Towards a Scalable SDN Hypervisors Framework

Aamir Hussain[1], Sajid Ali[2], Mubashir Ali[3], Sarfraz Hashim[4],

dr.mubashirali1@gmail.com

Department of Computer Science, Muhammad Nawaz Shareef University of Agriculture, Multan, Pakistan[1,4];
Department of Information Technology, University of Education, Lahore, Pakistan[2];
Department of Software Engineering, Lahore Garrison University, Lahore, Pakistan[3];

Abstract

Software-Defined Networking (SDN) is a new emerging networking paradigm that has adopted a logically centralized architecture to increase overall network performance agility and programmability. Combining network virtualization with SDN will guarantees for combined advantages of improved flexibility and network performance. Combining SDN with hypervisors divides the network physical resources into several logical transparent and isolated virtual SDN network (vSDN), where each has its virtual controller. However, SDN hypervisors bring several advantages as well as several challenges to its network operators as for the virtual appliances, their efficient placement, assurance of network performance is mandatory, and their dynamic instantiation with their migration. In this article, we provide a brief and concise review of network virtualization along with its implementation in the SDN network. SDN hypervisors types are discussed, and taxonomy is provided to demonstrate the importance of hypervisors in SDN. A comparison of SDN hypervisors is performed to elaborate on the vital hypervisor software along with their features, and different challenges are discussed faced by the SDN network. A framework is proposed to add combined functionalities of hypervisors to create a more effective and efficient virtual system. The purpose of the framework is to increase network performance through proper configuration of resources, software, control plane isolation functions with defined rules and policies.

Keywords

SDN virtualization, SDN Hypervisors, Hypervisor features, Taxonomy, SDN Hypervisor Framework

1. INTRODUCTION

Recently SDN networking paradigm is the new buzzword which decouples the network control plane from the underlying forwarding plane and results in revolving traditional wireless network, its application, and [1] complicated routing devices into simple switches. The intelligent, logically centralized controller is implemented to follow network policies. In addition to providing new networking services, SDN also supports network virtualization, where the network could be abstracted for the support of different controller applications such as all are tailored for the need of each particular 'tenant/traffic.' SDN virtualization function is basically projected to provide network service flexibility, to promote diversity, to increase manageability, to provide security, and to reduce the market time of new services.

A hypervisor is a major component to provide the virtualization in the network. It acts as an underlying software that has initially developed to monitor virtual machines that are running on it and also called Virtual Machine Monitors (VMM) [2]. Several virtual machines operate on a single particular computing platform. By adding hypervisors in the SDN network for network virtualization, available bandwidth is divided into multiple channels. It is divided in such a way where each channel is independent of each other and can be assigned to different servers [3]. SDN proposes extraordinary control to network administrators via running services on a logically centralized controller [4]. Hypervisor deals with the several challenges of stitching by making virtual SDN slice together from the physical SDN network [2]. Network Hypervisors contribute to the abstraction of several SDN networks as an independent virtual slice. Figure 1 shows the virtualization in SDN through Hypervisor. SDN hypervisor abstracts network physical resources into several logical transparent and isolated virtual SDN network (vSDN), where each has its own virtual controller. The hypervisor may be supported by the host Operating System (OS) and make use of host OS that allows the installation of the guest OS and secure management.

Along with the many benefits of creating vSDNs, sometimes network operators may face many issues while deploying its virtual appliances. vSDNs brings several challenges to its network operators as for the virtual devices, their efficient placement, assurance of network performance is mandatory, and their dynamic instantiation as well as their migration. A most critical challenge of a

Manuscript received February 5, 2024
Manuscript revised February 20, 2024
https://doi.org/10.22937/IJCSNS.2024.24.2.8

virtualized network is network performance, and with this IP addressing version is also has a role in network performance [5]. Research has shown that network virtualization may lead to latency and throughput in terms of providing speed for processing of data [6]. Ensuring network performance is the main issue for providing high-performance services to users. Besides the aspect of network performance, many other network issues are confronted that how smoothly the existing network migrates to network virtualization-based solutions.

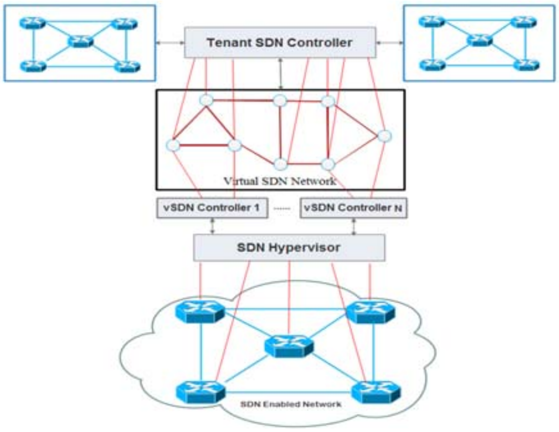


Fig. 1. Virtualization in SDN through Hypervisors

Moreover, the purpose of this paper is to identify the challenges of virtualization and discussing SDN hypervisors. The current state of virtualization in SDN, along with its hypervisors, is considered for getting a more precise image of SDN virtualization. Hypervisors types are discussed with the taxonomy that shows its importance in SDN. A comparison of SDN hypervisors is performed to analyze and conclude the best performance hypervisors along with features they are supporting. Different challenges faced by the research community are discussed in the paper. A framework is proposed to add combined functionalities of hypervisors in SDN to create a more effective and efficient virtual network. The purpose of the framework is to increase network performance through proper configuration of resources, software, control plane isolation functions, rules, and policies as defined. In the end, we conclude what challenges must be addressed to become conscious of a feasible distributed network virtualization environment.

This paper is structured as follows: Section 2 presents the background; section 3 presents SDN hypervisors, types, hypervisor taxonomy, and Comparative analysis of SDN hypervisors. In section 4, Challenges in the implementation of virtualization and Hypervisor in SDN are discussed. In section 5, the proposed framework and its components are discussed. In section 6, we conclude this paper and suggest challenges that should be addressed to become conscious of available distributed network virtualization in the SDN environment.

2. BACKGROUND

In recent years Network virtualization concept has been pressed forward by its promoters for the solution of long-term plodding ossification problem that is being faced by the existing Internet. It has been proposed for being an integral part of the networking paradigm. Memory was the first component that was virtualized. The concept of virtualization of memory was developed in the 1970s [7]. SDN will take place and used in all types of networks [8]. Virtualization provides ease of management, as shown in figure 2. Virtualization's main advantage is the implementation of time-sharing mechanisms using available physical resources that lead to the increased efficiency of the network [9]. It is also the primary need for sharing resources, and it is best to divide single resources into multiple resources so that everyone can share [7]. Another reason for virtualization is that users need isolation between them. The inherited concept of isolation from applications is adapted to provide separation between users. The personal data must not be shared with anyone. Vulnerability in-network can reduce reliability [10]. Aggregation of small resources into big is significant in virtualization. Virtualization provides a stable and suitable way of creating a reproducible environment for testing the software [9]. Processes execution can be performed transparently from one to another system. If the system fails, virtualization enables the migration of currently running processes to another virtual machine to be transparent, which provides higher availability of service for running multiple virtual machines. We need NIC (Network Interface Card) for each machine here comes the use of hypervisor software that provides many virtual NIC implementation and processors virtualization. These Virtual NICs (vNIC) are then interconnected via a virtual switch that is connected to the physical network containing NIC. VM software vendors proposed this first approach of providing software vNICs via hypervisor software. Most commercially available hypervisors use host operating systems that allow providing a secure management module as well as the installation of all guest operating systems. Software-Defined Networking (SDN) is a guaranteed 'paradigm' for flexible network communication. SDN networks are virtualized through hypervisors that provide the overall functionalities of virtualization in the SDN environment. The main functions that a hypervisor implements for the creation of a virtual SDN network are discussed in the paper. Now the SDN network is viewed as a single programmable entity, which means that the SDN network may be employed for the implementation of vSDNs [2].SDN network mostly virtualized with the insertion of a hypervisor layer between the physical network layer and the control plane layer, as illustrated in Figure 1. The hypervisors view an abstract whole physical SDN network through multiple interfaces, including multiple vSDN controllers. Some non-virtualized network controllers may also provide virtualization, but those controllers have limitations such as transparency and allow control of virtualization only via special interfaces. The hypervisor creates an isolated vSDN network that is controlled by vSDN controllers.

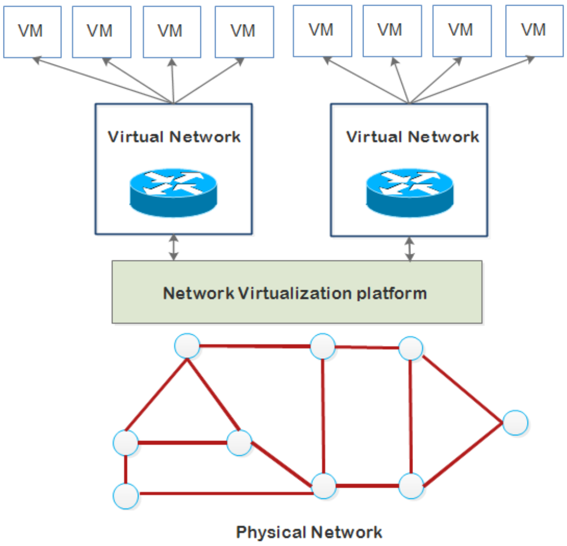
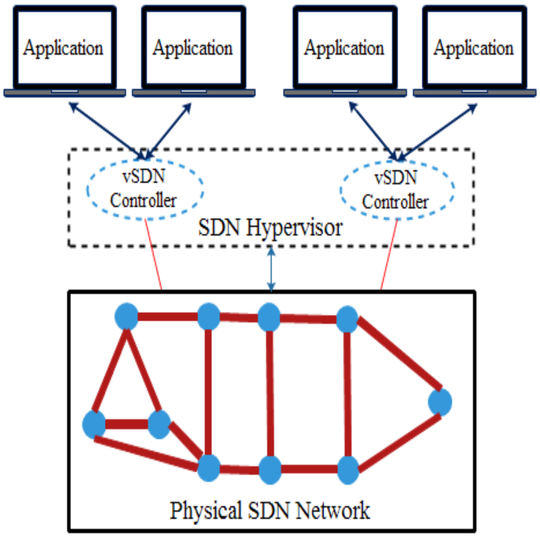


Fig. 2. View of Network Virtualization

Many challenges are discussed in this paper in the context of providing a scalable network. The biggest challenge of virtualization is overhead, which decreases performance; sometimes, performance is compromised by providing more flexibility. The development teams have been working hard to lower the hanging problem. Another challenge includes identifying a single point of failure in hardware. Virtual machines are decoupled from hardware, but they still dependent on devices. If the hardware failure occurs, it will lead towards virtual machine failure and will force a reboot. SDN hypervisors face challenges in the context of providing interfaces to each tenant and allocating each tenant resources on their demand. Managing overall virtual topologies so that it becomes easy for granting requests from different APIs. Discovering topologies and then maintaining through controllers need more mechanisms. We proposed a framework at the end of this paper to increase the scalability and performance of the overall network by providing new policies and rules. Different components are added in different blocks to overcome the current problems in the underlying system. We assume that this framework will increase the performance of the system and make it more scalable.



3. SDN HYPERVISOR: VIRTUAL MACHINE MONITOR

This section describes SDN hypervisors for providing a virtualization layer for logically distributing underlying physical resources. Section 3.1 defines the SDN hypervisor's taxonomy to highlight different attributes for abstraction and isolation of material resources. The distinct architecture proposed for SDN hypervisors is categorized as distributed and centralized architectures.

Section 3.2 provides a comparison of SDN hypervisors to outline the supported features by different architectures.

Hypervisors are sometimes known as virtual machine monitors that are the significant component to provide virtualization. It was initially developed in the area of virtual computing for monitoring running virtual machines [2, 3, 9]. Several virtual machines operate on a specific computing platform and have their own operating system (OS) to run on the underlying physical computing platform with full virtualization. Aside from monitoring, hypervisors allocate resources on these physical platforms. An SDN hypervisor monitors virtual machine networks and allocates resources to individual virtual networks like link and buffer capacity and switching nodes buffer capacity. It will increase the performance of the overall SDN network. There are two types of hypervisors known as type I and type II hypervisors shown in figure 3.

Type 1 includes hypervisors that run on top of the underlying hardware layer. Type I also called bare-metal or native hypervisor as it plays a vital role in scheduling and resource allocation to virtual machines because it is running without any OS. Physical resources of an SDN network are allocated to virtual machines by this hypervisor layer. It abstracts all the process and provides transparency for its users.

Type II hypervisors are running on OS as an application and host OS that does not have any knowledge about the hypervisor. It works the same as type I, but the main difference is its deployment on OS.

Fig. 3. (a) Type I of SDN Hypervisor

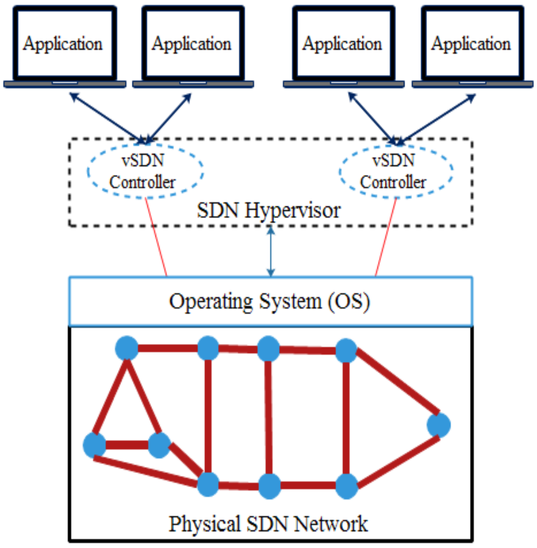


Fig. 3. (b) Type II of SDN Hypervisor

3.1 TAXONOMY

SDN hypervisors are nothing more than a data-plane node for its controller, and for the data-plane node, it is an SDN hypervisor. SDN hypervisors sit between SDN physical network and vSDN controllers [11]. Hypervisor implementation in the SDN network adds the entire virtualization stack to virtualized underlying physical resources of the SDN network [11]. The SDN hypervisor provides isolation slices for vSDNs that share a physical SDN network. In the SDN network, hypervisor abstracts underlying physical network-specific characteristics. The degree of abstraction in the network represents the level of virtualization. Figure 4 shows a taxonomy that defines the general tendency of hypervisors and their architectures. Taxonomy describes hypervisors in the context of providing the abstraction to the overall SDN network and providing security and resource isolation. The two types of hypervisors are further divided for centralized and distributed architectures that are discussed in the paper.

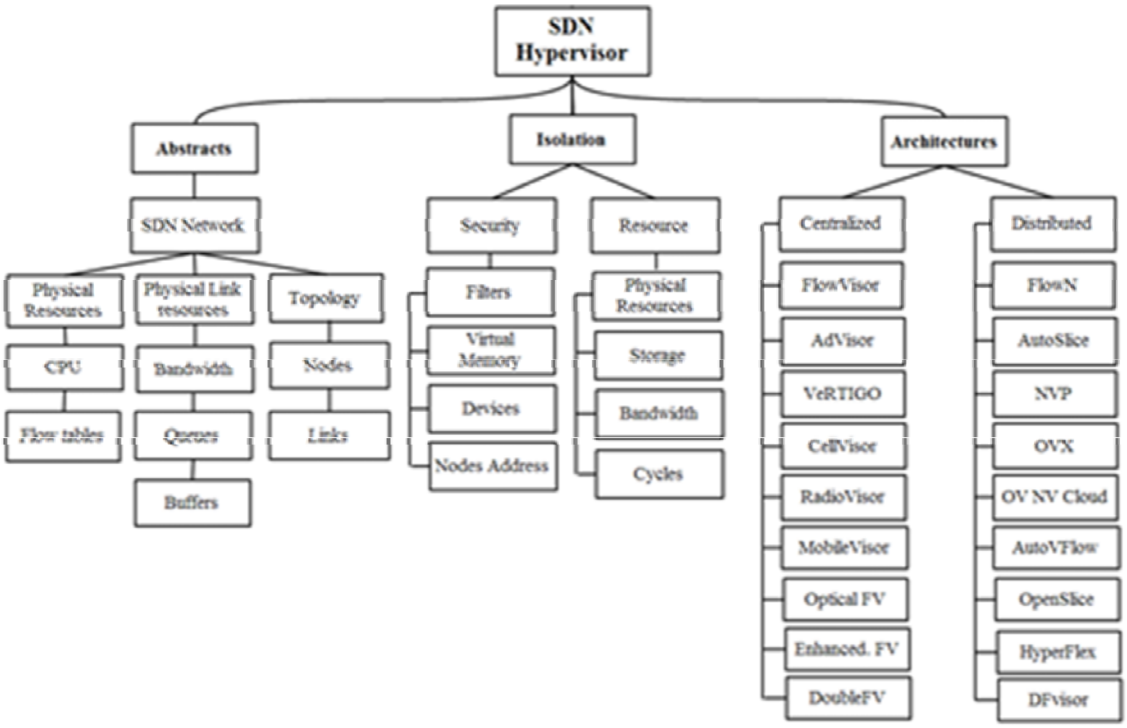


Fig. 4. Taxonomy of SDN Hypervisor

Table 1 provides a description of the taxonomy of hypervisors.

Table. 1. Various characteristics of SDN Hypervisors

Characteristics	Description
Abstraction:	There are three main attributes types of physical SDN networks that are considered for abstraction in physical SDN networks. These attributes include 1) physical node resources, 2) physical link, and 3) topology[2]. **Physical Node Resources:** It includes flow table resources, including flow messages, and CPU resources of the SDN switch link define its virtualization level.
Isolation:	**Physical Link Resources:** It includes bandwidth (BW) such as available link queues, transmission bit rate, link capacity, and link buffers define its virtualization level.
Architectures:	**Topology:** It includes the virtual nodes and links that define the virtualization level in the network. Mainly hypervisor abstracts network path and provides end-to-end traversing of physical links[2, 12]. The hypervisor provides isolation slices for vSDN sharing physical SDN networks. It provides isolation for all physical link transmission bit rate as a requirement for providing Quality of Service (QoS)[2, 4, 12]. **Resource isolation:** It includes disk storage limits in which user can state limits on a maximum number of nodes to be allocated. One can specify the following restrictions for memory storage 1) physical resources, 2) storage, 3) bandwidth, and cycles[2]. **Security isolation:** Hypervisor filter processes to hide all means outside the scope and for the prohibition of any unwanted interaction between inside the

Manuscript received February 5, 2024
Manuscript revised February 20, 2024
https://doi.org/10.22937/IJCSNS.2024.24.2.8

	operations of virtual memory and the processes belonging to other virtual machine's memory. It includes securing the network through filtering devices and the address of nodes connected with each other. **Centralized Architecture:** A hypervisor consisting of a single central entity is classified as centralized architecture. Centralized hypervisors have not to distribute their hypervisors' functionalities. The centralized controller controls several elements of the network. The primary object serves multiple application controllers, as shown in Figure 1. **Distributed Architecture:** Distributed hypervisors consist of logically separated running functions. These hypervisors include of the central controller as a centralized hypervisor; hence, distributed hypervisor consist of multiple distributed functionalities. Distributed hypervisors decouple management module functions from isolation functions.

3.2 COMPARISON OF HYPERVISORS

A comparison of different SDN hypervisors is performed for the performance evolution of hypervisors. The following table 2summarizesexisting hypervisors differences and helps to focus on strengths and areas that need attention. Different hypervisors used are briefly summarized as follows:

FlowVisor: FlowVisor (FV) hypervisor is SDN architecture to achieve virtualization. It is a very first hypervisor for virtualization, and sharing SDN resources reside on OpenFlow protocol. The hardware abstraction was the main problem that motivates for development of FlowVisor hypervisor. Flow Visor's goal is to run different networks in a transparent manner on the same physical hardware and to provide an extensible and flexible environment [2].

FlowN: It is a hypervisor that is distributed across the network for virtualization. FlowN architecture provides virtualization to container-based applications that host user controllers.

HyperFlex: HyperFlex is SDN architecture to achieve virtualization. HyperFlex is a hypervisor that operates in distributed behavior. It realizes the virtualization need according to its available capacities in the network. HyperFlex interconnects the needed virtualization functions [2].

AutoSlice: It is a centralized hypervisor used to improve scalability by the distribution of hypervisor workload. It is partitioned into multiple controllers, a single management module, and each physical domain having one controller.

VeRTIGO: VeRTIGO, it is a network virtualization architecture that simply extends Flow Visor to deal with scenarios of virtualization described before. The basic idea behind this was to extend Flow Visor with additional intelligence features to expose several views of the network to different network controllers that depend on the user's specific requirements.

Xen: It is a software layer hypervisor that resides on hardware by allowing multiple virtual user OS for communication in a secure, efficient, and resourceful manner. The management of CPU, memory, and scheduling virtual machines in visualize way is the responsibility of. Domain 0 and Domain U are key prestigious domains in Xen hypervisor [3].

VMware: The VMware hypervisor is a Mobile Virtualization Platform (MVP) that provides end-to-end solutions for the management of enterprise employee virtual devices, including phones, sensing devices, and remote devices [13]. Since the employee phone device is physical platform MVP hypervisor becomes essential to run on wide varieties of mobile devices [3].

Table 2: Comparison of SDN Hypervisor

Features	**Hypervisors**						
	FlowVisor [14, 15]	**FlowN [16]**	**HyperFlex [17]**	**Autoslice [15]**	**VeRTIGO [18]**	**Xen [3, 19]**	**VMware [3, 13]**
Architecture	Centralized	Distributed	Distributed	Distributed	Centralized	-	Centralized
Scalability	High	High	High	High	High	High	High
Reliability	-	-	-	-	-	✗	✗

Availability	High	High	High	High	High	High	High
Fault tolerance	-	-	-	-	-	✗	✗
Topology	-	✓		✓	✓	✓	✓
Physical node resource	-	-	✓	-	✓	✓	✓
Physical link resource	-	-	✓	-	-	✓	✓
Isolation Attributes: Control Plane Instances	-	Threads	CPU	-	-	CPU	-
Data Plane	BW, CPU	-	-	-	-	-	-
vSDN addressing	Configurable non-overlap Flow-space	VLAN	-	VLAN, MPLS	-	-	-
Transparency	✓	✓	✓	✓	✓	✓	✓
Software Implementation	✓	✓	✓	✓	✓	✓	✓
High Performance	-	✓	✓	✓	✓	-	✓
Support multiple tenants	✓	✓	✓	✓	✓	✗	✓
Support flow table control messages	✓	-	-	✓	✓	✓	-

4. CHALLENGES

There are a variety of SDN hypervisors that are heterogeneous in nature. Various data plane and control plane attributes of the network need to be selected for providing abstraction and isolation in the network. The grouping of SDN hypervisors along with abstraction and isolation functions directly impact network performance. Different studies have demonstrated only the concept of available SDN hypervisor rand discussed abstraction and isolation implemented results. In particular, we observe that there is no best SDN hypervisor design that fulfills all requirements above comparison shows that there is not a single hypervisor that has different features that may satisfy network requirements. Instead, this survey suggests detail performance evaluations of a vSDN hypervisor. Many vSDN hypervisors challenges are discussed as follows:

Bootstrapping: Network connectivity is a prerequisite for connecting to a virtual network. A clear procedure of bootstrapping is missing in vSDN [2]. Bootstrapping capabilities is must permit service providers to modify the allocated virtual nodes and links through appropriate interfaces. The user requires network connectivity that will be present all the time to handle user requests. Presently there are no predefined mechanisms for bootstrapping in vSDN. All connected components in vSDN must be connected with one another and bootstrapped. Hypervisors and controllers need to communicate for communication.

Interfacing: Interface Provider must provide interfaces to the service provider to communicate with each other and state their requirements [20]. In addition providing an appropriate interface between the users and, Service Providers (SP) and between multiple interface providers and SP must be standardized as well as identified [6]. Managing all tenant controllers, along with their interfaces, require new policies and rules for managing without latency and to improve performance.

Discovery of Resource and Topology: Virtualization hypervisors face challenges as discovering the presence of topologies adding these in the SDN network increases challenges. Discovery of resources should be present to allow communication, interaction, and collaboration to provide complex services [4, 20]. The topology of an underlying network must be managed, and the status of its corresponding elements must be arranged to allocate different resources on the request of Service providers. Resources include physical nodes, physical links, capacities in nodes and interconnections between them, etc.

Resource Allocation: Due to the heterogeneity of vSDN efficient allocation of hardware-based resources among multiple networks is more critical for maximizing coexisting virtual network numbers. Dynamic scheduling increases the utilization of Interface Providers [21]. Exploiting different topologies presence and opportunities leave sufficient space for research on the customize solutions and better approximation algorithms. Adequate resource allocation needs further consideration in the future.

Reliability and Fault Tolerance: The crucial aspect for actual deployment of SDN hypervisor layer needs investigation for reliability. Firstly, new mechanisms must be defined to recover from vSDN hypervisors' faults and failures that can cause significant damage. The hypervisor's development process must include redundancy and determine its levels to offer reliable vSDN [16].

Scalable Hypervisor Designing: In the SDN environment, hypervisor faces a lot of scalability challenges that lead to an enormous amount of load and demands for changing in the network [22]. An SDN hypervisor on its entity would not be sufficient, so designing distributes hypervisor should be considered in detail. The need for hypervisor scalability for distribution must be addressed and examined as a whole — more efficient controller algorithms for tenants and switches needed for assigning to hypervisors [23].

Self-Configuration and Self-Optimization: Hypervisors should provide the best performance for different virtual network topologies that are independent of their underlying vSDN network hardware. Hypervisors have to be designed for becoming highly adaptable for their continual performance. Hypervisors must implement a new mechanism for self-configuration and self-optimization. Hypervisors need to improve their operations. Self-reconfiguration and self-optimization must be transparent for better performance of vSDN. This will incur the minimal overhead of configuration for the hypervisor operator.

Hypervisor Security: Security is the main concern as many tenants are connected to the hypervisor needs to protect from attacks. It needs to define policies and procedures for all traffic types. Isolation between networks can provide a certain level of security by using secured tunnels and encryptions etc. Security and privacy issues must be identified and explored for ensuring the whole network [24]. Host machines are the control point in the virtual environment. It includes implications for enabling hosts to monitor and communicate with connecting running applications. Host machines should be strictly protected. Monitoring interfaces are so challenging, and hosts can shut down, start, pause, and restart the machine. Sometimes host machines monitor virtual machines and can modify available resources [25].

Hypervisor placement: Hypervisors lies between the physical network and controllers. Where to place hypervisor needs detail investigations. Tenants controllers placement and vSDN switch placement should be considered while setting a hypervisor in the network [9]. The careful installation of hypervisor increases performance through abstraction and isolation functions.

Resistance to Attacks: Physical resources, including CPU, disk memory, and different network resource, are shared through virtual machines. Sometimes it becomes possible for the guest to enforce a DoS attack on the system. In a virtual environment, the DoS attack occurs when a guest machine tries to take all possible resources in the network. Preventing the host machine from a single virtual machine is most important. If the attacker gains administrator privileges, it can break into the virtual machine. The attack is named as Guest-to-Guest because the attacker jumps from one virtual machine to another virtual machine.

5. FRAMEWORK FOR SDN HYPERVISORS

Different hypervisors, as discussed before having different functionalities that improve the performance of an overall network. We try to combine different hypervisors' features that will enhance the performance of the SDN network. Unlike works of literature discuss specific hypervisors their working and functionalities, the overall SDN network virtualization performance discussion is missing. Mostly data plane virtualization is provided, but control plane virtualization is missing. Figure 5 shows the proposed framework. The proposed framework can be defined as four different blocks. Each block includes functionalities that should be added to vSDN network for a more secure, effective, and efficient virtual network through Hypervisors. The purpose of this framework is to increase network performance through proper configuration of resources, software, control plane isolation functions, rules, and policies as defined. Four blocks of a framework are described as follows:

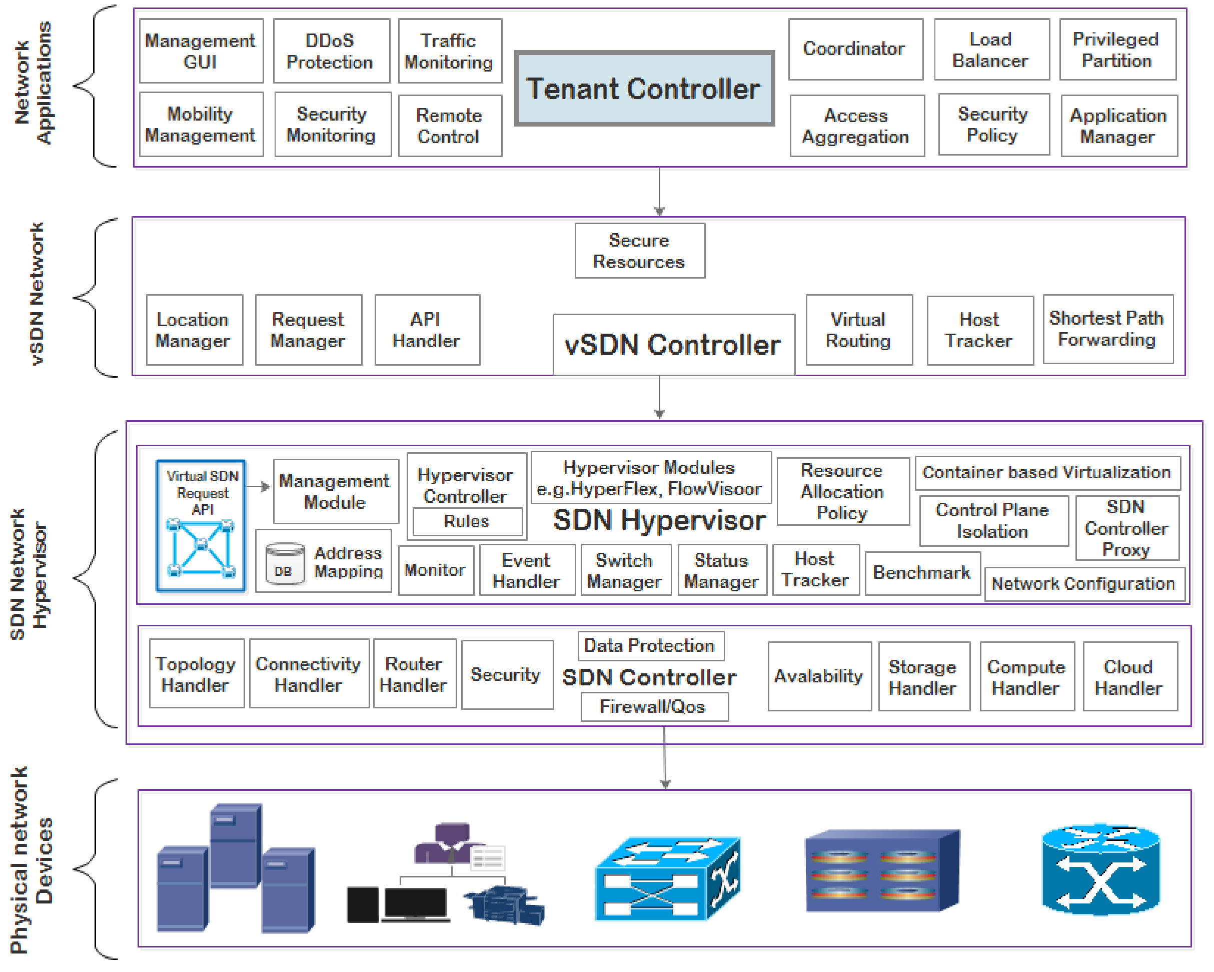


Fig. 5. Scalable SDN Hypervisor Framework

5.1 Physical Network Devices

The first block includes all the material resources of the network. The underlying SDN networks physical resources, including Ethernet switches, routers, servers, and networks. All these physical resources are part of the network and form a data plane [26].

5.2 SDN Network Hypervisor

The next module includes the SDN hypervisor. SDN hypervisors lie between SDN physical network and vSDN controllers. SDN hypervisors interconnect vSDN controllers or physical resources to create a virtual network. The hypervisor software consists of an SDN hypervisor and controller. The hypervisor is operated and controlled by the hypervisor SDN controller. The proposed framework includes hypervisor software to provide flexibility and efficiency and to allocate virtualization functions in the network, on a per-function basis, among servers and SDN network elements of the hypervisor network. Different modules are implemented to enforce control plane shaping policies and to set up isolation functions. These modules are discussed below.

SDN Control handler: The overall functionality of the controller here is to handle full applications and their availability along with a security measure for creating secure channels and managing integrity and confidentiality [27]. All controllers are responsible for setting up paths and rip downflows in a network for managing resources. The controllers handle all the information resources handling, capacity, topologies handling, and connectivity of open API [26].

Resource Allocation Policy: FV hypervisor is SDN architecture to achieve virtualization. It operates transparently as a proxy controller that lies between the physical resources and controllers of the network. The purpose of proposing FV was to make current SDN network management more efficient. Modification of the header fields for accessing resources is a significant threat to network security [28]. FV provides a solution for making policies of resource allocation by developing a priority-based mechanism by setting access privileges to controllers. Another solution is provided by setting rate limitations on both controllers and physical resources.

Management Modules: It is added to control the flow of all requests and manage the flow table, including flow table identifiers and packet identifiers [29].Management modules are added to control the load of the network by distributing the topologies mapping, resource assignment and coordination, SDN network segmentation, and connected physical devices, including switches migration [30]. The load that flows from all the attached resources must be managed to control the capacity of the network. All requests from API are handled by using different added modules, so the network is no flooded with requests waiting for the response.

SDN Controller Proxy: Controller proxies help the distributed SDN hypervisors to control load and to handle each domain's operations [29]. The functions of the controller proxy include infrastructure flow setup, flow cache management, flow space allocation, and message translation.

Control Plane Isolation: HyperFlex hypervisor ensures isolation of control plane slices for vSDN networks [28]. The primary purpose of this is to protect resources from exhaustion. If this function is added to the hypervisor, it will provide isolation with the protection of all funds.

Benchmark and Monitoring: Software module is added to monitor overall network CPU utilization for the process [31]. The experiential CPU values are provided to the isolation module. The monitoring function is added inspired by the previous work to incremental updating of computational SDN hypervisor [32]. It allows multiple nodes to collaborate for assessing the same processing traffic.

5.3 Virtual SDN Network

The virtual SDN network consists of virtual nodes, virtual interfaces, and virtual links. Virtual topologies in a network composed of all these components. The virtual nodes are connected through virtual interfaces with virtual links [4]. A server or SDN-based switches work as a virtual node with virtual interfaced that includes constraint in place of some processing capabilities. Managing all GUI requests is an essential responsibility of the controller. Tracking all unique host addresses and verifying their identity by handling all the network connected tenants.

5.4 Network Applications

The hypervisor software provides functions for large-scale networks interconnect with multiple network elements, and the hypervisor provides tenants controllers for these networks to communicate in a proper manner [17]. To control vSDN network switches, the controller software can be provided by each tenant. Each tenant can build up their controller applications, and after building, they can submit a request to run as software on a physical resource controller. Application block shows the functions of managing applications, including all APIs efficient networking management, monitoring flow of traffic, managing load, managing all interfaces that are part of the network [26]. Security monitoring is also the primary concern to maintain backward and forward security in the network. Access control for managing resources and access aggregation with privileged restrictions. A mapping is required for a hypervisor to process resources request rate and manage the distribution of resources between connected nodes [22]. By adding the mapping function, it guarantees that enough resources are allocated on demand to increase the performance on the market.

6. CONCLUSION

Recently SDN networking paradigm is the new buzzword that decouples the network control plane from the underlying forwarding plane and results in revolving traditional complicated routing devices into simple switches. The intelligent, logically centralized controller is implemented to follow network policies. By adding hypervisors in the SDN network for network virtualization, available bandwidth is divided into multiple channels and divided in such a way where each channel is independent of each other and can be assigned to different servers. SDN hypervisors are nothing more than a data-plane node for its controller and for the data-plane node, it is an SDN hypervisor. SDN hypervisors sit between SDN physical network and vSDN controllers. Hypervisor implementation in the SDN network adds the entire virtualization stack to virtualized underlying physical resources of the SDN network. Different pieces of literature discuss many challenges in the context of SDN hypervisors. Discussed problems of SDN hypervisor must be addressed to become conscious of a feasible distributed network virtualization environment. The current state of virtualization in SDN, along with its hypervisors, is vital to be considered for getting a more precise image of SDN virtualization. The comparison of SDN hypervisors analyzes and concludes that features supported by these hypervisors are essential to increase the performance of the network. The purpose of our framework is to consider the entire network needed functionalities to improve the efficiency and performance and SDN environment. The framework implementation will increase network performance through proper configuration of resources, software, control plane isolation functions, rules, and policies as defined.

REFERENCE

1. Latif, S., S. Akram, and M.A. Saleem, *Channel assignment using differential evolution algorithm in cognitive radio networks.* International Journal of Advanced and Applied Sciences, 2017. **4**: p. 160-166.
2. Blenk, A., et al., *Survey on network virtualization hypervisors for software-defined networking.* IEEE Communications Surveys & Tutorials, 2016. **18**(1): p. 655-685.
3. Malik, M.V. and C. Barde, *Survey on architecture of leading hypervisors and their live migration techniques.* International Journal of Computer Science and Mobile Computing, IJCSMC, 2014. **3**(11).
4. Drutskoy, D.A., *Software-defined network virtualization with flying*. 2012, Princeton University.
5. Basit, A. and R. Hussain, *Performance evaluation of simultaneous network configuration using dual-stack and tunnel transition techniques: An enterprise-level analysis.* INTERNATIONAL JOURNAL OF ADVANCED AND APPLIED SCIENCES, 2017. **4**(1): p. 102-109.
6. Chowdhury, N.M.K. and R. Boutaba, *Network virtualization: state of the art and research challenges.* IEEE Communications Magazine, 2009. **47**(7).
7. Jain, R. and S. Paul, *Network virtualization and software-defined networking for cloud computing: a survey.* IEEE Communications Magazine, 2013. **51**(11): p. 24-31.
8. Alam, S., et al., *Dynamic resource allocation for cognitive radio based smart grid communication networks.* Int. J. Adv. Appl. Sci., 2017. **4**(10): p. 76-83.
9. Khan, A., et al., *Network virtualization: a hypervisor for the Internet?* IEEE Communications Magazine, 2012. **50**(1).
10. Ahamad, T. and A. Aljumah, *Preventive mechanism against DDoS attacks in MANET.* International Journal of Advanced and Applied Sciences, 2017. **4**(5): p. 94-100.
11. Cziva, R., et al. *SDN-based virtual machine management for cloud data centers.* IEEE Transactions on Network and Service Management, 2016. **13**(2): p. 212-225.
12. Soltesz, S., et al. *Container-based operating system virtualization: a scalable, high-performance alternative to hypervisors*. in *ACM SIGOPS Operating Systems Review*. 2007. ACM.
13. Barr, K., et al., *The VMware mobile virtualization platform: is that a hypervisor in your pocket?* ACM SIGOPS Operating Systems Review, 2010. **44**(4): p. 124-135.
14. Sherwood, R., et al., *Flowvisor: A network virtualization layer.* OpenFlow Switch Consortium, Tech. Rep, 2009. **1**: p. 132.
15. Bozakov, Z., and P. Papadimitriou. *Autoslice: automated and scalable slicing for software-defined networks*. in *Proceedings of the 2012 ACM conference on CoNEXT student workshop*. 2012. ACM.
16. Drutskoy, D., E. Keller, and J. Rexford, *Scalable network virtualization in software-defined networks.* IEEE Internet Computing, 2013. **17**(2): p. 20-27.
17. Blenk, A., A. Basta, and W. Kellerer. *HyperFlex: An SDN virtualization architecture with flexible hypervisor function allocation*. in *Integrated Network Management (IM), 2015 IFIP/IEEE International Symposium on*. 2015. IEEE.
18. Corin, R.D., et al. *Vertigo: Network virtualization and beyond*. in *Software Defined Networking (EWSDN), 2012 European Workshop on*. 2012. IEEE.

19. Sailer, R., et al. *Building a MAC-based security architecture for the Xen open-source hypervisor*. in *Computer security applications conference, 21st Annual*. 2005. IEEE.

20. Rosenblum, M. and T. Garfinkel, *Virtual machine monitors: Current technology and future trends.* Computer, 2005. **38**(5): p. 39-47.

21. Bozakov, Z., and A. Rizk. *Taming SDN controllers in heterogeneous hardware environments*. in *Software Defined Networks (EWSDN), 2013 Second European Workshop on*. 2013. IEEE.

22. Sieber, C., et al. *hvbench: An open and scalable SDN network hypervisor benchmark*. in *NetSoft Conference and Workshops (NetSoft), 2016 IEEE*. 2016. IEEE.

23. Salvadori, E., et al. *Generalizing virtual network topologies in OpenFlow-based networks*. in *Global Telecommunications Conference (GLOBECOM 2011), 2011 IEEE*. 2011. IEEE.

24. Bouras, C., A. Kollia, and A. Papazois. *SDN & NFV in 5G: Advancements and challenges*. in *Innovations in Clouds, Internet, and Networks (ICIN), 2017 20th Conference on*. 2017. IEEE.

25. Sahoo, J., S. Mohapatra, and R. Lath. *Virtualization: A survey on concepts, taxonomy and associated security issues*. in *Computer and Network Technology (ICCNT), 2010 Second International Conference on*. 2010. IEEE.

26. Sezer, S., et al., *Are we ready for SDN? Implementation challenges for software-defined networks.* IEEE Communications Magazine, 2013. **51**(7): p. 36-43.

27. Ali, S.T., et al., *A survey of securing networks using software defined networking.* IEEE transactions on reliability, 2015. **64**(3): p. 1086-1097.

28. You, W., et al., *Towards security in virtualization of SDN.* Int J Comput Control, Quantum Inf Eng, 2014. **8**(8): p. 2014.

29. Bozakov, Z. and P. Papadimitriou. *Towards a scalable software-defined network virtualization platform*. in *Network Operations and Management Symposium (NOMS), 2014 IEEE*. 2014. IEEE.

30. Vilalta, R., et al., *Multidomain network hypervisor for abstraction and control of OpenFlow-enabled multitenant multitechnology transport networks.* Journal of Optical Communications and Networking, 2015. **7**(11): p. B55-B61.

31. Basta, A., et al. *HyperFlex: Demonstrating control-plane isolation for virtual software-defined networks*. in *Integrated Network Management (IM), 2015 IFIP/IEEE International Symposium on*. 2015. IEEE.

32. Jin, X., J. Rexford, and D. Walker. *Incremental update for a compositional SDN hypervisor*. in *Proceedings of the third workshop on Hot topics in software defined networking*. 2014. ACM.