 9 show departments, VLANs, IP addresses, domains, masks, broadcast IP addresses, useable IP ranges, and wildcard masks. After creating VLANs, IP addresses are assigned to them using subnetting and scaling allowances. The external network, specifically for internet-related tasks, solely uses the public IP numbers 175.10.0.1-2. Our private network uses 172.16.22.0–172.16.0.0 IP addresses. 192.168.2.1 is the inbound interface while 175.10.0.2 is the outbound interface. Each department has its own VLAN and subnet for future growth. Scalability is crucial to creating an appropriate network architecture, hence it must be considered throughout design. Figures 6, 7, 8, and 9 show the departments, VLANs, IP addresses, domains, masks, broadcast IP addresses, useable IP ranges, and wildcard masks. After VLANs are created, IP addresses are assigned to them using subnetting and scaling limitations. Only the public IP addresses 175.10.0.1-2 are used for the external network and internet-facing purposes. In our private network, IP addresses range from 172.16.0.0 to 172.16.22.0. 192.168.2.1 is the inbound interface while 175.10.0.2 is the outbound interface. Each department has its own VLAN and subnet for future growth. Scalability is crucial to an optimal network architecture; hence it must be carefully considered during design. Figures 6, 7, 8, and 9 show the VLANs, IP addresses, domains, masks, departments, broadcast IP addresses, useable IP ranges, and wildcard masks.

Department	VLAN	Hosts Needed	Hosts Available	Unused Hosts	Network Address	Slash	Mask	Usable Range	Broadcast	Wildcard
Computer Science (CS)	vlan 70-cs-faculty	62	62	0	172.16.8.0	/26	255.255.255.192	172.16.8.1 - 172.16.8.62	172.16.8.63	0.0.0.63
Computer Science (CS)	vlan 220-CS apis	254	254	0	172.16.11.0	/24	255.255.255.0	172.16.11.1 - 172.16.11.254	172.16.11.255	0.0.0.255
Computer Science (CS)	vlan 160	62	62	0	172.16.0.0	/26	255.255.255.192	172.16.0.1 - 172.16.0.62	172.16.0.63	0.0.0.63
Computer Science (CS)	vlan 170	62	62	0	172.16.0.64	/26	255.255.255.192	172.16.0.65 - 172.16.0.126	172.16.0.127	0.0.0.63
Computer Science (CS)	vlan 180	62	62	0	172.16.0.128	/26	255.255.255.192	172.16.0.129 - 172.16.0.190	172.16.0.191	0.0.0.63
Computer Science (CS)	vlan 190	62	62	0	172.16.0.192	/26	255.255.255.192	172.16.0.193 - 172.16.0.254	172.16.0.255	0.0.0.63

Figure 6 Descriptive Analysis Table with respect to Departments 1

Computer Science (CS)	vlan 200	62	62	0	172.16.1.0	/26	255.255.255.192	172.16.1.1 - 172.16.1.62	172.16.1.63	0.0.0.63
Computer Science (CS)	vlan 210	62	62	0	172.16.1.64	/26	255.255.255.192	172.16.1.65 - 172.16.1.126	172.16.1.127	0.0.0.63
Electrical Engineering	vlan 70-ee-faculty	62	62	0	172.16.8.64	/26	255.255.255.192	172.16.8.65 - 172.16.8.126	172.16.8.127	0.0.0.63
Electrical Engineering	vlan 240	62	62	0	172.16.1.192	/26	255.255.255.192	172.16.1.193 - 172.16.1.254	172.16.1.255	0.0.0.63
Electrical Engineering	vlan 250	62	62	0	172.16.2.0	/26	255.255.255.192	172.16.1.1 - 172.16.1.62	172.16.2.63	0.0.0.63
Electrical Engineering	vlan 260	62	62	0	172.16.2.64	/26	255.255.255.192	172.16.2.65 - 172.16.2.126	172.16.2.127	0.0.0.63
Electrical Engineering	vlan 270	62	62	0	172.16.2.128	/26	255.255.255.192	172.16.2.129 - 172.16.2.190	172.16.2.191	0.0.0.63
Electrical Engineering	vlan 230	62	62	0	172.16.1.128	/26	255.255.255.192	172.16.1.129 - 172.16.1.190	172.16.1.191	0.0.0.63
Mechanical	vlan 70-me-faculty	30	30	0	172.16.8.128	/27	255.255.255.224	172.16.8.129 - 172.16.8.158	172.16.8.159	0.0.0.31
Mechanical	vlan 330 ME apis	254	254	0	172.16.14.0	/24	255.255.255.0	172.16.14.1 - 172.16.14.254	172.16.14.255	0.0.0.255
Mechanical	vlan 290	62	62	0	172.16.2.192	/26	255.255.255.192	172.16.1.193 - 172.16.1.254	172.16.2.255	0.0.0.63
Mechanical	vlan 300	62	62	0	172.16.3.0	/26	255.255.255.192	172.16.3.1 - 172.16.3.62	172.16.3.63	0.0.0.63
Mechanical	vlan 310	62	62	0	172.16.3.64	/26	255.255.255.192	172.16.3.65 - 172.16.3.126	172.16.3.127	0.0.0.63
Mechanical	vlan 320	62	62	0	172.16.3.128	/26	255.255.255.192	172.16.3.129 - 172.16.3.190	172.16.3.191	0.0.0.63
Café (Both Floors)	vlan 340-cafe	254	254	0	172.16.9.0	/24	255.255.255.0	172.16.9.1 - 172.16.9.254	172.16.9.255	0.0.0.255
Café (Ground Floor)	vlan 350-café(ground	254	254	0	172.16.15.0	/24	255.255.255.0	172.16.15.1 - 172.16.15.254	172.16.15.255	0.0.0.255

Figure 7 Descriptive Analysis Table with respect to Departments 2

Café (First Floor)	vlan 360-café(first)	254	254	0	172.16.18.0	/24	255.255.255.0	172.16.18.1 - 172.16.18.254	172.16.18.255	0.0.0.255
Café (First Floor)	vlan 370-café(first)	254	254	0	172.16.19.0	/24	255.255.255.0	172.16.19.1 - 172.16.19.254	172.16.19.255	0.0.0.255
BBA	vlan 380-bba-std	126	126	0	172.16.5.0	/27	255.255.255.128	172.16.5.1 - 172.16.5.126	172.16.5.127	0.0.0.127
BBA	vlan 70-bba-faculty	30	30	0	172.16.8.224	/27	255.255.255.224	172.16.8.225 - 172.16.8.254	172.16.8.255	0.0.0.31
Sir Syed Block	vlan 70-sir-syed-faculty	30	30	0	172.16.8.192	/27	255.255.255.224	172.16.8.193 - 172.16.8.222	172.16.8.223	0.0.0.31
Hostel	vlan 90-hostel	126	126	0	172.16.5.128	/25	255.255.255.128	172.16.5.129 - 172.16.5.254	172.16.5.255	0.0.0.127
Hostel	vlan 100-hostel	254	254	0	172.16.12.0	/24	255.255.255.0	172.16.12.1 - 172.16.12.254	172.16.12.255	0.0.0.255
Hostel	vlan 110-hostel	254	254	0	172.16.13.0	/24	255.255.255.0	172.16.13.1 - 172.16.13.254	172.16.13.255	0.0.0.255
Library	vlan 120-library	126	126	0	172.16.6.0	/25	255.255.255.128	172.16.6.1 - 172.16.6.126	172.16.6.127	0.0.0.127
Library	vlan 130-library	126	126	0	172.16.6.128	/25	255.255.255.128	172.16.6.129 - 172.16.6.254	172.16.6.255	0.0.0.127
Library	vlan 140-library	126	126	0	172.16.7.0	/25	255.255.255.128	172.16.7.1 - 172.16.7.126	172.16.7.255	0.0.0.127
Library	vlan 150-library	126	126	0	172.16.7.128	/25	255.255.255.128	172.16.7.129 - 172.16.7.254	172.16.7.255	0.0.0.127
Secretariat	vlan 20	62	62	0	172.16.3.192	/26	255.255.255.192	172.16.3.193 - 172.16.3.254	172.16.3.255	0.0.0.63
Secretariat	vlan 50-dsa-qae	62	62	0	172.16.4.128	/26	255.255.255.192	172.16.4.129 - 172.16.4.190	172.16.4.191	0.0.0.63
Secretariat	vlan 40-exams	62	62	0	172.16.4.64	/26	255.255.255.192	172.16.4.65 - 172.16.4.126	172.16.4.127	0.0.0.63
Secretariat	vlan 30-accounts	62	62	0	172.16.4.0	/26	255.255.255.192	172.16.4.1 - 172.16.4.62	172.16.4.63	0.0.0.63

Figure 8 Descriptive Analysis Table with respect to Departments 3

Secretariat	vlan 10-vc	30	30	0	172.16.4.192	/27	255.255.255.224	172.16.4.193 - 172.16.4.222	172.16.4.223	0.0.0.31
Civil and Math's	Vlan faculty-70-civil-maths	30	30	0	172.16.8.160	/27	255.255.255.224	172.16.8.161 - 172.16.8.190	172.16.8.191	0.0.0.31
Civil and Math's	vlan 60	254	254	0	172.16.16.0	/24	255.255.255.0	172.16.16.1 - 172.16.16.254	172.16.16.255	0.0.0.255

Figure 9 Descriptive Analysis Table with respect to Departments 4

Following that, was entered the subsequent configuration phase, during which we trunked the required interfaces subsequent to allocating the corresponding VLANs and IP addresses to the designated blocks and devices. After configuring the vlans and selecting the IP addresses, this was executed. To facilitate communication between vlans, the intervlan protocol was also implemented on the layer 3 switch to enable intervlan communication. Only the authorized VLAN segments will engage in communication. We utilized a deployed server for DHCP services prior to configuring a pool in the DHCP server in accordance with the Vlan configuration. We created an individual pool for each vlan on that server.

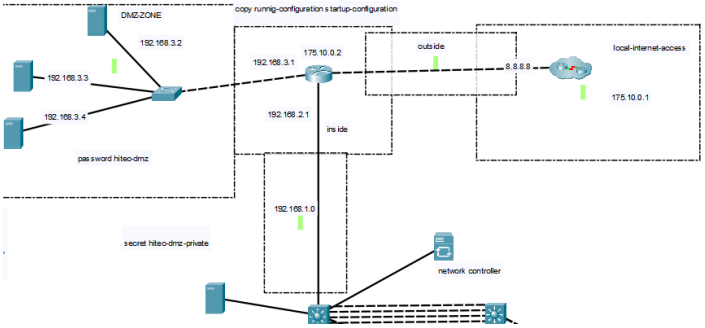

Figure 10 Network Architecture on CISCO Simulation Tool

The created DHCP pools are illustrated in figures 11, 12, 13, and 14. The servers' IP addresses have been configured as the auxiliary IP addresses for each VLAN subsequent to the creation of pools. This enables them to utilize the DHCP server for dynamic host configuration. We created a file transfer server, a DNS server, and a mail server subsequent to establishing the DNS. The hosting server for the university's website is the DNS.

Pool Name	Default Gateway	DNS Server	Start IP Address	Subnet Mask	Max User	TFTP Server	WLC Addres
vlan 170	172.16.1.65	192.168.3.2	172.16.1.65	255.255.2...	62	0.0.0.0	0.0.0.0
vlan220	172.16.3.1	192.168.3.2	172.16.3.1	255.255.2...	255	0.0.0.0	0.0.0.0
vlan380	172.16.22.1	192.168.3.2	172.16.22.1	255.255.2...	255	0.0.0.0	0.0.0.0
vlan80	172.16.21.1	192.168.3.2	172.16.21.1	255.255.2...	255	0.0.0.0	0.0.0.0
vlan150	172.16.19.1	192.168.3.2	172.16.19.1	255.255.2...	127	0.0.0.0	0.0.0.0
vlan140	172.16.18....	192.168.3.2	172.16.18....	255.255.2...	127	0.0.0.0	0.0.0.0
vlan130	172.16.18.1	192.168.3.2	172.16.18.1	255.255.2...	127	0.0.0.0	0.0.0.0
vlan120	172.16.17....	192.168.3.2	172.16.17....	255.255.2...	127	0.0.0.0	0.0.0.0
vlan90	172.16.17.1	192.168.3.2	172.16.17.1	255.255.2	127	0.0.0.0	0.0.0.0

Figure 11 IP Pools on DHCP Server According to their Respective VLANSR

Pool Name	Default Gateway	DNS Server	Start IP Address	Subnet Mask	Max User	TFTP Server	WLC Addres
vlan90	172.16.17.1	192.168.3.2	172.16.17.1	255.255.2...	127	0.0.0.0	0.0.0.0
vlan390	172.16.16.1	192.168.3.2	172.16.16.1	255.255.2...	255	0.0.0.0	0.0.0.0
vlan60	172.16.15.1	192.168.3.2	172.16.15.1	255.255.2...	255	0.0.0.0	0.0.0.0
vlan110	172.16.13.1	192.168.3.2	172.16.13.1	255.255.2...	255	0.0.0.0	0.0.0.0
vlan100	172.16.12.1	192.168.3.2	172.16.12.1	255.255.2...	255	0.0.0.0	0.0.0.0
vlan50	172.16.11.65	192.168.3.2	172.16.11.65	255.255.2...	63	0.0.0.0	0.0.0.0
vlan30	172.16.11.1	192.168.3.2	172.16.11.1	255.255.2...	63	0.0.0.0	0.0.0.0
vlan350	172.16.9.1	192.168.3.2	172.16.9.1	255.255.2...	255	0.0.0.0	0.0.0.0
vlan340	172.16.8.1	192.168.3.2	172.16.8.1	255.255.2	255	0.0.0.0	0.0.0.0

Figure 12 IP Pools on DHCP Server According to their Respective VLANs

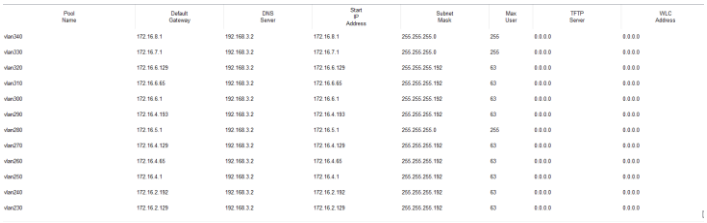

Figure 13 IP Pools on DHCP Server According to Their Respective VLANs

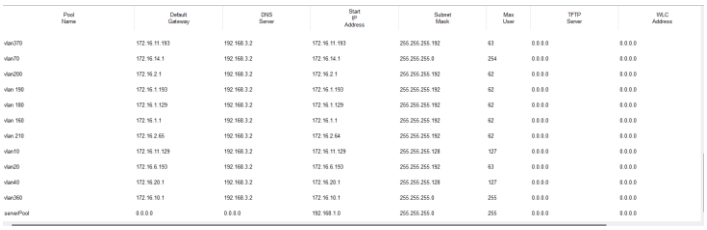

Figure 14 IP Pools on DHCP Server According to Their Respective VLANs

Configurations of the servers are shown in figure 15,16,17,18 and 19. for dhcp server the DHCP snooping is also enabled for security purposes so that no one else can become a dhcp server and only the real server configured by the admin is the real dhcp server.

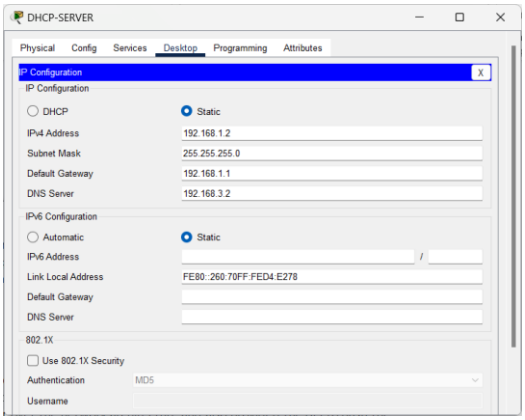

Figure 15 DHCP Server's IP Configurations

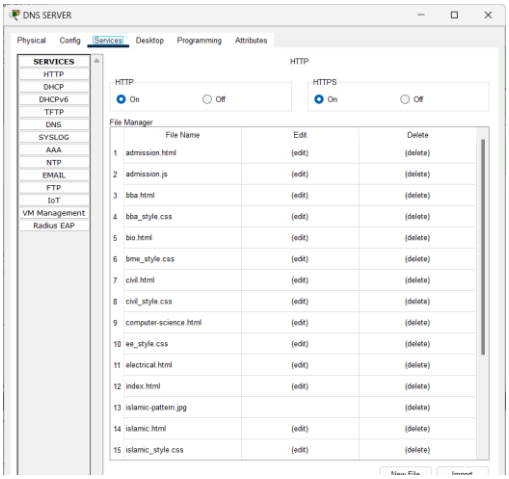

Figure 16 Website Pages and Their HTML and CSS Files

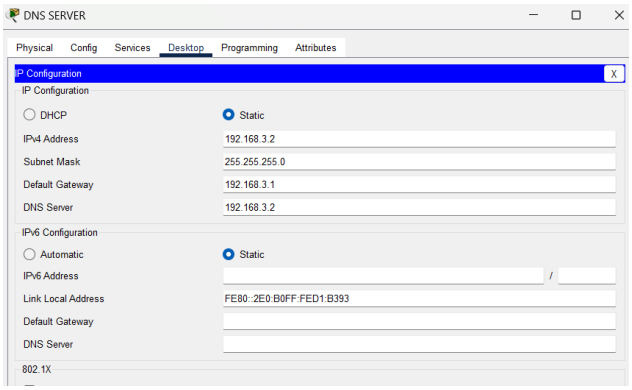

Figure 17 Domain Name Server's IP Configuration

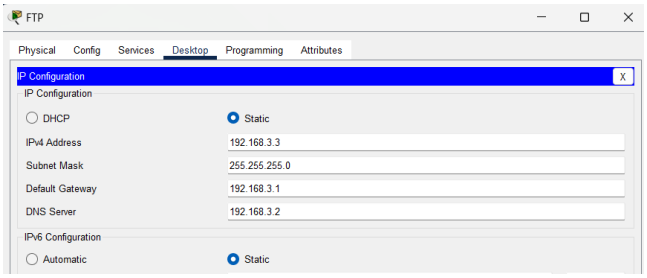

Figure 18 FTP Server's IP Configuration

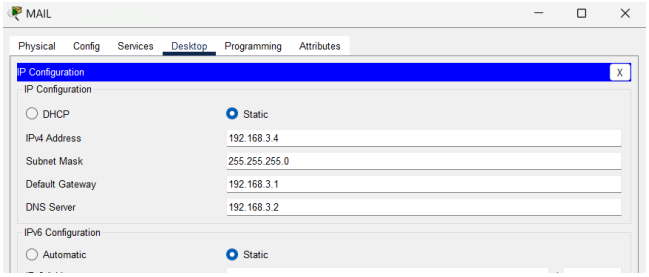

Figure 19 Mail Server's IP Configuration

Once the servers were configured, routers were established to allow private IP addresses to establish connections to both the router and the internet. The routes of the IP addresses leading to the router are illustrated in Figure 16.

```
Gateway of last resort is 175.10.0.1 to network 0.0.0.0

     172.16.0.0/24 is subnetted, 22 subnets
S       172.16.1.0/24 [1/0] via 192.168.2.2
S       172.16.2.0/24 [1/0] via 192.168.2.2
S       172.16.3.0/24 [1/0] via 192.168.2.2
S       172.16.4.0/24 [1/0] via 192.168.2.2
S       172.16.5.0/24 [1/0] via 192.168.2.2
S       172.16.6.0/24 [1/0] via 192.168.2.2
S       172.16.7.0/24 [1/0] via 192.168.2.2
S       172.16.8.0/24 [1/0] via 192.168.2.2
S       172.16.9.0/24 [1/0] via 192.168.2.2
S       172.16.10.0/24 [1/0] via 192.168.2.2
                       [1/0] via 192.168.3.2
S       172.16.11.0/24 [1/0] via 192.168.2.2
S       172.16.12.0/24 [1/0] via 192.168.2.2
S       172.16.13.0/24 [1/0] via 192.168.2.2
S       172.16.14.0/24 [1/0] via 192.168.2.2
S       172.16.15.0/24 [1/0] via 192.168.2.2
S       172.16.16.0/24 [1/0] via 192.168.2.2
S       172.16.17.0/24 [1/0] via 192.168.2.2
S       172.16.18.0/24 [1/0] via 192.168.2.2
S       172.16.19.0/24 [1/0] via 192.168.2.2
S       172.16.20.0/24 [1/0] via 192.168.2.2
S       172.16.21.0/24 [1/0] via 192.168.2.2
S       172.16.22.0/24 [1/0] via 192.168.2.2
     175.10.0.0/16 is variably subnetted, 2 subnets, 2 masks
C       175.10.0.0/24 is directly connected, GigabitEthernet0/0
L       175.10.0.2/32 is directly connected, GigabitEthernet0/0
     192.168.2.0/24 is variably subnetted, 2 subnets, 2 masks
C       192.168.2.0/24 is directly connected, GigabitEthernet0/1
L       192.168.2.1/32 is directly connected, GigabitEthernet0/1
     192.168.3.0/24 is variably subnetted, 2 subnets, 2 masks
C       192.168.3.0/24 is directly connected, GigabitEthernet0/2
L       192.168.3.1/32 is directly connected, GigabitEthernet0/2
S*   0.0.0.0/0 [1/0] via 175.10.0.1
```

Figure 20 IP Routes for the Whole Network

As soon as this was completed, network address translation was implemented. Figs. 21, 22, and 23 depict the access control and national control lists, respectively.

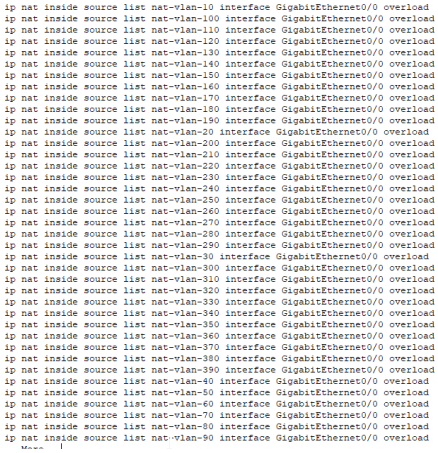

Figure 21 Access Lists for the Whole Network

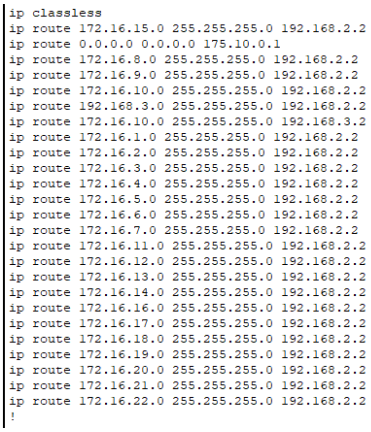

Figure 22 Static Routes for Network

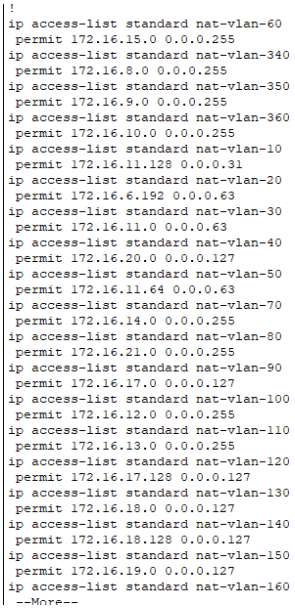

Figure 23 Access Lists on the Respective VLANs

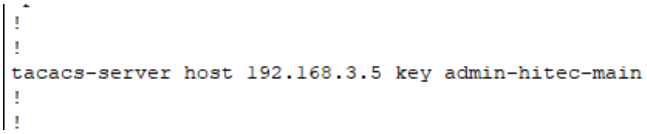

Figure 24 TACACS Server Configuration for Accessing Network Device More Securely

Once everything was in place, we established a Google server and connected the clustered wide area network to the internet after completing the necessary configurations. Once the network has been fully configured, SSH will be installed on each device. It establish a password and enable it for each device before generating credentials. Figure 25 illustrates the managed switches and hosts that are under the control of the controller, as depicted in Figure 28.

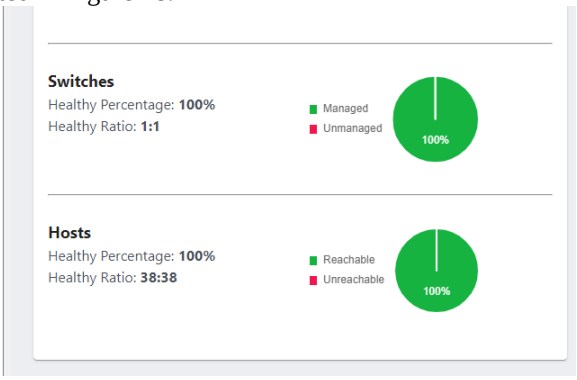

Figure 25 Switch and Costs Management and Reachability

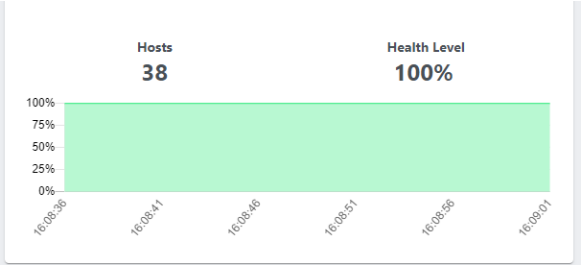

Figure 26 Hosts and Healthy Level

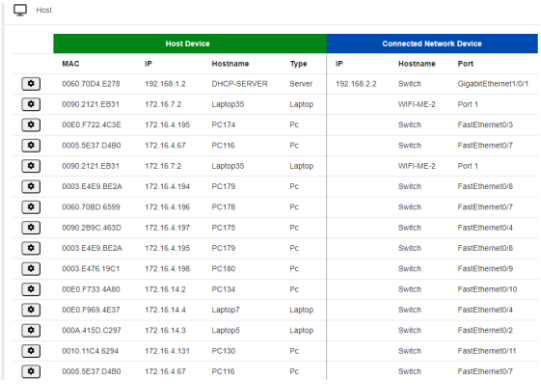

Figure 27 Host Devices and Their Connected Devices

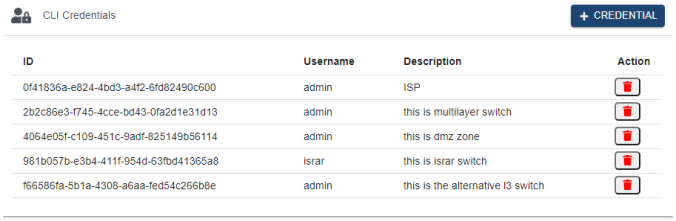

Figure 28 Credential Table to Access the Network Devices from Controller

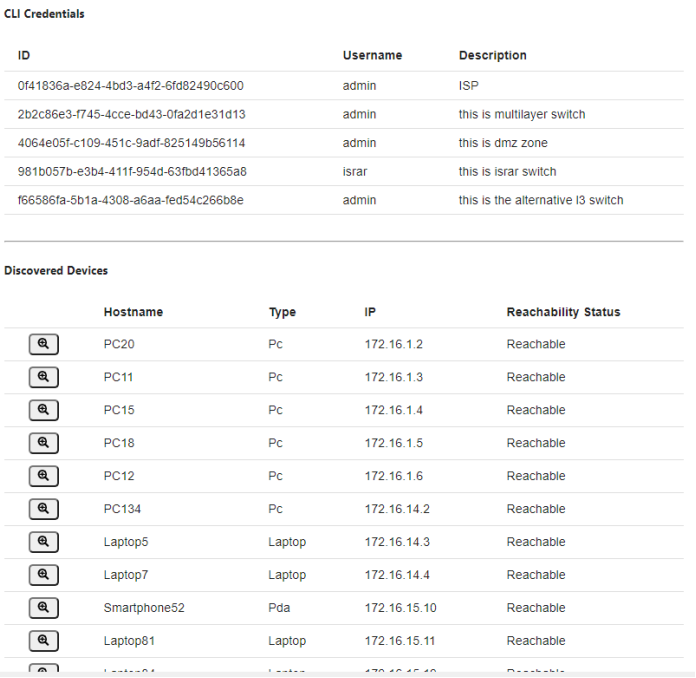

Figure 29 Hosts Connected and CLI Credentials with ID and Username Shown on the Network Controller

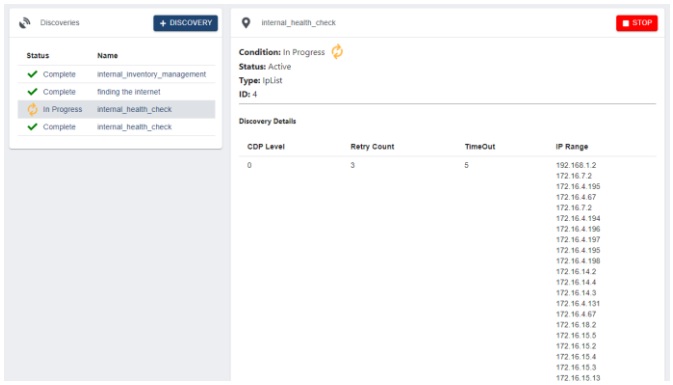

Figure 30 Internal Health Check and IP Range

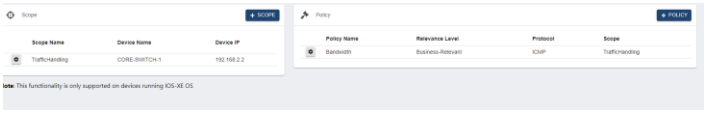

Figure 31 Policies for Traffic Handling at Core Switch

The network controller is now used for all administrator management; this is a good move for the optimized local area network design as it replaces the SDN controller with a Cisco network controller.

V. CONCLUSION

In conclusion, while not entirely transitioning to software-defined networks, the concept of software-defined networks can still be utilized. Users and network administrators now have greater flexibility and control over the network architecture due to this method. Furthermore, configurations can be pushed from the controller to network devices lower in the topology via an API. The utilization of API-driven automation and centralized control improves the agility and efficiency of a network, enabling swift adjustments to ever-changing business requirements and guaranteeing streamlined configuration management. The QoS policies were also implemented; instead of necessitating intricate configurations, we utilized the controller to implement said policies on the network architecture. Ultimately, enterprise-level local area networks ran more smoothly and effectively as a result. Another software-defined network concept that may be implemented in the future to optimize architecture and utilize various software applications with more realistic simulation capabilities for network performance is virtualization. Additionally, a variety of backup tools can be utilized to provide a long-term safety net in the event that server-level or virtualization-related data loss occurs.

REFERENCES

[1] Hauben, M. (2007). History of ARPANET. Site de l'Instituto Superior de Engenharia do Porto, 17, 1-20.

[2] DataReportal. (Year, Month Day). Global Digital Overview [Online]. Available: https://datareportal.com/global-digital-overview

[3] Unraveling Pakistan's Network Landscape -Legacy Structures vs. SDN Paradigms in the Internet Age in IoT Architecture. Available from: https://www.researchgate.net/publication/377382337_Unraveling_Pakistan's_Network_Landscape_-Legacy_Structures_vs_SDN_Paradigms_in_the_Internet_Age_in_IoT_Architecture [accessed Jan 31 2024]

[4] Sezer, S., Scott-Hayward, S., Chouhan, P. K., Fraser, B., Lake, D., Finnegan, J., ... & Rao, N. (2013). Are we ready for SDN? Implementation challenges for software-defined networks. IEEE Communications magazine, 51(7), 36-43

[5] Alfarsi, G., Jabbar, J., Tawafak, R. M., Malik, S. I., Alsidiri, A., & Alsinani, M. (2019). Using Cisco Packet Tracer to simulate smart home. International Journal of Engineering Research & Technology (IJERT), 8(12), 670-674.

[6] Shemsi, I., Uramová, J., Segec, P., & Kontšek, M. (2017). Boosting campus network design using cisco packet tracer. International Journal of Innovative Science and research Technology, 2(11), 43-54.

[7] Nagendram, S., & Rao, K. R. H. (2019). Hybrid security and energy aware routing for wireless ad hoc networks. International Journal of Recent Technology and Engineering, 8(2), 5594-5597.

[8] Tarkaa, N. S., Iannah, P. I., & Iber, I. T. (2017). Design and simulation of local area network using cisco packet tracer. The International Journal of Engineering and Science, 6(10), 63-77.

[9] M. Casado, M. J. Freedman, J. Pettit, J. Luo, N. McKeown, and S. Shenker, "Ethane: Taking Control of the Enterprise," in Proceedings of the 2007 conference on Applications, technologies, architectures, and protocols for computer communications (SIGCOMM), 2007.

[10] Ahmed, D., Dillshad, V., Danish, A. S., Jahangir, F., Kashif, H.,&Shahbaz,T. (n.d.). Enhancing Home Automation through Brain-ComputerInterfaceTechnology. http://xisdxjxsu.asia.

[11] Guo, A., & Yuan, C. (2021). Network intelligent control and traffic optimization based on SDN and artificial intelligence. Electronics, 10(6), 700.

[12] Shridevi Murthy, P., Ramaiah, K. V., & Govindarajan, L. (2015). A survey of software defined networking: Future directions and open research issues. Journal of Network and Computer Applications, 55, 30-47.

[13] Hu, Y., Wang, W., Lou, W., & Fang, Y. (2014). Dynamic traffic management with SDN in cloud data centers. Computer Networks, 72, 36-48.

[14] Kreutz, D., Osterlund, F., Raszukiewicz, M., & Nilsson, S. (2015). Software-defined networking: A comprehensive survey. IEEE Communications Surveys & Tutorials, 17(4), 1493-1530.

[15] A. S. Samad Danish et al., "Implementation of Smart Aquarium SystemSupporting Remote Monitoring and Controlling of FunctionsusingInternetof Things," Journal of Multidisciplinary Approaches inScience(JMAS),2019.

[16] Gudipati, A., Perry, J., & Puthalpattu, S. (2013). Software-defined networking (SDN): Applications to cloud computing. IEEE Network, 27(2), 26-32.

[17] Ford, L. R., & Fulkerson, D. R. (1956). Maximum flow through a network. Canadian Journal of Mathematics, 8(4), 399-404.

[18] Kirkpatrick, S., Gelatt, C. D., & Vecchi, M. P. (1983). Optimization by simulated annealing. Science, 220(4598), 671-680.

[19] Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A., Greg Brockman, J., Wainwright, M. A., & Hassabis, D. (2015). Human-level control through deep reinforcement learning. Nature, 518(7540), 529-533.

[20] Ahuja, R. K., Magnanti, T. L., & Orlin, J. B. (1993). Network flows: theory, algorithms, and applications. Prentice-Hall, Inc.

[21] Non-cooperative resource allocation in wireless networks: Axiomatic foundations and computational models" by S. Nash and M. Manjunath (IEEE Journal on Selected Areas in Communications, 2006)

[22] Orchestrating the cloud: Automatic deployment and management of network services" by M. Yannuzzi et al. (IEEE Communications Magazine, 2017)

[23] Multi-objective optimization in network design and management" by S. Srinivasan et al. (IEEE Communications Magazine, 2004)

[24] Distributed optimization and learning for wireless networks" by S. Boyd et al. (Foundations and Trends in Networking, 2012)

[25] A survey of nature-inspired algorithms for network optimization" by M. Dorigo et al. (IEEE Communications Surveys & Tutorials, 2011)

AUTHORS

First Author – Agha Muhammad Yar Khan, Student, HITEC University Taxila.
Second Author – Saima Shaheen, Assistant Professor, HITEC University Taxila.
Third Author – Abdul Samad Danish, Lecturer, HITEC University Taxila.
Forth Author – Umul Warah, Student, NUST Islamabad.
Fifth Author – Onib -Ur- Rehman, Student, HITEC University Taxila.
Sixth Author – Muhammad Faizan Hassan, Student, HITEC University Taxila.

Correspondence Author – Abdul Samad Danish,