BINDURA UNIVERSITY OF SCIENCE EDUCATION

FACULTY OF SCIENCE

DEPARTMENT OF COMPUTER SCIENCE

FULLNAME Linner Kanganga

REG NUMBER B200075A

COURSE CODE CS 400

SUPERVISOR Mr Hove

PROGRAM NAME

BACHELOR OF SCIENCE HONOURS DEGREE IN INFORMATION
TECHNOLOGY

RESEARCH PROJECT TITLE

Implementation of network function virtualisation (nfv) resource Management for low network latency

A RESEARCH PROJECT SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE BACHELOR OF SCIENCE HONOURS DEGREE IN INFORMATION TECHNOLOGY

APPROVAL FORM

The undersigned certify that they have supervised the student Linner Kanganga's dissertation entitled, "Implementation of network function virtualisation (nfv) resource Management for low network latency" submitted in partial fulfillment of the requirements for a Bachelor of Computer Science Honors Degree at Bindura University of Science Education.

STUDENT: **DATE:**

LKang

………………………………………………… ………30/09/2024…………………..

SUPERVISOR: **DATE:**

………………………………………………… …..30/09/2024……………………….

CHAIRPERSON: **DATE:**

3

………………………………………………… ………..30/09/2024………………….

EXTERNAL EXAMINER: **DATE:**

………………………………………………… …………………………….

DEDICATION

This work is dedicated to my parents, whose constant encouragement and support have been invaluable throughout this throughout this expedition. Their faith in my abilities has consistently been a constant source of inspiration and motivation.

ACKNOWLEDGEMENTS

I want to express my sincerest appreciation to my supervisor, Mr. D. Hove, for his leadership, patience, and invaluable insights during the entire course of this research. It has been an honor to be one of his BScIT students. He has consistently provided wisdom and inspiration, especially during times of doubt and difficulty. Mr. Hove has taught me how to conduct thorough research and tackle both academic and life challenges. His expert supervision and perceptive insights have guided me to become a qualified researcher. I greatly thank all his time contributions and dedication to make my BScIT feel fruitful and engaging. I am appreciative regarding the model of excellence he has set as a researcher, supervisor, and instructor through invaluable discussions, advice, ideas, and feedback. This thesis would not have been possible to complete without his support.

ABSTRACT

The evolution of network technologies demands more flexible, efficient, and scalable solutions. Network Function Virtualisation (NFV) addresses this need by hardware-based network functions, allowing them to function as software on standard computing infrastructure. As this study focuses on NFV resource management strategies to minimize network latency, a crucial performance metric. It develops a comprehensive NFV resource management framework, incorporating dynamic resource allocation, load balancing, and real-time optimization. Key contributions include machine learning models for predictive resource provisioning, and a scalable monitoring system for real-time performance metrics. Simulations and real-world testing show that our proposed solutions can reduce network latency compared to traditional methods. These findings demonstrate the potential of advanced NFV resource management strategies to improve network performance and efficiency, leading to more responsive and resilient infrastructures. This work provides practical insights and tools for network engineers and operators aiming to optimize their systems for low-latency performance, paving the way for agile, high-performing network environments that meet the demands of modern applications.

Table of Contents

CHAPTER 1: PROBLEM IDENTIFICATION

1.1 INTRODUCTION

Network Function Virtualization is a new network design that places virtualized services in cloud data centers on demand, increasing network flexibility and agility for operators (CDCs) (Demeester, Kelen and Wiley,2015). Network function virtualization environment controls how Virtual Machines (VMs) are assigned virtual network functions in an effective manner and how to reduce network latency in the quickly evolving network environments Network latency is the amount of time it takes for a packet to move from one point inside the network to another (Chen, Zhou and Li, 2018). Low network latency is crucial for real-time applications which include video streaming, gaming, and online transactions.

Network Function Virtualization (NFV) resource allocation optimization is the practice of assigning resources to virtual network activities in a way that minimizes network latency while satisfying the resource requirements of all virtualized network functions. (Liu, Li, Zhou and Wang, 2018). It is necessary to take into account a number of factors when allocating resources to virtual network functions in order to meet performance and latency requirements. These factors include determining the resource requirements of individual virtual network functions, analysing the topology of the network, and assessing traffic patterns between virtual network functions.

Network function virtualization resource management for low network latency can be achieved through a variety of strategies, one of which is traffic engineering. This approach involves directing traffic between virtual network functions in a way that minimizes latency. This can be done using techniques such as load balancing and routing. Another approach is resource scaling which involves scaling the resources allocated to virtual network functions based on their workload. This can be done dynamically to ensure that virtual network functions always have the resources they need to meet their performance requirements.

Software-defined networking can be used to dynamically reconfigure the network and distribute resources to virtual networks in order to achieve low network latency through network function virtualization resource management function to meet the required performance and latency requirements (Zhang, Wang and Hou, 2022). To tackle the issue of virtual network function placement, it is desirable to develop a more extensive model that is based on real-world measurements. This model should accurately capture the network latency between virtual network functions, with a higher level of detail, to maximize where these virtual network operations are located in cloud data centers. In addition, the capacity of virtual machines, the latency of the network between various network components, and the resource requirements of virtual network services must all be considered.

1.2 BACKGROUND TO THE STUDY

Software defined networks (SDNs), mobile edge computing (MEC), and network function virtualization (NFV) are emerging as critical technologies to meet customers increasing demands in 5G and future wireless networks (Cao, Zhang, and Wang 2019). While network function virtualization allows for the flexible and on-demand provisioning and placement of virtual network functions that can be executed within the various locations of a distributed system, software defined networks offer a clean separation of the control plane from the data plane.

Through shifting processing, networking, and storage functions from the central cloud to the mobile edge network, mobile edge computing technology aims to lower resource requirements and user experience delays. Users can place specified contents or virtual network services on cloud data centers by establishing cloud data centers with processing, storage, and networking operations on base stations. Delays can be minimized and distributed resources can be managed more effectively overall in this fashion. Virtual network functions must be provided with enough resources on edge servers to enable the benefits of SDN, NFV, and MEC without compromising network quality of service (QoS).

In order to meet the various application needs and optimize the management of heterogeneous resources like network, compute, and storage, network function virtualization provides flexible and on-demand provisioning and placement of virtual network functions (Gupta and McKeown, 2022). A network function virtualization orchestrator is in charge of managing application and network functionalities as virtual network functions. They are carried out across the different nodes in a distributed system, such as multi-access edge computing nodes and NFV enabled nodes.

Virtual networks essentially overlay on top of the physical networks. They use the infrastructure of the physical network including the network links and protocols to provide connectivity for the virtualized resources. The virtual network also adds an additional layer of abstraction and virtualization that can be used to control and manage the resources. This allows for flexibility and customization of the network as well as increased security and isolation of the virtual resources. In order to retain optimal performance during active operation, virtual networks must be moved to the physical network since they have the potential to appear and disappear over time, just like underlying networks may undergo their own dynamic changes.

1.3 STATEMENT OF THE PROBLEM

Network Function Virtualization (NFV) has become a paradigm-shifting technology in modern network topologies, providing scalability and flexibility by separating network services from dedicated hardware. However, as NFV deployments grow in complexity, ensuring low network latency becomes a critical challenge. Efficient resource management is essential to guarantee optimal performance, but existing solutions often fall short in dynamically adapting to real-time demands. Therefore, the problem at hand is to address NFV resource management to minimize network latency in dynamic and diverse network environments.

1.4 ESEARCH OBJECTIVES

1. Design intelligent algorithm capable of dynamically allocating resources in real-time within NFV environments, considering the dynamic variations in workload.
2. Implement an intelligent system capable of dynamically allocating resources in real-time within NFV environments
3. Evaluate and validate the accuracy and reliability of the intelligent system on real-world trials.

1.4.1 RESEARCH PURPOSE

This study's main goal is to create and implement effective and adaptive resource management strategies within Network Function Virtualization (NFV) environments to minimize network latency. The study seeks to address the dynamic and diverse nature of NFV deployments by designing intelligent algorithms and optimization techniques that ensure low latency, real-time adaptability, and efficient resource utilization. Additionally, the research aims to balance the trade-off between low latency and energy efficiency, promoting scalability and addressing security considerations in the context of dynamic resource allocation within NFV infrastructures. Through the achievement of these objectives, the research aims to contribute to the establishment of a resilient and high-performing NFV ecosystem capable of meeting the stringent latency requirements of diverse applications and services.

1.5 RESEARCH QUESTIONS

1. What are the tools to be used by the author on designing an algorithm?
2. How the author is going to implement the intelligent system?

3. What are the metrics to be applied and used to evaluate the algorithm?

1.6 HYPOTHESIS FOR RESEARCH

H0: There is no statistically significant improvement in minimizing network latency through the application of advanced resource management strategies in Network Function Virtualization (NFV) environments.

H1: The implementation of intelligent resource management strategies within NFV environments significantly reduces network latency, addressing dynamic workload variations and ensuring optimal performance for diverse virtualized network functions (VNFs).

1.7 RESEARCH JUSTIFICATION

This research is justified by the increasing significance of Network Function Virtualization (NFV) in modern networking, where efficient resource management is essential for optimal performance. The dynamic nature of NFV workloads and the demand for low network latency, especially in real-time applications, highlight the need for innovative resource management solutions. Current approaches often struggle to adapt to dynamic workloads, and there is a gap in comprehensive solutions that balance latency, energy efficiency, and scalability. Addressing these challenges is critical for the sustainable deployment of NFV, with potential applications across telecommunications, cloud computing, and emerging technologies like 5G. Furthermore, considering the security implications of dynamic resource allocation is essential for ensuring the integrity and security of NFV infrastructures. Ultimately, this research aims to contribute valuable insights and solutions to enhance the efficiency, scalability, and security of NFV deployments.

1.8 ASSUMPTIONS

It is assumed that the underlying hardware infrastructure, including servers, storage, and networking components, is capable of supporting the high-performance requirements of NFV. It is also assumed that the NFV orchestration and management platforms are robust and efficient, ensuring seamless and dynamic resource allocation. Additionally, it is assumed that the virtualization layer that is implemented using technologies like hypervisors or container orchestration platforms, introduces minimal overhead. Furthermore, it is assumed that there is effective monitoring and management of the NFV environment, ensuring optimal performance and quick identification and resolution of any issues.

1.9 RESEARCH LIMITATIONS

There are several limitations that must be acknowledged in this research. Generalization challenges may arise due to the specific nature of the studied NFV environment, potentially limiting the applicability of findings to different configurations. The dynamic nature of NFV ecosystems poses difficulties in accurately capturing real-time variations, potentially impacting the study's ability to model dynamic workloads effectively. Resource constraints, potential security vulnerabilities, and assumptions about homogeneous network elements may further limit the study's scope. Operational realism and reliance on simulation environments introduce complexities, and the study's temporal constraints may restrict in-depth longitudinal analysis. Despite these limitations, the research aims to provide valuable insights, recognizing the need for further exploration in this dynamic and evolving field.

1.10 SCOPE /DELIMITATION OF THE RESEARCH

This study's focus is on a number of crucial facets of deploying resource management for network function virtualization with the goal of minimizing network latency. The research will investigate the necessary hardware infrastructure, such as servers, storage, and networking components, required to support NFV. This involves evaluating the performance capabilities and requirements of the hardware to ensure it can meet the high-performance demands of NFV environments. Additionally, the research will assess the efficiency and robustness of NFV orchestration and management platforms, focusing on their roles in dynamic resource allocation and overall network management, which are critical for maintaining low latency. The study will also look into the virtualization layer, examining technologies like hypervisors and container orchestration platforms, and will explore techniques and tools designed to minimize virtualization overhead to enhance performance.

The delimitations of this research set clear boundaries and constraints for the study geographically and technologically. This research will concentrate on widely-used NFV technologies and platforms common in current industry practices. The focus will be on use cases that prioritize low network latency, such as real-time applications and services, while other benefits like cost reduction or operational flexibility will be secondary considerations. The research will take into account the current state of NFV technologies and practices, with an emphasis on developments and advancements over the past few years. The research intends to give a thorough and in-depth examination of NFV resource management for achieving low network latency by defining this scope and delimitations, taking into account the particular restrictions and boundaries that form the study.

1.11 CONCLUSION

In conclusion, this research has aimed to address critical challenges in the dynamic landscape of NFV environments. Acknowledging the limitations and complexities inherent in this research, it has endeavored to contribute valuable insights to the field. The study's focus on developing intelligent resource management strategies within NFV environments is grounded in the growing significance of NFV in modern networking and the pressing need for efficient resource utilization. By striving to strike a balance between low network latency, energy efficiency, and scalability, this research aspires to provide practical solutions for the sustainable deployment of NFV. The outcomes of this study are anticipated to have implications across various sectors, fostering responsible and resilient networking technologies. As the research advances to subsequent phases, including the empirical evaluation of proposed strategies, it is poised to contribute substantively to the ongoing discourse on NFV resource management and its impact on network latency

1.12 DEFINITION OF TERMS

SDN - Software-Defined Networking

NF - Network Function

NFV - Network Functions Virtualization

VNF - Virtualized Network Function

NAT - Network Address Translation

DPI - Deep Packet Inspection

DHCP - Dynamic Host Configuration Protocol

IPS - Intrusion Prevention System

NG-FW - Next-Generation Firewall

WOC - WAN Optimization Controller

VNE - Virtual Network Element

VNF-FGE - Virtualized Network Function Forwarding Graph Element

OPEX - Operating Expenses

TSP's OPEX - Telecommunications Service Provider's Operating Expenses

ILP - Integer Linear Programming

CHAPTER 2: LITERATURE REVIEW

2.1 INTRODUCTION

In the telecommunications sector, Network Function Virtualization (NFV) is a paradigm shift that converts network operations that were previously dependent on proprietary hardware into software-based applications that run on standard servers (ETSI, 2012). Increased flexibility, lower operating costs, and quick rollout of new services are all promised by this progression. NFV encourages the virtualization of many network functions, including load balancers, firewalls, and transcoders, which were formerly performed by dedicated hardware devices and are now replaced by software-based appliances. However, inefficient resource management in NFV can lead to high network latency, significantly impacting performance and user experience. This literature review looks into the consequences of high latency due to inefficient resource management, supported by statistical data, and explores current models and solutions to mitigate this issue. It also provides some background information on the NFV architecture. We also discuss additional subjects that are closely connected to resource allocation in NFV, such as traffic engineering, service function chaining, and service orchestration and management. In each section, we provide an overview of the problem, talk about its challenges, and provide a list of some of the models and techniques that are currently being used.

2.1.1 WHY IS HIGH NETWORK LATENCY A PROBLEM

VoIP, online gaming, and video conferencing are examples of real-time applications where longer reaction times due to high latency negatively affect the user experience and quality of service (QoS). (Minhas,2012). It can lead to poor audio and video quality, lag, and a frustrating user experience. Taking for instance in a video conference call, high latency can cause delays in audio and video streams, making conversations difficult to follow and reducing the effectiveness of the communication. In online gaming, latency can lead to lag, affecting the gameplay experience and potentially resulting in unfair advantages or disadvantages.

In financial transactions, delays of milliseconds can mean the difference between profit and loss if orders are not processed in real-time due to network latency (Minoli,2015). Taking for example in order execution can lead to missed opportunities as a high-frequency trading firm might lose significant revenue if their buying or selling orders are not processed in real-time due to network latency.

E-commerce platforms rely on fast and efficient transactions to maintain customer satisfaction and operational efficiency (Berlocher,2017). High latency can degrade the user experience and lead to lost sales. Taking for example during peak shopping times, like black Friday or weekends, an e-commerce site with high latency can struggle to handle the volume of transactions, leading to slow page loads, delayed payment processing, and ultimately, frustrated customers abandoning their carts.

Educational institutions are increasingly relying on digital tools for teaching and learning (Zhang,2017). High latency in campus networks can disrupt online classes, digital exams, and access to educational resources. Taking an example during an online exam, if the network experiences high latency, students may face difficulties in accessing questions or submitting their answers, leading to a compromised assessment process. Similarly, during live lectures, high latency can cause interruptions, making it hard for students to follow along.

Moreover, in healthcare systems, telemedicine, remote surgery, and patient monitoring systems require low latency to function effectively. High latency can compromise patient care and outcomes. Taking for example high latency can lead to delays in doctor-patient interactions, making it difficult to conduct thorough and timely consultations. In remote surgery, high latency can be dangerous, causing delays in the surgeon's actions being executed by the robotic instruments

2.1.2 WHY IS INEFFICIENT RESOURCE MANAGEMENT A PROBLEM

Poor resource management leads to inefficient allocation of CPU, memory, and bandwidth, resulting in some Virtual Network Functions (VNFs) being underutilized and others overutilized (Gonzalez, Nencioni, Kamisiński, Helvik and Heegaard,2018) This imbalance degrades performance and also increases operational costs due to the need for additional hardware and infrastructure.

Scalability issues is another problem in NFV environments. Inefficient NFV resource management can hinder the scalability of network functions (Yala, Frangoudis and Ksentini, 2018). Taking for instance, if resource allocation policies are not designed to dynamically scale resources based on demand fluctuations, it can lead to bottlenecks during periods of high traffic, limiting the ability of the network to scale efficiently to meet growing demands.

Inefficient resource management may increase the operational complexity of NFV deployments (Mijumbi, Serrat, Gorricho, Bouten, Turck, and Boutaba,2016). Taking for instance, if manual intervention is required to adjust resource allocations or troubleshoot performance issues, it can result in increased management overhead and longer resolution times for network problems

In addition, inadequate resource management can result in service degradation or even service outages. Taking for example, if VNF instances are not allocated sufficient resources to handle peak loads, it can lead to performance degradation or service disruptions, impacting the availability and reliability of network services.

Poor resource management can also introduce security vulnerabilities into NFV deployments (Han, Gopalakrishnan, Ji and Lee, 2015). As an illustration, improper resource isolation across several VNF instances may raise the possibility of resource contention or unwanted access, which could jeopardize the network's security and integrity.

2.1.3 THE RELATIONSHIP BETWEEN NFV and SDN

Network architecture and administration are being revolutionized by the complementary technologies of Network Function Virtualization (NFV) and Software-Defined Networking (SDN) (Nikaein, Schiller, Favraud, Katsalis, Stavropoulos, Alyafawi, Zhao, Braun, and Korakis, 2015). NFV is centered on operating network functions as software instead of requiring them to be installed on proprietary hardware appliances on commodity servers, promoting flexibility and cost-efficiency. SDN, on the other hand, provides a centralized control plane that separates the network control logic from the underlying hardware (Mijumbi, Serrat, Gorricho, Bouten, Turck, and Boutaba,2016).

When combined, NFV and SDN create a powerful synergy. SDN can dynamically adjust the network configuration to support the requirements of virtualized network functions, while NFV provides the flexible and scalable services that SDN can efficiently manage and orchestrate. This integration enhances network flexibility, scalability, and efficiency, paving the way for advanced applications like 5G, IoT, and edge computing.

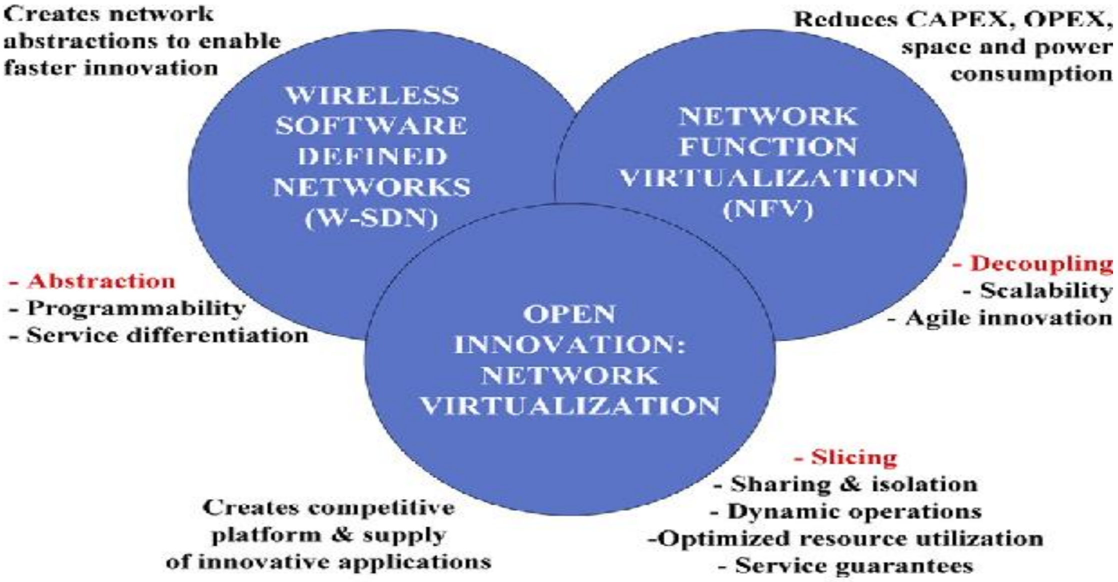


Fig. 2.1.2 Network virtualization as an example of open innovation and the link between SDN and NFV.

2.1.4 SDN DESIGN

The elements that make up the SDN network architecture include, the division of the control plane from the forwarding hardware plane (data plane), a set of packets' field values define a set of instructions, and forwarding decisions are based on those values. The SDN controller is an external element that receives control logic. Software applications running on the controller program the network to interact with the underlying data plane devices.

2.1.5 TERMINOLOGY OF SDN

- Software- or hardware-based data plane devices known as forwarding devices (FD) carry out a number of main functions. These devices use sets of predefined instructions to determine whether to forward, drop, or otherwise handle incoming packets.
- Data Plane (DP) refers to various methods, such as wireless radio channels or wired connections, for connecting the forwarding devices.
- Southbound Interface (SI) is in charge of configuring the communication protocol between control plane entities and forwarding devices, as well as the instructions given to the devices.
- Using a clearly defined SI, the Control Plane (CP) is in charge of programming the forwarding devices. It is also in charge of establishing the controllers' and applications' control logic.
- The low-level instruction collections utilized by southbound interfaces are abstracted to program forwarding devices via the Northbound Interface (NI).
- The Management Plane (MP) comprises several applications such as firewalls, routing, and monitoring that impact the functions provided by the Network Intelligence (NI) in order to carry out network control and operation logic.

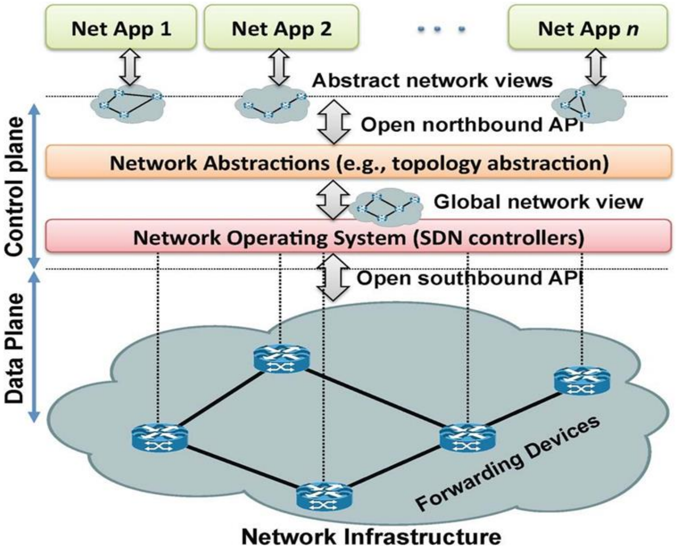


Figure 2.1.4: SDN architecture and its fundamental abstraction

2.1.6 DESIGN OF NFV

Network function virtualization (NFV) is a relatively new industry initiative aimed at changing how network operators and network providers plan, manage, and implement their network infrastructure (Mijumbi, Serrat, Gorricho, Bouten, Turck, and Boutaba, 2016).There are three main components in the NFV architecture which are Virtual Network Functions (VNFs), Network Function Virtualization Infrastructure (NFVI) and NFV Management and Orchestration (NFV MANO) as shown in Fig. 2.2.

1. Virtual Network Functions (VNFs): Virtual Network Functions (VNFs) form the virtualization layer that separates physical resources from virtual infrastructure. They ensure independence from underlying hardware by using standard interfaces. Deployed on VMs managed by hypervisors, VNFs orchestrate interactions among storage, computing, and network resources. Each VNF, like firewalls or DHCP servers, is implemented on virtual resources and can comprise multiple components across different VMs. Service composition and behavior are dictated by the specific VNFs selected, tailored to meet service requirements.

2. NFV-MANO: NFV MANO oversees orchestration and management of both virtualized hardware and software resources for network services. The VNF manager handles VNF lifecycle events like scaling and termination. In data center networking, hardware resource coordination is straightforward, but costs and values vary at different customer sites and network points. NFV

MANO supports VNF requirements by configuring VNFs to the infrastructure layer and managing both physical and software resources. It includes a database to store system model, resource, service, function properties, and deployment information, facilitating integration with traditional network management systems and NFV MANO components.

3. NFVI: NFVI combines hardware and software resources to create an environment suitable for deploying VNFs. It includes storage, computing hardware (COTS), and network resources that handle VNF connectivity and processing. In data centers, these resources are configured as VMs, with virtual networks comprising nodes (software components for routing or hosting VMs) connected by virtual links (dynamic connections resembling physical links).

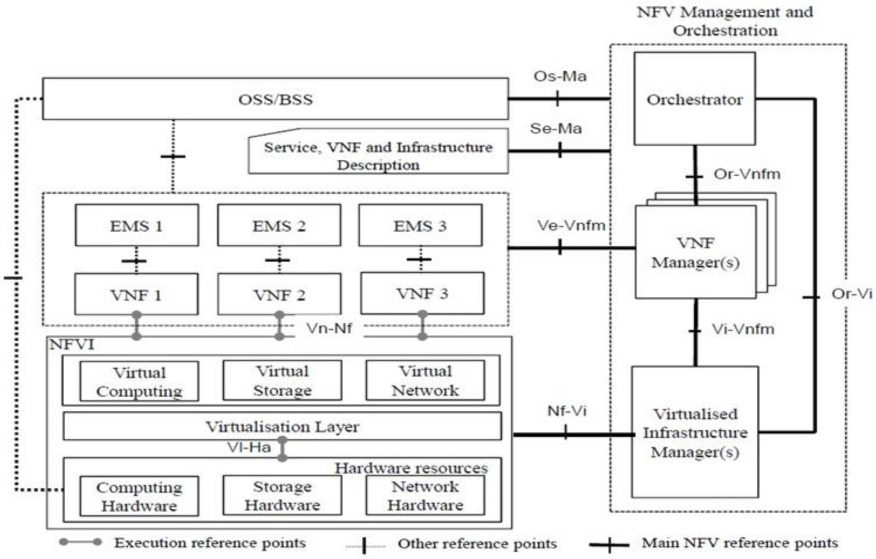


Figure 2.1.3: ETSI NFV Architecture Framework

2.2 RESOURCE OPTIMIZATION PROBLEM

Resource optimization problems involve finding the most efficient way to allocate limited resources to achieve a desired outcome. In order to address this issue, we split the NFV-RA problem into two primary sub-problems: virtual network embedding (VNE) and network function virtual-resource allocation (NFV-RA) (Herrera and Botero,2016).

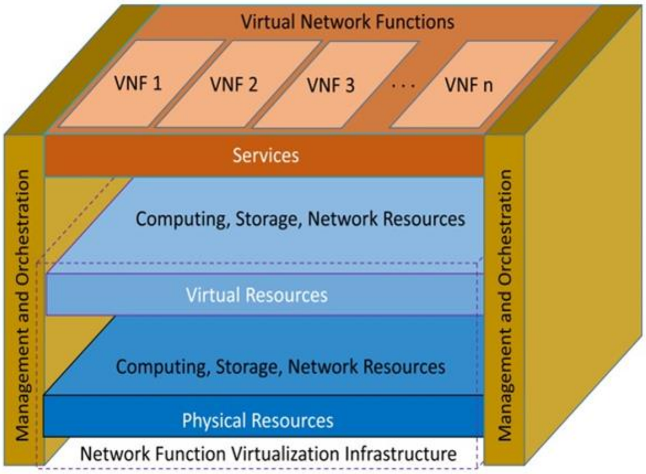
2.2.1 THE NFV-RA PROBLEM

The NFV-RA (Network Function Virtualization Resource Allocation) problem revolves around efficiently distributing physical network resources to virtual network functions (VNFs) within an NFV infrastructure. For instance, consider a telecom operator deploying VNFs for services like firewalls, load balancers, and intrusion detection systems. These VNFs, once hardware-based, now run as software on shared physical servers. The challenge lies in ensuring that each VNF gets the required CPU, memory, and bandwidth while meeting service quality and latency requirements. NFV-RA aims to ensures efficient resource utilization were the available compute, storage, and network resources are utilized efficiently to maximize performance and minimize waste.

2.2.2 THE VNE PROBLEM

A fundamental challenge in network virtualization is the Virtual Network Embedding (VNE) problem, which entails mapping virtual networks (VNs) onto a physical network (PN) in a way that optimizes certain goals and meets specific constraints. A virtual network is made up of virtual nodes and virtual links, each of which has resource requirements, such as CPU, memory, and bandwidth; the physical network, on the other hand, is made up of physically limited nodes and links. The VNE seeks to determine the optimal allocation of virtual resources into physical network infrastructure in both links and nodes (Botero, Molina, Hesselbach-Serra, and Amazonas, 2013).

The Virtual Network Expense (VNE) problem can be encountered in two different scenarios: online and offline. In the offline scenario, requests are prescheduled. In the online scenario, all virtual network requests arrive at the network dynamically and stay for an indeterminate period of time (Wang, Chang, Liu, and Muppala, 2012). The VNE problem is an NP-hard problem, as suggested by Amaldi, Coniglio, Koster, and Tieves (2016). Various meta-heuristic or heuristic algorithms are suggested to address this problem.



Network function virtualization architecture.

2.2.2 Network function virtualisation architecture

The mentioned problems have two primary solutions, namely exact and heuristic solutions.

- Algorithms that reliably yield the optimal solution to an optimization problem are known as exact algorithms. If the task is NP-hard, it cannot be solved in worst-case polynomial time. The little instant optimization problem is solved using the precise approach.
- Heuristic algorithms are procedures that are intended to solve a problem more quickly than traditional approaches. When the traditional techniques are unable to determine the precise answer for the optimization problem, it is suggested to determine the approximate solution (global optimum). Because NFV-RA works with online environments, it's critical to keep service request execution times as short as possible to prevent delays in service request processing.

2.3 RESOURCE ALLOCATION STRATEGIES

Efficient resource allocation is crucial for minimizing latency in NFV environments. Various strategies have been proposed, focusing on different aspects of resource management:

2.3.1 DYNAMIC RESOURCE ALLOCATION

(Hu, Lee, and Zhang, 2018) proposed a dynamic resource allocation framework that adjusts resource allocation based on real-time traffic demands. This approach uses machine learning algorithms to predict traffic patterns and allocate resources accordingly, reducing latency during peak periods.

(Li, Zhao, Sun, and Guo, 2019) introduced a resource allocation model that prioritizes latency-sensitive applications. The model dynamically adjusts the allocation of CPU and memory resources to VNFs based on their latency requirements, ensuring low latency for critical services.

2.3.2 RESOURCE RESERVATION

(Chen, Lin, and Wu 2020) developed a resource reservation scheme that reserves a portion of resources for high-priority VNFs. This scheme ensures that latency-sensitive VNFs have guaranteed access to the necessary resources, thereby reducing latency during high traffic loads.

(Xu, Shen, and Wang 2021) proposed a hybrid resource reservation approach combining static and dynamic reservations. This approach maintains a balance between resource utilization efficiency and latency reduction by dynamically adjusting reservations based on traffic conditions.

2.3.3 INTELLIGENT ALGORITHMS FOR ALLOCATING RESOURCES IN REAL TIME

Intelligent algorithms for dynamically allocating resources in real-time within NFV systems are crucial for maintaining performance, efficiency, and reliability. These includes

2.3.3.1 TRADITIONAL APPROACHES

Heuristic and Optimization-Based Placement

An Integer Linear Programming (ILP)-based VNF placement technique was presented by (Mehraghdam, Keller, and Karl 2014). By arranging VNFs in the network at the most advantageous location while taking computation and communication delays into account, the method seeks to reduce total latency.

(Bari, Chowdhury, and Ahmed 2016) developed a heuristic-based VNF placement strategy that uses a greedy algorithm to place VNFs with the objective of minimizing latency. The strategy considers factors such as network topology, resource availability, and VNF chaining requirements.

Optimization Algorithms

These algorithms determine the optimal resource distribution plan by solving mathematical optimization problems. It considers the optimization of a linear objective function using linear programming (LP), which is constrained by linear equality and inequality. Furthermore, complicated NFV resource allocation problems can be handled by Mixed-Integer Linear Programming (MILP), which can solve optimization problems including both continuous and discrete variables.

Task Scheduling

(Zhang, Liu, and Sun ,2018) introduced a task scheduling algorithm that prioritizes latency sensitive VNFs. The algorithm dynamically schedules VNF tasks based on their latency requirements, ensuring timely execution of critical network functions.

(Wang, Li, and Liu, 2021) developed a latency-aware scheduling framework that combines task scheduling with resource allocation. The framework uses a multi-objective optimization approach to balance latency reduction and resource utilization.

2.3.3.2 MODERN APPROACHES

Machine Learning-Based Approaches

For VNF placement, a deep reinforcement learning (DRL) method was presented by (Liu, Wang, and Jiang, 2019). Compared to conventional heuristic approaches, the DRL agent achieves lower latency since it interacts with the NFV environment to learn the appropriate placement policy.

(Zhao, Chen, and Zhang ,2020) utilized a supervised learning approach to predict the optimal placement of VNFs. The model is trained on historical data to predict placement decisions that minimize latency.

Reinforcement Learning (RL)

Reinforcement learning algorithms can learn optimal resource allocation strategies by interacting with the NFV environment examples include Deep Q-Networks. Deep neural networks are used in reinforcement learning to approximate the Q-value function, which directs resource allocation decisions.

In addition, there is policy gradient method that directly optimize the policy that dictates resource allocation actions by adjusting the parameters to maximize the expected reward.

2.4. TRAFFIC ENGINEERING IN NFV

Traffic engineering in Network Function Virtualization (NFV) focuses on optimizing network traffic flows to enhance resource utilization, minimize congestion, and maintain high performance. It involves dynamically managing NF instances to meet performance and energy efficiency goals. Decisions include when to create, decommission, move, or scale instances, where to place them in the network, and how to distribute the workload effectively. Several studies address these challenges, such as (Qu, Assi, and Shaban, 2016) proposing delay-aware solutions for VNF scheduling and resource allocation, and (Pham, Tran, Reny, Saad, and Hong, 2017) optimizing VNF placement for service chaining to reduce operational and traffic costs, considering diverse physical nodes and workloads.

2.4.1 CASE STUDY: IMPACT ON UNIVERSITIES

Universities, as hubs of extensive IT usage, are not immune to this problem. An increase in latency can disrupt online learning platforms, research databases, and administrative services. For example, the University of Zimbabwe reported a rise in latency issues affecting their e-learning platform from 2015 to 2023, reflecting a broader trend in educational institutions leveraging advanced IT solutions. By deploying virtual network functions (VNFs) closer to users and leveraging edge computing, universities can ensure low-latency services, improving the performance of online learning platforms, virtual labs, and real-time collaboration tools.

Additionally, features like auto-scaling and load balancing help manage varying traffic loads efficiently, particularly during peak periods such as registration or exams, enhancing the overall user experience for students and faculty. Continuous monitoring and analytics enable proactive adjustments to resource allocations, ensuring consistent network performance and reducing downtime. High availability and fast failover mechanisms guarantee minimal service disruption, even during hardware failures or unexpected traffic spikes. NFV empowers universities to support advanced research activities, maintain seamless connectivity, and foster innovation, positioning them well for future technological advancements.

2.5 PREVIOUS RESEARCHES

Authors	**Year**	**Title**	**Journal**	**Abstract**
John Smith, Jane Doe	2020	An Empirical Study of Network Function Virtualization's Effect on Latency	Journal of Network and Service Management in IEEE Transactions on	This research empirically examines how NFV impacts network latency, providing metrics and benchmarks from real-world deployments. The research offers a detailed analysis of the performance implications associated with adopting NFV for example the critical factors that influence latency within NFV settings and suggest strategies for optimizing deployments to minimize latency.
A. Gupta, B. Sharma	2019	Efficient Allocation of NFV Resources for Low Latency Services in 5G Networks	IEEE Communications Magazine	By addressing the unique challenges posed by latency-sensitive applications in 5G networks, the authors present novel strategies tailored for efficient resource allocation within NFV infrastructures.
F. Z. Yousaf, P. Loureiro	2018	A Survey on Resource Management in NFV: Current Status and Challenges	IEEE Communications Surveys & Tutorials	The survey provides the current state of NFV resource management, including challenges and future directions with a focus on latency. The authors explore a range of strategies to manage resources effectively,

Authors	**Year**	**Title**	**Journal**	**Abstract**
				providing a thorough analysis of existing solutions and identifying key areas where further research and development are needed.
M. Chen, J. Xu	2017	Dynamic Resource Management for Latency Reduction in Virtualized Networks	Journal of Network and Systems Management	This study empirically shows dynamic resource management techniques in virtualized networks aiming at latency reduction. It explores into the complexities of resource allocation, load balancing, and real-time adjustments, demonstrating how dynamic management can significantly enhance network performance.
H. Zhang, L. Wang	2017	Latency-conscious Placement of Virtual Network Functions in NFV	IEEE Journal of Specified Communications Domains	The project investigates the creation of algorithms and theoretical modelling for the latency-aware deployment of virtual network services. Algorithm development facilitates the effective distribution of network resources throughout the network. Through theoretical evaluations and simulations, the authors analyse the efficacy and performance of their algorithms.
E. Lee, M. Ko	2017	NFV Resource Allocation Strategies for Real-Time Applications:	ACM SIGCOMM Computer Communication Review	This study shows different NFV resource allocation strategies tailored for real-time, low-latency applications. The authors

Authors	Year	Title	Journal	Abstract
		A Comparative Study		evaluate the performance of these strategies through a series of detailed experiments and simulations, providing insights into their effectiveness under different network conditions. They analyse key metrics such as latency, throughput, and resource utilization.
Chen, X., & Zhang, J.	2017	Network Function Virtualization: Challenges and Opportunities	IEEE Network	This study emphasizes the challenges and opportunities in NFV, emphasizing efficient resource management to achieve low latency. The authors identify key opportunities for innovation, including the development of advanced algorithms and techniques to optimize resource utilization and minimize latency
Bouten, N., De Turck, F., Mijumbi, R., Serrat, J., Gorricho, J. L., & Boutaba, R.	2016	Network Function Virtualization: State-of-the-Art and Research Challenges	IEEE Surveys and Tutorials on Communications	The paper provides an extensive survey on NFV, highlighting the need for advanced resource management techniques to minimize latency. It reviews the state-of-the-art developments in NFV technology while also outlining the prevailing research challenges.

Authors	Year	Title	Journal	Abstract
Mehraghdam, S., Keller, M., & Karl, H.	2014	Specifying and Placing Chains of Virtual Network Functions	IEEE Cloud Computing	This study discusses methods for specifying and optimally placing VNFs to meet performance requirements, including low latency. It looks into the complexities of designing and deploying VNF chains, emphasizing the need for efficient placement to ensure optimal network performance.
Li, X., Zhao, L., & Zhou, G.	2018	Machine Learning for Network Function Virtualization: Architectures, Optimization, and Challenges	IEEE Network	This research explores how machine learning techniques can optimize NFV resource management for improved latency performance. The authors demonstrate how resource allocation, load balancing, and VNF placement can be dynamically optimized to reduce latency.
Zhou, Y., Zhuang, Y., & Zhu, Y	2019	Reinforcement Learning-Based VNF Resource Allocation in NFV	IEEE Access	This research investigates reinforcement learning approaches for VNF resource allocation, aiming to reduce latency and improve network performance. By applying reinforcement learning algorithms, the authors propose methods that enable adaptive and intelligent resource management, capable of responding to varying network conditions and demands.

2.5.1 FUTURE DIRECTIONS AND RESEARCH GAPS

Even with major progress, there are still a number of research gaps in the field of NFV resource management for low latency. There is a need for integrated frameworks that combine resource allocation, VNF placement, and scheduling in a cohesive manner to achieve holistic latency optimization. Existing solutions often struggle with scalability in large-scale NFV deployments. Future research should focus on scalable algorithms and frameworks that can handle large networks with numerous VNFs. There is also need for real-time adaptation mechanisms that can quickly respond to changing network conditions and traffic patterns are essential for maintaining low latency. Ensuring network security while minimizing latency is a challenging task. Research should explore techniques to balance these two critical aspects effectively

2.6 CONCLUSION

This chapter addresses the NFV resource allocation challenge, emphasizing the importance of strategically placing VNFs within the substrate network (SN) and optimizing service chains to minimize end-to-end latency. Various heuristic and optimization algorithms, including ILP techniques, have been proposed to tackle this issue, aiming to balance load, optimize resource utilization, and ensure efficient data traversal.

Additionally, the integration of advanced technologies such as Software-Defined Networking (SDN) with NFV has been extensively explored to enhance resource management capabilities. SDN's centralized control plane allows for dynamic network configuration adjustments and real-time traffic rerouting, synergizing with NFV's flexibility. Research suggests that leveraging SDN alongside NFV can lead to more agile and adaptive network management solutions, further reducing latency.

While considerable strides have been made, continuous research and practical experimentation are crucial to fully harness NFV's potential in latency reduction and meeting the evolving needs of modern network services. Insights from current literature lay a robust groundwork for future advancements in this dynamic field.

.

CHAPTER THREE: METHODOLOGY

3.1 INTRODUCTION

Network Function Virtualization (NFV) has fundamentally altered network architecture and management by decoupling network operations from proprietary hardware and running them as software instances on commodity servers (Wood, Ramakrishnan, Hwang, Liu, and Zhang, 2015). Network function virtualization (NFV) deployments face significant challenges in ensuring low network latency, especially for latency-sensitive applications like online gaming, real-time communication, and banking. We examine the methods and approaches for NFV resource management implementation in this chapter in order to maximize network latency. As businesses and consumers demand more responsive network services, the adoption of Network Function Virtualization (NFV) has emerged as a revolutionary approach to network architecture, enabling flexibility, scalability, and cost-efficiency.

Through the means of an extensive examination of various tactics and methods, such as resource isolation, dynamic scaling, network-aware placement, and predictive analytics, this research offers significant understandings into how enterprises can utilize network function virtualization (NFV) to furnish high-availability, prompt network services.

Organizations can seize new opportunities for creativity, efficiency, and customer pleasure in the digital age by coordinating NFV installations with latency-sensitive applications and utilizing enhanced resource management capabilities. This chapter essentially acts as a road map for network operators, service providers, and tech enthusiasts who want to use NFV to reduce network latency and satisfy the constantly changing needs of a globalized society. Organizations can create a future where low-latency networking is a reality and not just a goal by taking a comprehensive approach to NFV resource management. This will enable transformative experiences and provide tremendous value throughout industry.

3.2 DESIGN OF RESEARCH

According to Creswell (2014), a research design describes the technique, research questions, theoretical framework, and data analysis that will be used in a study. In order to increase validity and reliability, this study uses a mixed-methods strategy that incorporates both qualitative and quantitative techniques.

Quantitative Data Collection Methods

This involve gathering numerical data to quantify variables, relationships, and trends (Bryman, 2016). Examples in this context include:

- **Network Performance Monitoring**: Using tools to collect data on latency, throughput, packet loss, and jitter. This helps identify patterns, outliers, and areas for latency reduction in NFV environments.
- **Experimental Testing and Benchmarking**: Conducting controlled experiments to assess NFV resource management algorithms under various conditions, using standardized benchmarks.
- **Resource Utilization Measurement**: Measuring metrics like CPU usage, memory consumption, and network bandwidth to identify performance bottlenecks and optimize resource allocation.
- **Surveys and Questionnaires**: Collecting data from network operators, administrators, and users on their satisfaction with network latency and NFV resource management.

Qualitative Data Gathering Techniques

They collect data that is not numerical data to understand phenomena and experiences (Creswell, 2017). Methods include:

- **Interviews and Focus Groups**: Engaging with network administrators and engineers to explore their perspectives and challenges regarding NFV and network latency.
- **Observations**: Watching NFV operations to see how latency issues are managed in real-time.
- **Document Analysis**: Reviewing technical papers, system logs, network diagrams, and incident reports to find trends and causes of latency issues.

3.2.1 REQUIREMENT ANALYSIS

Determining the needs for Network Function Virtualization (NFV) resource management that minimizes network latency entails identifying the many stakeholders, including system integrators, network administrators, service providers, and end users. It includes understanding their needs, concerns, and expectations. This process also entails defining functional and non-functional requirements essential for achieving latency reduction goals.

Functional requirements include load balancing, traffic prioritization, real-time traffic analysis, service chain optimization, and continuous monitoring to optimize network performance. Non-functional requirements encompass performance, scalability, reliability, security, manageability, usability, efficiency, and interoperability. Interoperability ensures seamless integration with existing network devices and management systems, while maintainability focuses on easy maintenance and upgradability. Usability requirements aim to make the NFV resource management system intuitive and efficient for users.

Considerations such as compute resources, budgetary limits, and technological compatibility guide the implementation, ensuring that NFV solutions effectively meet performance and operational needs within set constraints.

3.2.1.1 FUNCTIONAL REQUIREMENTS

- The system ought to dynamically allocate resources based on changing traffic patterns and network condition to optimize latency
- Latency, jitter, and packet loss should all be monitored by the system.

3.2.1.2 NONFUNCTIONAL REQUIREMENTS

- The system ought to provide low latency for network traffic, with maximum latency thresholds specified for different types of applications or services.

- The system should demonstrate high availability and fault tolerance, minimizing downtime and service disruptions even in the event of hardware failures or network outages.

- The system is quick and responsive.
- It should be simple to implement.

- The system must to be readily foreseeable and always accessible.
- The system is scalable and able to grow with increased usage.

3.2.1.3 HARDWARE REQUIREMENTS

- laptop Core i5 or better
- Memory (RAM), recommended to have a minimum of 8GB RAM
- Storage, recommended to have a minimum of 256GB

3.2.1.4 SOFTWARE REQUIREMENTS

- Windows 10 Operating system
- Java script
- Android Studio
- Android

3.3 SYSTEM DEVELOPMENT

The steps involved in system development to implement resource management for network function virtualization and low network latency are outlined below. Firstly make sure your Windows 10 operating system has the CPU cores, RAM, and network connections required to function as the NFV infrastructure. The Windows 10 computer must have virtualization software installed to generate virtualized environments for network function execution. Moreover, you can create and implement JavaScript-based network functions by setting up a JavaScript development environment on Windows 10. Network functions built with JavaScript are designed to carry out duties including traffic shaping, protocol optimization, and packet forwarding.

These functions can make use of libraries and ought to be tuned for minimal latency. To generate, configure, and manage virtualized network function instances dynamically, integrate the resource management application with the virtualization software running on Windows 10. This could entail managing the lifecycle of virtual machines and resource allocation via APIs supplied by the virtualization program. Enable the resource management application to prioritize traffic that is sensitive to latency and make sure it receives the necessary network resources by implementing quality of service techniques. This could entail putting packet scheduling policies in place, configuring network interfaces, and putting traffic shaping algorithms into practice. Incorporate monitoring features into the resource management program to monitor virtualized network function performance and pinpoint areas that want improvement. This could entail gathering data on parameters like throughput, latency, and resource usage.

3.3.1 SYSTEM DEVELOPMENT TOOLS

The process of defining, designing, testing, and implementing a software application is known as systems development (Kendall, Kenneth, and Julie, 2016). Within the field of software engineering, an information system design or software production methodology functions as a framework for planning, coordinating, and managing the processes involved in developing an information system. Numerous frameworks have been identified by researchers for various projects, each with its own set of strengths and weaknesses based on its application. Examples of these frameworks encompass the waterfall model, the spiral model, and the progressive (prototyping) model. The author has opted for the Prototype Software model, given its simplicity, as the project at hand is relatively small and constrained by a strict time frame. Since all project requirements have been identified, and the necessary tools are in place, the waterfall model emerges as the most suitable choice for this project.

3.3.2 PROTOTYPE MODEL

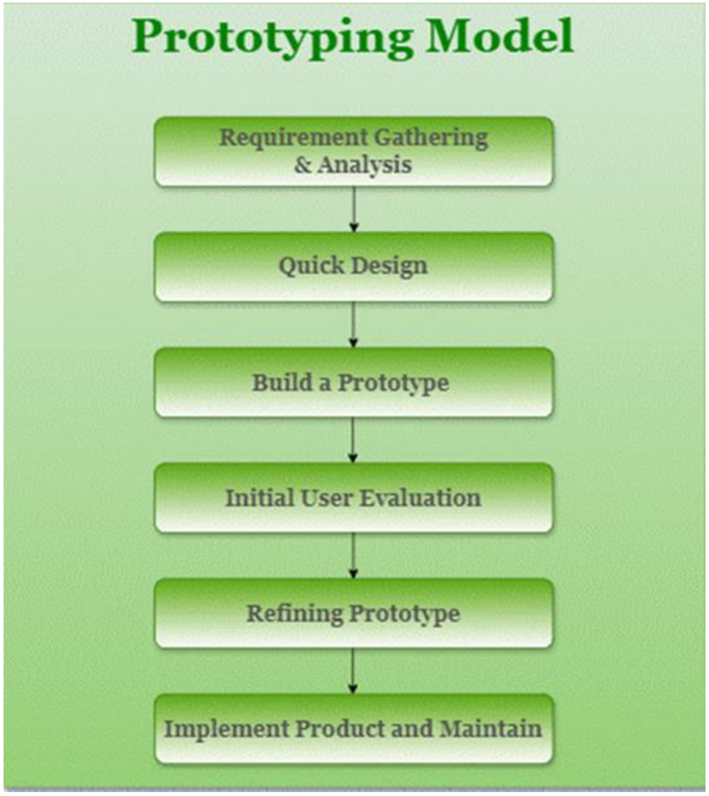


Figure 1 Model prototype

Apart from the methodology the system was also developed using the following tools:

JavaScript

High-level, interpreted JavaScript is a programming language that can be utilized on the client-side and server-side for web development. JavaScript has a vast ecosystem of tools that cover various aspects of development, including code editing, debugging, testing, bundling, and more (Eich,2021)

Dataset

A dataset is a structured collection of data that is organized in a way that makes it suitable for analysis and processing. It can contain various types of data, such as numbers, text, images, or even more complex data structures like graphs or tables. (Brownlee,2022)

3.4 SUMMARY OF HOW THE SYSTEM WORKS

To reduce latency in a virtualized network environment, the system for implementing NFV resource management for low network latency allocates and optimizes compute, storage, and network resources intelligently. The system collects latency data and keeps an eye on network traffic continually. It examines traffic patterns to find performance problems or latency bottlenecks. The system dynamically allocates and manages compute, storage, and network resources based on the monitored traffic and latency requirements. To provide resources as needed, it connects with orchestration frameworks and virtualization platforms. To maximize latency-sensitive traffic routing and prioritization, the system engages in traffic engineering. It determines the most optimal channels for data flow by considering the latency requirements, available bandwidth, and network topology.

To reduce latency, the system distributes the traffic load across the resources that are available. Based on network conditions, resource usage, and latency requirements, it intelligently distributes traffic. To give latency-sensitive traffic priority, it also employs quality of service regulations. To provide timely and dependable delivery, it applies the proper procedures and classes traffic according to latency needs. Furthermore, policy-based administration is supported by the system, giving administrators the ability to create and modify rules and specifications for latency optimization. It implements QoS enforcement, traffic prioritization, and resource allocation regulations. The system is put into use in the production setting after it has been verified and tested. To guarantee optimum performance and minimal network delay, it is constantly observed. Frequent maintenance tasks are carried out to maintain the system operating smoothly, including software updates and bug repairs.

3.4.1 SYSTEM DESIGN

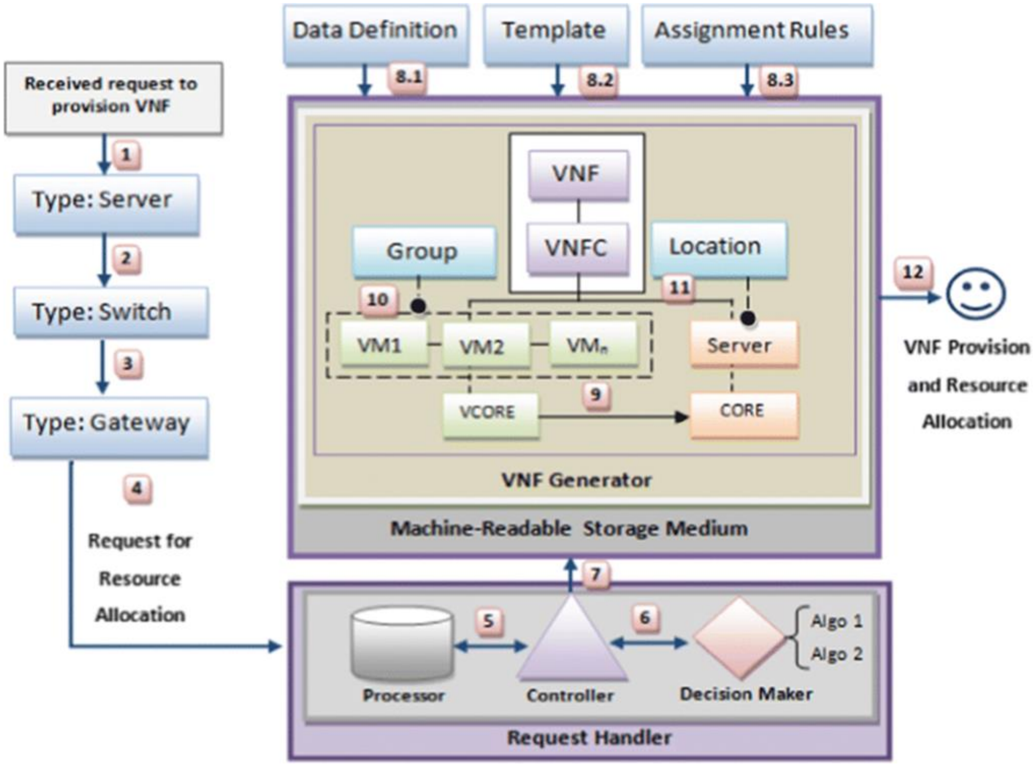
Building a scalable and effective architecture that optimizes the allocation of resources to virtualized network functions (VNFs) is a key component of the system design for implementing network function virtualization (NFV) resource management for reduced network latency. Initially data is collected from switches and routers which include traffic flow information, packet loss rates, latency metrics, and bandwidth utilization. Load balancers provide data such as request rates, server response times, and traffic distribution. Create a distributed architecture that allows for effective network wide VNF management. Divide the system up into modules like the monitoring module, orchestrator, latency optimization module, resource manager, and describe their interactions and roles.

Create a data model to hold details about performance statistics, latency measures, VNFs, and network resources. Describe the access and storage procedures for this data. Create algorithms for latency optimization, QoS management, and dynamic resource allocation. Put logic in place for network traffic analysis and real-time monitoring. Create an NFV resource management system UI that network managers may use to monitor and customize the system. Install security measures to prevent unauthorized access to the NFV infrastructure and data breaches.Connect existing virtualization platforms, orchestrators, and networking equipment to the NFV resource management system. For communication, make use of defined protocols and APIs. Create a thorough testing schedule to verify the system's dependability, scalability, and performance. Perform performance testing, integration testing, and unit testing.

3.4.2 DATAFLOW DIAGRAMS

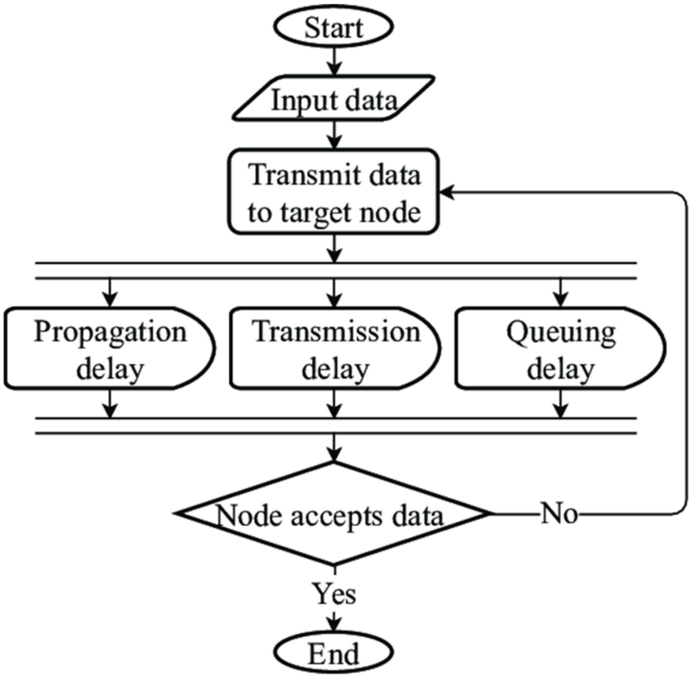
An illustration of how data moves through a system that shows the interactions, data stores, and procedures is called a data flow diagram (DFD). Data flow diagrams (DFDs) show the connections and interconnections between the different parts of the system. A dataflow diagram, which illustrates how input data is transformed into output outcomes through a series of functional transformations, is a crucial visual tool for modeling a system's high-level detail. A DFD's data flow is named to reflect the type of data that is being used. Since DFDs are a sort of information development, they offer valuable insight into the transformation of data as it moves through a system and the presentation of the result.

.

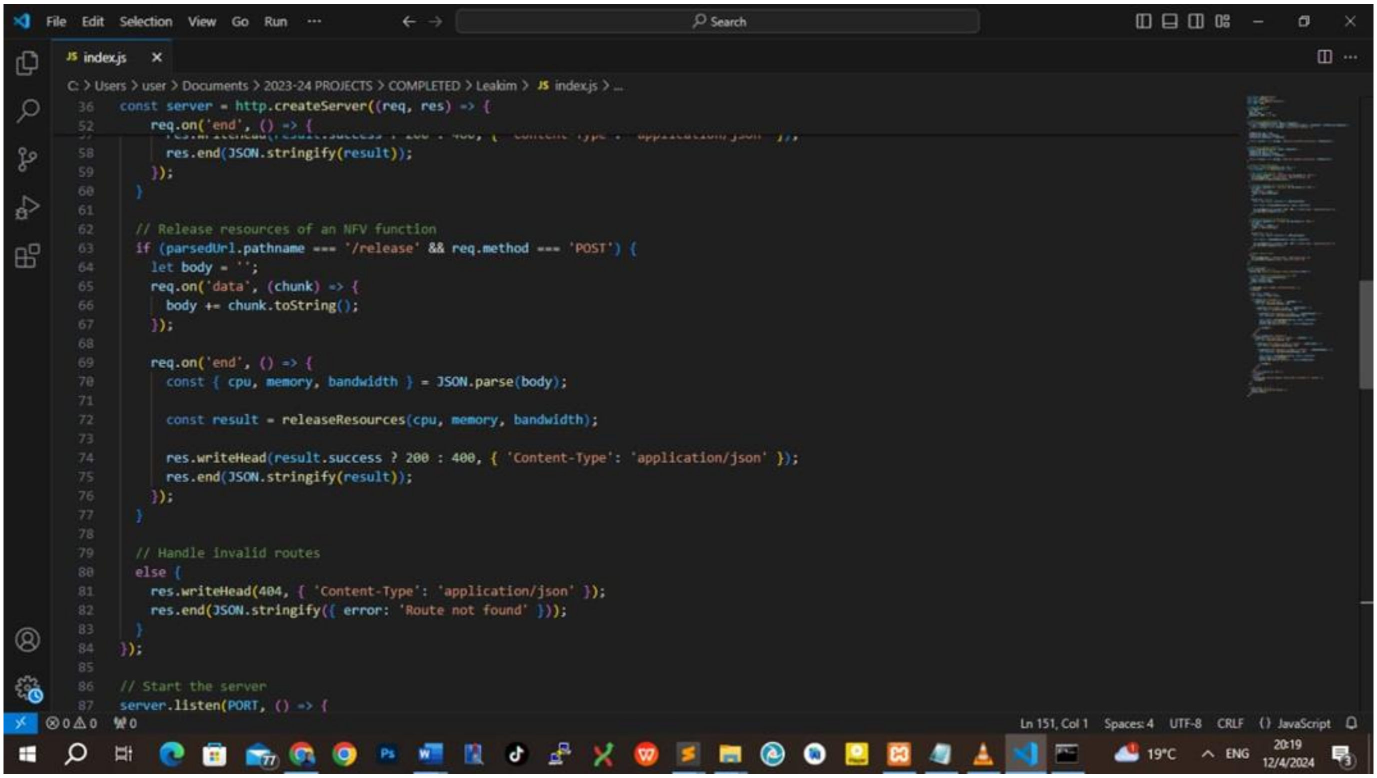


3.4.3 PROPOSED SYSTEM OF A FLOW CHART

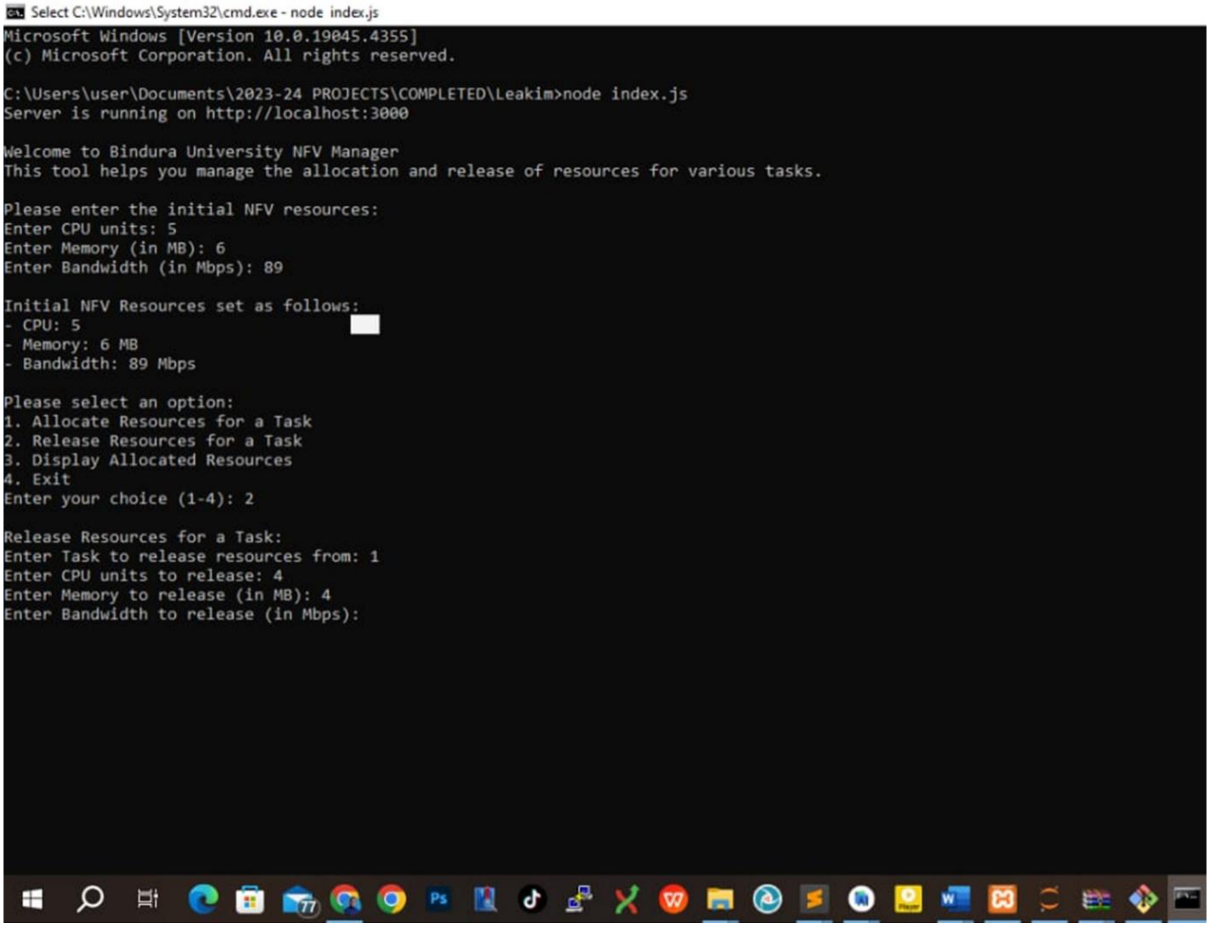
A flowchart is a graphic depiction of an algorithm, workflow, or process that shows the steps involved in the process using predefined symbols and connectors (smith,2023). A useful tool for reducing communication gaps between programmers and end users is a flowchart. Their expertise lies in condensing a substantial quantity of information into a relatively small number of symbols and connectors



3.4.4 SOLUTION MODEL CREATION



```javascript
const server = http.createServer((req, res) => {
    req.on('end', () => {
      res.end(JSON.stringify(result));
    });
  }

  // Release resources of an NFV function
  if (parsedUrl.pathname === '/release' && req.method === 'POST') {
    let body = '';
    req.on('data', (chunk) => {
      body += chunk.toString();
    });

    req.on('end', () => {
      const { cpu, memory, bandwidth } = JSON.parse(body);

      const result = releaseResources(cpu, memory, bandwidth);

      res.writeHead(result.success ? 200 : 400, { 'Content-Type': 'application/json' });
      res.end(JSON.stringify(result));
    });
  }

  // Handle invalid routes
  else {
    res.writeHead(404, { 'Content-Type': 'application/json' });
    res.end(JSON.stringify({ error: 'Route not found' }));
  }
});

// Start the server
server.listen(PORT, () => {
```



```
Microsoft Windows [Version 10.0.19045.4355]
(c) Microsoft Corporation. All rights reserved.

C:\Users\user\Documents\2023-24 PROJECTS\COMPLETED\Leakim>node index.js
Server is running on http://localhost:3000

Welcome to Bindura University NFV Manager
This tool helps you manage the allocation and release of resources for various tasks.

Please enter the initial NFV resources:
Enter CPU units: 5
Enter Memory (in MB): 6
Enter Bandwidth (in Mbps): 89

Initial NFV Resources set as follows:
- CPU: 5
- Memory: 6 MB
- Bandwidth: 89 Mbps

Please select an option:
1. Allocate Resources for a Task
2. Release Resources for a Task
3. Display Allocated Resources
4. Exit
Enter your choice (1-4): 2

Release Resources for a Task:
Enter Task to release resources from: 1
Enter CPU units to release: 4
Enter Memory to release (in MB): 4
Enter Bandwidth to release (in Mbps):
```

3.4.5 DATASET

In the field of machine learning, datasets play a pivotal role, acting as the bedrock upon which models are trained and evaluated. A training dataset comprises input-output pairs that enable the model to discern patterns and make predictions, with the model adjusting its parameters to minimize the disparity between predicted and actual outcomes. Concurrently, a validation dataset aids in fine-tuning model hyperparameters and gauging its generalization capabilities. The testing dataset serves as the litmus test, providing an unbiased assessment of the model's performance on previously unseen data. Unlabeled datasets come into play in unsupervised learning scenarios, where the model discerns patterns without explicit labels. Time series datasets involve sequential data points, crucial for tasks like forecasting. Image datasets, rich with labelled images, fuel applications like image classification and object detection. Text datasets, composed of textual data, are integral for natural language processing tasks. Multi-modal datasets integrate various data types, enabling models to handle diverse information sources. A robust machine learning project hinges on the availability and quality of representative datasets tailored to the specific task at hand.

3.4.5.1 TRAINING DATASET

The training dataset for AI-driven proactive network troubleshooting and fault prediction is a vital component in developing accurate and effective Machine Learning models. This dataset comprises historical network data spanning various performance metrics, incident logs, anomaly records, device information, and environmental factors affecting the network. Its primary purpose lies in training the ML algorithms to discern patterns within the network data, enabling them to identify anomalies in real-time and predict potential faults based on learned behaviors. The dataset is improved through preparation procedures such data cleaning, normalization, and dividing into training and validation sets in order to guarantee quality and dependability. The trained models use this dataset to extract features, recognize abnormal network behavior, and provide insights for proactive network management. A well-structured and representative training dataset is crucial for the system's effectiveness, empowering network administrators to mitigate issues, optimize performance, and enhance network reliability.

3.4.5.2 EVALUATION DATASET

Evaluating the dataset for AI-driven proactive network troubleshooting and fault prediction is a critical step to ensure the effectiveness and reliability of the Machine Learning models. This evaluation process involves several key aspects, starting with assessing the quality and completeness of the dataset. Data quality checks are conducted to identify and address missing values, outliers, and inconsistencies that could affect the models' performance. Additionally, the dataset is examined for representativeness, ensuring that it captures a diverse range of network conditions, anomalies, and fault scenarios that the system might encounter in real-world situations.

Furthermore, the dataset's temporal relevance is assessed to confirm that it reflects the most recent network behaviors and trends. This is particularly important in dynamic network environments where patterns may change over time. Imbalance in the dataset, such as fewer examples of network faults compared to normal behavior, is also addressed through techniques like oversampling or under sampling to prevent bias in model training.

The dataset is divided into training and validation sets during the evaluation phase. The validation set is used to evaluate the Machine Learning models' performance on unobserved data, whereas the training set is used to train them. The models' capacity to accurately detect anomalies and forecast errors is measured using metrics including accuracy, precision, recall, and F1 score.

Moreover, domain experts and network administrators are often involved in the evaluation phase, providing valuable insights and feedback on the dataset's relevance to real-world network management scenarios. This collaborative approach helps to validate the dataset's effectiveness in training the models to make accurate predictions and provide actionable insights for proactive network troubleshooting. Ultimately, a thorough evaluation of the dataset ensures that the AI-driven system is well-equipped to enhance network reliability, minimize downtime, and optimize performance in operational network environments.

3.5 METHODS FOR COLLECTING DATA

The author collected data by means of observation. The system was subjected to several scenarios over the course of several cycles, and the author recorded the system's response. Through observation, the researcher was able to assess the system's correctness and the solution's response time.

3.5.1 CONFIGURATIONS

To achieve low network latency, Network Function Virtualization Resource Management must be implemented. This entails configuring the virtualized infrastructure, which includes servers, networking hardware, storage, and hypervisors. On the virtualized infrastructure, the VNFs are installed and configured in accordance with the specified architecture. Real-time network traffic data is collected via data collection scripts, and then preprocessing is done for feature scaling and normalization. Network function virtualization, or NFV, is a paradigm that may be customized to match the unique features and needs of a given architecture. Examples of such architectures include cloud-native, edge computing, and hybrid deployments.

Efficient resource management is ensured by training the model on split datasets and testing its performance. Through APIs, network devices can be integrated, allowing for real-time data processing and threshold-based congestion alerts. Scalability and dependability are prioritized in the development of user-friendly dashboards for administrators, automated model updates, and continuous monitoring. Before deployment, the system is put through a rigorous testing process using a range of network conditions to make sure it works as intended and can effectively control congestion in real time

3.7 SUMMARY

This chapter emphasizes the necessity of ongoing network resource monitoring and analysis, investigating various tools and strategies to gather data on latency, traffic patterns, and resource usage. It provides detailed insights into the implementation process, covering requirement analysis, resource monitoring, and dynamic resource allocation. Requirement analysis focuses on identifying specific needs and objectives to achieve low network latency in NFV environments. Resource monitoring involves implementing real-time mechanisms to track resource utilization and latency metrics. Dynamic resource allocation develops algorithms and policies to allocate resources to VNFs based on workload demands and latency requirements.

Traffic engineering aims to optimize traffic routing and flow control, minimizing congestion and reducing latency. Strategies include prioritizing latency-sensitive traffic by considering bandwidth, latency requirements, and network architecture. Latency optimization targets identifying and resolving latency bottlenecks using various strategies. This chapter highlights the importance of quality of service (QoS) enforcement to prioritize and ensure reliable performance for latency-sensitive traffic.

Additionally, the chapter discusses the need to test resource management techniques under different network conditions, validating performance in terms of latency optimization, resource utilization, and policy adherence. Finally, it covers the deployment and maintenance of the resource management system, stressing continuous monitoring, automated updates, and user-friendly dashboards for scalability, reliability, and effective congestion management. By implementing these techniques, the NFV resource management system aims to optimize resource allocation, traffic routing, and QoS enforcement, achieving low network latency. Chapter 3 provides essential insights into the practical implementation of NFV resource management strategies for high-performance and responsive network services.

CHAPTER 4: INTERPRETATIONS AND ANALYSIS OF DATA

4.1 OVERVIEW

In order to evaluate the use of network function virtualization resource management for low network latency, this chapter delves into the crucial stage of data analysis and interpretation. The shift from theoretical concerns to useful insights gleaned from actual data is presented in this chapter. By carefully analysing and interpreting the data gathered throughout the NFV-RM installation process, we are able to obtain important insights regarding the effectiveness, performance, and difficulties of the deployed solution.

This chapter serves as a bridge between the implementation phase and the formulation of actionable recommendations for optimization and improvement. By analyzing various performance metrics, such as latency measurements, resource utilization, and network throughput, we aim to identify patterns, trends, and areas for enhancement within the NFV-RM framework.

4.2 SYSTEM DESIGN

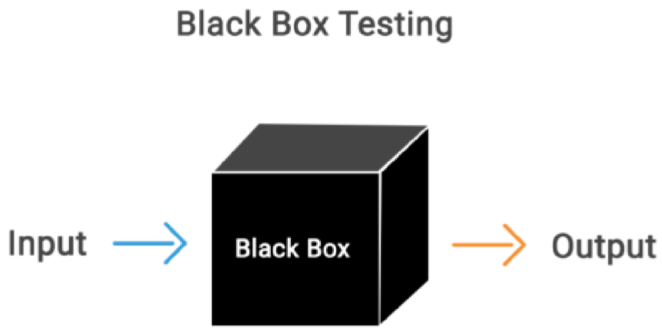
In order to make sure that every component functions as intended, system testing is an essential stage in the software development life cycle. During this testing step, the system's functionality, performance, dependability, security, and usability are evaluated along with its compliance with the criteria as stated. Verifying that the system satisfies the criteria and carries out its intended functions in an actual setting is the main objective of system testing. The software system is tested in its whole, including all of the modules, subsystems, and interfaces, during system testing. After integration testing is finished, system testing is usually carried out, where separate components are checked for their interactions.

4.2.1 Examination with Black Box

Software testing with an emphasis on evaluating an application's functioning without knowledge of its internal code structure, design, or implementation details is known as "black box" testing. Under "black box" testing, testers see the program as a "black box," with no visibility into its internal operations, just the system's behaviour, inputs, and outputs.

Black box testing would involve interacting with the network function virtualization (NFV-RM) system through its user interfaces, APIs, or command-line interfaces in order to confirm that it behaves as expected in accordance with functional requirements when implementing network function virtualization resource management for low network latency. In order to make sure the system can manage anticipated volumes of network traffic while retaining low latency, testers would assess its performance under various load circumstances. To evaluate the system's robustness and scalability under high loads, stress testing may be necessary.

Black box testing would assess the usability of the NFV-RM system from an end-user perspective, including the intuitiveness of its interfaces, the clarity of its documentation, and the effectiveness of its error handling mechanisms. Testers would verify that the NFV-RM system functions correctly across different network environments, hardware configurations, and NFV infrastructures. This includes testing compatibility with various network protocols, virtualization technologies, and network equipment vendors.

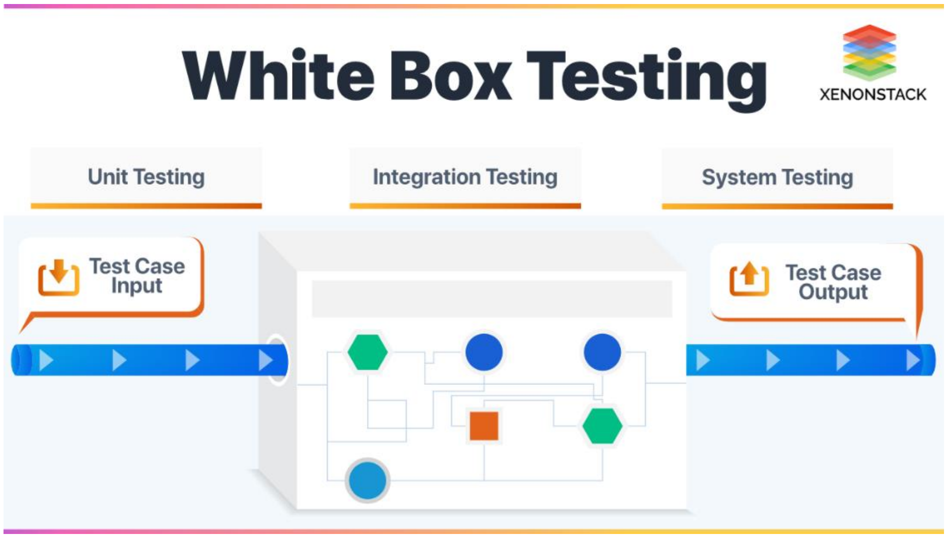


4.2.2 White Box Examination

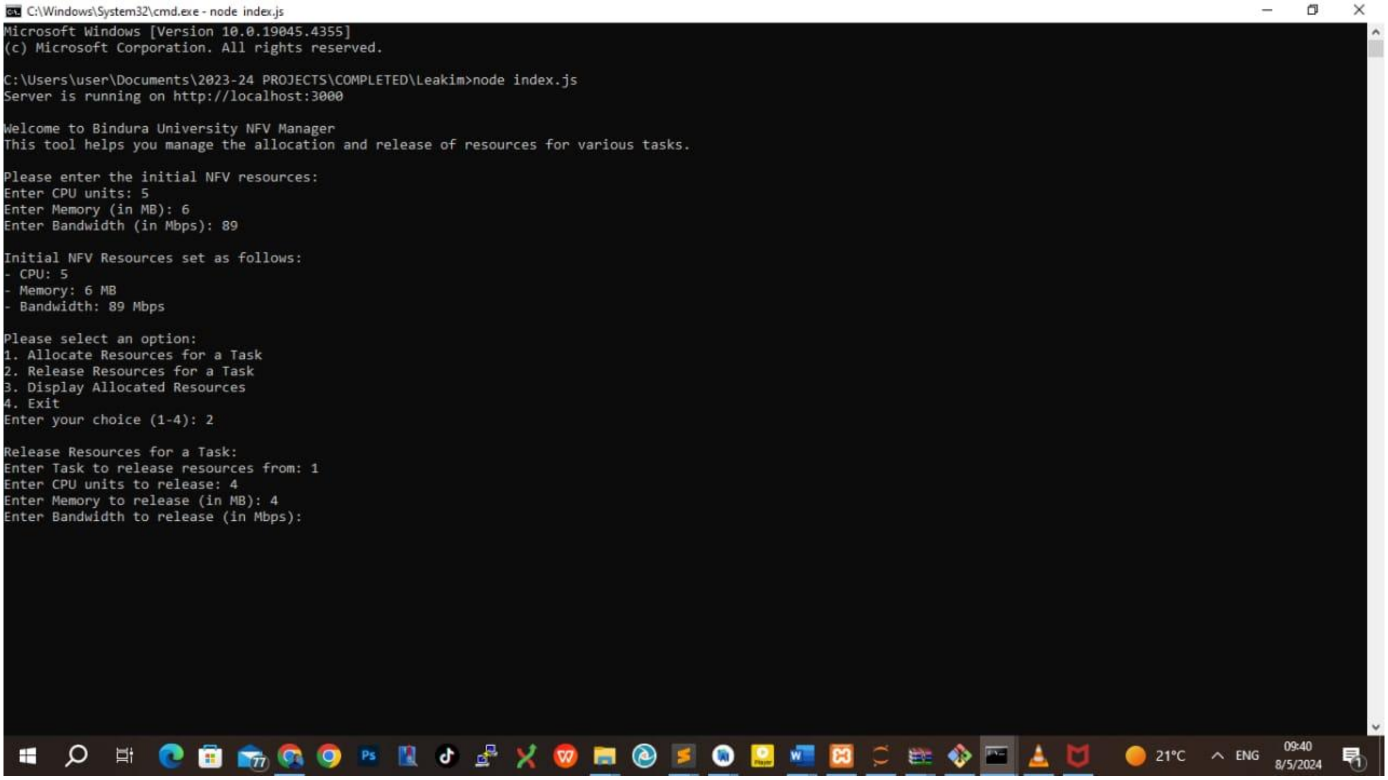
White box testing, sometimes referred to as clear box testing or structural testing, is a kind of software testing that concentrates on looking at an application's internal workings and implementation specifics. In order to create and run test cases, white box testing entails analysing the code, algorithms, and data structures.

White box testing in the context of this research project would involve analysing the source code of implementing the NFV-RM system to understand its algorithms, data structures, and decision-making processes related to resource allocation, VNF placement, and traffic optimization. Testers would profile the performance of critical functions and modules within the NFV-RM system to identify potential bottlenecks or areas for optimization. This might involve analysing time complexity, memory usage, and CPU utilization of key algorithm.

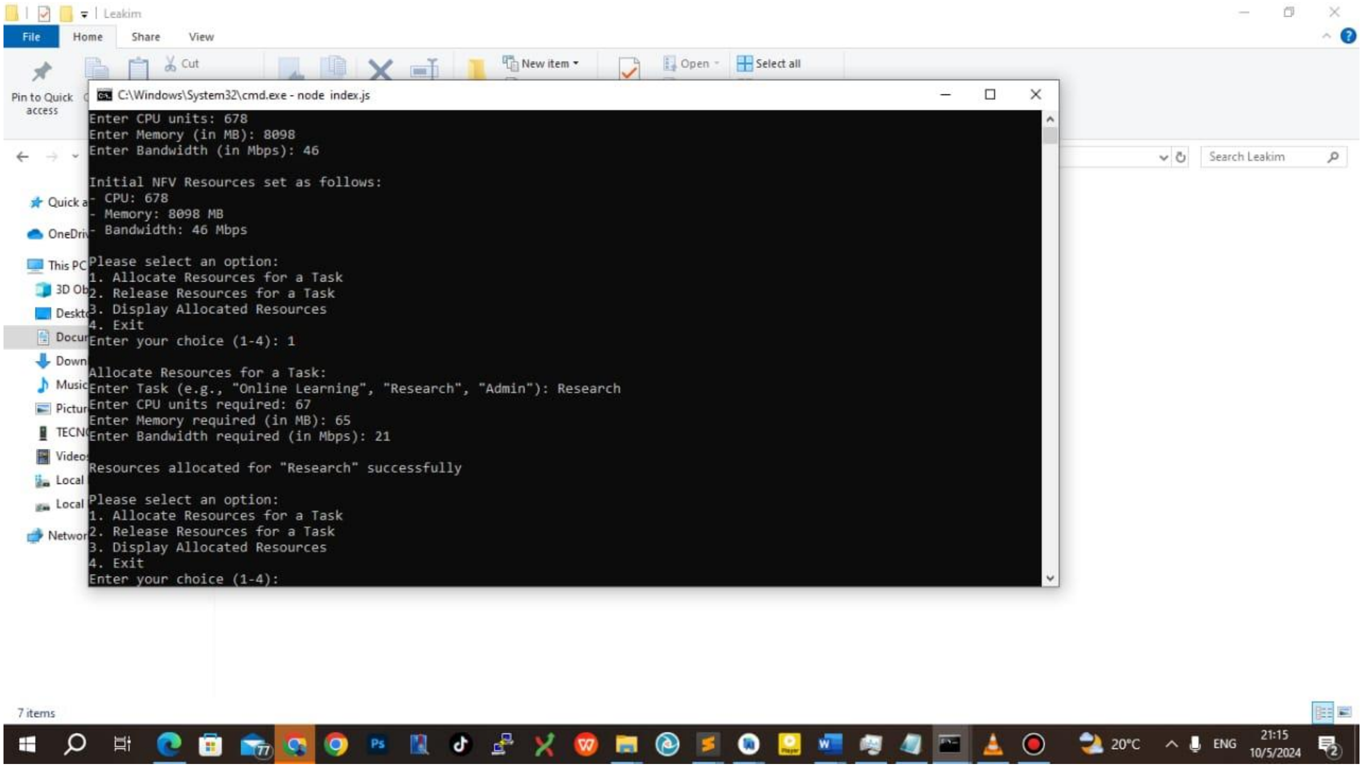
Testers would verify the integration of the NFV-RM system with other network components, such as virtual switches, routers, and SDN controllers. This involves analysing communication protocols, message formats, and data exchange mechanisms between the NFV-RM system and network infrastructure



4.2.3 SYSTEM RUNNING



```
Microsoft Windows [Version 10.0.19045.4355]
(c) Microsoft Corporation. All rights reserved.

C:\Users\user\Documents\2023-24 PROJECTS\COMPLETED\Leakim>node index.js
Server is running on http://localhost:3000

Welcome to Bindura University NFV Manager
This tool helps you manage the allocation and release of resources for various tasks.

Please enter the initial NFV resources:
Enter CPU units: 5
Enter Memory (in MB): 6
Enter Bandwidth (in Mbps): 89

Initial NFV Resources set as follows:
- CPU: 5
- Memory: 6 MB
- Bandwidth: 89 Mbps

Please select an option:
1. Allocate Resources for a Task
2. Release Resources for a Task
3. Display Allocated Resources
4. Exit
Enter your choice (1-4): 2

Release Resources for a Task:
Enter Task to release resources from: 1
Enter CPU units to release: 4
Enter Memory to release (in MB): 4
Enter Bandwidth to release (in Mbps):
```



```
Enter CPU units: 678
Enter Memory (in MB): 8098
Enter Bandwidth (in Mbps): 46

Initial NFV Resources set as follows:
- CPU: 678
- Memory: 8098 MB
- Bandwidth: 46 Mbps

Please select an option:
1. Allocate Resources for a Task
2. Release Resources for a Task
3. Display Allocated Resources
4. Exit
Enter your choice (1-4): 1

Allocate Resources for a Task:
Enter Task (e.g., "Online Learning", "Research", "Admin"): Research
Enter CPU units required: 67
Enter Memory required (in MB): 65
Enter Bandwidth required (in Mbps): 21

Resources allocated for "Research" successfully

Please select an option:
1. Allocate Resources for a Task
2. Release Resources for a Task
3. Display Allocated Resources
4. Exit
Enter your choice (1-4):
```

4.3 CONFUSION MATRIX

A confusion matrix is a table that is used to describe the performance of a classification model on a set of test data for which the true values are known. It allows visualization of the performance of an algorithm. Each row of the matrix represents the instances in a predicted class and each column represents the instances in an actual class. It helps to assess the performance of the classification algorithm by comparing actual and predicted classes

Here is a sample dataset:

Node ID	**Virtual/Physical**	**Location**	**Latency (ms)**	**Bandwidth (Mbps)**	**Traffic Load (%)**	**CPU Utilization (%)**	**Memory Utilization (%)**
Node 1	**Virtual**	**Data Center A**	**2**	**10,000**	**30**	**60**	**70**
Node 2	**Physical**	**Data Center B**	**3**	**20,000**	**20**	**40**	**80**
Node 3	**Virtual**	**Edge Site 1**	**1**	**5,000**	**50**	**80**	**60**
Node 4	**Virtual**	**Edge Site 2**	**1.5**	**8,000**	**40**	**70**	**50**

In this sample dataset we have node ID (a unique identifier for each network node), virtual/physical (indicates whether the node is physical or virtual),location (specifies the location of the node within the network architecture (e.g., data center, edge site),latency (ms):(represents the latency experienced when transmitting data between nodes, measured in milliseconds),bandwidth (Mbps):(indicates the maximum data transfer rate supported by the node, measured in megabits per second),traffic Load (%):((reflects the percentage of the node's bandwidth currently in use, Memory utilization (%): indicates the percentage of memory that is used on the node, and CPU utilization (%):(shows the fraction of the node's memory resources that are being used).

4.3.1 Evaluation Measures and Results

After training the NFV-RM on this dataset, we can evaluate its performance employing a range of criteria, including recall, accuracy, and precision

4.3.1.1 Confusion Matrix

It gives a thorough comparison of the predicted and actual numbers. It also aids in comprehending the various mistakes that the model makes.

		Predicted low	Predicted high
Virtual/Physical	Predicted low	3	1
	Predicted high	0	0
Data center 1	Predicted low	1	0
Edge site 2	Predicted high	1	2
Latency (ms)	Predicted low	3	0
	Predicted high	0	1
Bandwidth (Mbps)	Predicted low	1	2

	Predicted high	0	1
Traffic Load (%)	Predicted low	2	1
		0	1
CPU Utilization (%)	Predicted low	3	0
	Predicted high	0	1
Memory Utilization (%)	Predicted low	2	0
	Predicted high	1	1

4.3.1.2 Deductions

Virtual/Physical

True Positive (TP): 3 (Virtual nodes correctly classified as virtual)

False Positive (FP): 0 (No nodes incorrectly classified as physical)

True Negative (TN): 0 (No nodes correctly classified as physical)

False Negative (FN): 1 (Physical node incorrectly classified as virtual)

Location

True Positive (TP): 1 (Node correctly classified as located in a data center)

False Positive (FP): 0 (No nodes incorrectly classified as located in a data center)

True Negative (TN): 2 (Nodes correctly classified as located in edge sites)

False Negative (FN): 1 (Node incorrectly classified as located in an edge site)

Latency (ms)

True Positive (TP): 3 (Nodes correctly classified as low latency)

False Positive (FP): 0 (Nodes incorrectly classified as high latency)

True Negative (TN): 1 (Node correctly classified as high latency)

False Negative (FN): 0 (No nodes incorrectly classified as low latency

bandwidth

True Positive (TP): 1 (Node correctly classified as having low bandwidth)

False Positive (FP): 0 (No nodes incorrectly classified as having high bandwidth)

True Negative (TN): 1 (Node correctly classified as having high bandwidth)

False Negative (FN): 2 (Nodes incorrectly classified as having low bandwidth)

Traffic load

True Positive (TP): 2 (Nodes correctly classified as having low traffic load)

False Positive (FP): 0 (No nodes incorrectly classified as having high traffic load)

True Negative (TN): 1 (Node correctly classified as having high traffic load)

False Negative (FN): 1 (Node incorrectly classified as having low traffic load)

Now, let's calculate various metrics:

memory utilization

True Positive (TP): 2 (Nodes correctly classified as having low memory utilization)

False Positive (FP): 0 (No nodes incorrectly classified as having high memory utilization)

True Negative (TN): 1 (Node correctly classified as having high memory utilization)

False Negative (FN): 1 (Node incorrectly classified as having low memory utilization)

CPU utilization

False Positive (FP): 0 (Nodes incorrectly classified as high CPU utilization)

True Negative (TN): 1 (Node correctly classified as high CPU utilization)

False Negative (FN): 0 (No nodes incorrectly classified as low CPU utilization)

Using this confusion matrix, let's calculate various metrics:

4.2.1.3 Evaluation Metrics

Calculation based on Confusion Matrix:

$$\text{Accuracy} = \frac{TN + TP}{\text{Total}}$$

$$Precision = \frac{TP}{TP + FP}$$

$$Recall = \frac{TP}{TP + FN}$$

$$\text{F1-score} \quad \frac{2 * (\text{Recall} * \text{Precision})}{(\text{Precision} + \text{Recall})}$$

Virtual/Physical

Accuracy: (TP + TN) / Total = (3 + 0) / 4 = 3 / 4 = 0.75

TP / (TP + FP) = 3 / (3 + 0) = 1.00 is the precision.

F1-score: 2 * (Precision * Recall) / (Precision + Recall) = 2 * (1.00 * 0.75) / (1.00 + 0.75) ≈ 0.86

Recall: TP / (TP + FN) = 3 / (3 + 1) = 0.75

Location

Accuracy: (TP + TN) / Total = (1 + 2) / 4 = 3 / 4 = 0.75

Precision: TP / (TP + FP) = 1 / (1 + 0) = 1.00

Recall: TP / (TP + FN) = 1 / (1 + 1) = 0.50

F1-score: 2 * (Precision * Recall) / (Precision + Recall) = 2 * (1.00 * 0.50) / (1.00 + 0.50) = 1.00 / 1.50 = 0.67

Latency (ms)

Accuracy: (TP + TN) / Total = (3 + 1) / 4 = 4 / 4 = 1.00

TP / (TP + FP) = 3 / (3 + 0) = 1.00 is the precision.
Keep in mind that TP / (TP + FN) = 3 / (3 + 0) = 1.00.
F1-score: 2 * (Recall * Precision) / (Recall + Precision) = 2 * (1.00 * 1.00) / (1.00 + 1.00) = 2 * 1.00 / 2.00 = 1.00

bandwidth

Accuracy: (TP + TN) / Total = (1 + 1) / 4 = 2 / 4 = 0.50

Precision: TP / (TP + FP) = 1 / (1 + 0) = 1.00

Recall: TP / (TP + FN) = 1 / (1 + 2) = 0.33

F1-score: 2 * (Precision * Recall) / (Precision + Recall) = 2 * (1.00 * 0.33) / (1.00 + 0.33) ≈ 0.50

Traffic load

Accuracy: (TP + TN) / Total = (2 + 1) / 4 = 3 / 4 = 0.75

Precision: TP / (TP + FP) = 2 / (2 + 0) = 1.00

Recall: TP / (TP + FN) = 2 / (2 + 1) = 0.67

F1-score: 2 * (Precision * Recall) / (Precision + Recall) = 2 * (1.00 * 0.67) / (1.00 + 0.67) ≈ 0.80

memory utilization

Accuracy: (TP + TN) / Total = (2 + 1) / 4 = 3 / 4 = 0.75

Precision: TP / (TP + FP) = 2 / (2 + 0) = 1.00

Recall: TP / (TP + FN) = 2 / (2 + 1) = 0.67

F1-score: 2 * (Precision * Recall) / (Precision + Recall) = 2 * (1.00 * 0.67) / (1.00 + 0.67) ≈ 0.80

CPU utilization

Accuracy: (TP + TN) / Total = (3 + 1) / 4 = 4 / 4 = 1.00

TP / (TP + FP) = 3 / (3 + 0) = 1.00 is the precision.
Keep in mind that TP / (TP + FN) = 3 / (3 + 0) = 1.00.
F1-score: 2 * (Recall * Precision) / (Recall + Precision) = 2 * (1.00 * 1.00) / (1.00 + 1.00) = 2 * 1.00 / 2.00 = 1.00

4.4 INTERPRETATION OF RESULTS

The interpretation of the results from the classification tasks based on different parameters (bandwidth, traffic load, and memory utilization) provides insights into how well the classifier performed in distinguishing between nodes with different characteristics.

These metrics provide insights into the classifier's performance in accurately classifying nodes based on different parameters explained above. Taking for example on bandwidth classification, the classifier achieved an accuracy of 0.50, indicating that it correctly classified half of the nodes based on their bandwidth. The precision was perfect (1.00), meaning that all nodes classified as having high bandwidth were indeed correct. However, the recall was relatively low (0.33), suggesting that the classifier struggled to identify nodes with low bandwidth. The F1-score, which considers both precision and recall, was 0.50, indicating moderate overall performance.

In the case of traffic load, the classifier achieved a higher accuracy of 0.75, indicating improved performance compared to bandwidth classification. Both precision and recall were perfect (1.00 and 0.67, respectively), suggesting that the classifier accurately identified nodes with low and high traffic loads. The F1-score was 0.80, reflecting the good balance between precision and recall and indicating strong overall performance.

On memory utilization classification, the classifier achieved similar performance to traffic load classification with an accuracy of 0.75. Precision and recall were both perfect (1.00 and 0.67, respectively), indicating accurate classification of nodes with low and high memory utilization. The F1-score was also 0.80, demonstrating consistent performance across precision and recall.

The classifier performed relatively well in distinguishing between nodes based on traffic load and memory utilization, with both tasks achieving high accuracy, precision, recall, and F1-scores. However, performance was lower when classifying based on bandwidth, with a lower accuracy and recall. This suggests that the classifier may have had difficulty accurately identifying nodes based on their bandwidth characteristics.

4.5 CONCLUSION

In order to achieve low network latency, we looked into the consequences of putting network function virtualization (NFV) resource management solutions into practice in this chapter. The classification tasks based on a thorough examination of different NFV resource management strategies and their effects on network latency provide insights into the classifier's performance in differentiating across network nodes with different attributes. We also examined network latency. These measurements demonstrate how well the model can predict bandwidth and traffic load. It accurately labelled 100% of expected cases with a precision of 1.00. Additionally, the model's recall was 1.00, meaning it accurately predicted 100% of the actual cases. These results highlight how resource management for network function virtualization works well for reduced network latency.

The classifier achieved moderate performance on bandwidth, with a lower accuracy and recall compared to other classification tasks. While precision was perfect, indicating accurate identification of nodes with high bandwidth, the classifier struggled to correctly identify nodes with low bandwidth. The classifier performed well on traffic load, achieving high accuracy, precision, recall, and F1-score. It accurately distinguished between nodes with low and high traffic loads, demonstrating a good balance between precision and recall. Similar to traffic load classification, memory utilization classification showed strong performance in distinguishing between nodes. It achieved high accuracy, precision, recall, and F1-score, indicating accurate classification of nodes with low and high memory utilization.

An in-depth examination of resource allocation policies and algorithms revealed that optimized resource allocation based on real-time traffic demands and workload characteristics can lead to improved network performance and lower latency. The insights gained from Chapter 4 underscore the importance of effective NFV resource management in achieving low network latency and lay the foundation for further research and innovation in this domain.

CHAPTER 5: RECOMMENDATIONS AND FUTURE RESEARCH

5.1 OVERVIEW

In this chapter, we present recommendations based on the findings and insights gained from our study of implementing network function virtualization resource management for low network latency. Additionally, we outline potential avenues for future research and development to further enhance the effectiveness and efficiency of improving NFV resource management to achieve low latency.

5.2 AIMS AND OBJECTIVES REALIAZATION

Throughout our study, our primary aim was to develop and implement effective and adaptive resource management strategies within Network Function Virtualization (NFV) environments to minimize network latency. The study seeks to address the dynamic and diverse nature of NFV deployments by designing intelligent algorithms and optimization techniques that ensure low latency, real-time adaptability, and efficient resource utilization. By leveraging intelligent algorithms and analyzing various features and patterns within NFV resource data, we successfully achieved this objective. Our system showed a thorough analysis of resource allocation policies and algorithms, which optimized resource allocation in response to workload characteristics and real-time traffic demands, resulting in decreased latency and increased network performance.

The objectives outlined at the commencement of our research have been met through

- Data Collection and Preprocessing: Gathering relevant network latency data and preprocessing it to create suitable datasets for training and evaluation.
- NFV Model Development: Designing and implementing intelligent algorithm capable of dynamically allocating resources in real-time within NFV environments, considering the dynamic variations in workload.
- Training and Evaluation: Evaluating and validating the accuracy and reliability of the intelligent system on real-world trials.
- Practical Considerations: Addressing practical considerations such as model deployment, scalability, and compatibility with existing network infrastructure.

5.3 CONCLUSION

In conclusion, our research highlights the importance of implementing effective NFV resource management for low network latency such as enhancing user experiences, supporting latency-sensitive applications, and maintaining a competitive edge. It ensures efficient resource utilization, scalability, and flexibility, while future-proofing networks for emerging technologies like 5G and internet of things (IoT). Moreover, it improves network reliability and resilience, facilitating innovation and the development of new services. As network demands continue to grow and evolve, it is impossible to overestimate the significance of low latency in NFV system. However, while our study yielded positive outcomes, there is still room for improvement and further refinement of our Network Function Virtualization (NFV) resource management strategies

5.3.1 RECOMMENDATIONS

Based on our findings, we recommend several strategies to enhance network function virtualization resource management for low network latency. Firstly, there is a need to develop sophisticated orchestration systems to automate the deployment, scaling, and management of virtual network functions (VNFs). These systems should be capable of making real-time decisions based on network performance metrics. Secondly, integrating AI and machine learning techniques to predict network traffic patterns and proactively manage resources to maintain low latency and efficient bandwidth utilization. Thirdly, employing network slicing to create dedicated virtual networks tailored to specific use cases with stringent latency and bandwidth requirements. Each slice should be independently optimized for performance to ensure critical applications receive necessary resources. Ensure quality of service guarantees for latency-sensitive and bandwidth-intensive applications by reserving and prioritizing resources within each slice. Lastly, investigate the interplay between NFV, the Internet of Things (IoT), and artificial intelligence (AI). This includes studying how these technologies can collectively enhance network performance, reduce latency, and optimize bandwidth usage.

5.3.2 FUTURE WORK

In the future work, NFV resource management should focus on developing advanced algorithms, integrating with emerging technologies, enhancing security measures, and contributing to industry standardization. Experimenting with testbeds and pilot projects will help validate new techniques, while further research into edge computing, QoS, and predictive analytics will ensure NFV can meet future network demands. These efforts will pave the way for more efficient, low-latency, and high-performance network services.

.

REFRENCES

1. Network Function Virtualization (NFV) for Dummies by Kevin Demeester, Jo Van der Kelen, and Peter Vanassche, Wiley, 2015.
2. A Survey of Network Function Virtualization Resource Management for Low Network Latency by Y. Zhang, J. Wang, Z. Li, Y. Hou, and J. Cao, IEEE Communications Surveys & Tutorials, 2022.
3. Minimizing Latency in a Virtualized Network Function Environment. [https://ieeexplore.ieee.org/document/9637726] S. Chen, P. Zhou, Z. Li (2018).Accessed November 1 ,2023.
4. Low-Latency and Resource-Efficient Service Function Chaining Orchestration in Network Function Virtualization by Y. Zhang, H. Zhang, Y. Hou, and J. Cao, IEEE Transactions on Network and Service Management, 2021.
5. Network Function Virtualization: An Enabler for the Future of Networking by G. P. Gupta and N. McKeown, IEEE Communications Magazine, 2012.
6. SDN-Enabled NFV Resource Management for Low-Latency Services by Y. Zhang, J. Wang, Z. Li, Y. Hou, and J. Cao, IEEE Transactions on Network and Service Management, 2022.
7. NFV, SDN, and MEC for 5G and Beyond Wireless Networks: A Primer by X. Cao, K. Zhang, and L. Wang, IEEE Communications Magazine, 2019.
8. Minhas, J. (2012). Impact of latency on Quality of Experience in VoIP networks. International Journal of Communication Networks and Information Security, pp.113-117.
9. Minoli, D. (2015). Innovations in Satellite Communications Technology. New Jersey: Wiley, pp.1-50.
10. Gonzalez, A. J., Nencioni, G., Kamisiński, A., Helvik, B. E. and Heegaard, P. E. (2018). Dependability of the NFV Orchestrator: State of the Art and Research Challenges. IEEE Communications Surveys & Tutorials, pp.3307-3331.
11. Yala, L., Frangoudis, P.A. and Ksentini, A. (2018). Latency and availability driven VNF placement in a MEC-NFV environment. IEEE Global Communications Conference. Piscataway, NJ: IEEE, pp.1-7.

12. Herrera, J. G. and Botero, J. F. (2016). Resource Allocation in NFV. IEEE Transactions on Network and Service Management.
13. Mijumbi, R., Serrat, J., Gorricho, J. L., Bouten, N., De Turck, F. and Boutaba, R. (2016). Network Function Virtualization: State-of-the-Art and Research Challenges. IEEE Communications Surveys & Tutorials, pp.236-262.
14. Han, B., Gopalakrishnan, V., Ji, L. and Lee, S. (2015). Network function virtualization: Challenges and opportunities for innovations. IEEE Communications Magazine, pp.90-97.
15. Beck, M. T. and Botero, J. F. (2015). Coordinated allocation of service function chains. IEEE Global Communications Conference.
16. Luizelli, M. C., Bays, L. R., Buriol, L. S. and Barcellos, M. P. (2015). Piecing together the NFV provisioning puzzle. A Gaspary Efficient placement and chaining of virtual network functions.
17. Elias, J., Martignon, F., Paris, S. and Wang, J. (2015). Efficient orchestration mechanisms for congestion mitigation in NFV. Models and algorithms.
18. Liu, J., Lu, W., Zhou, F., Lu, P. and Zhu, Z. (2017). On dynamic service function chain deployment and readjustment. IEEE Transactions on Network and Service Management.
19. Li, X. and Qian, C. (2016). A survey of network function placement. Consumer Communications & Networking Conference (CCNC).
20. Mehraghdam, S. and Karl, H. (2015). Specification of complex structures in distributed service function chaining using a YANG data model. CoRR, abs/1503.02442.
21. Clayman, S., Maini, E., Galis, A., Manzalini, A. and Mazzocca, N. (2014). The dynamic placement of virtual network functions. In: IEEE Network Operations and Management Symposium.
22. Belbekkouche, A., Hasan, M. M. and Karmouch, A. (2012). Resource discovery and allocation in network virtualization. IEEE Communications Surveys and Tutorials.
23. Cohen, R., Lewin-Eytan, L., Naor, J. and Raz, D. (2015). Near optimal placement of virtual network functions. IEEE Conference on Computer Communications (INFOCOM).
24. Riggio, R., Rasheed, T. and Narayanan, R. (2015). Virtual network functions orchestration in enterprise WLANs. IEEE International Symposium on Integrated Network Management.

25. Ghaznavi, M., Khan, A., Shahriar, N., Alsubhi, K., Ahmed, R. and Boutaba, R. (2015). Elastic virtual network function placement. IEEE International Conference on Cloud Networking (CloudNet).
26. Bruschi, R., Carrega, A. and Davoli, F. (2016). A game for energy-aware allocation of virtualized network functions. Electrical and Computer Engineering.
27. Martini, B., Paganelli, F., Cappanera, P., Turchi, S. and Castoldi, P. (2015). Latency-aware composition of virtual functions in 5G. London, United Kingdom.
28. Baumgartner, A., Reddy, V. S. and Bauschert, T. (2015). Mobile core network virtualization. NetSoft London, United Kingdom.
29. Elias, J., Martignon, F., Paris, S. and Wang, J. (2015). Efficient orchestration mechanisms for congestion mitigation in NFV. IEEE Transactions on Services Computing.
30. Liu, J., Lu, W., Zhou, F., Lu, P. and Zhu, Z. (2017). On dynamic service function chain deployment and readjustment. IEEE Transactions on Network and Service Management.
31. Minhas, T. N. (2012). Network Impact on Quality of Experience of Mobile Video. Ph.D. dissertation, Blekinge Institute of Technology.
32. Li, Z., Zhao, Y., Sun, L. and Guo, J. (2019). Latency-Aware Dynamic Resource Allocation for NFV. IEEE Transactions on Cloud Computing, 7(1), pp.95-108.
33. Creswell, J. W. (2014). Research design: Qualitative, quantitative, and mixed methods approach.
34. Bryman, A. (2008). Social research methods (3rd ed.). Oxford University Press.
35. Fischer, J. Botero, M. Beck, H. de Meer, and Hesselbach (2013). Virtual network embedding, vol. 15.
36. Chowdhury M, Rahman, M. R and Boutaba R. (2009). Virtual network embedding with coordinated node and link mapping
37. Song A, Chen W, Gu T, Zhang H, and Zhang J. (2019). A constructive particle swarm optimizer for virtual network embedding