BULLETIN of Cherkasy State Technological University

Journal homepage: https://bulletin-chstu.com.ua/en

Vol. 29 No. 4. 2024

UDC 004.72
DOI: 10.62660/bcstu/4.2024.10

Comparative analysis of different virtual switches to improve the efficiency of networks of different configurations

Oleksandr Berestovenko[*]

Postgraduate Student
National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute"
03056, 37 Beresteiskyi Ave., Kyiv, Ukraine
https://orcid.org/0000-0003-4887-4674

Abstract. The aim of the study was to determine the optimal type of virtual switch to ensure maximum efficiency of computer networks of various configurations, taking into account their technical characteristics, capabilities, and level of integration. A comparative analysis of the performance and functionality of virtual switches was conducted. The main results showed that the Cisco Nexus 1000V provides excellent performance and low latency (0.5-2 milliseconds), making it ideal for environments where network speed and responsiveness are critical. Open vSwitch is highly scalable and memory efficient, with up to 9 gigabits per second of bandwidth and moderate Central Processing Unit usage, making it suitable for scalable virtualised environments. VMware vSwitch, with a bandwidth of 6-8 gigabits per second, has good integration into the VMware environment and easy configuration. Moreover, Virtual Packet Processing was found to provide the best throughput, reaching values between 20 and 50 gigabits per second, and also exhibits low latency in the range of 0.3-0.5 milliseconds, making it the optimal choice for environments with high bandwidth requirements. At the same time, the Bridge Virtual Switch has the lowest Central Processing Unit load (5-10%), which allows maintaining performance even with limited hardware resources. The other switches, namely Hyper-V Virtual Switch, Juniper Contrail Virtual Router, CloudStack Virtual Router, and Huawei CloudEngine vSwitch, demonstrated good performance and can be useful for environments with lower bandwidth and scalability requirements. The results showed that the choice of a virtual switch depends on specific requirements, as each switch has its own advantages and limitations that determine its optimality for different network configurations

Keywords: bandwidth; component scalability; system integration and performance; data processing; latency reduction

INTRODUCTION

With the rapid development of information technology, virtualisation is becoming an integral part of computer networks. One of the key elements of virtualisation is virtual switches, which provide data transfer between virtual machines and the physical network. Thanks to their ability to optimise the use of hardware resources and support high scalability, virtual switches play an important role in building Software-Defined Networking (SDN) and cloud environments. At the same time, the constant growth of data transmitted, the need to minimise latency and optimise resource utilisation create the need for systematic analysis of their performance.

Despite the significant attention paid to network virtualisation, the issues of choosing the optimal virtual switch for different configurations remain poorly understood. There is a wide range of solutions available, but the lack of clear guidelines for their use in specific environments makes it difficult for network engineers and architects to make decisions. The problem lies in the need to take into account many parameters such as

Article's History: Received: 15.08.2023; Revised: 22.10.2024; Accepted: 16.12.2024.

Suggested Citation:
Berestovenko, O. (2024). Comparative analysis of different virtual switches to improve the efficiency of networks of different configurations. *Bulletin of Cherkasy State Technological University*, 29(4), 10-20. doi: 10.62660/bcstu/4.2024.10.

[*]Corresponding author

throughput, latency, CPU (Central Processing Unit) and memory usage, and scalability. Existing research often focuses on specific aspects of switches, leaving gaps in understanding their overall performance.

R. Olifirenko (2021) emphasised that virtualisation of network resources remains the "missing link" in cloud computing, despite the automation of other components. The researcher pointed out that hypervisors increase the flexibility of SDN networks, although this is often accompanied by a decrease in performance, which requires improved methods of integration with SDN controllers. The results of V. Dumitrak (2020) demonstrated methods for implementing Network Function Virtualisation (NFV) in modern infrastructures, focusing on the use of Information-Centric Networking and SDN. This has made it possible to increase the efficiency of using network resources and introducing new services, improving the adaptability of networks to changing loads.

In turn, G. Bueno *et al.* (2022) showed how integrating NFV with SDN can improve the efficiency of network infrastructure. They pointed out that the use of a common SDN controller to manage NFV allows for optimising resource allocation and improving scalability by reducing data transmission delays. Researchers K. Wang *et al.* (2024) proposed an automated tool for diagnosing and localising errors in virtual private cloud networks. The tool uses a modular network model to accurately reflect the functions of real-world components of such networks and can analyse large networks with tens of thousands of components, making it effective for detecting errors in complex configurations.

Moreover, Y. Wang *et al.* (2022) pointed out the problem of dynamic resource allocation and packet scheduling for virtual switches in vehicle Internet networks. They demonstrated a mathematical model of this problem and optimisation algorithms that significantly improve the efficiency of resource allocation and data transmission in complex environments with high dynamicity of cognitive services. The results of the study by K. Yalda *et al.* (2024) compared centralised and distributed SDN architectures for Internet of Things (IoT) networks. The results showed that distributed architectures provide better fault tolerance and reduced use of the controller's CPU.

On the other hand, M.C. Lucas-Estañ *et al.* (2024) investigated the impact of different 5G network architectures and configurations on telemanipulated driving. The results showed that Mobile Edge Computing networks are the best due to high bandwidth requirements compared to centralised networks, and control channel settings can help reduce the impact of video processing time on the scalability of the service. Authors Y. Yang *et al.* (2021) proposed models for the effective implementation of virtual data centres, taking into account the possibility of placing virtual switches on physical switches. At the same time, researchers F. Ahmmed *et al.* (2024) showed that the use of virtual Multiple Input Multiple Output technologies for routing in wireless sensor networks significantly improves energy efficiency compared to traditional methods, in particular Single Input Single Output. This allows optimising network paths and reducing energy costs for data transmission, which can help improve the performance of networks with high energy saving requirements.

The purpose of the study was to identify the most effective virtual switches for optimising the operation of computer networks of various configurations, which was not fully covered by previous research. The tasks included conducting a comparative analysis of virtual switches and studying the effectiveness of their integration into existing network infrastructures with different configurations.

MATERIALS AND METHODS

To achieve the research goal, a detailed review and comparative analysis of the most common virtual switches, including Open vSwitch (OVS), Cisco Nexus 1000V (CNV), and VMware vSwitch (VMS), was carried out. It was found out how these switches affect the efficiency of computer networks of various configura tions, evaluating their performance, functionality and architectural features. A review of various tools for OVS environments, namely OpenFlow, Virtual Local Area Network (VLAN), Generic Routing Encapsulation (GRE), and Virtual Extensible Local Area Network (VXLAN), is conducted (Rashelbach *et al.*, 2022). For a CNV switch, these are VLAN, Private VLAN, and VXLAN, and for a VMS, VLAN and Network Interface Card (NIC) (Mehta, 2015). The review of the architectures of these switches included an analysis of their structural components and mechanisms that ensure their effective operation in different network configurations.

Additionally, architecture diagrams of the listed virtual switches were developed. The OVS switch diagram included various components, including the SDN controller, OVS User-Space Components, which included the Command Line Interface and Application Programming Interface (API), and the OVS Kernel Module (Pfaff *et al.*, 2015). The CNV switch circuit was based on the Virtual Supervisor Module (VSM) and Virtual Ethernet Module (VEM). It included such components as Access Control List, Quality of Service (QoS), and Application Centric Infrastructure. As for the VMS architecture diagram, it focused on the NIC component. The comparative analysis of the performance and functionality of OVS, CNV and VMS switches included such parameters as throughput (Gbps), average latency (ms), CPU usage, Random Access Memory and scalability. In addition, the advantages and disadvantages of using each of these virtual switches were outlined.

Other, less popular virtual switches such as Hyper-V Virtual Switch (HVVS), Juniper Contrail Virtual Router (JCVR), Vector Packet Processing (VPP), CloudStack Virtual Router (CSVR), OpenContrail vRouter (OCR), Big Virtual Switch (BVS), and Huawei CloudEngine vSwitch (HCES) were analysed. The platforms for using these switches were specified, namely Hyper-V and Azure for the HVVS switch (Detecting bottlenecks in..., 2022), Juniper Contrail for JCVR (Contrail Networking and..., 2023), Apache CloudStack for CSVR (Configuring AutoScale with..., n.d.), OpenContrail for OCR (OpenContrail vRouter, 2024), and cloud platforms for HCES (CloudEngine 1800V virtual..., 2018). Moreover, companies and projects such as the Fast Data Project for VPP (Vector Packet Processing, n.d.) and Big Switch Networks for BVS (Poller, 2017) were reviewed. These switches were compared by such parameters as throughput (Gbps), latency (ms), CPU utilisation (%), memory (MB), and scalability. Based on the comparison of all the virtual switches analysed in this study, namely OVS, CNV, VMS, HVVS, JCVR, VPP, CSVR, OCR, BVS, HCES, recommendations were formulated to select the most suitable solutions for optimising the operation of virtual networks depending on the specifics of the tasks and performance requirements.

RESULTS

Virtualised switches are key components of modern SDNs that provide efficient traffic management, scalability, security, and performance. OVS, for example, is one of the most popular software-defined switches designed to meet the requirements of virtualised environments and SDN. Its functionality, modular architecture, and cross-platform compatibility make it widely used in modern data centres. The OVS supports a wide range of network protocols, making it a flexible and versa tile solution for a variety of environments, including:

- OpenFlow is an SDN management protocol that allows dynamically changing traffic routing and provides centralised network management;
- VLAN – provides isolation of network segments within the physical infrastructure;
- GRE and VXLAN are tunnelling protocols that enable the creation of scalable networks on top of existing physical networks;
- integration with virtualisation platforms, as OVS is compatible with Elastic Sky X Integrated Virtual Machine (VMware ESXi), Kernel-based Virtual Machine and other platforms, allowing to effectively manage traffic between virtual machines.

The OVS architecture demonstrates the interaction between the operating system kernel, user space, SDN controller, and virtual machines (Fig. 1). This architecture has a modular structure consisting of several main components:

- OVS Kernel Module is a component that works at the operating system kernel level, is responsible for fast network traffic processing and provides low latency and high performance;
- the OVS User-Space Daemon is a component that performs management, configuration, and monitoring functions, as well as provides an API for interacting with SDN controllers such as OpenDaylight or Ryu;
- SDN controller – a component that allows centrally managing network flows using the OpenFlow protocol.

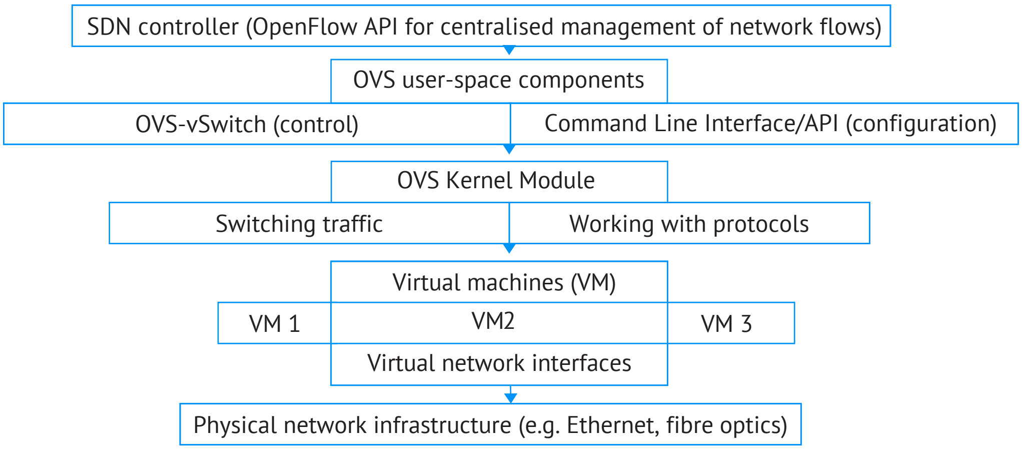
SDN controller (OpenFlow API for centralised management of network flows)
OVS user-space components
OVS-vSwitch (control)
Command Line Interface/API (configuration)
OVS Kernel Module
Switching traffic
Working with protocols
Virtual machines (VM)
VM 1
VM2
VM 3
Virtual network interfaces
Physical network infrastructure (e.g. Ethernet, fibre optics)

Figure 1. Diagram of the OVS architecture

Source: created by the author

OVS is widely used in a variety of areas, including data centres, cloud and container environments, SDN, and telecommunications. In data centres, OVS is used to communicate between virtual machines on the same server or between servers, as well as to build tunnel networks using VXLAN or GRE. This allows efficiently scaling one's network infrastructure and providing traffic isolation across VLANs. In cloud environments, OVS helps to create logical networks on top of the physical infrastructure. For example, in OpenStack, it acts as the main network element, providing convenient management through the Neutron module.

In container platforms such as Kubernetes, OVS orchestrates networks for containers, providing connectivity between them and integrating with the Container Networking Interface for scalable management.

Moreover, in SDN environments, OVS integrates with controllers such as OpenDaylight for centralised traffic management. This allows for the implementation of complex routing policies, adapting to changes in the network. In telecommunications, OVS is used to build NFV and organise high-speed services in IoT or 5G networks.

In turn, the CNV virtual switch is an important element of modern SDN, providing high performance, flexibility, and security in virtualised environments. It is an enterprise solution specifically designed to integrate virtual networks with physical infrastructure, making it a popular choice in large data centres. CNV supports a number of functionalities that ensure its adaptability to the needs of modern virtualised environments, including:

- VLAN – network segmentation to increase security and isolate traffic between virtual machines;
- Private VLAN is an additional layer of isolation for traffic between separate organisational units;
- VXLAN is a protocol for building scalable virtual networks on top of physical infrastructure;
- Standardised port settings that are automatically applied to virtual interfaces, which simplifies network management.

The CNV architecture is based on two main components: VSM and VEM (Fig. 2). The VSM is the central management unit that acts as the controller for all switches in the network. It provides centralised management, configuration, monitoring, and provides APIs for integration with virtualisation management platforms such as VMware vCenter. For its part, the VEM is located on each server running virtual machines. It directly handles traffic between virtual machines on the server and transmits it to the physical network or other hosts.

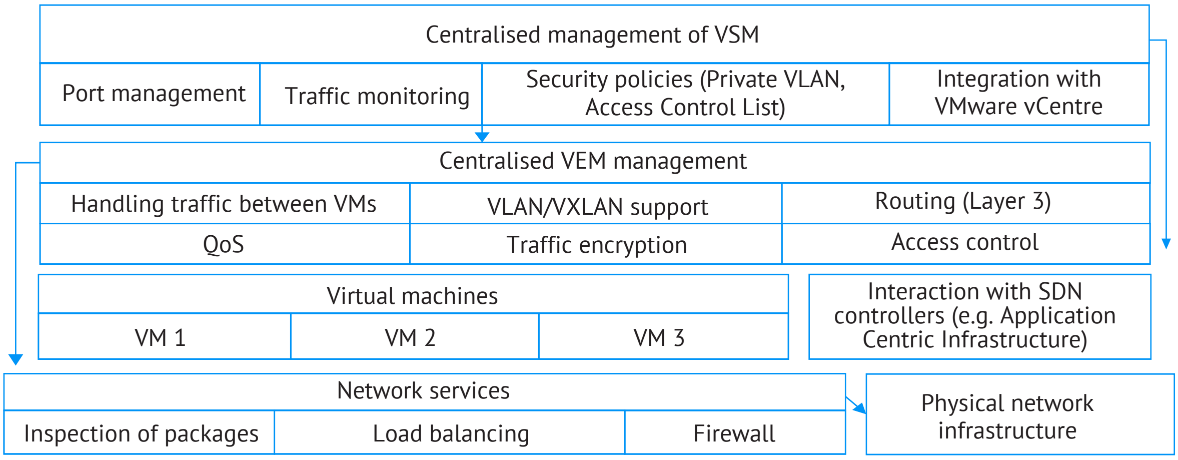
Centralised management of VSM
Port management
Traffic monitoring
Security policies (Private VLAN, Access Control List)
Integration with VMware vCentre
Centralised VEM management
Handling traffic between VMs
VLAN/VXLAN support
Routing (Layer 3)
QoS
Traffic encryption
Access control
Virtual machines
VM 1
VM 2
VM 3
Interaction with SDN controllers (e.g. Application Centric Infrastructure)
Network services
Inspection of packages
Load balancing
Firewall
Physical network infrastructure

Figure 2. Diagram of the CNV architecture

Source: created by the author

CNV is widely used in data centres, cloud environments and enterprise networks. In data centres, it integrates virtual networks with the physical infrastructure, allowing operators to centrally manage traffic between virtual machines and configure security and segmentation policies using VLANs and Private VLANs. For example, in multi-zone data centres, CNV is used to clearly distinguish zones by access level and isolate critical resources.

In cloud environments, this virtual switch is often used to build virtual networks that scale with VXLAN, allowing logical networks to be created on top of physical infrastructure. In enterprise networks, the CNV provides standardised configuration through standardised port settings, simplifying the deployment of new services and minimising human error. In addition, CNV supports integration with management platforms such as VMware vCenter and SCVMM (Microsoft System Center Virtual Machine Manager), which allows automating network management processes. This makes it indispensable in large corporate environments where it is necessary to dynamically scale resources and ensure stable network operation.

On the other hand, the VMS virtual switch is a key element of VMware's virtualisation network infrastructure that allows virtual machines on the same ESXi server to exchange traffic with each other or with a physical network. Thanks to its ease of integration, scalability, and high performance, vSwitch has become the standard in enterprise and data centre environments. VMS supports the following functionalities:

- VLAN, which provides logical network segmentation, increasing security and isolating traffic between virtual machines;
- NIC Teaming, which combines multiple physical network adapters to increase bandwidth and provide fault tolerance;
- traffic shaping, which allows the administrator to restrict incoming or outgoing traffic for load balancing;
- security policies that control access to network resources through features such as Promiscuous Mode, MAC Address Changes, and Forged Transmissions.

A VMS functions as the virtual equivalent of a physical switch (Fig. 3). Its structure includes virtual ports that are used to connect virtual network interfaces of virtual machines to the virtual switch. It also contains port groups with defined configuration parameters, such as VLAN IDs or security policies, and physical network adapters that connect the vSwitch to the physical network.

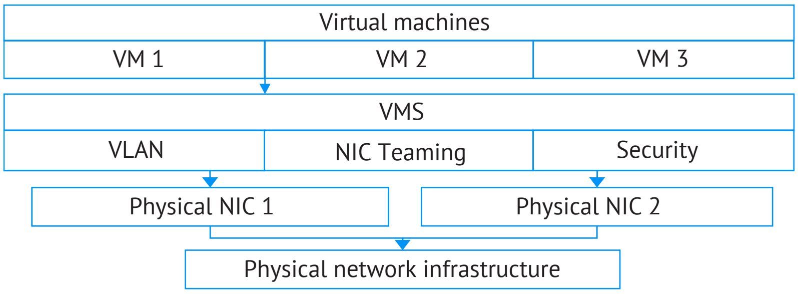

Figure 3. VMS architecture diagram

Source: created by the author

Overall, VMS is the basic solution for network virtualisation in large data centres. Its VLAN and NIC Teaming features provide the flexibility to build segmented and fault-tolerant networks. In cloud environments, such as VMware Cloud Foundation, vSwitch is used to aggregate virtual machines into logical networks. This allows quickly configuring and scaling resources as needed. On the other hand, a VMS is suitable for creating isolated networks needed for software or configuration testing. In SDN, the VMS is the component that routes traffic between virtual machines and hosts, integrating with solutions such as Network Security eXtension to implement complex security and routing policies.

In addition to the virtual switches already mentioned, it is worth paying attention to HVVS, JCVR, VPP, CSVR, OCR, BVS, and HCES. For example, HVVS is a virtual switch from Microsoft that is integrated into the Hyper-V platform. It provides VLAN support, traffic isolation, QoS, and security features such as Address Resolution Protocol and Dynamic Host Configuration Protocol protection. It is suitable for building enterprise networks and integrating with the Azure platform. JCVR is a software-based router and switch that is part of the Juniper Contrail platform. It provides routing functionality, network isolation, and traffic management in SDN and cloud environments. It supports integration with OpenStack. VPP is a high-performance software switch and router developed by the Fast Data Project. It is focused on low latency and high throughput. It is widely used in NFV and SDN. CSVR is a solution integrated into the Apache CloudStack platform that provides routing, Network Address Translation, Virtual Private Network, and firewall functions for cloud environments. It is effective for creating isolated networks in multi-tenant environments. OCR is a component of the OpenContrail platform that provides virtual switch and router functions. It supports VXLAN tunnelling and Multiprotocol Label Switching, providing scalability and flexibility in building SDN. BVS is an SDN solution from Big Switch Networks that provides centralised traffic management and scalability. It is used to build private and hybrid clouds with VLAN and VXLAN support. HCES is a virtual switch that provides high performance, integration with cloud management platforms, and support for SDN controllers. It is designed for enterprise environments and large data centres.

In summary, OVS improves the efficiency of networks in various configurations with its modular architecture and support for a wide range of protocols such as VLAN, GRE, and VXLAN, enabling scalable and optimised networks in cloud, container, and on-premises environments. Its integration with SDN controllers provides centralised management in large distributed networks, while in small environments it makes efficient use of resources. CNV delivers high performance in networks of varying complexity through clear segmentation and centralised management, which is important for large data centres and cloud environments. Its support for VXLAN allows networks to scale over physical infrastructure, and its standardised port settings simplify management in enterprise networks, providing adaptability to changing loads. As for VMS, it integrates seamlessly with VMware vSphere to automate traffic management and simplify virtual machine deployment in virtualisation environments, optimising network performance in enterprise configurations. Other switches, such as HVVS, JCVR, VPP, CSVR, OCR, BVS, and HCES, also offer specific solutions for different needs, from SDN and NFV scalability to security and traffic isolation in multi-tenant environments. This demonstrates the wide range of functionality that virtual switches offer to improve the efficiency of networks in various configurations.

To improve the efficiency of networks of various configurations, it is important to evaluate the performance and functionality of virtual switches. To determine the optimal solutions for different network environments, a comparative analysis of the most popular virtual switches should be conducted: OVS, CNV, and VMS (Table 1).

Table 1. Comparison of OVS, CNV and VMS

Parameter	OVS	CNV	VMS
Bandwidth (Gbps)	Up to 9	Up to 10	6-8
Average latency (ms)	1-3	0.5-2	2-5
CPU usage	Moderate	Low	Moderate
Using Random Access Memory	Effective	High	Moderate
Scalability	High	High	Medium

Source: created by the author based on R. Mehta (2015), A. Rashelbach *et al.* (2022)

Consequently, OVS has high scalability and memory efficiency, making it a flexible solution for SDN environments. CNV demonstrates the best performance with low latency and efficient CPU usage, but requires significant memory resources. And VMS integrates well into the VMware ecosystem and is suitable for virtualised environments, but is limited in scalability and slightly inferior in performance. In addition, it is worth considering the advantages and limitations of each switch.

OVS has a number of advantages that make it a popular choice for modern virtualised environments. High performance is achieved through optimisation at the operating system kernel level, which allows for efficient network traffic processing even in complex and busy environments. Configuration flexibility is provided by support for multiple network protocols and compatibility with SDN controllers, which allows OVS to adapt to the needs of different users. Scalability is achieved through support for tunnelling, such as VXLAN and GRE, which allows for the integration of virtual and physical networks. In addition, the OVS architecture is optimised for efficient resource utilisation, making it suitable for environments with limited hardware. At the same time, this switch also has disadvantages, such as complexity of configuration, which requires high skills, and increased CPU requirements when working with numerous virtual machines or tunnels.

CNV also offers significant benefits, including deep integration with corporate networks, allowing it to be easily integrated with existing infrastructure. CNV's security is backed by support for Private VLANs and flexible access policies, which ensures reliable data protection. VXLAN technology allows creating scalable virtual networks on top of physical infrastructure, and centralised management through VSM makes it easy to administer. However, the disadvantages of this solution are its high cost, the need for in-depth knowledge of networking technologies for configuration, and limited compatibility with non-VMware platforms.

The VMS stands out for its seamless integration with the VMware ecosystem, including ESXi, vCenter and Network Security eXtension, for easy management. An intuitive interface and standard templates simplify configuration, while support for security and redundancy policies increases network reliability. VMS flexibility is provided through the use of VLAN and NIC Teaming, which allows for segmentation and load balancing. The main disadvantages are limited analytics and integration with SDN solutions, high dependence on VMware infrastructure, and the need for additional licences to use advanced features such as VMware Distributed Switch. It is also important to compare other virtual switches (Table 2).

Table 2. Comparison of HVVS, JCVR, VPP, CSVR, OCR, BVS and HCES

Parameter	HVVS	JCVR	VPP	CSVR	OCR	BVS	HCES
Bandwidth (Gbps)	5-15	10-30	20-50	5-20	10-25	15-50	20-40
Delay (ms)	~0.8	~0.5	~0.3	~0.9	~0.4	~0.2	~0.3
CPU load (%)	10-20	15-25	5-10	10-30	10-20	5-15	5-20
Memory (MB)	30-50	40-70	25-60	30-80	35-70	25-50	30-60
Scalability	Medium	High	Very high	Medium	High	Very high	High

Source: created by the author based on Configuring AutoScale with using CloudStack virtual router (n.d.), Detecting bottlenecks in a virtualized environment (2022), Contrail Networking and Security User Guide (2023)

This means that the VPP switch is best suited for networks with high bandwidth requirements, such as those in data centres or telecommunications environments. The BVS offers the lowest latency, making it the best choice for applications that require fast response times, such as financial systems or real-time. VPP and BVS also have the lowest CPU load, which is important for maintaining performance in systems with limited hardware resources. They are leaders in terms of RAM efficiency, which is suitable for systems with limited resources. In addition, these switches demonstrate the highest level of scalability, making them effective for large distributed environments. On the other hand, HVVS is the easiest to configure, making it a good choice for small teams or environments with minimal specialisation requirements. For distributed networks or SDN environments, JCVR or OCR should be considered due to their high scalability.

OVS, CNV, and VMS were chosen because of their popularity, versatility, and wide integration into various network configurations. They are the standard for many organisations due to their proven efficiency and ease of use. At the same time, VPP and BVS switches achieve the best technical performance. The VPP is the leader in terms of throughput and low CPU load, while the BVS provides the lowest latency. JCVR and OCR have the highest scalability, which is best suited for distributed and cloud environments.

DISCUSSION

The results of the study showed that the choice of a virtual switch depends on the specific requirements of the network configuration, as each switch has its own advantages and limitations in terms of performance, scalability, and integration in different environments. S. Rush (2020) focused on the use of virtual switches in automotive systems to replace physical components, in particular in the context of functional safety and network security. Compared to the current study, which focuses on comparing the performance of virtual switches in networks of different configurations, the aforementioned study focuses on specific applications in automotive systems. Therefore, this study covers more general networking solutions for virtualised environments. In addition, A. Alnaim (2024) focused on the security of virtual networks in the context of 5G, emphasising the importance of ensuring the security of virtual networks through the flexibility of virtual functions. Whereas the current study focuses on analysing the performance of virtual switches for different network configurations, other work focuses on the security of virtualised systems. Thus, the current work complements A. Alnaim (2024) research by focusing on switch performance and integration into different environments where security is an important consideration.

The authors O.G. Lira *et al.* (2024) focused on automating the network configuration process by using large language models to generate and verify settings with minimal human involvement. Whereas the current work analyses the performance of virtual switches, the results of the above researchers focus on the automation of network configurations. Therefore, the current study complements the approach of the above work by emphasising the importance of switch performance and its integration into different network environments. Meanwhile, Y. Wang *et al.* (2018) analysed the optimisation of hash tables for flow classification in virtual switches, which is an important component of network efficiency, in particular in the context of SDN. However, the current study focuses on the performance of switches, while the above work on hash tables focuses more on the theoretical aspects of flow classification and hash table optimisation to improve switch performance. The results of the present study extend this by considering not only optimisation but also a practical comparison of different virtual switches for different types of network configurations.

In turn, P.M. Rekha & M. Dakshayini (2015) focused on managing virtual networks with SDN and using the OpenFlow architecture to improve QoS in cloud data centres. Similar to the current work, these researchers focus on configuring virtual switches to optimise network performance, and analyse the role of SDN and OpenFlow in the context of network efficiency. However, their approach is more focused on QoS management in a multi-user environment, while the current study focuses on comparing switch performance for different types of network configurations. Therefore, the current study complements the aforementioned work by considering the broader aspects of virtual switch integration and network performance in different environments. Regarding the work of C. Wang *et al.* (2024), they investigated the use of large language models to simplify the configuration of network devices and the development of routing algorithms, minimising errors due to the translation of high-level policies and requirements into low-level network configurations. Compared to the current research, which focuses on analysing the performance of virtual switches in different networks, the study of language models focuses on automating configurations using artificial intelligence. Therefore, this study complements the work of these authors by focusing on the efficiency of switches and their integration into various network configurations.

On the other hand, V.K. Tchendji *et al.* (2018) focused on the use of virtual switches to increase the resilience of virtual networks to failures by proposing traffic redirection schemes to ensure QoS. This work examines the effectiveness of virtual switches, but focuses on solutions for network recovery in the event of a failure. Similarly, the current work analyses the performance of switches in different network configurations, which also demonstrates the importance of virtual switches in ensuring network efficiency and reliability. Thus, the study confirms the conclusions of these authors, emphasising the importance of virtual switches for network scalability and stability. Additionally, S. Sadrhaghighi *et al.* (2022) presented Open Virtual Tap (OVT), which uses OpenFlow switches to monitor traffic in virtual networks, focusing on the efficiency of flow mirroring. In the current study, OpenFlow is considered as a management protocol in OVS, providing flexibility and centralised traffic management in SDN. Thus, OpenFlow is a common aspect of both works, but the current work focuses on switch performance, while the authors of the other work investigated its role in traffic analysis.

The results of the study by Z. Guo *et al.* (2023) presented ConfigReco, a configuration recommendation tool that uses graph neural networks to create templates based on the network operator's intentions. The difference from the current study is the emphasis on automating manual configuration, as the work in this paper focuses on analysing the performance of virtual switches in different network environments. Both approaches complement each other, as ConfigReco provides templates for efficient network configuration, which can be applied to virtual switches to optimise their integration into networks. In addition, L. Zhu *et al.* (2020) investigated the effectiveness of SDN controllers, particularly in the context of specialised networks such as IoT and blockchain, and compared their performance across different networks. In contrast, the current work focuses on the effectiveness of virtual switches such as OVS, VMS, and CNV, as well as their integration with different networks. In other words, this study complements the aforementioned work by extending their findings by comparing the performance of switches in the context of different network configurations and virtualisation.

For their part, I. Alam *et al.* (2020) focused on the integration of SDN and NFV for IoT, analysing their architecture, security, and management, with a focus on IoT challenges. They also highlighted key issues such as scalability and flexibility. Similarly, current work has looked at the use of SDN and NFV, specifically through virtual switches, namely OVS and CNV, to provide scalability, security, and performance in virtualised environments. Hence, the current work focuses more on the performance of specific virtual switches in different network configurations, including telecoms, IoT and 5G, which is an extension of the second study. The authors D. Bringhenti *et al.* (2023) proposed a method for automating firewall configuration in virtualised networks to improve security by reducing the number of firewalls and configuration settings required. Their approach focuses on optimising network security by minimising configuration errors. The current work focuses on virtual switch configurations, particularly in the context of traffic management, scalability, and security. This work has the advantage of evaluating not only security, but also performance and scalability, which is critical for environments with high traffic and resource requirements, such as data centres and 5G networks.

As for the work of J.V.G. de Oliveira *et al.* (2021), they proposed the implementation of NFV as programmable rules distributed among SDN switches to improve performance and scalability in packet-intensive environments. Compared to the current study, which focuses on comparing switch performance, the work of the above researchers focuses on combining hardware and software SDN switches to optimise processing speed and instantiation flexibility. This approach complements the current study by increasing the performance and scalability of virtual networks under high traffic loads. In turn, K. Marzuki *et al.* (2023) focused on the use of OVS in the context of Proxmox to manage traffic between virtual machines and external communications using VLANs. The study focused on automating virtual network configuration with Ansible, which reduces configuration time and human error. The current study also examined OVS, but focused on its performance and capabilities in the context of scalability and efficiency of virtual switches in network configurations. In addition, various aspects of virtual switches such as throughput, latency, and resource utilisation were compared, providing a deeper understanding of the functional features of OVS compared to other switches such as CNV and VMS.

A. Singh (2019) looked at CNV as a virtualisation solution targeting VMware environments, with a particular focus on automating the management of VEM and VSM components through the Python API, which significantly reduces manual intervention. The current study also analysed CNVs, but in a broader context. It not only investigated the management features of this switch, but also compared its performance, latency, resource utilisation, and scalability with other popular virtual switches. This made it possible to formulate comprehensive recommendations for choosing the optimal solution depending on the specifics of the network environment, making the approach of the current study more universal. Moreover, A. Abdou *et al.* (2018) and L. Patrão (2024) focused on a basic overview of VMware vSphere, its functionality, and basic concepts such as the differences between vSphere and ESXi hosts.

Compared to other works, the current study focuses on a detailed performance analysis of popular virtual switches and compares them. The uniqueness of the work lies in the emphasis on practical comparison of different switches in the context of different network scenarios, making it an addition to existing virtualisation and network management approaches.

CONCLUSIONS

The study identified the optimal virtual switches for different network configurations depending on key technical parameters. In particular, it was found that CNV provides the best performance with low latency (0.5-2 ms) and a high level of integration, making it optimal for environments with critical speed and reliability requirements. OVS proved to be highly efficient with scalability, up to 9 Gbps throughput, and moderate CPU usage, which is suitable for scalable virtualised environments. VMS has shown good integration into VMware environments with 6-8 Gbps bandwidth. The highest throughput (20-50 Gbps) was demonstrated by Virtual Packet Processing, making it the best choice for environments with high bandwidth requirements, while Bridge Virtual Switch has the lowest CPU load (5-10%). Other switches, such as Juniper Contrail Virtual Router and Huawei CloudEngine vSwitch, performed satisfactorily for environments with lower scalability requirements.

Limitations include the lack of testing of switches on platforms with Advanced Risc Machines processors, which are becoming increasingly popular in cloud and embedded solutions due to their energy efficiency. In addition, the study did not analyse switch performance in HPC environments, which could have yielded a wider range of results. And the sample of switches selected was limited to popular and less popular solutions, while there are other models that may also be important for specialised environments.

To improve the efficiency of virtual switches in different environments, it is advisable to focus on expanding the analysis of their integration with technologies such as next-generation networks that require low latency and high throughput. Tests on energy-efficient processors of the Advanced Risc Machines architecture will help determine their effectiveness in resource-constrained environments. Additionally, it is important to develop standardised benchmarking methodologies to take into account the specific requirements of different network configurations, which will allow for the creation of optimal solutions for specialised environments such as IoT or HPC.

ACKNOWLEDGEMENTS

None.

CONFLICT OF INTEREST

None.

REFERENCES

[1] Abdou, A., van Oorschot, P.C., & Wan, T. (2018). Comparative analysis of control plane security of SDN and conventional networks. *IEEE Communications Surveys & Tutorials*, 20(4), 3542-3559. doi: 10.1109/COMST.2018.2839348.

[2] Ahmmed, F., Rahman, A., Hossain Emon, M., & Rahman Enam, M. (2024). Enhancing energy efficiency in wireless sensor networks using virtual MIMO technology. *Global Mainstream Journal of Innovation, Engineering & Emerging Technology*, 3(2), 27-42. doi: 10.62304/jieet.v3i02.93.

[3] Alam, I., Sharif, K., Li, F., Latif, Z., Karim, M.M., Biswas, S., Nour, B., & Wang, Y. (2020). A survey of network virtualization techniques for Internet of Things using SDN and NFV. *ACM Computing Surveys*, 53(2), article number 35. doi: 10.1145/3379444.

[4] Alnaim, A.K. (2024). Securing 5G virtual networks: A critical analysis of SDN, NFV, and network slicing security. *International Journal of Information Security*, 23(6), 3569-3589. doi: 10.1007/s10207-024-00900-5.

[5] Bringhenti, D., Marchetto, G., Sisto, R., Valenza, F., & Yusupov, J. (2023). Automated firewall configuration in virtual networks. *IEEE Transactions on Dependable and Secure Computing*, 20(2), 1559-1576. doi: 10.1109/TDSC.2022.3160293.

[6] Bueno, G., Saquetti, M., Rodrigues, P., Lamb, I., Gaspary, L., Luizelli, M.C., Zhani, M.F., Azambuja, J.R., & Cordeiro, W. (2022). Managing virtual programmable switches: Principles, requirements, and design directions. *IEEE Communications Magazine*, 60(2), 53-59. doi: 10.1109/MCOM.001.2100363.

[7] CloudEngine 1800V virtual switch. (2018). Retrieved from https://carrier.huawei.com/~/media/CNBG/Downloads/Product/Fixed%20Network/b2b/0920/1800-en.pdf.

[8] Configuring AutoScale with using CloudStack virtual router. (n.d.). Retrieved from https://docs.cloudstack.apache.org/en/4.19.1.3/adminguide/autoscale_with_virtual_router.html.

[9] Contrail Networking and Security User Guide. (2023). Retrieved from https://www.juniper.net/documentation/us/en/software/contrail-networking19/contrail-networking-security-user-guide/contrail-networking-security-user-guide.pdf.

[10] De Oliveira, J.V.G., Bellotti, P.C.P., de Oliveira, R.M., Borges Vieira, A., & Chaves, L.J. (2021). Virtualizing packet-processing network functions over heterogeneous OpenFlow switches. *IEEE Transactions on Network and Service Management*, 19(1), 485-496. doi: 10.1109/TNSM.2021.3112403.

[11] Detecting bottlenecks in a virtualized environment. (2022). Retrieved from https://learn.microsoft.com/en-us/windows-server/administration/performance-tuning/role/hyper-v-server/detecting-virtualized-environment-bottlenecks.

[12] Dumitrak, V. (2020). *Methods and means of increasing the efficiency of implementing virtualisation of network functions in modern network infrastructures*. (Master's dissertation, Ternopil Ivan Puluj National Technical University, Ternopil, Ukraine).

[13] Guo, Z., Li, F., Shen, J., Xie, T., Jiang, S., & Wang, X. (2023). ConfigReco: Network configuration recommendation with graph neural networks. *IEEE Network*, 38(1), 7-14. doi: 10.1109/MNET.2023.3336239.

[14] Lira, O.G., Caicedo, O.M., & da Fonseca, N.L.S. (2024). Large language models for zero touch network configuration management. *ArXiv*. doi: 10.48550/arXiv.2408.13298.

[15] Lucas-Estañ, M.C., Coll-Perales, B., Khan, M.I., Gozalvez, J., Avedisov, S.S., Altintas, O., & Sepulcre, M. (2024). 5G network architecture and configuration choices to support teleoperated driving at scale. In *Proceedings of the 100th vehicular technology conference* (pp. 1-6). Washington: IEEE. doi: 10.1109/VTC2024-Fall63153.2024.10758026.

[16] Marzuki, K., Kholid, M.I., Hariyadi, I.P., & Mardedi, L.Z.A. (2023). Automation of open VSwitch-based virtual network configuration using ansible on Proxmox virtual environment. *International Journal of Electronics and Communications Systems*, 3(1), 11-20. doi: 10.24042/ijecs.v3i1.16524.

[17] Mehta, R. (2015). *Network improvements in vSphere 6 boost performance for 40G NICs*. Retrieved from https://surl.li/pmnzmm.

[18] Olifirenko, R. (2021). *An improved way of NFV hypervisor functioning in SDN networks*. (Bachelor's dissertation, National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute", Kyiv, Ukraine).

[19] OpenContrail vRouter. (2024). Retrieved from https://docs.mirantis.com/mcp/q4-18/mcp-ref-arch/opencontrail-plan/contrail-vrouter.html.

[20] Patrão, L. (2024). *VMware vSphere essentials: A practical approach to vSphere deployment and management*. Berkeley: Apress. doi: 10.1007/979-8-8688-0208-9.

[21] Pfaff, B., *et al*. (2015). The design and implementation of open vSwitch. In *Proceedings of the 12th USENIX symposium on networked systems design and implementation* (pp. 117-130). Oakland: USENIX.

[22] Poller, J. (2017). *Big switch networks and Dell EMC: Next-generation data center networking*. Retrieved from https://i.dell.com/sites/csdocuments/Shared-Content_data-Sheets_Documents/en/ESG-Lab-Review-Big-Switch-and-Dell-April-2017.pdf.

[23] Rashelbach, A., Rottenstreich, O., & Silberstein, M. (2022). Scaling open vSwitch with a computational cache. In *Proceedings of the 19th USENIX symposium on networked systems design and implementation* (pp. 1359-1374). Renton: USENIX.

[24] Rekha, P.M., & Dakshayini, M. (2015). Dynamic network configuration and virtual management protocol for open switch in cloud environment. In *Proceedings of the international advance computing conference* (pp. 143-148). Bangalore: IEEE. doi: 10.1109/IADCC.2015.7154687.

[25] Rush, S. (2020). Virtual switches and indicators in automotive displays. *SAE International Journal of Advances and Current Practices in Mobility*, 2(4), 2418-2424. doi: 10.4271/2020-01-1362.

[26] Sadrhaghighi, S., Dolati, M., Ghaderi, M., & Khonsari, A. (2022). Monitoring OpenFlow virtual networks via coordinated switch-based traffic mirroring. *IEEE Transactions on Network and Service Management*, 19(3), 2219-2237. doi: 10.1109/TNSM.2022.3149734.

[27] Singh, A. (2019). *Development of Python API for a network switch*. (Bachelor's dissertation, Jaypee University of Information Technology Waknaghat, Waknaghat, India).

[28] Tchendji, V.K., Yankam, Y.F., & Myoupo, J.F. (2018). Conflict-free rerouting scheme through flow splitting for virtual networks using switches. *Journal of Internet Services and Applications*, 9(1), article number 13. doi: 10.1186/s13174-018-0085-4.

[29] Vector Packet Processing. (n.d.). Retrieved from https://www.netgate.com/resources/articles-vector-packet-processing.

[30] Wang, C., Scazzariello, M., Farshin, A., Ferlin-Reiter, S., Kostic, D., & Chiesa, M. (2024). NetConfEval: Can LLMs facilitate network configuration? *Proceedings of the ACM on Networking*, 2, article number 7. doi: 10.1145/3656296.

[31] Wang, K., Zhao, C., Chu, J., Shi, Y., Lu, J., Lyu, B., Zhu, S., Cheng, P., & Chen, J. (2024). LFVeri: Network configuration verification for virtual private cloud networks. *IEEE/ACM Transactions on Networking*, 32(6), 5475-5490. doi: 10.1109/TNET.2024.3469386.

[32] Wang, Y., Gobriel, S., Wang, R., Tai, T.-Y.C., & Dumitrescu, C. (2018). Hash table design and optimization for software virtual switches. In *Proceedings of the 2018 afternoon workshop on Kernel bypassing networks* (pp. 22-28). New York: Association for Computing Machinery. doi: 10.1145/3229538.3229542.

[33] Wang, Y., Wang, X., Huang, Z., Li, W., & Xu, S. (2022). Joint optimization of dynamic resource allocation and packet scheduling for virtual switches in cognitive internet of vehicles. *EURASIP Journal on Advances in Signal Processing*, 2022, article number 32. doi: 10.1186/s13634-022-00862-7.

[34] Yalda, K., Hamad, D.J., & Tapus, N. (2024). Comparative analysis of centralized and distributed SDN environments for IoT networks. *Journal of Control Engineering and Applied Informatics*, 26(3), 84-91. doi: 10.61416/ceai.v26i3.9164.

[35] Yang, Y., Guo, S., Liu, G., & Yi, L. (2021). Fine granularity resource allocation of virtual data center with consideration of virtual switches. *Journal of Network and Computer Applications*, 175, article number 102916. doi: 10.1016/j.jnca.2020.102916.

[36] Zhu, L., Karim, M., Sharif, K., Xu, C., Li, F., Du, X., & Guizani, M. (2020). SDN controllers: A comprehensive analysis and performance evaluation study. *ACM Computing Surveys*, 53(6), article number 133. doi: 10.1145/3421764.

Порівняльний аналіз різних віртуальних комутаторів для підвищення ефективності функціонування мереж різної конфігурації

Олександр Берестовенко
Аспірант
Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського»
03056, просп. Берестейський, 37, м. Київ, Україна
https://orcid.org/0000-0003-4887-4674

Анотація. Мета роботи полягала у визначенні оптимального типу віртуального комутатора для забезпечення максимальної ефективності роботи комп'ютерних мереж різних конфігурацій, з огляду на їхні технічні характеристики, можливості та рівень інтеграції. Було проведено порівняльний аналіз продуктивності та функціональних можливостей віртуальних комутаторів. Основні результати показали, що Cisco Nexus 1000V забезпечує відмінну продуктивність та низьку затримку (0,5-2 мілісекунд), що робить його ідеальним для середовищ, де критичні швидкість та реакція мережі. Open vSwitch характеризується високою масштабованістю і ефективним використанням пам'яті, з пропускною здатністю до 9 гігабіт на секунду і помірним використанням процесора, що робить його підходящим для масштабованих віртуалізованих середовищ. VMware vSwitch, із пропускною здатністю 6-8 гігабіт на секунду, має хорошу інтеграцію у середовище VMware та зручне налаштування. Більше того, виявлено, що Virtual Packet Processing забезпечує найкращу пропускну здатність, досягаючи значень від 20 до 50 гігабіт на секунду, а також демонструє низьку затримку в межах 0,3-0,5 мілісекунд, що робить його оптимальним вибором для середовищ із високими вимогами до пропускної здатності. У той самий час, Bridge Virtual Switch має найменше навантаження на процесор (5-10 %), що дозволяє зберігати продуктивність навіть за обмежених апаратних ресурсів. Інші комутатори, а саме Hyper-V Virtual Switch, Juniper Contrail Virtual Router, CloudStack Virtual Router та Huawei CloudEngine vSwitch продемонстрували хорошу ефективність і можуть бути корисними для середовищ з меншими вимогами до пропускної здатності та масштабованості. Отримані результати показали, що вибір віртуального комутатора залежить від специфічних вимог, оскільки кожен комутатор має свої переваги та обмеження, що визначають його оптимальність для різних мережевих конфігурацій

Ключові слова: пропускна здатність; масштабованість компонентів; інтеграція та продуктивність систем; обробка даних; зниження затримок