International Journal of Computer Information Systems and Industrial Management Applications
ISSN 2150-7988 Volume 18(3s) 2026 pp. 1307-1316
https://doi.org/10.70917/ijcisim-2026-2462
Article

Evaluating the Impact of Network Function Virtualization (NFV) on End-to-End Latency

Bhargavi S.[1], Reshma Verma[2], N. Sathisha[3], Anil Kumar C.[4], Harish S.[5], Lavanya Vaishnavi D. A.[6]

[1] Department of Electronics and Communication Engineering, SJC Institute of Technology (Autonomous), Visvesvaraya Technological University (VTU), Belagavi, Chikkaballapur, Karnataka, India.
Email: bhargavisunil@gmail.com
[2] Department of Electronics and Communication Engineering, Ramaiah Institute of Technology, Bengaluru, Karnataka, India.
Email: reshmaverma@msrit.edu
[3] Department of Electronics and Communication Engineering, Government Engineering College, K. R. Pete, Karnataka, India (Affiliated to Visvesvaraya Technological University, Belagavi).
Email: nsathisha@gmail.com
[4] Department of Electronics and Communication Engineering, R. L. Jalappa Institute of Technology, Doddaballapur, Karnataka, India (Affiliated to Visvesvaraya Technological University, Belagavi).
Email: canilkumarc22@gmail.com
[5] Department of Electronics and Communication Engineering, R. L. Jalappa Institute of Technology, Doddaballapur, Karnataka, India (Affiliated to Visvesvaraya Technological University, Belagavi).
Email: harishsrinivasaiah@gmail.com
[6] Department of Electronics and Communication Engineering, R. L. Jalappa Institute of Technology, Doddaballapur, Karnataka, India (Affiliated to Visvesvaraya Technological University, Belagavi).
Email: lavanyavaishnavi@gmail.com

Abstract: — Network Function Virtualization (NFV) has become an important facilitator of scalable, expendable, cost-efficient network designs through the decoupling of network functions with proprietary hardware and their implementation as software instances on virtualized platforms. Regardless of these benefits, there are concerns about the effect of NFV on the end-to-end (E2E) latency, especially when it comes to the latency-sensitive applications, including real-time communications, industrial automation, and edge computing. In this paper, the effect of NFV on E2E latency has been assessed by examining the underlying latency elements that have been presented by virtualization and service chaining. An exhaustive analysis of processing, queuing, transmission and delay caused by virtualization is also introduced to determine important latency bottlenecks. To create a realistic traffic and workload environment, an experimental NFV testbed is developed based on a virtualized cluster and NFV management and orchestration system. Several service chain implementations and traffic profiles are tested to quantify the variation of latency at varying system loads, and deployment plans. The findings prove that although NFV may introduce extra processing and context-switching overhead over traditional hardware-based implementations, smart service placement, service chaining optimization, and effective orchestration can go a long way to reduce latency-penalties. The analysis also shows trade-offs among flexibility and performance with the focus on the significance of lightweight virtualization and resource-aware orchestration. In general, this paper has offered useful information on the latency behavior of NFV-enabled networks and has also given recommendations on how to design low-latency virtualized network services that can be used in the next-generation communication systems.

Keywords: — Network Function Virtualization, End-to-End Latency, Virtual Network Functions, Service Chaining, Network Orchestration

1. Introduction

The modern communication networks have radically changed with the rapid development of cloud computing, 5G/6G networks, Internet of Things (IoT), and latency-sensitive applications. The classic network architectures that

Copyright: © 2026 by the authors

are largely dependent on proprietary and hardware-based network appliances cannot be flexible and scalable enough to address the dynamic service needs. Network Function Virtualization (NFV) has been proposed to overcome these constraints as a paradigm shift decoupling network functions and hardware, and making them software-based objects that are executed on virtualized infrastructure. NFV has also formed the basis of the next-generation network design by facilitating fast service rollout, scaling at will, and minimized capital and operational spending. Even though NFV has some architectural benefit, it has severe performance issues, especially regarding the end-to-end (E2E) latency. Numerous new applications such as real-time multimedia streaming, autonomous systems, industrial automation, tactile internet and edge-assisted services have extremely rigorous latency requirements that have to be met to ensure reliability and quality of experience [1]. In these situations, even minimal processing or queuing delay can cause immense deterioration to the performance of applications. As a result, it is imperative to know the impact of NFV on E2E latency as a measure of its appropriateness in delay-sensitive network services. NFV-enabled networks have various factors which determine the E2E latency other than traditional transmission delays. Introduction of virtualization layers, hypervisors, container run time and software based packet processing impose new overheads which are not seen in the legacy hardware-based networks [2]. Moreover, inter-VNF communication, context switching, and resource contention may exaggerate latency because of chaining of multiple Virtual Network Functions (VNFs) that may be located in data centers or edge nodes in geographically dispersed locations. These issues make it necessary to analyze the sources of latency that are peculiar to virtualized networks [3]. The recent developments in the NFV management and orchestration architecture have tried to solve the problem of latencies with smart placing of VNFs, dynamic resource allocation, and traffic-aware service chaining. Nevertheless, the success of these mechanisms is very dependent on the underlying hardware capabilities, virtualization technologies and workload characteristics. Consequently, the latency behavior during the deployment of NFVs can be significantly different in various network conditions, and empirical assessment is important in this case [4].

2. Related Work

The effect of Network Function Virtualization (NFV) on end-to-end (E2E) latency has been actively studied in recent years, as this issue has become more and more concerned with the appropriateness of the virtualized network to the delay-sensitive services. Initial research was mostly concerned with conceptual NFV design and performance trade-offs, with the focus on flexibility and economic efficiency, and not on absolute latency assurances [5]. ETSI standardization work provided the basis of the NFV model, with virtualization overhead, service chaining, and orchestration delays as some of the possible causes of performance bottlenecks, but lacked a quantitative analysis of the latencies. A number of experimental analyses have been made concerning the NFV-based deployments versus traditional hardware middleboxes [6]. In these works, always there was higher processing delays on virtual environments because of hypervisor overhead, virtual switching and shared resource contention. The scholars proved that the latency of packing processing is significantly higher when various Virtual Network Functions (VNFs) are configured in a chaining fashion, particularly when the traffic load is high. They noted, however, that latency penalties can be minimized by providinging resources with careful care and pinning CPUs [7]. The more recent studies have discussed the application of the technologies of virtualization in the performance of latency. Comparative studies between virtual machines and container-based NFV deployments indicated that container-based deployments typically have lower E2E latency since their abstraction layers are typically smaller and because they have short context switching times. Research also investigated the effects of faster I/O models and methods of bypassing the kernel, and reported significant improvements in latency with these over more traditional virtual network interfaces [8]. The other significant research area is on NFV management and orchestration. A number of works suggested the latency-conscious VNF placement and service chaining algorithms to reduce the inter-VNF communication delays and minimize the number of network hops. These methods proven to be at least measurable, in terms of latency reduction, using VNFs being deployed to be in proximity to the end user, or service chains pulled together within edge data centers. However, these solutions usually imply a trade-off between the optimization of latency and the effectiveness of the use of resources. Simulation studies have also provided useful information since they model the large-scale NFV environments with various traffic pattern conditions [9]. The NFV latency studies in Table 1 distance the virtualization overheads, advantages, and optimization gaps. Although simulations can be used to analyze scalability, they can be overly simplistic in terms of overheads of such virtualization, and can not accurately model the latency behavior in the real world.

Table 1. Comparative Review of Related Work on NFV and End-to-End Latency

NFV Environment	Service Chain Length	Latency Focus	Key Benefits	Impact
Conceptual NFV Framework	Single VNF	Processing Delay	Identified core NFV latency sources	Lacked experimental validation
ETSI NFV Architecture	Multi-VNF	End-to-End	Standardized latency components	No quantitative latency results
Cloud NFV Testbed [10]	2–3 VNFs	Processing & Queuing	Demonstrated feasibility of NFV	High VM overhead under load
Data Center NFV [11]	3–5 VNFs	Service Chaining	Showed latency growth with chain length	Limited scalability analysis
Containerized NFV	2–4 VNFs	Processing Delay	Reduced latency vs. VMs	Security isolation concerns
SDN–NFV Integrated [12]	Dynamic Chains	Transmission Delay	Improved traffic steering	Simplified network models
Edge NFV Deployment	1–3 VNFs	End-to-End	Significant latency reduction	Edge resource constraints
OpenStack-Based NFV [13]	Multi-VNF	Queuing Delay	Flexible orchestration support	Orchestration overhead
Latency-Aware NFV	Optimized Chains	End-to-End	Intelligent VNF placement	Increased control complexity
5G Core NFV [14]	3–6 VNFs	URLLC Latency	Supports low-latency services	Sensitive to traffic bursts
AI-Assisted NFV	Adaptive Chains	Predictive Latency	Proactive latency optimization	Training overhead

3. Fundamentals of Network Function Virtualization

A. NFV architecture and core components

Network Function Virtualization (NFV) is a new network architecture model that presents the traditional networking functions that are based on software and run on a virtualized infrastructure, as opposed to hardware-based models. NFV architecture was first standardized by ETSI, and consists of three closely linked layers which in combination provide virtualization of network services, control and management of network services. The core of it, is the NFV Infrastructure (NFVI) that contains the physical computing, storage, and networking resources, and the virtualization layer, that abstracts the resources. Figure 1 depicts a layered architecture of the NFV, which consists of MANO, VNFs, virtualization, and infrastructure. The virtualization layer is the layer that is normally done with the help of hypervisor or container run time to allow various virtualized network functions to share a single physical hardware effectively.

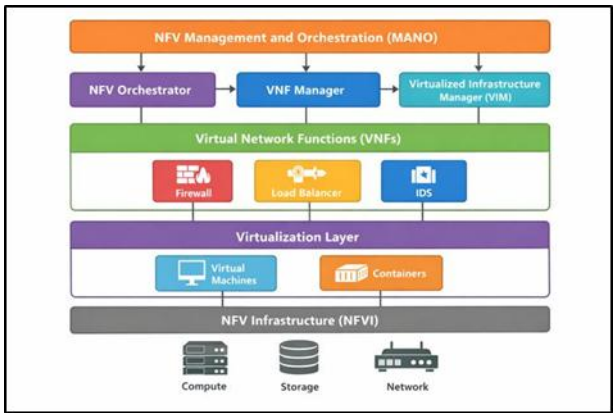

Fig.1. NFV Architecture and Core Components Block Diagram

The Virtual Network Functions (VNFs) occupying the top of the NFVI are software-based implementations of network functions (i.e., firewalls, load balancers, intrusion detection systems, and gateways). The VNFs do not rely on the underlying hardware as they can be deployed quickly and can be scaled dynamically. The third element is the NFV Management and Orchestration (NFV Management and Orchestration) which is the coordination of resource allocation, VNF lifecycle management, and service composition. NFV architecture is also based on logical interfaces which facilitate exchange of messages among the components of the architecture which makes it interoperable and modular.

B. Virtual Network Functions (VNFs) and Service Chaining

NFV is composed of Virtual Network Functions (VNFs) which are substitutes of dedicated hardware appliances used as software-based network services. VNFs are network functions, i.e. packet inspection, traffic filtering, encryption or routing and are deployed as instances of a virtual machine or container. VNFs are dynamically instantiated, scaled, migrated or terminated, enabling networks to respond quickly to the changing traffic flow and service needs. Service chaining is defined in terms of an ability to chain various VNFs in a sequence in order to create an end-to-end network service. One such example is the service chain comprising of a firewall, a load balancer and then a deep packet inspection capability. These VNFs have a logical flow of the traffic despite the fact that the VNFs may be located physically on different servers or data centers. Service chaining allows traffic processing to be finely controlled, and allows service compounds to be constructed without reconfiguring hardware. Although service chaining will increase flexibility, latency will also be a problem. Every VNF creates processing delay and, between VNFs, there are additional transmission and queuing delays particularly when the VNFs are deployed on different physical hosts. Latency overhead is also contributed by context switching, virtual switching and packet encapsulation.

C. NFV Management and Orchestration (MANO)

NFV Management and Orchestration (MANO) ensures the process of coordination and control of the lifecycle of virtualized network services and resources. MANO offers the wisdom to deploy, set, monitor, scale, and terminate VNFs and also ensure sound utilization of underlying infrastructure. It is the control plane of NFV, which turns service requirements into specific actions in terms of resource allocation and operations. Normally, the MANO framework is made of three functional entities. VNF Manager (VNFM) serves as a management accountable in the lifecycle of VNFs, that is, instantiation, scaling, updates, and fault recovery. The Virtualized Infrastructure Manager (VIM) links and governs the compute, storage, and networking resources in the NFVI, such that the VNFs get the resources that they need. NFV Orchestrator (NFVO), works on the next level, as it coordinates various VNFs and domains of infrastructure to provide full network services resultant of service chaining. Optimization of performance on MANO is critical as far as latency is concerned. The decisions that affect processing and transmission delays are decisions that pertain to the placement of VNFs, reserving of resources and scaling.

4. End-to-End Latency in Virtualized Networks

A. Definition and latency components

End-to-end (E2E) latency is defined as the sum of time spent by a data packet in a network by an originating source to a destination and all the processing and transmission stages between the source and destination. E2E latency is an important performance metric in virtualized networks especially in the case of applications that need real-time or near-real-time communication. Wall (2012) states that the latency components of traditional networks are mostly fixed and hardware-imposed, whereas virtualized environments place dynamic and software-sensitive delay components. E2E latency may be broken down into a number of basic elements. Transmission delay is the delay needed to force the bits of packets into the communication medium and is a factor of the size of a packet and the bandwidth of a link. Propagation delay is the delay associated with transmission of a signal over the physical medium, it is mainly dependent on distance. Processing delay Processing delay is the delay caused by network devices or VNFs in reading packet header, enforcing policies or performing transformation, like encryption or checking. Queuing delay occurs when the packets are held in buffers as a result of congestion or low processing capability. In networks that have NFV, physical infrastructure as well as virtualization layers affect these components. Other delays can be added by software-based packet forwarding, virtual switching and resource scheduling.

B. Processing, Queuing, and Transmission Delays

Processing, queuing, and transmission delay are the significant components of the E2E latency in virtualized networks. The processing delay is the amount of time taken to process packets that are checked and processed by network functions, such as routing, filtering, or deep packet inspection. NFV environments rely on software-based VNFs to operate on a set of shared computing resources, which can add delay to process requests since they can switch context, schedule their use on a CPU, and have a lower processing capacity when heavily loaded. Queuing delay is created when packets come faster than the processing and transmission capabilities resulting in their queuing in buffers. Queuing may occur at various different locations in virtualized networks such as virtual switches, VNF input queues, and physical network interfaces. The issue of resource contention between co-located VNFs and the changing nature of traffic patterns can considerably increase the level of queuing delay, which reduces predictability of latency. Transmission delay is defined as the time required to pass packets between links in the network and it depends on the bandwidth of a link and the size of packets. Whereas transmission delay between different nodes is comparable in virtualized and traditional networks, deployments of NFV can add levels of transmission delay through inter-VNF communications, particularly when a service chain of multiple physical hosts or data centers are involved.

C. Virtualization-Induced Latency Sources

The latency caused by virtualization is the added duration because of the abstraction layers that support NFV. These delays do not exist or are only very small in the traditional hardware-based networks but are very large in the context of a software-based environment. The virtualization layer overhead is one of the key sources which comprise hypervisors or container runtimes which coordinate resource sharing among various VNFs.

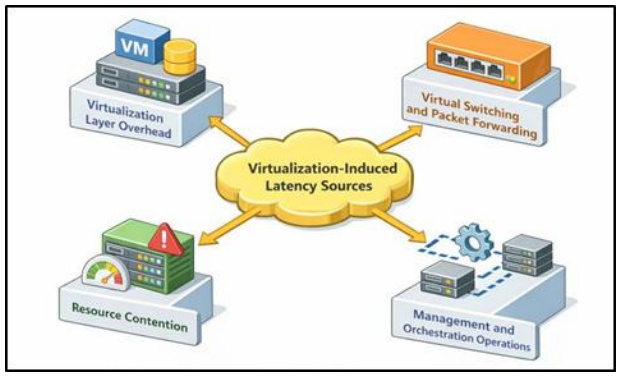

Fig.2. Virtualization-Induced Latency Sources in NFV-Enabled Networks

Switching between virtual instances and the operating system of the host may add time to the processing of packets. Another important factor is virtual switching and packet forwarding. Such methods as packet encapsulation and tunneling that are usually employed to provide network isolation and mobility add more latency. In Figure 2, the

sources of latency include important virtualization, switching, contention in resources and orchestration sources. Another significant cause of delay due to virtualization is resource contention. VNFs operate on identical physical equipment and thus they are in competition regarding access to CPU, memory and I/O resources, associating with unpredictable latency during periods of heavy load. Also, the effects of NUMA and delays in access of memory can occur when VNFs are not located optimally on a physical host.

5. Experimental Setup and Methodology

A. NFV testbed or simulation environment

In order to measure the effects of Network Function Virtualization (NFV) on end-to-end latency, the controlled experimental testbed will be built to resemble realistic virtualized network scenario. The testbed adheres to a modular NFV architecture, which allows flexible network functionality configuration, service chain, and traffic flow configuration. A virtualized infrastructure is implemented to run various Virtual Network Functions (VNFs), which are connected with each other by using virtual switches to create end-to-end service chains. The arrangement can provide accurate measurements of latency added at every level of handling packets. The experimental setup promotes single-VNF and multi-VNF services chain conditions to examine the cumulative effect of chaining on the latency. VNFs are implemented on various virtual machines or containers to replicate distributed NFV deployments that are typical of cloud and edge deployment. Besides the physical testbed implementation, simulation based evaluation is also included to test the scalability in large scale conditions.

B. Hardware, Virtualization Platform, and Orchestration Tools

The experimental environment lies on commodity server hardware in order to represent realistic NFV deployment environments. The servers have multi-core processors, adequate memory and high-speed network interfaces to enable simultaneous running of several VNFs. The hardware setup can be varied in availability of resources with control, making it possible to analysis the effects of CPU and memory constraints on latency performance. to host VNFs, a virtualization platform is used, both based on the virtual machines and containing deployments. Hypervisors offer high isolation of VNFs whereas container run times offer lightweight virtualization with low overhead. VNFs communicate with each other using virtual switches that control the packet forwarding in the NFV infrastructure. This two-layer-virtualization solution allows one to compare the latency features of various virtualization solutions. In the case of NFV management and orchestration, an open-source MANO framework is applied to automate VNF lifecycle management and deploy service chain. The orchestration platform manages the VNF instantiation, scaling, placement, and monitoring, which allow dynamic response to the workload changes. Policies related to resource allocation and placement are set to examine their effect on end-to-end latency.

C. Traffic Patterns and Workload Scenarios

In order to properly measure the end-to-end latency in NFV-enabled networks, various traffic patterns and workload conditions are created in the experimental setting. Traffic sources are simulated to model the behaviour of a realistic network by emulating both latency sensitive and throughput sensitive applications. Base Latency Constant bit rate traffic is used to measure the latency with a steady load and variable rate traffic represents bursty and dynamic application behavior that is generally seen in real networks. There are a number of workload cases to investigate the impacts of traffic intensity and complexity of service chains on latency. Low-load conditions are lightly loaded networks, which can be used to see natural virtualization overhead. Scenario settings of medium and high load gradually increase the rate of packet arrivals to examine the congestion impact, queuing time and resource contention between VNFs. Moreover, mixed traffic conditions are the mixed type of traffic that combines various types of traffic to test the performance of a mixed service demand. Service chains of different length are implemented to investigate the effect of VNF number on the end to end latency. Another set of scenarios is the distributed VNF that is deployed on several physical hosts to evaluate the inter-node transmission delay. Monitoring systems are synchronized with traffic generation tools to have correct timestamping and measurement of packet delays.

6. Result and Discussion

The results of the experiments indicate that NFV based networks cause quantifiable overhead on end-to-end latency relative to the traditional hardware based deployments. Delays in the service chain caused by VNFs are proportional to the number of VNFs in the chain mainly because it adds processing and inter-VNF communication delays. Disruption in low traffic loads is small but as load increases, queuing and resource contention are much bigger contributors of latency. Container-based VNFs are always better than those deployed using virtual machines, with less

processing delay and less variance. Placement of distributed VNFs also adds to latency because of extra transmission hops.

Table 2. End-to-End Latency Comparison Across Service Chain Lengths

Service Chain Configuration	Avg. Processing Delay (ms)	Avg. Queuing Delay (ms)	End-to-End Latency (ms)
Legacy Network (No NFV)	1.8	0.9	7
Single VNF	3.4	1.6	9.5
Two-VNF Chain	6.1	2.9	14.1
Three-VNF Chain	8.9	4.2	18.9
Five-VNF Chain	13.6	6.8	27.1

Table 2 quantitatively shows how the chain length of the service increases the end-to-end latency of NFV-enabled networks. In the legacy network with no NFV, the end-to-end latency is low (7 ms), the processing (1.8 ms) and queuing delay are low (0.9 ms).

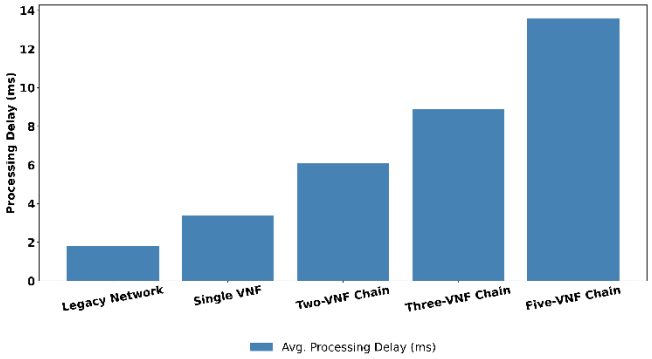

Fig.3. Average Processing Delay Across Service Chain Configurations

Figure 3 depicts that the processing delays are directly proportional to the length of NFV service chains. The addition of one VNF adds processing and queuing delay of 3.4 ms and 1.6 ms, respectively, and leads to an end-to-end design of 9.5 ms; this is a 35.7 percentage point higher than the base case. The figure 4 indicates that queuing delays increase very rapidly with NFV service chain length.

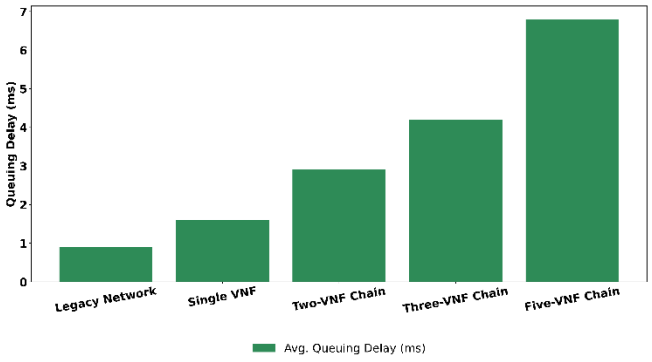

Fig.4. Average Queuing Delay Across Service Chain Configurations

With the expansion of the service chain to two VNFs, end-to-end latency increases drastically to 14.1 ms, almost 2 times that of the legacy situation. Figure 5 illustrates that end to end latency increases nonlinearly with problem chain length of VNF. The three-VNF chain causes even more latency to increase 18.9 ms, whereas the five-VNF design increases it to 27.1 ms, which is an increase of 287% relative to non-NFV.

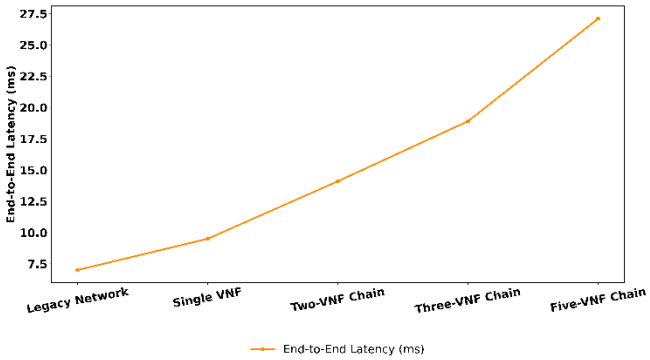

Fig.5. End-to-End Latency Trend With Increasing VNF Chain Length

These numerical patterns have a clear indication that processing and queuing delays increase nonlinearly with service chain length, which shows that service chaining is a significant source of latency in NFV deployments.

Table 3. Latency Impact of Virtualization and Deployment Strategies

Deployment Strategy	Avg. CPU Utilization (%)	Packet Loss (%)	Avg. E2E Latency (ms)	Latency Jitter (ms)
Bare Metal (Baseline)	42	0.01	7	0.6
NFV on Virtual Machines	68	0.12	16.4	2.3
NFV on Containers	61	0.07	12.8	1.5
Distributed VNFs	73	0.15	21.6	3.2

Table 3 shows the latency effect of various virtualization and deployment strategies that have evident performance tradeoffs. The lowest value of end-to-end latency 7 ms, lowest packet loss (0.01 percent), and least latency jitter (0.6 ms) are observed with the bare-metal base, which indicates the lack of virtualization overhead. Figure 6 indicates that there is increased CPU usage and the loss of packets in virtualized NFV.

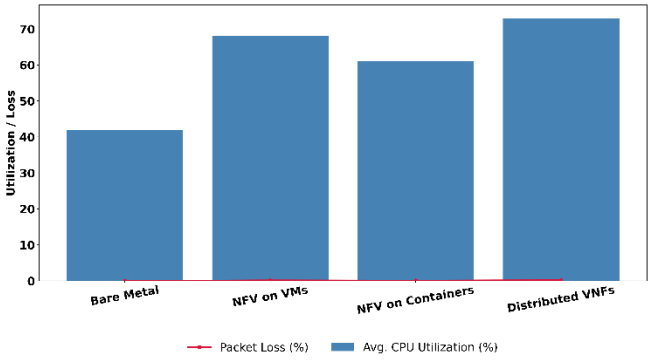

Fig.6. CPU Utilization and Packet Loss Across Deployment Strategies

NFV on virtual machine is found to achieve substantial CPU usage of 68% and end-to-end latency of 16.4 ms, which is 134 percent higher than bare metal, with additional jitter (2.3 ms). Figure 7 compares increased latency and jitter with the various NFV deployment strategies.

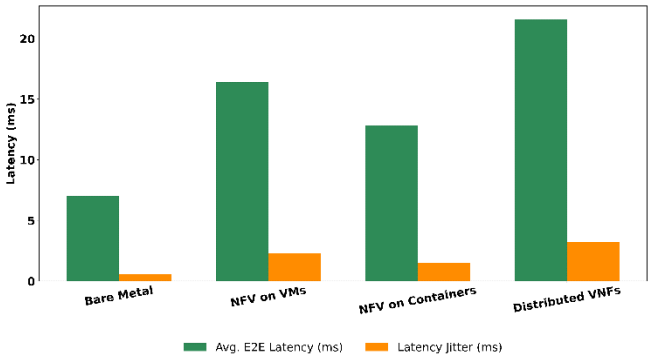

Fig.7. End-to-End Latency and Jitter Comparison

NFV with containers enhances the performance of the technology by lowering the average latency to 12.8 ms and jitter to 1.5 ms and about 22% of latency reduction compared to VM-based NFV. inter-node communication and resource contention cause the highest latency (21.6 ms) and jitter (3.2 ms), with the CPU utilization reaching its highest level of 73%. These findings prove that lightweight virtualization and local VNF placement play a crucial role in reducing latency in networks with NFV.

7. Conclusion

Network Function Virtualization has become a revolutionary technology to the contemporary communication networks, which is more flexible, scalable and cost-effective because it separates network functions and proprietary hardware. Nevertheless, its effect on end-to-end latency is still a critical issue, especially when it comes to latency-sensitive applications, which include the real-time communication, edge computing, and industrial automation. This paper has provided a systematic evaluation of the influence of NFV to end-to-end latency through the study of architectural elements, virtualization levels, service chaining, and actual traffic workloads. The experimental results validate that with NFV, it is observed that there is added latency to traditional networks, which is primarily caused by software based packet processing, virtualization overhead and inter-VNF communication. These findings also indicate that latency is growing with the length of service chain and traffic intensity, as is the contribution of several VNFs and contention of shared resources. However, the paper also shows that the existence of these latency penalties is not an intrinsic constraint of NFV. The processing and queuing delays are minimized with the help of container-based virtualization, optimized resource assignment, and intelligent orchestration. Furthermore, VNF placement models with latency sensitivity, especially models that use edge resources, can be effectively used to reduce the transmission delays and enhance the overall responsiveness. Design wise, the findings reveal the need to focus on the proper choice of virtualization technology and orchestration policies depending on the needs of applications. Although NFV can be highly effective with the elastic and scalable services, the performance optimization should be carefully considered to ensure strict latency requirements. Future NFV implementations should thus consider lightweight virtualization, hardware acceleration and real time monitoring in order to achieve predictable latency behavior.

References

1. Sun, J.; Zhang, Y.; Liu, F.; Wang, H.; Xu, X.; Li, Y. A Survey on the Placement of Virtual Network Functions. J. Netw. Comput. Appl. 2022, 202, 103361.
2. Zhu, M.; Gu, J.; Shen, T.; Zhang, J.; Gu, P. Delay-Aware and Resource-Efficient Service Function Chain Mapping in Inter-Datacenter Elastic Optical Networks. J. Opt. Commun. Netw. 2022, 14, 757–770.
3. Yu, H.; Taleb, T.; Zhang, J. Deterministic Latency/Jitter-Aware Service Function Chaining Over Beyond 5G Edge Fabric. IEEE Trans. Netw. Serv. Manag. 2022, 19, 2148–2162.
4. Yang, S.; Li, F.; Yahyapour, R.; Fu, X. Delay-Sensitive and Availability-Aware Virtual Network Function Scheduling for NFV. IEEE Trans. Serv. Comput. 2022, 15, 188–201.
5. Thiruvasagam, P.K.; Kotagi, V.J.; Murthy, C.S.R. A Reliability-Aware, Delay Guaranteed, and Resource Efficient Placement of Service Function Chains in Softwarized 5G Networks. IEEE Trans. Cloud Comput. 2022, 10, 1515–1531.
6. Eramo, V.; Catena, T. Application of an Innovative Convolutional/LSTM Neural Network for Computing Resource Allocation in NFV Network Architectures. IEEE Trans. Netw. Serv. Manag. 2022, 19, 2929–2943.
7. Alnaim, A.K.; Alwakeel, A.M.; Fernandez, E.B. Towards a security reference architecture for NFV. Sensors 2022, 22, 3750.
8. Shang, X.; Liu, Z.; Yang, Y. Online service function chain placement for cost-effectiveness and network congestion control. IEEE Trans. Comput. 2022, 71, 27–39.

9. Zhang, Z.; Wang, C. Service Function Chain Migration: A Survey. Computers 2025, 14, 203.
10. Rzym, G.; Masny, A.; Chołda, P. Dynamic Telemetry and Deep Neural Networks for Anomaly Detection in 6G Software-Defined Networks. Electronics 2024, 13, 382.
11. Ammar, S.; Lau, C.P.; Shihada, B. An in-depth survey on virtualization technologies in 6G integrated terrestrial and non-terrestrial networks. IEEE Open J. Commun. Soc. 2024, 5, 3690–3734.
12. Hu, Y.; Min, G.; Li, J.; Li, Z.; Cai, Z.; Zhang, J. VNF migration in digital twin network for NFV environment. Electronics 2023, 12, 4324.
13. Erbati, M.M.; Tajiki, M.M.; Schiele, G. Service Function Chaining to Support Ultra-Low Latency Communication in NFV. Electronics 2023, 12, 3843.
14. D. P. Chavan, “FJSO-Based Optimized Multipath Routing in Mobile Ad-Hoc Networks”, IJACECT, vol. 14, no. 2, pp. 48–52, Dec. 2025.