INTERNATIONAL TELECOMMUNICATION UNION
TELECOMMUNICATION STANDARDIZATION SECTOR
STUDY PERIOD 2022-2024

SG11-TD1030/GEN
STUDY GROUP 11
Original: English

Question(s): 14/11 Geneva, 1-10 May 2024

TD

Source: Editors

Title: Consent – draft new Recommendation ITU-T Q.4047 (ex. Q.Scvh-iopt) "Interoperability testing between software-defined networking (SDN) and hypervisor based computing virtualization" (Geneva, 1-10 May 2024)

Contact: Huilei Ren
China Telecom
China
Tel: + 86 17688860805
E-mail: renhl1@chinatelecom.cn

Contact: Linze Wu
China Telecom
China
Tel: + 86 20 38639090
E-mail: wulinz@chinatelecom.cn

Contact: Baohong Lin
China Telecom
China
Tel: + 86 20 38639103
E-mail: linbaoh@chinatelecom.cn

Abstract: This document is the Recommendation ITU-T Q.4047 (ex. Q.Scvh-iopt) "Interoperability testing between software-defined networking (SDN) and hypervisor based computing virtualization". This document includes the results of discussion during the Q14/11 meeting, which was held in Geneva, 1-10 May 2024. It is proposed for consent at the SG11 plenary meeting (10 May 2024).

The following table shows discussion results for input documents:

DOC No.	Source	Title	Result and action
SG11-C437R1	China Telecom	ITU-T Q.Scvh-iopt - Proposal to advance the baseline text and consent	Accepted with modification

Major issues of the draft Recommendation are summarized as follows.

- Modify the summary.
- Update the abbreviations in clause 4.
- Add descriptions of the interaction process in clause 6.
- Editorial enhancement of the sentences from clause7 to 9.
- Other editorial changes.

Draft new Recommendation ITU-T Q.4047 (ex. Q.Scvh-iopt)

Interoperability testing between software-defined networking (SDN) and hypervisor based computing virtualization

Summary

This Recommendation specifies the interoperability testing between software-defined networking (SDN) and hypervisor based computing virtualization. Firstly, this Recommendation introduces the framework for interoperability between SDN and hypervisor based computing virtualization, which includes the target areas, components and interoperability in different scenarios. The corresponding requirements and the data model framework which should be considered during the interoperability testing between SDN and hypervisor based computing virtualization are presented in the next. And test cases of interoperability testing between SDN and hypervisor based computing virtualization are provided as appendix, which describe the involved test procedures.

Keywords

SDN, hypervisor based computing virtualization, interoperability testing.

Table of Contents

Page

Draft new Recommendation ITU-T Q.4047 (ex Q.Scvh-iopt)

Interoperability testing between software-defined networking (SDN) and hypervisor based computing virtualization

1. Scope

The scope of this Recommendation consists of:

– Framework for interoperability between SDN and hypervisor based computing virtualization.

– Interoperability testing (IOPT) requirements between SDN and hypervisor based computing virtualization.

– Data model framework considered during SDN and hypervisor based computing virtualization IOPT.

– Test cases of SDN and hypervisor based computing virtualization IOPT.

2. Reference

The following ITU-T Recommendations and other references contain provisions which, through reference in this text, constitute provisions of this Recommendation. At the time of publication, the editions indicated were valid. All Recommendations and other references are subject to revision; users of this Recommendation are therefore encouraged to investigate the possibility of applying the most recent edition of the Recommendations and other references listed below. A list of the currently valid ITU-T Recommendations is regularly published. The reference to a document within this Recommendation does not give it, as a stand-alone document, the status of a Recommendation.

[ITU-T Y.3300] Recommendation ITU-T Y.3300 (2014), *Framework of software-defined networking.*

[ITU-T Y.3502] Recommendation ITU-T Y.3502 (2014), *Information technology – Cloud computing – Reference architecture.*

[ITU-T Y.3510] Recommendation ITU-T Y.3510 (2016), *Cloud computing infrastructure requirements.*

[ITU-T Y.4040] Recommendation ITU-T Y.4040 (2016), *The framework and overview of cloud computing interoperability testing.*

[ITU-T Y.4043] Recommendation ITU-T Y.4043 (2019), *Interoperability testing requirements of a virtual switch.*

3. Definition

3.1 Terms defined elsewhere

This Recommendation uses the following terms defined elsewhere:

3.1.1 interoperability [b-ITU-T Y.101]: The ability of two or more systems or applications to exchange information and to mutually use the information that has been exchanged.

3.1.2 interoperability testing [b-ITU-T Y.4500.15]: Activity of proving that end-to-end functionality between (at least) two devices is as required by the base standard(s) on which those devices are based.

3.1.3 hypervisor [ITU-T Y.3510]: A type of system software that allows multiple operating systems to share a single hardware host.

3.1.4 Network Virtualization Edge [b-IETF RFC7365]: An Network Virtualization Edge (NVE) is the network entity that sits at the edge of an underlay network and implements L2 and/or L3 network virtualization functions.

3.1.5 virtual switch [ITU-T Q.4043]: Resource abstraction and control function abstracting physical network resources to offer virtual network capabilities.

3.2 Terms defined in this Recommendation

None.

4. Abbreviations and acronyms

This Recommendation uses the following abbreviations:

API	Application Programming Interface
CME	Compute management entity
CSP	Cloud service provider
HM	Hypervisor manager
HTTP	Hyper Text Transfer Protocol
IaaS	Infrastructure as a Service
IOPT	Interoperability Testing
L2	Layer 2
L3	Layer 3
MAC	Media Access Control
NFV	Network Function Virtualization
NIC	Network Interface Card
NME	Network management entity
NVE	Network Virtualization Edge
QinQ	802.1Q-in-802.1Q
SDN	Software-Defined Network
SDNC	SDN Controller
ToR	Top of Rack
VIM	Virtualized Infrastructure Manager
VLAN	Virtual Local Area Network
VM	Virtual Machine
vNIC	Virtual Network Interface Card
vSwitch	Virtual switch
VXLAN	Virtual Extensible LAN
WAN	Wide Area Network

5. Conventions

In this Recommendation:

The keywords "is required" indicate a requirement which must be strictly followed and from which no deviation is permitted if conformance to this Recommendation is to be claimed.

The keywords "is recommended" indicate a requirement which is recommended but which is not absolutely required. Thus, this requirement need not be present to claim conformance.

6. Framework for interoperability between SDN and hypervisor based computing virtualization

According to [ITU-T Y.3502], the resource abstraction and control function provides access to the physical computing resources through software abstraction. Resource abstraction needs to ensure efficient, secure and reliable usage of the underlying infrastructure. SDN [ITU-T Y.3300] and hypervisor based computing virtualization both belong to resource abstraction and control function.

SDN and hypervisor based computing virtualization are capabilities of the IaaS. The IOPT in different cloud capabilities type is different. According to [ITU-T Y.4040], there are three major IOPT scenarios: infrastructure capabilities type IOPT, platform capabilities type IOPT, application capabilities type IOPT. IOPT between SDN and hypervisor based computing virtualization should be concerned as an infrastructure capabilities type IOPT.

Hypervisor based computing virtualization which is composed of hypervisor, hypervisor manager (HM), and vSwitch, abstracts physical compute resources to enable CSP to offer virtualized compute resources. And SDN which is composed of SDN Controller and SDN NVE, abstracts physical network resources to enable CSP to offer virtual network connectivity for VMs or containers. SDN's capabilities include rapid elasticity, resource pooling and on-demand self-service of network.

IOPT between SDN and hypervisor based computing virtualization is to verify that SDN can interact with hypervisor based computing virtualization as expected. The user could initiates requests related to the VM life cycle, including VM creation, startup, shutdown, deletion, migration, adding vNIC or deleting vNIC, and so on. The IOPT between SDN and hypervisor based computing virtualization can ensure that the requests receive corresponding interaction results. Meanwhile, virtualized infrastructure manager (VIM) including Compute management entity (CME) and Network management entity (NME) also play indispensable role in the IOPT between SDN and hypervisor based computing virtualization.

NOTE - The SDN mentioned in this Recommendation refers to the SDN in the cloud computing data center, excluding the WAN, the Access Network and other scenarios.

6.1. Target areas for IOPT between SDN and hypervisor based computing virtualization

According to the way SDN and hypervisor based computing virtualization works, there are 10 target areas that need to be considered in SDN and hypervisor based computing virtualization IOPT as follows:

- Target area 1: "**CME-NME**", dealing with interaction between compute management entity and network management entity;
- Target area 2:" **SDNC-NME**", dealing with interaction between SDN Controller and network management entity;
- Target area 3: "**CME-HM**", dealing with interaction between compute management entity and hypervisor manager;
- Target area 4: "**SDNC-VM**", dealing with interaction between SDN Controller and VM;
- Target area 5: "**NVE-vSwitch**", dealing with interaction between virtual switch and SDN NVE;
- Target area 6: "**NVE – VM**", dealing with interaction between SDN NVE and VM;

- Target area 7: "**SDNC-vSwitch**", dealing with interaction between SDN Controller and virtual switch;
- Target area 8: "**SDNC-NIC**", dealing with interaction between SDN Controller and intelligent NIC;
- Target area 9: "**NVE-Hypervisor**", dealing with interaction between SDN NVE and hypervisor;
- Target area 10: "**SDNC-ToR**", dealing with interaction between SDN Controller and ToR.

Meanwhile, different implementation scenarios of SDN NVE will lead to different target areas that need to be considered in SDN and hypervisor based computing virtualization IOPT. Referring to the form of NVE, four scenarios can be respectively named as virtual machine (VM) NVE, vSwitch NVE, intelligent Network Interface Card (NIC) NVE and ToR NVE.

- Scenario 1: "**VM NVE**", deploying the NVE as a VM on hypervisor based computing virtualization.
- Scenario 2: "**vSwitch NVE**", deploying the NVE in the form of vSwitch on the hypervisor based computing virtualization, which merge the original functions of vSwitch.
- Scenario 3: "**intelligent NIC NVE**", realizing overlay encapsulation/decapsulation packet forwarding services of NVE by the physical network card of the host where the hypervisor based computing virtualization is located.
- Scenario 4: "**ToR NVE**", realizing the function of NVE on ToR corresponding to the host where the hypervisor based computing virtualization is located.

Chapters 6.2 to 6.5 describe the four scenarios and their corresponding IOPT target areas respectively.

6.2. Scenario 1: "VM NVE"

As shown in the figure 6-1, NVE is deployed in hypervisor based computing virtualization in the form of a VM, and vSwitch directs the traffic of VMs to NVE to realize overlay encapsulation/decapsulation packet forwarding services. The interaction between SDN and hypervisor based computing virtualization are as following:

- **CME:** deploy event processing module and its virtualization event interface. The CME responds to the VM life cycle related requests and sends the VM life cycle related requests to the HM (which is the virtualization management node). Through virtualization event interface of the virtualization event processing module, the CME retrieves the VM life cycle related operational information reported by HM.
- **HM**: call the hypervisor to perform the user requested VM life cycle related operations on the specific physical host. After the operation, HM calls the virtualization event interface of the virtualization event processing module, and report the VM life cycle related operation results and corresponding parameter information to CME.
- **SDNC:** subscribes event of the VM life cycle related operation information from the virtualization event processing module of CME and updates the network configuration of the SDN NVE according to the operation information.

In scenario 1, the IOPT between SDN and hypervisor based computing virtualization involve target area 1, 2, 3, 4 and 5.

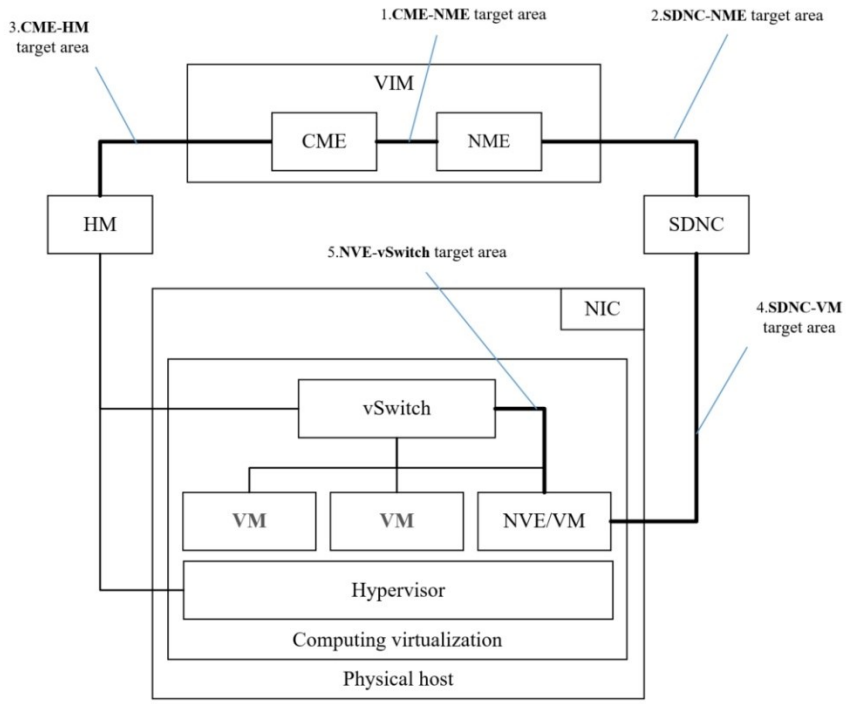

Figure 6-1 –Scenario 1: "virtual machine NVE"

6.3. Scenario 2: "vSwitch NVE"

As shown in the figure 6-2, NVE is embedded into vSwitch to achieve services of overlay encapsulation/decapsulation packet forwarding. The interaction between SDN and hypervisor based computing virtualization are as following:

- **vSwitch NVE:** vSwitch NVE interacts with hypervisor for data forwarding, and provides network connection for VMs in or between hosts instead of vSwitch;
- **VIM:** including CME and NME, standardize the interoperability between SDN and hypervisor based computing virtualization in different VM lifecycle management services. The CME responds to the VM life cycle related requests and sends the VM life cycle related requests to the HM. NME interacts with HM to perform NIC related operations, including vNIC addition/deletion and so on.
- **hypervisor**: receives instructions from HM to perform the VM life cycle related operations on the specific physical host.
- **SDNC:** subscribes event of the VM life cycle related operation information from CME, and interact with NME to receive NIC operation commands, call the corresponding vSwitch to complete vNIC addition/deletion operations, and synchronize network configuration information with NME.

In scenario 2, the IOPT between SDN and hypervisor based computing virtualization involve target area 1, 2, 3, 6, 7 and 9.

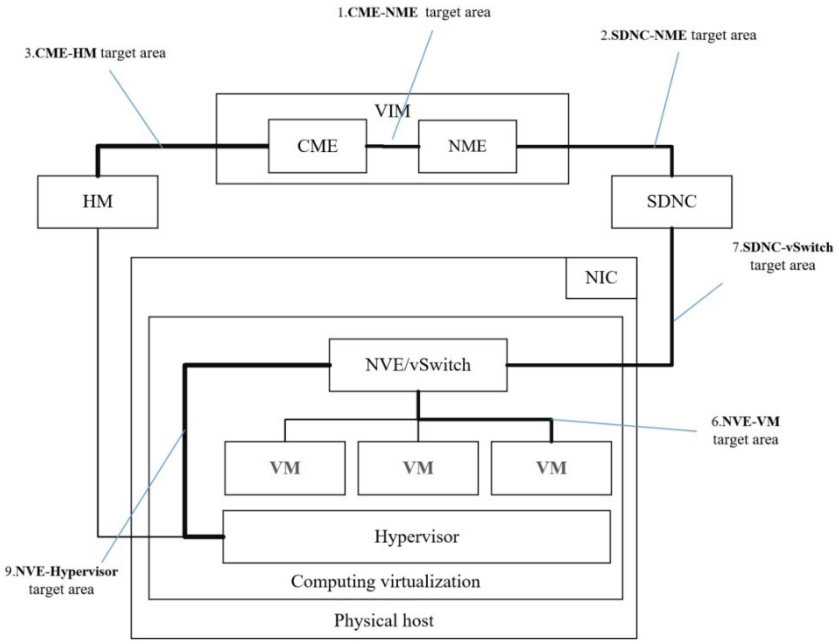

Figure 6-2 –Scenario 2: "vSwitch NVE"

6.4. Scenario 3: "intelligent NIC NVE"

As shown in the figure 6-3，SDN NVE is deployed in the intelligent NIC, the interaction between SDN NVE and hypervisor based computing virtualization are as following:

- **Intelligent NIC,** deploying SDN NVE and its corresponding kernel dependencies, runnable software library and standard interface library. Intelligent NIC NVE can perform the functions of the regular NIC, and receive the related operation commands of NIC from SDNC, synchronize network configuration information to hypervisor after completing those operations.
  NOTE - The standard interface library of SDN NVE includes standard interface and interface plug-ins, the standard interface defines the operation process and parameters between the vNIC of VM and the virtual port of NVE. The interface plug-ins inherit from the standard interfaces of SDN NVE, and realize the conversion between the virtual port of NVE and the standard interface, which are updated with the version of the virtual port of NVE.
- **Hypervisor**, include its corresponding kernel dependencies and runnable software library, receive instructions from HM to perform the VM life cycle related operations on the specific physical host, and synchronize network configuration information with intelligent NIC.
- **VIM:** same as Scenario 2, the CME responds to the VM life cycle related requests and sends the VM life cycle related requests to the HM. NME interacts with HM to perform NIC related operations, including vNIC addition/deletion and so on.

In scenario 3, the IOPT between SDN and hypervisor based computing virtualization involve target area 1, 2, 3, 6, 8 and 9.

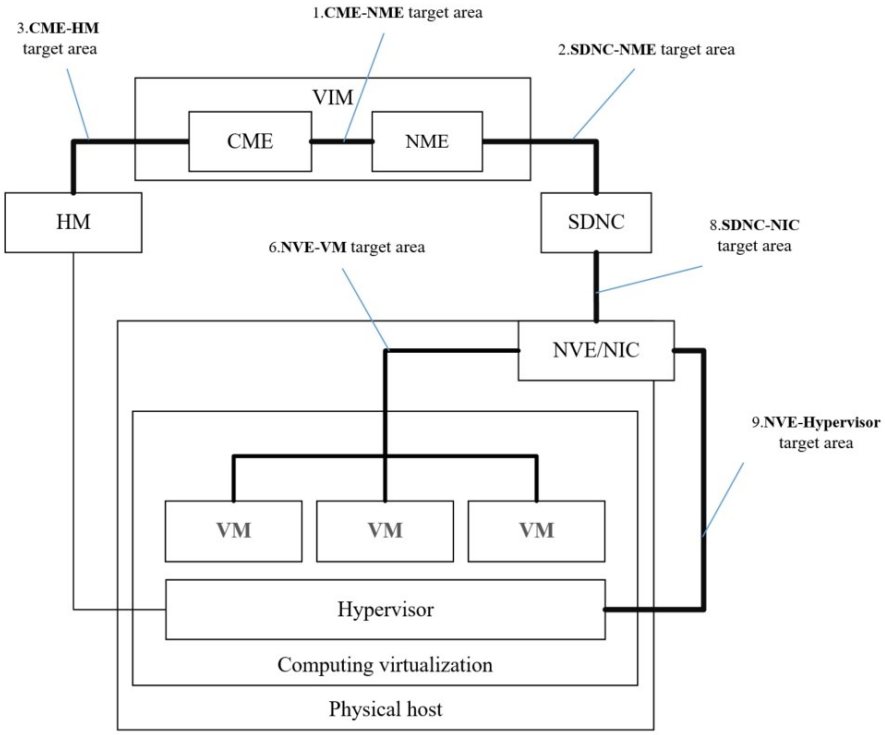

Figure 6-3 – Scenario 3: "intelligent NIC NVE"

6.5. Scenario 4: "ToR NVE"

As shown in the figure 6-4, ToR can also conduct the overlay encapsulation/decapsulation packet forwarding services which is the function of NVE, the interaction between SDN NVE and hypervisor based computing virtualization are as following:

The interaction between SDN and hypervisor based computing virtualization are as following:

- **VIM:** achieves the interoperability between SDN and hypervisor based computing virtualization by port binding and same as Scenario 2, The CME responds to the VM life cycle related requests and sends the VM life cycle related requests to the HM. NME interacts with HM to perform NIC related operations, including vNIC addition/deletion, and so on.
  NOTE - Port Binding schemes is a typical method to achieve the interoperability. Hierarchical port binding allows NME to collaborate with SDN and hypervisor to complete port binding operations, configure VLAN network on vSwitch, and configure corresponding tunnel network on TOR NVE.
- **ToR NVE:** receive data frames from hypervisor, delete VLAN Tags from the data frames based on the mapping between VLAN and tunnel ID, encapsulate the data frames as corresponding tunnel messages, and forwards data frames from virtual ports based on address information in data frames;
- **SDNC:** configures correct tunnel ID for ToR NVE which should be consistent with the values of API during VM lifecycle management services.

In scenario 4, the IOPT between SDN and hypervisor based computing virtualization involve target area 1, 2, 3, 9 and 10.

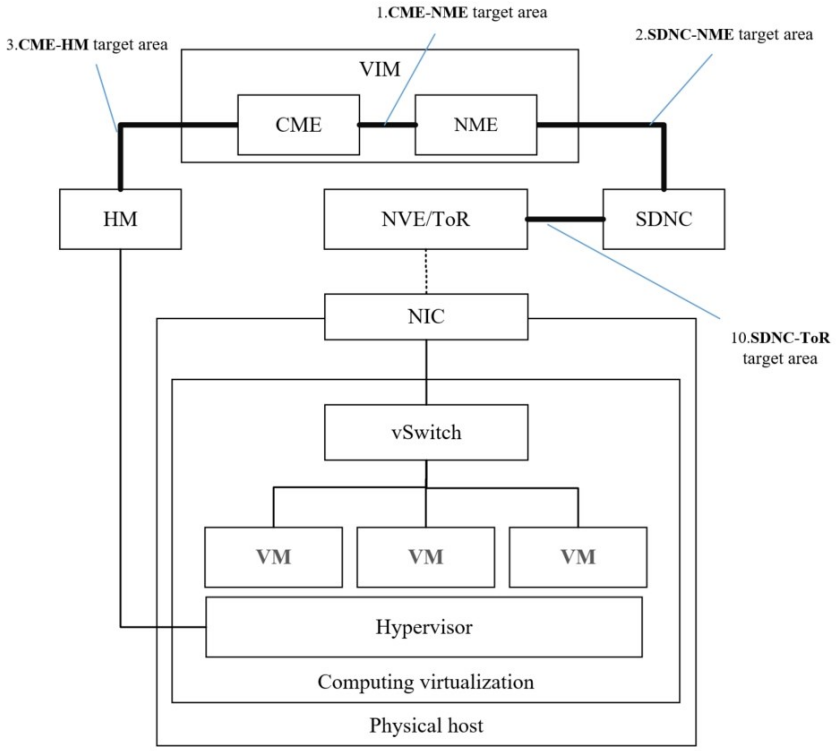

Figure 6-4 –Scenario 4: "ToR NVE"

7. IOPT requirements between SDN and hypervisor based computing virtualization

Clauses 7.1 to7.3 provide the IOPT requirements between SDN and hypervisor based computing virtualization in different categories.

7.1. IOPT requirements for interface compatibility aspect

The interface compatibility of the interaction provides consistent interaction operations and interface types. Verification of interface compatibility capabilities between SDN NVE and hypervisor based computing virtualization should be considered in target area 1, 2, 5, 6 and 9.

- **Permission setting**: It is recommended that computing virtualization open permission of hypervisor to deploy SDN NVE.
- **Version compatibility**: It is recommended that the operating system version of the hypervisor based computing virtualization are compatible with SDN NVE. And meanwhile it is recommended that the SDN be configured with the compute driver version and storage driver version which required by the VIM.

Interface compatibility is a prerequisite for IOPT between SDN and hypervisor based computing virtualization, so all test cases in appendix I.2 can be used to test these requirements.

7.2. IOPT requirements for VM lifecycle management aspect

The capabilities of VM lifecycle management include VM creation, deletion, migration, adding/deleting vNIC and so on. Verification of VM lifecycle management capabilities between SDN NVE and hypervisor based computing virtualization should be considered in target area 2, 3, 4, 7, 8, 9 and 10.

- **Data frame forwarding**: It is required that SDN NVE, VIM and hypervisor based computing virtualization provide data frame forwarding function, which forwards data frames from virtual ports based on address information in data frames. The corresponding test case can be found in appendix I.2.1.

- **Interface attribute extension**: It is required that SDN NVE and VIM process and fill in interface attributes of VM lifecycle management according to interface attribute conventions to ensure the consistency of interaction. The corresponding test case can be found in appendix I.2.2.

7.3. IOPT requirements for network connectivity aspect

The capabilities of network connectivity includes VLAN, transport protocol and QinQ. Verification of network connectivity capabilities should be considered in target area 1, 5, 6 and 9.

- **VLAN:** It is required that vSwitch/NVE provides VLAN function in order to create partitioned and isolated broadcast domain in a computer network at the data link layer. The corresponding test case can be found in appendix I.2.3.
- **Transport protocol**: It is required that SDN NVE and hypervisor based computing virtualization support the Hyper Text Transfer Protocol (HTTP) protocol. The corresponding test case can be found in appendix I.2.4.
- **QinQ**: It is recommended that vSwitch/NVE support encapsulating the private network VLAN tag in the public network VLAN so that the packet can be forwarded with two VLAN tags [b-IEEE Std 802.1q]. The corresponding test case can be found in Aappendix I.2.5.

8. Data model framework considered during SDN and hypervisor based computing virtualization IOPT

Clauses 8.1 to 8.2 provide the resource data models and corresponding operations that needs to be considered in IOPT between SDN and hypervisor based computing virtualization.

The elements listed in Table 8-1 and Table 8-2 could be used as the request and response parameters of the operations listed in Table 8-3 and Table 8-4.

8.1. Resource data model description

The data models includes VM and vNIC resource data model.

The data model description for VM resource is shown in Table 8-1.

Table 8-1 – Data model description for VM resource

Element	Type	Necessity	Description
VmName	String	Mandatory	Name of VM.
VmID	String	Mandatory	Unique identification of VM.
VmStatus	String	Mandatory	Status of VM, including running, stop, paused, migration, etc.
InstanceName	String	Optional	Name of VM registered on VIM.
InstanceID	String	Mandatory	Unique identification of VM registered on VIM.
HostName	String	Optional	Name of host where the VM is located.
HostID	String	Mandatory	Unique identification of host where the VM is located. SDN can refer to this value to match the corresponding hypervisor.
vNICList	Array	Mandatory	List of vNIC. More description can be found in vNIC resource data model description (Table 8-2).

The data model description for vNIC resource is shown in Table 8-2.

Table 8-2 – Data model description for vNIC resource

Element	Type	Necessity	Description
vNICName	String	Mandatory	Name of vNIC.
vNICID	String	Mandatory	Unique identification of vNIC.
vNICType	String	Mandatory	Type of vNIC, including normal, direct, bare-metal, direct-physical, virtio-forward and smart-nic, etc. The default value is normal.
VIFType	String	Optional	Type of VIF which corresponding to the vNIC. In "vSwitch NVE" Scenario, SDNC updates its attributes with the network configuration of the SDN NVE corresponding to VM lifecycle management.
PortGroupName	String	Optional	Name of port group.
PortGroupID	String	Mandatory	Unique identification of port group.
PortName	String	Optional	Name of port.
PortID	String	Mandatory	Unique identification of port.
NetworkName	String	Optional	Name of network.
NetworkID	String	Mandatory	Unique identification of network.
MACID	String	Mandatory	MAC address.
PortGroupVLANID	Integer	Mandatory	VLAN ID of port group.
vNICIP	String	Mandatory	IP address of vNIC.

8.2. Operations of the resource data models

Accordingly, the operations of resource data models include VM and vNIC resource data models related operations.

Table 8-3 provides the operations of the VM resource data model.

Table 8-3 – Operations of the VM resource data model

No	Operation	HTTP Action	Description
1	List VMs	GET	The information of all the VMs is listed.
2	Create VM	POST	One VM is created.
3	Shutdown VM	PUT	The specific VM is shutdown.
4	Startup VM	PUT	The specific VM is started.
5	Delete VM	DELETE	The specific VM is deleted.
6	Migrate VM	PUT	The specific VM is migrated.
7	Suspend VM	PUT	The specific VM is suspended.
8	Snapshot VM	PUT	The status of the specific VM is preserved as a snapshot.

Table 8-4 provides the operations of the vNIC resource data model.

Table 8-4 – Operations of the vNIC resource data model

No	Operation	HTTP Action	Description
1	Add vNIC	POST	The specific vNIC of VM is added in specific port of

			vSwitch/NVE.
2	Delete vNIC	DELETE	The specific vNIC of VM is removed from specific port of vSwitch/NVE.

Appendix I. Test cases of IOPT between SDN and hypervisor based computing virtualization

(This appendix does not form an integral part of this Recommendation.)

This appendix provides test cases for IOPT between SDN and hypervisor based computing virtualization.

I.1 Test case template

Table I.1 provides a test case template to describe IOPT between SDN and hypervisor based computing virtualization. The test case template is designed with reference to relevant technical specifications. As shown in the table, an interoperability test case consists of test purpose, reference, test procedures, expected results and test verdict.

– The test purpose specifies which test case to verify.

– The reference of the test case provides a list of references to the base specification clause(s), use case(s), requirement(s), etc. They are either used in the test or define the functionality being tested.

– The test sequences provide the stimuli required to perform the test. A stimulus corresponds to an event, it triggers a specific action on the object under test. There is no need to provide a result for a stimulus step.

– Expected results consist of check steps and are used to review whether the test phenomena of the test object are as expected. A result must be provided for every check step. If the object under test behaves as described in the description of the check step, the result should be recorded as OK, otherwise the result should be recorded as failed.

– The notes column can record any special conditions that occurred during the test which should be valued before test verdict.

– For every test case, test the verdict should be provided based on the result(s) of the check step(s) to indicate whether the test is passed, failed or partly passed.

Table I.1 – Test case template

test description of interoperability between SDN and hypervisor based computing virtualization				
Test purpose		A concise summary of the test reflecting its purpose and allowing readers to easily distinguish this test from any other test in the document.		
Reference		List of references to the base specification clause(s), requirement(s), etc. They are either used in the test or define the functionality being tested.		
Test procedures	Step	Type	Description	
	1	Stimulus	A stimulus corresponds to an event, it triggers a specific action on the object under test.	
Expected results	Step	Type	Description	Result
	1	Check	A check is used to review whether the test phenomena of the test object are as expected. A result must be provided for every check step. If the object under test behaves as described in the description of the check step, the result should be recorded as OK, otherwise the result should be recorded as failed.	A result must be provided for every check step.
Notes				

Table I.1 – Test case template

test description of interoperability between SDN and hypervisor based computing virtualization	
Test verdict	If all checks described in the test case are successful, it is deemed successfully terminated. If at least one check is successful but at least one check is failed, the test is deemed partially passed. If no checks are successful, the test is deemed failed.

I.2 Test cases for IOPT between SDN and hypervisor based computing virtualization

I.2.1 Test case: Data frame forwarding

Table I.2 shows the test case for data frame forwarding.

Table I.2 – Test case: Data frame forwarding

Data frame forwarding test description				
Test purpose		To verify that the SDN NVE, VIM and hypervisor based computing virtualization provide the data frame forwarding function, which forwards data frames from virtual ports base on address information in data frames.		
Reference		clause 7.2		
Test procedures	Step	Type	Description	
	1	Stimulus	Login to the configuration interface of VIM.	
	2	Stimulus	Create VM1 and VM2 in different hosts under the same subnet, configure VM1 with IP1 and VM2 with IP2, which should produce expected result1.	
	3	Stimulus	From VM1 ping IP2 and from VM2 ping IP1, which should produce expected result2.	
	4	Stimulus	View the configuration information of the VMs on the SDN controller and VIM respectively, which should produce expected result3.	
	5	Stimulus	Perform the operations in table 8-3 and table 8-4 of chapter 8 on VMs, including shutdown, startup, deletion, suspension, adding/deleting vNIC etc.	
	6	Stimulus	View the VM's status on the SDN controller and VIM respectively, which should produce expected result4.	
Expected results	Step	Type	Description	Result
	1	Check	The VMs are successfully created.	
	2	Check	VM1 and VM2 can ping each other.	
	3	Check	The configuration information of VMs seen on the SDN controller and VIM is consistent, including VmName、VmID、InstanceID、HostID、vNICList and other parameters in table 8-3 and table 8-4.	
	4	Check	The VM's status seen on the SDN controller and VIM is consistent, including stop, running, paused, etc.	
Notes				

Table I.2 – Test case: Data frame forwarding

Data frame forwarding test description	
Test verdict	If all checks described in the test case are successful, it is deemed successfully terminated. If at least one check is successful but at least one check is failed, the test is deemed partially passed. If no checks are successful, the test is deemed failed.

I.2.2 Test case: Interface attribute extension

Table I.3 shows the test case for interface attribute extension.

Table I.3 –Test case: Interface attribute extension

Interface attribute extension test description				
Test purpose		To verify that SDN NVE and VIM process and fill in interface attributes corresponding to VM lifecycle management, which can ensure the consistency of interaction within special interface attribute conventions.		
Reference		clause 7.2		
Test procedures	step	Type	Description	
	1	Stimulus	Login to the configuration interface of VIM.	
	2	Stimulus	Create network1 and create port1 in network1, which should produce expected result1.	
	3	Stimulus	Create VM1 under network1, attach port1 to VM1, which should produce expected result2 and result3.	
	4	Stimulus	View the network configuration information of VM1 on the SDN controller and VIM respectively, which should produce expected result4.	
Expected results	step	Type	Description	Result
	1	Check	The network and port are successfully created.	
	2	Check	The VM is successfully created.	
	3	Check	The port is successfully attached.	
	4	Check	The network configuration information of VMs seen on the SDN controller and VIM is consistent, including vNICType、PortGroupID, and other parameters in table 8-2.	
Notes		This test case can be considered together with VLAN, transport protocol test cases, see clauses I.2.3 and I.2.4.		
Test verdict		If all checks described in the test case are successful, it is deemed successfully terminated. If at least one check is successful but at least one check is failed, the test is deemed partially passed. If no checks are successful, the test is deemed failed.		

I.2.3 Test case: VLAN

Table I.4 shows the test case for VLAN.

Table I.4 –Test case: VLAN

VLAN test description

Test purpose		To verify that vSwitch/NVE provides VLAN function in order to create partitioned and isolated broadcast domain in a computer network at the data link layer.		
Reference		clause 7.3		
Test procedures	step	Type	Description	
	1	Stimulus	Login to the configuration interface of VIM.	
	2	Stimulus	Create VM1 and configure VM1 with IP1, which should produce expected result1.	
	3	Stimulus	View the network configuration of VM1 on the SDN controller and VIM respectively, which should produce expected result2.	
Expected results	step	Type	Description	Result
	1	Check	The VM is successfully created.	
	2	Check	The PortGroupVLANID of VM1 seen on the SDN controller and VIM is consistent.	
Notes				
Test verdict		If all checks described in the test case are successful, it is deemed successfully terminated. If at least one check is successful but at least one check is failed, the test is deemed partially passed. If no checks are successful, the test is deemed failed.		

I.2.4 Test case: Transport protocol

Table **I**.5 shows the test case for transport protocol.

Table I.5 –Test case: Transport protocol

Transport protocol test description				
Test purpose		To verify that SDN NVE and hypervisor based computing virtualization support the Hyper Text Transfer Protocol (HTTP) protocol, which can be used for operations of the resource data models registration / deregistration and notification.		
Reference		clause 7.3		
Test procedures	step	Type	Description	
	1	Stimulus	Login to the testing tool which can perform HTTP message testing.	
	2	Stimulus	Send HTTP messages to hypervisor based computing virtualization refer to the Data model description and operations in clause8, which should produce expected result1.	
	3	Stimulus	View the VM's status on the SDN controller and VIM respectively, which should produce expected result2.	
Expected results	step	Type	Description	Result
	1	Check	The testing tool can receive the corresponding correct response message	
	2	Check	The configuration information of VMs seen on the SDN controller and VIM is consistent, including VmName、VmID、InstanceID、HostID、vNICList, etc.	
Notes				

Test verdict	If all checks described in the test case are successful, it is deemed successfully terminated. If at least one check is successful but at least one check is failed, the test is deemed partially passed. If no checks are successful, the test is deemed failed.

I.2.5 Test case: QinQ

Table I.6 shows the test case for QinQ.

Table I.6 –Test case: QinQ

QinQ test description				
Test purpose		To verify that vSwitch/NVE support encapsulating the private network VLAN tag in the public network VLAN so that the packet can be forwarded with two VLAN tags [b-IEEE Std 802.1q].		
Reference		clause 7.3		
Test procedures	Step	Type	Description	
	1	Stimulus	Login to the configuration interface of vSwitch.	
	2	Stimulus	Create VLAN1, add virtual port1 in VLAN1. Configure virtual port1 to support QinQ.	
	3	Stimulus	Create virtual port2 inVLAN1 on another host.	
	4	Stimulus	Login to the configuration interface of VIM.	
	5	Stimulus	Create VM1 and connect it to virtual port1，create VM2 and connect it to virtual port2, which should produce expected result1.	
	6	Stimulus	From VM1, using virtual instrumentation, send packet encapsulated with VLAN2 header to VM2. Capture packet between the hosts, which should produce expected result2.	
Expected results	step	Type	Description	Result
	1	Check	The VMs are successfully created.	
	2	Check	The captured packet has 2 VLAN tags. The inner tag VLAN2 is encapsulated in the outer tag VLAN1.	
Notes				
Test verdict		If all checks described in the test case are successful, it is deemed successfully terminated. If at least one check is successful but at least one check is failed, the test is deemed partially passed. If no checks are successful, the test is deemed failed.		

Bibliography

[b- ITU-T Y.101] Recommendation ITU-T Y.101 (2000), *Global Information Infrastructure terminology: Terms and definitions.*

[b-ITU-T Y.4500.15] Recommendation ITU-T Y.4500.15/Q.3955 (2018), o*neM2M – Testing framework.*

[b-IEEE Std 802.1q] IEEE Std 802.1q (2022), *802.1Q - Virtual LANs*.

[b-IETF RFC7365] IETF RFC7365(2014), *Framework for Data Center (DC) Network Virtualization*.

_____