Computers, Materials & Continua
DOI: 10.32604/cmc.2023.035960
Article

Tech Science Press

Performance Evaluation of Virtualization Methodologies to Facilitate NFV Deployment

Sumbal Zahoor[1], Ishtiaq Ahmad[1], Ateeq Ur Rehman[2], Elsayed Tag Eldin[3], Nivin A. Ghamry[4] and Muhammad Shafiq[5],*

[1]Department of Electrical Engineering, The University of Lahore, Lahore, 54000, Pakistan
[2]Department of Electrical Engineering, Government College University, Lahore, 54000, Pakistan
[3]Faculty of Engineering and Technology, Future University in Egypt, New Cairo, 11835, Egypt
[4]Faculty of Computers and Artificial Intelligence, Cairo University, Giza, 3750010, Egypt
[5]Department of Information and Communication Engineering, Yeungnam University, Gyeongsan, 38541, Korea
*Corresponding Author: Muhammad Shafiq. Email: shafiq@ynu.ac.kr

Received: 11 September 2022; Accepted: 04 November 2022

Abstract: The development of the Next-Generation Wireless Network (NGWN) is becoming a reality. To conduct specialized processes more, rapid network deployment has become essential. Methodologies like Network Function Virtualization (NFV), Software-Defined Networks (SDN), and cloud computing will be crucial in addressing various challenges that 5G networks will face, particularly adaptability, scalability, and reliability. The motivation behind this work is to confirm the function of virtualization and the capabilities offered by various virtualization platforms, including hypervisors, clouds, and containers, which will serve as a guide to dealing with the stimulating environment of 5G. This is particularly crucial when implementing network operations at the edge of 5G networks, where limited resources and prompt user responses are mandatory. Experimental results prove that containers outperform hypervisor-based virtualized infrastructure and cloud platforms' latency and network throughput at the expense of higher virtualized processor use. In contrast to public clouds, where a set of rules is created to allow only the appropriate traffic, security is still a problem with containers.

Keywords: NFV; hypervisors; cloud computing; containers

1 Introduction

We live in an era of the Internet of Everything (IoE), with exponential growth in the number of new operators and billions more to come as 5G systems evolve and become more accessible. In comparison to the existing commercial 4G cell design, it is expected that NGWN will need to deliver new features and time limitations by 2022 [1]. Even though the idea and goals of NGWNs are strong, the exploration demands for the organization of systems, application developments, and empowering advancements are still present. This encourages international decisions from the government, academia, and significant industries to offer creative solutions and identify fundamental research questions, particularly regarding agility, scalability, and security. Concepts of softwarization and virtualization of resources and services are undeniably among the driving factors of NGWN. They will provide a network strategy, enabling network flexibility and agility, besides supporting network maintenance, and updating all networks with ease. Paradigms like SDN, NFV, big data, and machine learning have recently been positioned as emerging technologies and important 5G components. However, with the commercialization of 5G, scientists are wondering how far NFV and SDN can assist as the two are interrelated but distinct technologies advancing the digitalization of network infrastructure in the telecom industry [2].

NFV, an innovation created by the European Telecommunication Standards Institute (ETSI), will play a prominent part in executing the idea of 5G [3]. To speed up the creation of new network services with flexible scale and automation, NFV incorporates cloud and virtualization technologies such as virtual machines (VMs), containers, and unikernels to deliver virtualized network tasks. The significant benefits of virtualization technologies have pushed Telecommunication Service Providers (TSPs) to select them as a solution for service provision. Because each virtualization platform has unique benefits and drawbacks, it is necessary to evaluate its achievement.

Prior articles [4–7] examined the concept of virtualization technologies conceptually, providing the scientific community with a broad understanding of the subject. Furthermore, a few articles [8–11] attempted to evaluate the performance of containers, hypervisors, and explicit cloud platforms but the analysis and comparisons were limited. To the best of our knowledge, we are the first to perform an in-depth analysis of virtualization techniques (such as containers, private clouds, and public clouds) using a unified platform to assess the potential of virtualization techniques using important network attributes such as latency, throughput, and processor utilization through a variety of experiments, which will act as a base for the positioning of the challenging concepts of future networks.

The key contribution of this paper is summarized as follows:

- We propose a framework that uses the leading hypervisors available at present, including VMware vSphere, Microsoft Hyper-V, and Citrix XenServer, to examine network virtualization performance in various scenarios, including labeled and unlabeled connections.
- Another goal of this performance analysis study is to build a prototype system to compare the efficiency of various cloud computing platforms, such as Amazon Web Services and Microsoft Azure.
- We then established a foundation for evaluating the Docker container's effectiveness in a similar configuration, followed by thoroughly examining all discussed virtualization technologies to determine their competency.

The structure of the article is as follows. Section 2 summarizes and discusses the relevant literature. Section 3 provides an overview of NFV's significance. The virtualization methods needed for network virtualization are organized in Section 4. Experimental arrangements are presented in Section 5. The results are precisely summarized in Section 6. The paper concluded in Section 7.

2 Related Works

Different NFV-associated research projects have explored the possibilities for future networking environments to fulfill various objectives. In [12], the authors addressed NFV needs, design goals, and important considerations critical to accelerating NFV implementation. The authors [13] investigated the efficacy of Remote Desktop Virtualization (RDV) based on Hyper-V and connected it to remote desktop services. For all cases, they ran memory, CPU, and storage management tests. They concluded that Relational Database Service (RDS) memory and storage organization produced better results than CPU management, which produced poor outcomes.

The authors [14] have offered a paradigm for an NFV virtual system environment, which specifies the architecture where VMs generated and maintained for NFV purposes; the created VMs may utilize to build VNFs. The authors [15] used four unique strategies: hyperthreading, virtual CPU core allocation, virtual CPU affinity, and virtual CPU isolation, all of which provide recommendations for improving VM and container performance and overall speed. In [16], authors applied different samples from the three largest cloud computing industry leaders and their well-known products. The use of serverless services depends on the intended scope and operational circumstances.

Additionally, several studies have proposed simulation tools for testing NFV capabilities and applications. In [17], authors described the mini-nfv framework for NFV Orchestration, which enables the deployment and management of VNFs and network services on Mininet using common TOSCA NFV templates built on the ETSI MANO architectural framework. It is an all-purpose VNF Manager developed to ease the transition from virtualized to real-world exploration by bringing agility, standardization, and replicability to the NFV experimentation world. The paper [18] established a system model that integrates SDN and NFV to facilitate resource abstractions and offer NFV-enabled edge FL collection servers for improving integration and stability. The epsilon-greedy algorithm, which utilizes a deep neural network (DNN) value function approximator, balances development and exploration. The proposed scheme outperformed in terms of Quality of Service (QoS) performance statistics like packet decline ratio, packet drop amounts, packet transfer ratio, lag, and throughput.

3 Significance of Network Function Virtualization

Network virtualization embraced by 5G as a powerful technique both in the access and core network reliant on conventional server's virtualization methods, such as those found in business IT, provides a solid basis for NGWNs by lowering equipment costs and increasing functional efficiency. The concept is that the user plane must separate from the control plane at every point in the relevant network areas. This separation of the user and control planes is also one of the fundamental underlying tenants of SDN/NFV (i.e., network virtualization) [19].

NFV uses virtualization features to transform network functions into virtualized networks that may integrate to provide various network services [20]. These services are installed as virtual machines (VMs) on physical servers, allowing operators to run their networks on shared rather than private servers, which lowers operational and capital expenditures. Network functions like fixed and mobile networks can benefit from NFV. The most well-known NFV applications include video servers, network slicing, service delivery, network monitoring, content delivery networks, and several security features like firewalls, intrusion prevention, and detection systems [21]. Standards for NFV layout developed with the support of the NFV architecture suggested by the ETSI presented in Fig. 1.

To support improved interoperability, each element of the design is based on these standards; the following are the components of NFV architecture:

- VNFs: software programs inside VMs provide network services, including file allocation, directory amenities, and IP setup. A single VNF can position over multiple VMs.
- Network Functions Virtualization Infrastructure (NFVI): involves network equipment and software necessary for telecom operators to build and connect virtual network services such as hypervisors or containers.

- Management, Automation, and Network Orchestration (MANO): a strategy for managing NFV architecture and adding new VNFs.

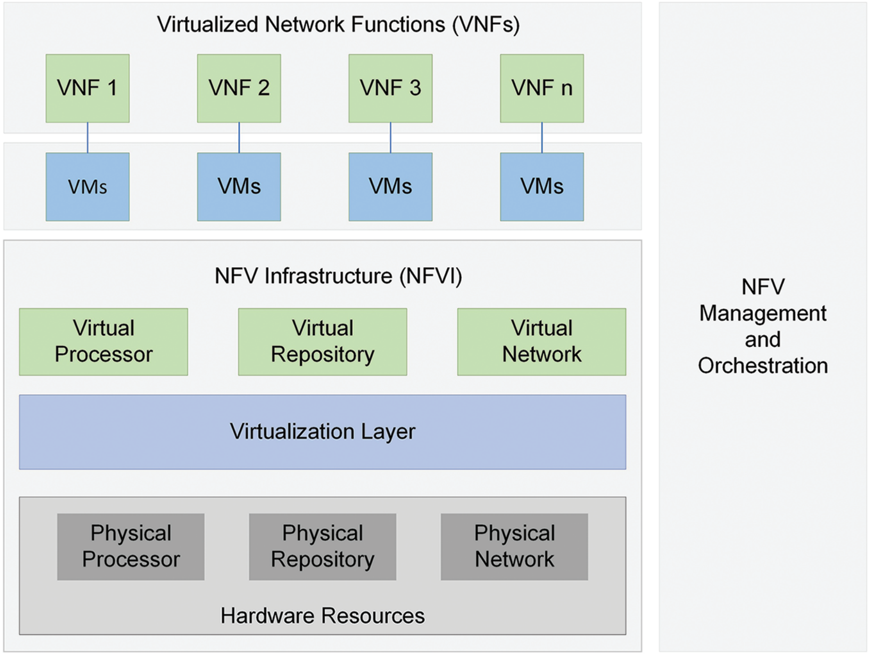

Figure 1: ETSI's proposed NFV conceptual framework

VNFs are critical for NFV since most VNFs run on hypervisors. NFV architecture also contains a framework for MANO of VNF components and hardware and software-based processing, storage, and networking hardware that enables network services to be virtualized. As a result, NFV design is no longer confined to VNFs or software [22]. Each virtualization platform, however, has unique benefits and drawbacks, so it is worthwhile to investigate how well they function.

4 Categories of Virtualization Technologies

Communication networks are rapidly incorporating the cloud model, revolutionizing the IT world by offering adaptability and increased efficiency through infrastructure control. This section provides an overview of various NFV-enabling virtualization technologies currently utilized in the cloud and at the edge to isolate many programs running on the same base server.

4.1 Hypervisors

The hypervisor is the underappreciated powerhouse for Communications Service Providers (CSPs) who are expanding their RANs to handle 5G better. Cloud computing is hypothetical without virtualization, but virtualization is impossible without the hypervisor. This thin layer of software is the foundation for the whole cloud ecosystem that facilitates running systems to collaborate while sharing the same physical computer resources. The NFV approach enables network expansion without including new hardware by virtualizing all physical networking resources under a hypervisor.

Multiple VMs successfully operated and seen on the same host by using a hypervisor layer to increase utilization and maximizes productivity. Regardless of the benefits, employing hypervisors imposes constraints that drastically slow down servers, even though they are designed to enhance resource allocation. Typically, there are two types of hypervisors. It includes type-1 and type-2 hypervisors. Although they are executed differently, both types accomplish the same task. The type-1 hypervisor can control the guest OS directly on the host's hardware, whereas type-2 hypervisors are hosted on the primary operating system. We have selected three of the most well-known type-1 hypervisors in the market, including VMware vSphere, Microsoft Hyper-V, and Citrix XenServer, as they all run directly on top of the host's physical layer and are well-known for their competence in virtualization and cloud computing [23].

4.2 Containers

Containers are designed to deliver a better and more flexible program execution as they operate on less expensive hardware and make better use of resources and separate function contexts. Applications can work independently within the same OS without hypervisors or VMs. A cloud native network function (CNF) is a network function that is intended and built to run within containers. Each program uses a distinctive set of resources without affecting the server's overall performance. As a result, they are ideal for companies that run various tasks concurrently on a single server.

To validate the performance metrics used in our study, we used a Docker container, an operating system released in 2013. Docker containers are more user-friendly and prevalent than other container operating systems. The most sustainable solutions integrate containers and hypervisors into a single architecture because each technology has benefits and drawbacks of its own. Therefore, the functionality of hypervisors is not replaced by containers; it is somewhat enhanced [24].

4.3 Cloud Computing

Virtualization and cloud computing are rapidly growing, with new features constantly added. Public clouds' extraordinary technological and financial benefits have recently accelerated their adoption with the help of cloud computing; consumers can lease their IT equipment instead of buying it [25]. This article examines the performance of the cloud services provided by the two most well-known public cloud providers.

- Amazon AWS: division of the massive technology and e-commerce company Amazon. It provides over two hundred services from data centers worldwide and has millions of users, including startups, giant corporations, and prominent government organizations.
- Microsoft Azure: is a leading cloud computing service that facilitates the management of apps using data centers run by Microsoft. The platform offers a wide range of cloud services, comprising networking, statistics, computation, and memory.

Although VNFs are increasingly used in conventional network designs, they still have drawbacks, such as the capability to optimize VMs as digital service providers strive to have more flexible services. Both CNFs and VNFs replace specific physical devices in telco cloud functions. However, CNFs are intended and constructed to operate within containers as an evolution from VNFs [26,27].

5 Experimental Evaluation

The proposed framework lays out the implications and specifications for creating scenarios to investigate how VNFs and CNFs function to offer crucial cost-effective, hardware- and energy-saving solutions. Utilizing IT virtualization techniques, NFV breaks down entire classes of network node functions into modular components that can be connected or chained together to produce and deliver communication services. The following significant network attributes were included in the objective classifications of our research design:

- Latency (Round Trip Time (RTT)): a statistic that counts in milliseconds (ms) the time it takes to send a data packet and the time it takes to obtain the response to that signal.
- Processor usage: the amount of time that a processor is being used overall.
- Throughput: calculates the number of jobs accomplished in time.

The Windows network throughput benchmark tool (Ntttcp) was used in all three setups to measure network throughput and processing time. However, the Ping and Address Resolution Protocol (ARP) tests to address the latency.

5.1 Case 1: Hypervisor-Based Performance Metrics

VMs are essential components of a 5G network because they allow present monolithic network topologies to further split into customizable microservices. IT departments of all sizes have used VMs to reduce expenses and increase productivity since they are isolated and safer.

- Citrix XenServer: initially, we installed XenServer 6.2.0 on two separate physical machines. This edition comes with Open vSwitch 1.4.6 and may only establish on 64-bit workstations. Since XenCenter has centralized control, we needed one device to communicate after installing it and connecting it to the XenServer.
- Microsoft Hyper-V: unlike XenServer, it is a licensed and closed-source hypervisor that cannot be configured independently. The Windows server provides a built-in Graphical User Interface (GUI) in contrast to XenServer and vSphere, which are command-line Linux interfaces that require the XenCenter and vSphere client GUIs to communicate with the server. The Hyper-V manager is employed to make virtual connections labeled with Virtual Local Area Network (VLAN) IDs.
- VMware vSphere: VMware vSphere ESXi is free for non-commercial use on the VMware platform. We installed the vSphere Client on another machine with the same characteristics linked with the server through the IP address and administrative credentials.

Table 1 shows the hardware requirements for VMs, hypervisors, and the software and tools required to connect with them.

Table 1: Hypervisor hardware and software configurations

Component	Hardware	Software
Server (XenServer)	Intel core i3, 4 GB RAM, 500 GB disk	Citrix XenServer 6.2 open vSwitch
Server (Hyper-V)	Intel core i3, 4 GB RAM, 500 GB Disk	Windows server 2012
Server (vSphere)	Intel core i3. 4 GB RAM, 500 GB disk	VMware vSphere ESXi 5.0
Virtual machine (Windows)	Two i3 cores, 1 GB RAM, 100 GB disk	Windows 7 (64-bit)
Virtual machine (Linux)	Two i3 cores, 1 GB RAM, 100 GB disk	Ubuntu 12.04 (64-bit)
Remote host	Intel core i3, 4 GB RAM, 500 GB disk	Windows 7 (64-bit) XenCenter vSphere center

Each server is set up with two VMs running Windows and Linux. The basic layout of the experiment is depicted in Fig. 2. VMs in the cloud do not have direct access to virtual network ports; instead, they are linked to the software layer, which connects VMs on the same host or to the outside world via physical and Virtual Network Interfaces (VIFs). A crucial aspect of network security is isolation, and we may use VLANs to divide traffic among devices connected to the same physical network.

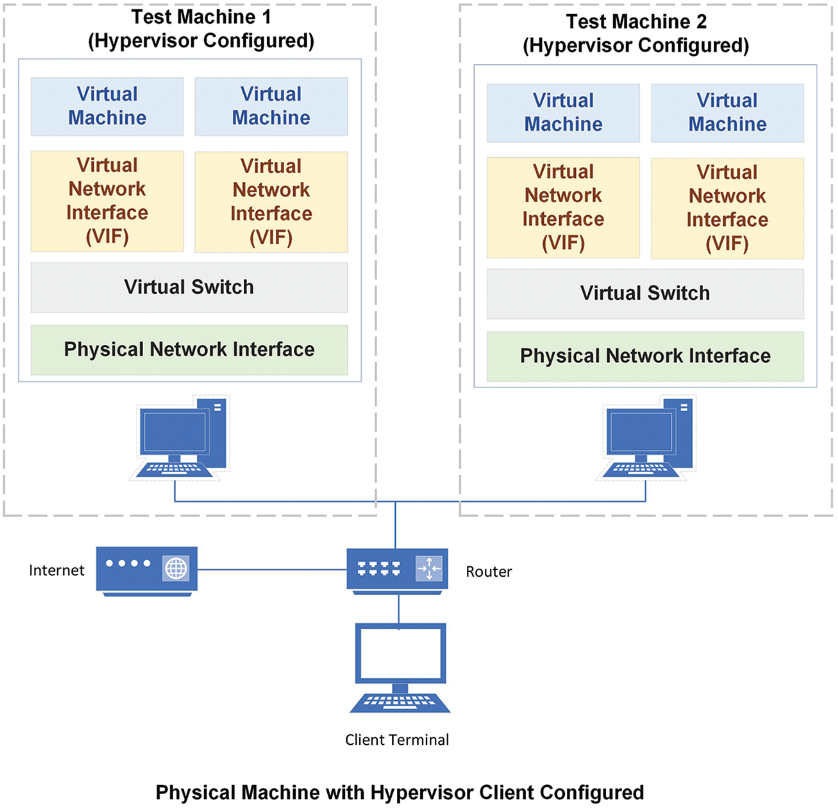
Test Machine 1
(Hypervisor Configured)
Virtual Machine
Virtual Machine
Virtual Network Interface (VIF)
Virtual Network Interface (VIF)
Virtual Switch
Physical Network Interface
Test Machine 2
(Hypervisor Configured)
Virtual Machine
Virtual Machine
Virtual Network Interface (VIF)
Virtual Network Interface (VIF)
Virtual Switch
Physical Network Interface
Internet
Router
Client Terminal
Physical Machine with Hypervisor Client Configured

Figure 2: An illustration of the experimental setup using hypervisors

Tables 2 and 3 demonstrate hypervisor and VMs placement factors.

Table 2: Hypervisors' network configurations

Network settings	Hypervisor host 1	Hypervisor host 2
IP address	10.50.2.10	10.50.2.20
Subnet mask	255.255.254.0	255.255.254.0
Access (gateway)	10.50.2.1	10.50.2.1
Time region	Asia/Karachi	Asia/Karachi

Table 3: Specifications of deployed VMs

	Virtual machines			
Operating systems	Ubuntu (1)	Windows (2)	Ubuntu (3)	Windows (4)
Hypervisor	Host (1)	Host (1)	Host (2)	Host (2)
IP address	10.50.2.11	10.50.2.12	10.50.2.21	10.50.2.22
Subnet mask	255.255.254.0	255.255.254.0	255.255.254.0	255.255.254.0
Default time zone	Asia/Karachi	Asia/Karachi	Asia/Karachi	Asia/Karachi

The tests were carried out in three different scenarios: (1) unlabeled VLAN connections, (2) VMs with the same labels on every server, and (3) various labeled connections attached to the VMs. It is interesting to note that the outcomes in each case are comparable. In any of these situations, a VM cannot communicate with another VM unless both are linked to the network via a VIF with and without labeling.

- Network throughput: TCP and UDP data streams are employed to check the quality of virtual networks. On the same hypervisor, TCP performance without a VLAN connection was better than that of a labeled link. Whereas vSphere achieved a transmission rate of 11.31 Mbps without labeling, this completely matched the non-virtual throughput. There was no noticeable difference in UDP throughput between hypervisors with or without labels. This implies that enabling network virtualization or security does not affect throughput inside the same hypervisor, as indicated in Fig. 3.

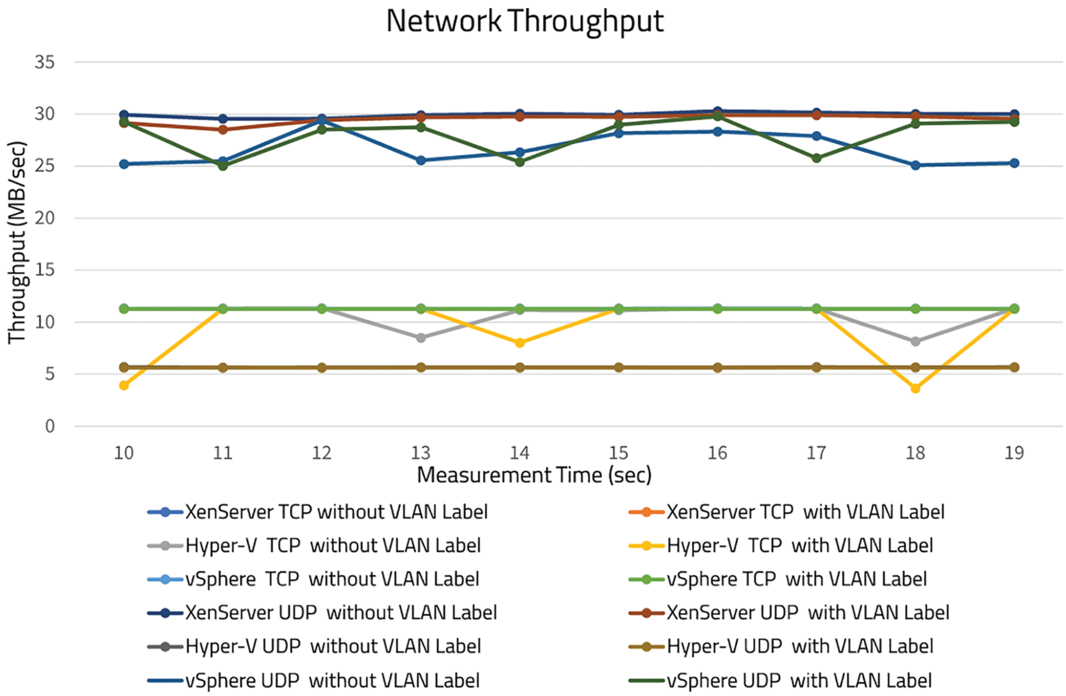

Figure 3: Plots of TCP and UDP throughput

- Latency: two situations were examined (1) both VMs hosted on the same hypervisor server, and (2) hosted on separate hypervisor servers. The findings indicated that the RTT in the event (2) was more significant than the RTT in the event (1) because both virtual switch control and actual network operations need time. When compared to vSphere, XenServer's average RTT was 0.38 ms, while vSphere and Hyper-V produced comparable results, as shown in Fig. 4.

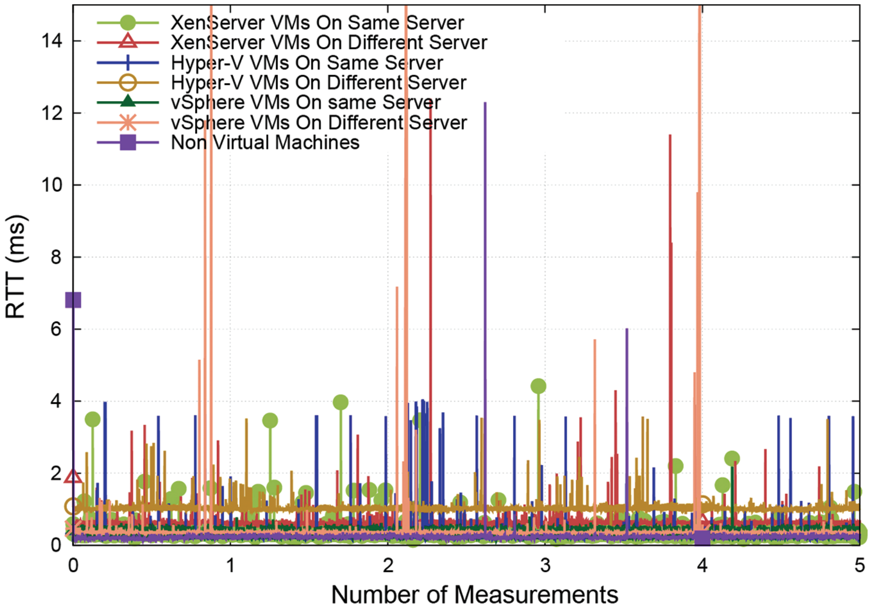

Figure 4: An illustration of RTT plots for (1) when both VMs hosted on the same hypervisor server and (2) when they hosted on separate hypervisor servers

- Processor utilization: During TCP throughput, both labeled and unlabeled connections in vSphere consumed the most processing power, while XenServer consumed the least. On the other hand, unlabeled connections used up more processors than labeled connections. Fig. 5 shows that increased UDP throughput resulted in increased CPU usage.

5.2 Case 2: Cloud Computing-Based Performance Metrics

To compare the experimental investigations with our private cloud setup, we leased two of the industry's most prominent cloud servers, Azure, and AWS, to create an equivalent situation on public clouds. To analyze VNF and network performance, we selected the Ireland location and EC2 instances to function as VMs in AWS. Table 4 indicates the configurations used to frame AWS and Azure.

Different security groups were established and arranged to manage all network firewalls and security rules. Fig. 6 illustrates the overall layout of AWS.

To avoid conflicts during Azure investigations, Windows and Linux VMs created, and the VM virtual network interface was set to accept all types of inbound data. We used a remote desktop to log in to our Windows instances and SSH to access our Linux VMs. The overall arrangement of Azure is presented in Fig. 7.

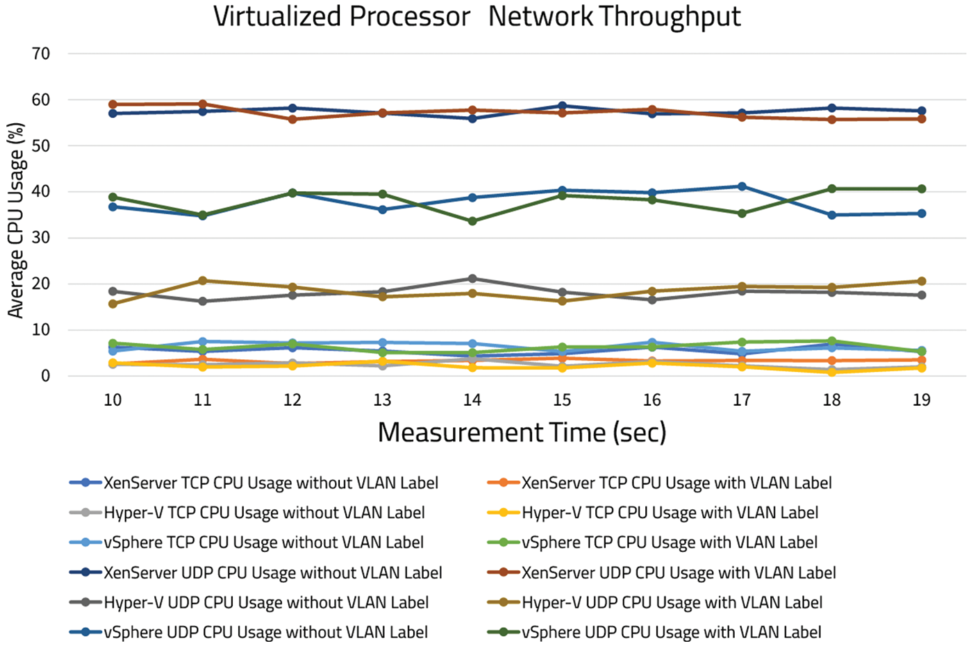

Figure 5: Average CPU usage for TCP and UDP using hypervisors

Table 4: Specification of Amazon AWS EC2 and Azure VM setup

	Amazon AWS EC2 configurations	
Operating system	Windows server	Ubuntu server
Instance type	T2 medium	T2 micro
Zone	EU-West-1b (VM 1) EU-West-1a (VM 2)	EU-West-1b (VM 1) EU-West-1a (VM 2)
CPUs	2	1
RAM	4 GB	1 GB
Storage	30 GB SSD	30 GB SSD
	Azure VM configurations	
Instance Type	Standard_B2s	Standard_B2s
Zone	Zone 1 (VM1) Zone 2 (VM2)	Zone 1 (VM1) Zone 2 (VM2)
CPUs	2	2
RAM	4 GB	4 GB
Storage	127 GB	30 GB SSD

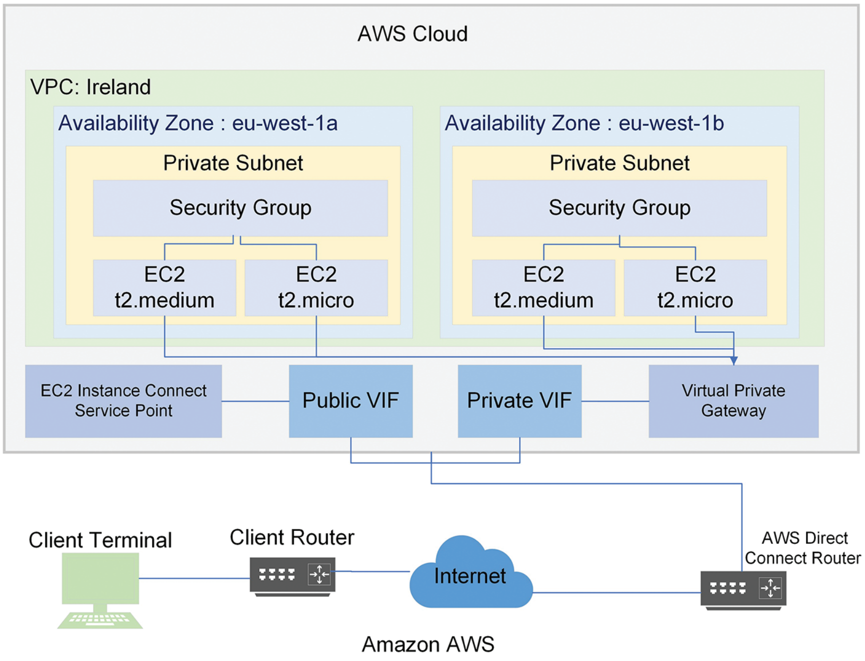

Figure 6: Experimental arrangement of AWS cloud

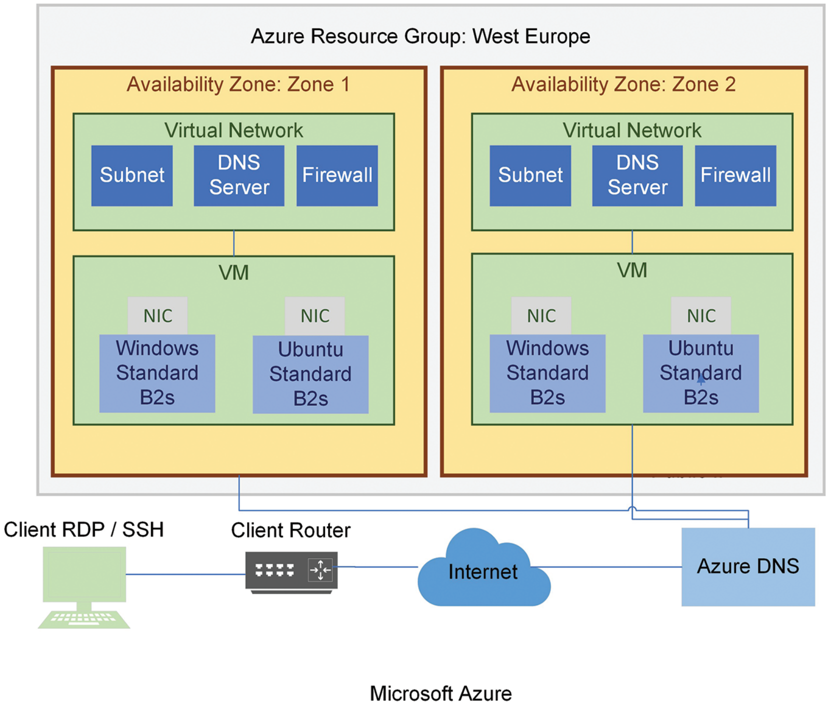

Figure 7: Experimental setup of Azure cloud

Both clouds provide more security regarding extensibility, allowing identifying rules for mutually inward and outgoing traffic. We may also use advanced safety systems like virtual security structures and other features over the cloud.

- Network throughput: transmitter and receiver were evaluated in different availability zones while the primary region remained constant. Even though Azure outlasted AWS in both scenarios (TCP and UDP). The average TCP throughput in AWS was 106.83 Mbps, whereas it was 111.32 Mbps in Azure. TCP throughput decreased in AWS as test duration increased, while it remained consistent in Azure, another critical factor observed. In the case of UDP, there was a significant variance in performance. The most considerable throughput in AWS was 2.62 Mbps, whereas the average in Azure was 25.62 Mbps, with the lowest recorded result being 22.46 Mbps, as shown in Fig. 8.

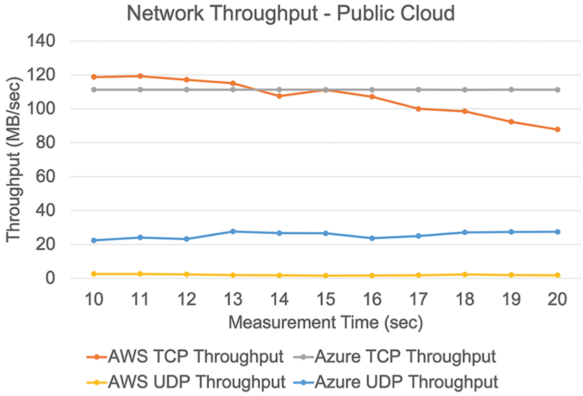

Figure 8: Network throughput using public clouds

- Latency: we opted VMs and hosted them in various availability zones. In this assessment, AWS outperforms Azure by a wide margin, as shown in Fig. 9. Azure had multiple variations, while AWS remained virtually constant.

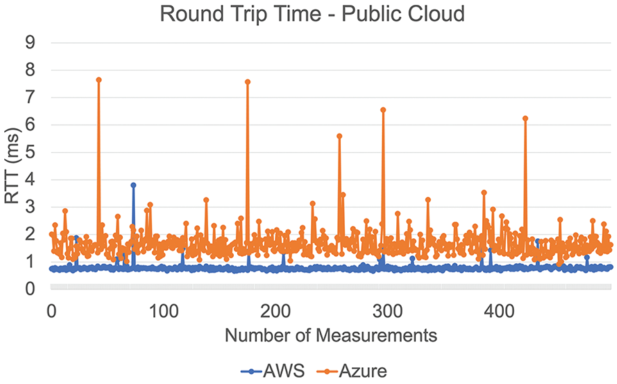

Figure 9: An illustration of RTT plots for Azure and AWS VMs hosted in different zones

- Processor utilization: there was no substantial difference in TCP throughput between the two public clouds, yet AWS consumed more CPU than Azure. The most significant CPU use on Azure was 12.75%, whereas the lowest was 33.12% on AWS. Even though throughput was very modest, Azure required much more CPU than AWS during the operation. The average UDP impact on CPU consumption in Azure and AWS was 52.36% and 20.74%, respectively. Despite using 50% more CPU than AWS, we obtained greater throughput, as presented in Fig. 10.

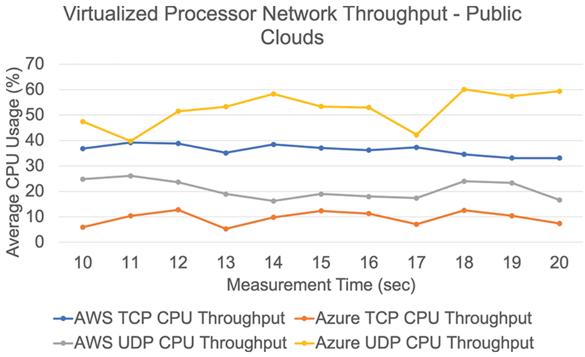

Figure 10: Processor utilization using public clouds

5.3 Case 3: Container-Based Performance Metrics

We installed the Docker desktop on a Windows PC with the specifications provided in Table 5. Docker launched in Linux mode, and the Ubuntu basic images were obtained from Docker Hub. Once acquiring the idea, we used that computer to build containers.

Table 5: Specification of Docker system

Docker system configurations	
Operating system	Windows 10
CPUs	8
RAM	16 GB
Storage	500 GB SSD

Since there is no separate layer for network protection, integrating security mechanisms in containers is more challenging than in private and public clouds based on hypervisors. The representation of the experimental setup is presented in Fig. 11.

- Network throughput: effects analyzed using the same benchmarking methodology. It is worth noting that the Docker container produced the maximum network throughput, as shown in Fig. 12.
- Latency: we used Ubuntu containers with a shared network connection. After five hundred tests, the average RTT was 0.05 ms, with a maximum value of 0.15 ms, as presented in Fig. 13.
- Processor utilization: container's high throughput necessitates more processing time. The average CPU consumption during TCP operation was 70.85%. However, in UDP, the moderate CPU use was 18.80%. Because resource utilization varies so little, we can assume it is constant. There is minimal movement in the utilization of resources; we may infer that they are almost endless depicted in Fig. 14.

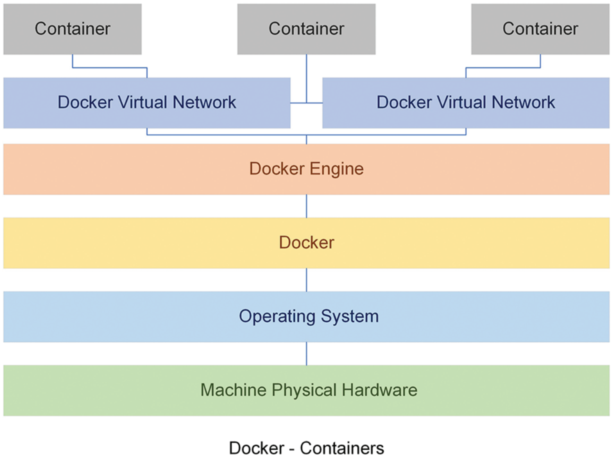

Figure 11: Experimental setup using Docker container

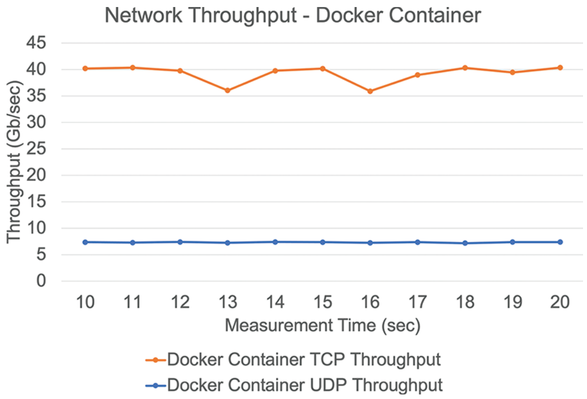

Figure 12: Network throughput using Docker container

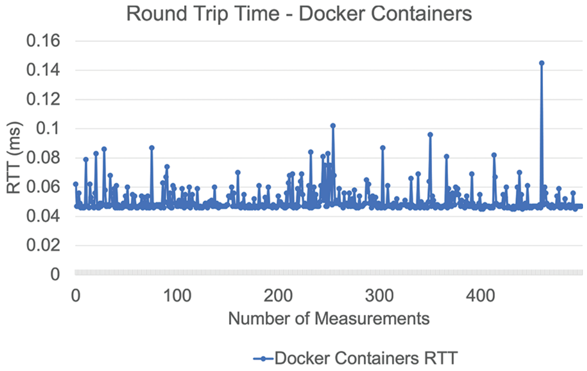

Figure 13: An illustration of RTT plots using the Docker container

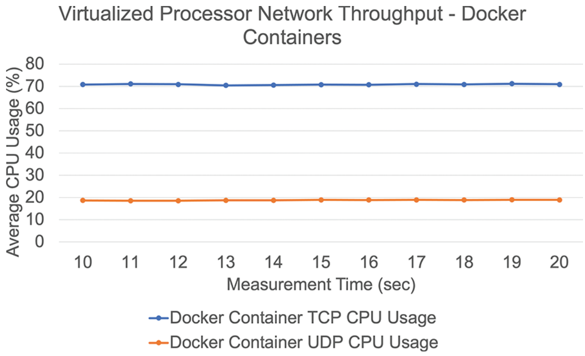

Figure 14: Processor utilization using Docker container

6 Results Summary

Considering the projected characteristics of NGWNs, the ability to select the appropriate technology to host the network function based on slice service requests is significant. An experimental methodological approach is employed to acquire data from leading virtualization technologies while operating under identical settings. The findings presented usually indicate how virtualization solutions would perform; we focused on viable TCP traffic, as used in several recent papers [28–31].

To the best of our understanding, the efficacy of the utilized technologies and their performance metrics such as throughput, latency, and CPU usage will have no significant impact on the type of traffic. Tables 6 and 7 provide the outcomes of the complete investigation to examine how VNFs and CNFs perform to offer essential cost-effective, hardware- and energy-saving details.

Table 6: Network throughput results summary

	TCP Throughput (Mbps)			UDP Throughput (Mbps)		
	Max	Min	Avg	Max	Min	Avg
XenServer	11.32	11.28	11.30	30.28	29.54	29.54
XenServer∗	11.28	11.26	11.27	29.90	28.51	29.53
Hyper-V	11.35	8.15	10.69	5.70	5.63	5.66
Hyper-V∗	11.29	3.63	9.46	5.65	5.62	5.64
vSphere	11.31	11.31	11.31	29.39	25.08	26.67
vSphere∗	11.28	11.26	11.28	29.78	25.02	27.97
AWS	119.31	87.86	106.83	2.62	1.61	2.09
Azure	111.36	111.28	111.32	22.46	27.67	25.62
Docker container	40330	35900	39190	7.42	7.18	7.34
Physical network	11.32	11.31	11.31	7.55	7.37	7.42

Note: [Labeled connections depicted with (∗) sign].

Table 7: Latency and processor usage results summary

	RTT (ms)			TCP (CPU Usage%)			UDP (CPU Usage%)		
	Max	Min	Avg	Max	Min	Avg	Max	Min	Avg
XenServer	4.42	0.15	0.38	6.98	4.34	5.60	58.68	55.90	57.41
XenServer∗	12.40	0.34	0.64	3.90	2.65	3.32	59.06	55.68	57.13
Hyper-V	4.05	0.11	0.31	1.43	8.15	2.49	21.19	16.26	18.08
Hyper-V∗	3.57	0.49	1.03	0.82	3.63	2.14	20.74	15.70	18.51
vSphere	2.18	0.13	0.30	7.51	5.35	6.44	41.19	34.57	37.78
vSphere∗	25.40	0.28	0.40	7.67	5.12	6.33	40.69	33.63	38.08
AWS	3.80	0.67	0.78	39.18	33.12	36.35	26.17	16.28	20.74
Azure	7.65	0.93	1.69	12.75	5.28	9.56	60.10	39.80	52.36
Docker container	0.15	0.05	0.05	71.20	70.44	70.85	18.96	18.58	18.80
Physical network	12.30	0.11	0.23	6.04	3.39	3.94	5.41	2.82	4.35

Note: [Labeled connections depicted with (∗) sign].

Docker provides the highest throughput of 40.33 Gbps, but AWS outperforms all others in the public cloud with the highest TCP throughput of 119.31 Mbps. Similarly, in the case of UDP, Docker gives the maximum throughput of 7.42 Gbps, whereas, in the private cloud, XenServer offers the maximum throughput of 30.28 Mbps. Azure is close to the value of 27.67 Mbps, whereas AWS has the lowest UDP throughput of 2.62 Mbps. Docker, on the other hand, has the lowest latency of 0.05 ms. Figs. 15–17 demonstrate relative analysis of the findings obtained from different virtualization technologies.

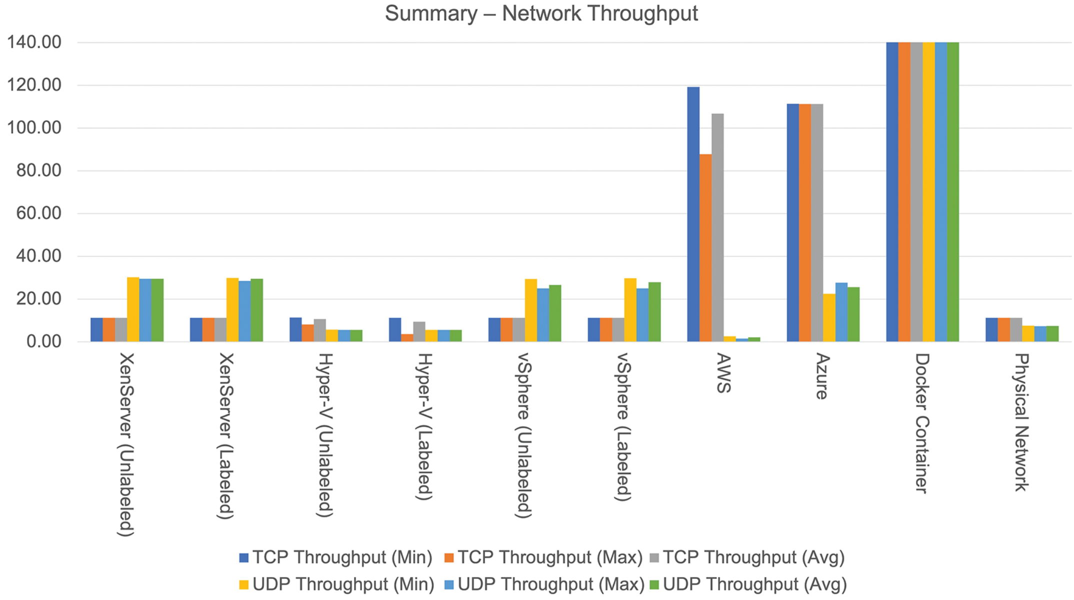

Figure 15: Result summary of network throughput

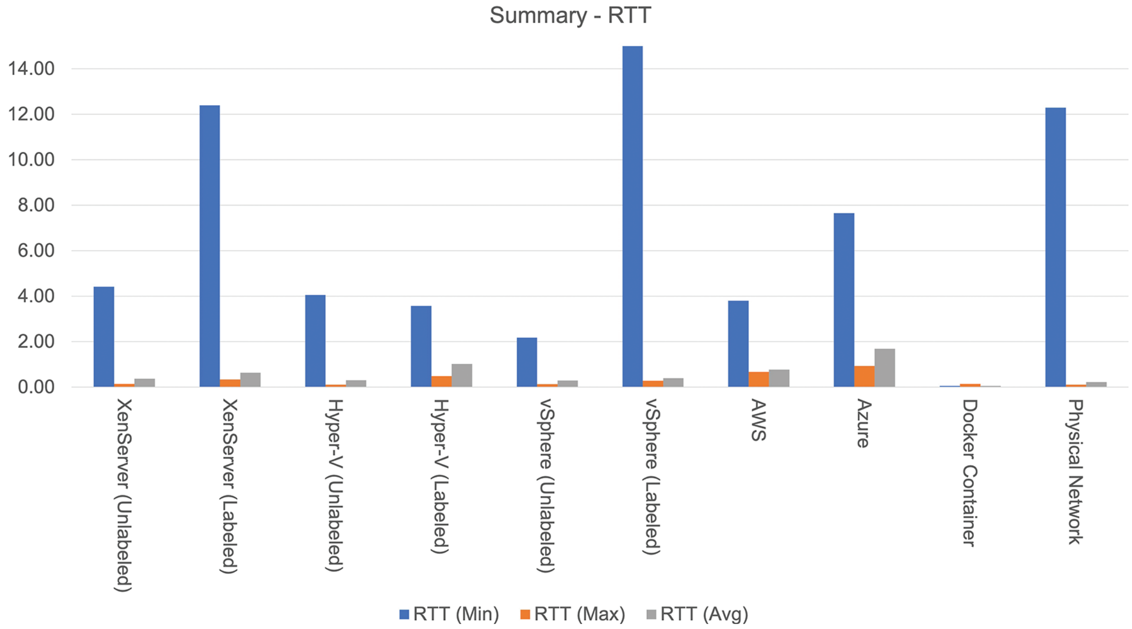

Figure 16: Result summary of latency

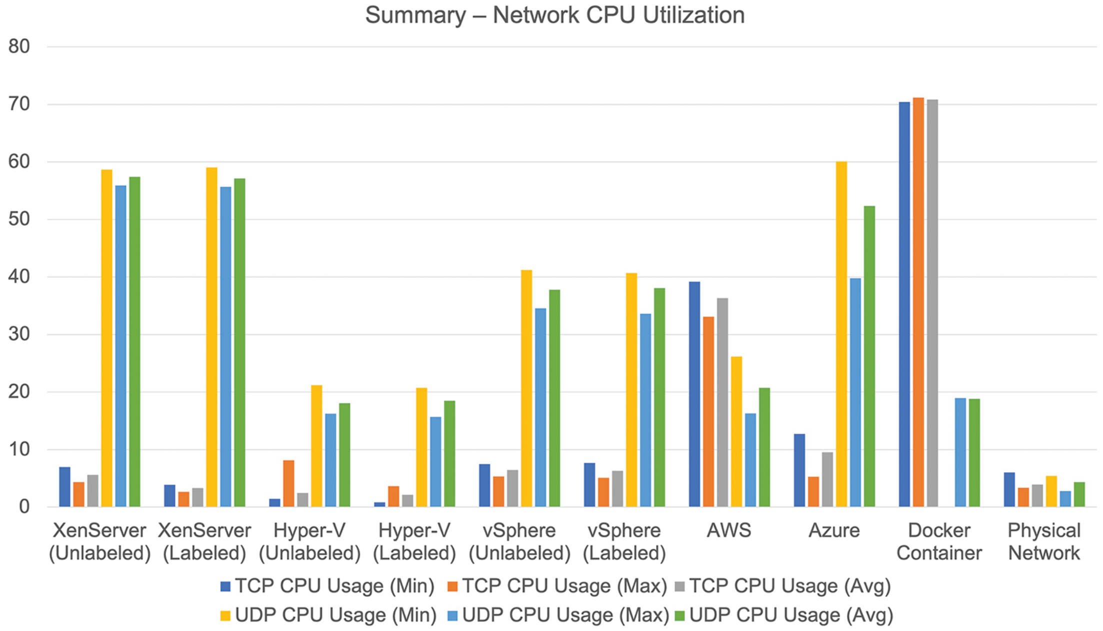

Figure 17: Result summary of processor utilization

7 Conclusions

The state-of-the-art technologies SDN and NFV are regarded as essential tools for future networks. Virtualization technologies such as VMs, containers, and cloud platforms, as well as more advanced solutions like unikernels, are paving the way for NFV. Selecting the correct technology is crucial for making the best use of underlying hardware and virtual environments. This article comprehensively analyzes leading virtualization technologies to achieve cost-effective, hardware- and energy-saving solutions. Experimental results suggest that containers outperform private and public clouds in terms of latency and network throughput but at the cost of more virtualized processor utilization. However, containers are still concerned with safety compared to public clouds. Top cloud platforms, including AWS and Azure, encourage operating containers without sacrificing speed and providing noticeably better security. The primary benefits would be real-time availability, enhanced security, and management of the provided services. The outcomes of this study will enable us to propose answers to the challenges of using containers in a virtualized environment, such as security, and expedite future network modeling.

Funding Statement: This research was supported by Future University Researchers Supporting Project Number FUESP-2020/48 at Future University in Egypt, New Cairo 11845, Egypt.

Conflicts of Interest: The authors declare they have no conflicts of interest to report regarding the present study.

References

[1] Osseiran, F. Boccardi, V. Braun, K. Kusume, P. Marsch *et al.,* "Scenarios for 5G mobile and wireless communications: The vision of the METIS project," *IEEE Communications Magazine*, vol. 52, no. 5, pp. 26–35, 2014.

[2] M. Condoluci and T. Mahmoodi, "Softwarization and virtualization in 5G mobile networks: Benefits, trends and challenges," *Computer Networks*, vol. 146, pp. 65–84, 2018.

[3] M. García-Valls, T. Cucinotta and C. Lu, "Challenges in real-time virtualization and predictable cloud computing," *Journal of Systems Architecture*, vol. 60, no. 9, pp. 726–740, 2014.

[4] V. G. da Silva, M. Kirikova and G. Alksnis, "Containers for virtualization: An overview," *Applied Computer Systems*, vol. 23, no. 1, pp. 21–27, 2018.

[5] M. Maule, J. Vardakas and C. Verikoukis, "5G RAN slicing: Dynamic single tenant radio resource orchestration for eMBB traffic within a multi-slice scenario," *IEEE Communications Magazine*, vol. 59, no. 3, pp. 110–116, 2021.

[6] S. Yang, F. Li, S. Trajanovski, R. Yahyapour, X. Fu *et al.,* "Recent advances of resource allocation in network function virtualization," *IEEE Transactions on Parallel and Distributed Systems*, vol. 32, no. 2, pp. 295–314, 2021.

[7] D. M. Naranjo, S. Risco, C. de Alfonso, A. Pérez, I. Blanquer *et al.,* "Accelerated serverless computing based on GPU virtualization," *Journal of Parallel and Distributed Computing*, vol. 139, pp. 32–42, 2020.

[8] S. T. Arzo, R. Bassoli, F. Granelli and F. H. P. Fitzek, "Study of virtual network function placement in 5G cloud radio access network," *IEEE Transactions on Network and Service Management*, vol. 17, no. 4, pp. 2242–2259, 2020.

[9] B. P. R. Killi and S. V. Rao, "On placement of hypervisors and controllers in virtualized software defined network," *IEEE Transactions on Network and Service Management*, vol. 15, no. 2, pp. 840–853, 2018.

[10] P. Wankhede, M. Talati and R. Chinchamalatpure, "Comparative study of cloud platforms-microsoft azure, google cloud platform and Amazon EC2," *International Journal of Research in Engineering and Applied Sciences*, vol. 5, no. 2, pp. 60–64, 2020.

[11] A. K. Alnaim, A. M. Alwakeel and E. B. Fernandez, "A pattern for an NFV virtual machine environment," in *Proc. IEEE Int. Systems Conf. (SysCon)*, Orlando, FL, USA, pp. 1–6, 2019.

[12] T. M. Pham, "Optimization of resource management for NFV-enabled IoT systems in edge cloud computing," *IEEE Access*, vol. 8, pp. 178217–178229, 2020.

[13] Q. Duan, S. Wang and N. Ansari, "Convergence of networking and cloud/edge computing: Status, challenges, and opportunities," *IEEE Network*, vol. 34, no. 6, pp. 148–155, 2020.

[14] P. J. Maenhaut, B. Volckaert, V. Ongenae and F. De Turck, "Resource management in a containerized cloud: Status and challenges," *Journal of Network and Systems Management*, vol. 28, no. 2, pp. 197–246, 2020.

[15] S. Giallorenzo, J. Mauro, M. G. Poulsen and F. Siroky, "Virtualization costs: Benchmarking containers and virtual machines against bare-metal," *SN Computer Science*, vol. 2, no. 5, pp. 1–20, 2021.

[16] C. Kelly, N. Pitropakis, A. Mylonas, S. McKeown and W. J. Buchanan, "A comparative analysis of honeypots on different cloud platforms," *Sensors*, vol. 21, no. 7, pp. 2433, 2021.

[17] J. Castillo-Lema, A. Venâncio Neto, F. de Oliveira and S. Takeo Kofuji, "Mininet-NFV: Evolving mininet with OASIS TOSCA NVF profiles towards reproducible NFV prototyping," in *2019 IEEE Conf. on Network Softwarization (NetSoft)*, Paris, France, pp. 506–512, 2019.

[18] P. Tam, S. Math, A. Lee and S. Kim, "Multi-agent deep q-networks for efficient edge federated learning communications in software-defined IoT," *Computers, Materials & Continua*, vol. 71, no. 2, pp. 3319–3335, 2022.

[19] A. U. Rehman, R. L. Aguiar and J. P. Barraca, "Network functions virtualization: The long road to commercial deployments," *IEEE Access*, vol. 7, pp. 60439–60464, 2019.

[20] S. Zahoor, I. Ahmad, M. T. B. Othman, A. Mamoon, A. U. Rehman *et al.,* "Comprehensive analysis of network slicing for the developing commercial needs and networking challenges," *Sensors*, vol. 22, no. 17, pp. 6623, 2022.

[21] R. Mijumbi, J. Serrat, J. -L. Gorricho, N. Bouten, F. De Turck *et al.,* "Network function virtualization: State-of-the-art and research challenges," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 1, pp. 236–262, 2016.

[22] Y. Liu, D. Lan, Z. Pang, M. Karlsson and S. Gong, "Performance evaluation of containerization in edge-cloud computing stacks for industrial applications: A client perspective," *IEEE Open Journal of the Industrial Electronics Society*, vol. 2, pp. 153–168, 2021.

[23] S. A. R. Shah, A. Waqas, M. -H. Kim, T. -H. Kim, H. Yoon *et al.,* "Benchmarking and performance evaluations on various configurations of virtual machine and containers for cloud-based scientific workloads," *Applied Sciences*, vol. 11, no. 3, pp. 993, 2021.

[24] M. Sollfrank, F. Loch, S. Denteneer and B. Vogel-Heuser, "Evaluating docker for lightweight virtualization of distributed and time-sensitive applications in industrial automation," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 5, pp. 3566–3576, 2020.

[25] K. L. Lim, J. Whitehead, D. Jia and Z. Zheng, "State of data platforms for connected vehicles and infrastructures," *Communications in Transportation Research*, vol. 1, pp. 100013, 2021.

[26] M. Zoure, T. Ahmed and L. Réveillére, "Network services anomalies in NFV: Survey, taxonomy, and verification methods," *IEEE Transactions on Network and Service Management*, vol. 19, no. 6, pp. 1567–1584, 2022.

[27] N. Kazemifard and V. Shah-Mansouri, "Minimum delay function placement and resource allocation for open RAN (O-RAN) 5G networks," *Computer Networks*, vol. 188, pp. 107809, 2021.

[28] T. Zhang, H. Qiu, L. Linguaglossa, W. Cerroni and P. Giaccone, "NFV platforms: Taxonomy, design choices and future challenges," *IEEE Transactions on Network and Service Management*, vol. 18, no. 1, pp. 30–48, 2020.

[29] A. Bhardwaj and C. R. Krishna, "Virtualization in cloud computing: Moving from hypervisor to containerization—a survey," *Arabian Journal for Science and Engineering*, vol. 46, no. 9, pp. 8585–8601, 2021.

[30] E. Casalicchio and S. Iannucci, "The state-of-the-art in container technologies: Application, orchestration and security," *Concurrency and Computation: Practice and Experience*, vol. 32, no. 17, pp. e5668, 2020.

[31] L. Zhu, M. M. Karim, K. Sharif, C. Xu, F. Li *et al.,* "SDN controllers: A comprehensive analysis and performance evaluation study," *ACM Computing Surveys (CSUR)*, vol. 53, no. 6, pp. 1–40, 2020.