I n t e r n a t i o n a l   T e l e c o m m u n i c a t i o n   U n i o n

ITU-T

TELECOMMUNICATION
STANDARDIZATION SECTOR
OF ITU

Q.3716

(01/2018)

SERIES Q: SWITCHING AND SIGNALLING, AND ASSOCIATED MEASUREMENTS AND TESTS

Signalling requirements and protocols for SDN – Resource control protocols

Signalling requirements for mapping between physical and virtual networks

Recommendation ITU-T Q.3716

ITU-T

International Telecommunication Union

ITU-T Q-SERIES RECOMMENDATIONS

SWITCHING AND SIGNALLING, AND ASSOCIATED MEASUREMENTS AND TESTS

SIGNALLING IN THE INTERNATIONAL MANUAL SERVICE	Q.1–Q.3
INTERNATIONAL AUTOMATIC AND SEMI-AUTOMATIC WORKING	Q.4–Q.59
FUNCTIONS AND INFORMATION FLOWS FOR SERVICES IN THE ISDN	Q.60–Q.99
CLAUSES APPLICABLE TO ITU-T STANDARD SYSTEMS	Q.100–Q.119
SPECIFICATIONS OF SIGNALLING SYSTEMS No. 4, 5, 6, R1 AND R2	Q.120–Q.499
DIGITAL EXCHANGES	Q.500–Q.599
INTERWORKING OF SIGNALLING SYSTEMS	Q.600–Q.699
SPECIFICATIONS OF SIGNALLING SYSTEM No. 7	Q.700–Q.799
Q3 INTERFACE	Q.800–Q.849
DIGITAL SUBSCRIBER SIGNALLING SYSTEM No. 1	Q.850–Q.999
PUBLIC LAND MOBILE NETWORK	Q.1000–Q.1099
INTERWORKING WITH SATELLITE MOBILE SYSTEMS	Q.1100–Q.1199
INTELLIGENT NETWORK	Q.1200–Q.1699
SIGNALLING REQUIREMENTS AND PROTOCOLS FOR IMT-2000	Q.1700–Q.1799
SPECIFICATIONS OF SIGNALLING RELATED TO BEARER INDEPENDENT CALL CONTROL (BICC)	Q.1900–Q.1999
BROADBAND ISDN	Q.2000–Q.2999
SIGNALLING REQUIREMENTS AND PROTOCOLS FOR THE NGN	Q.3000–Q.3709
SIGNALLING REQUIREMENTS AND PROTOCOLS FOR SDN	Q.3710–Q.3899
Resource control protocols	**Q.3710–Q.3739**
Network signalling and signalling requirements for services	Q.3740–Q.3779
TESTING SPECIFICATIONS	Q.3900–Q.4099

For further details, please refer to the list of ITU-T Recommendations.

Recommendation ITU-T Q.3716

Signalling requirements for mapping between physical and virtual networks

Summary

Recommendation ITU-T Q.3716 specifies signalling requirements for mapping between software-defined networking (SDN) based physical underlay networks and virtual overlay networks, by architecturally adding dedicated functional components and the corresponding interfaces in the software-defined networking framework.

History

Edition	Recommendation	Approval	Study Group	Unique ID*
1.0	ITU-T Q.3716	2018-01-13	11	11.1002/1000/13485

Keywords

Mapping between overlay and underlay networks, SDN, signalling requirements.

* To access the Recommendation, type the URL http://handle.itu.int/ in the address field of your web browser, followed by the Recommendation's unique ID. For example, http://handle.itu.int/11.1002/1000/11830-en.

FOREWORD

The International Telecommunication Union (ITU) is the United Nations specialized agency in the field of telecommunications, information and communication technologies (ICTs). The ITU Telecommunication Standardization Sector (ITU-T) is a permanent organ of ITU. ITU-T is responsible for studying technical, operating and tariff questions and issuing Recommendations on them with a view to standardizing telecommunications on a worldwide basis.

The World Telecommunication Standardization Assembly (WTSA), which meets every four years, establishes the topics for study by the ITU-T study groups which, in turn, produce Recommendations on these topics.

The approval of ITU-T Recommendations is covered by the procedure laid down in WTSA Resolution 1.

In some areas of information technology which fall within ITU-T's purview, the necessary standards are prepared on a collaborative basis with ISO and IEC.

NOTE

In this Recommendation, the expression "Administration" is used for conciseness to indicate both a telecommunication administration and a recognized operating agency.

Compliance with this Recommendation is voluntary. However, the Recommendation may contain certain mandatory provisions (to ensure, e.g., interoperability or applicability) and compliance with the Recommendation is achieved when all of these mandatory provisions are met. The words "shall" or some other obligatory language such as "must" and the negative equivalents are used to express requirements. The use of such words does not suggest that compliance with the Recommendation is required of any party.

INTELLECTUAL PROPERTY RIGHTS

ITU draws attention to the possibility that the practice or implementation of this Recommendation may involve the use of a claimed Intellectual Property Right. ITU takes no position concerning the evidence, validity or applicability of claimed Intellectual Property Rights, whether asserted by ITU members or others outside of the Recommendation development process.

As of the date of approval of this Recommendation, ITU had not received notice of intellectual property, protected by patents, which may be required to implement this Recommendation. However, implementers are cautioned that this may not represent the latest information and are therefore strongly urged to consult the TSB patent database at http://www.itu.int/ITU-T/ipr/.

Table of Contents

Recommendation ITU-T Q.3716

Signalling requirements for mapping between physical and virtual networks

1 Scope

The scope of this Recommendation consists of:

– Overview of mapping between physical and virtual networks (PVMapping);
– Data model description for PVMapping;
– Interfaces requirements for PVMapping;
– Signalling requirements for PVMapping.

The appendices to this Recommendation also provide examples of:

– SDN-based slice abstraction;
– Restful API for PVMapping;
– Typical scenarios and procedures.

2 References

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.3011] Recommendation ITU-T Y.3011 (2012), *Framework of network virtualization for future networks.*

[ITU-T Y.3100] Recommendation ITU-T Y.3100 (2017), *Terms and definitions for IMT-2020 network.*

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

[ITU-T Y.3302] Recommendation ITU-T Y.3302 (2017), *Functional architecture of software-defined networking.*

[ITU-T Y.3500] Recommendation ITU-T Y.3500 (2014), *Information technology – Cloud computing – Overview and vocabulary.*

3 Definitions

3.1 Terms defined elsewhere

This Recommendation uses the following terms defined elsewhere:

3.1.1 network slice [ITU-T Y.3100]: A logical network that provides specific network capabilities and network characteristics.

NOTE 1 – Network slices enable the creation of customized networks to provide flexible solutions for different market scenarios which have diverse requirements, with respect to functionalities, performance and resource allocation.

NOTE 2 – A network slice may have the ability to expose its capabilities.

NOTE 3 – The behaviour of a network slice is realized via network slice instance(s).

3.1.2 network virtualization [ITU-T Y.3011]: A technology that enables the creation of logically isolated network partitions over shared physical networks so that heterogeneous collections of multiple virtual networks can simultaneously coexist over the shared networks. This includes the aggregation of multiple resources in a provider and appearing as a single resource.

3.1.3 software-defined networking [ITU-T Y.3300]: A set of techniques that enables to directly program, orchestrate, control and manage network resources, which facilitates the design, delivery and operation of network services in a dynamic and scalable manner.

3.1.4 tenant [ITU-T Y.3500**]**: One or more cloud service users sharing access to a set of physical and virtual resources.

3.2 Terms defined in this Recommendation

None.

4 Abbreviations and acronyms

This Recommendation uses the following abbreviations and acronyms:

API	Application Programming Interface
APP	Application
C-Core	Cloud Core
C-RAN	Cloud Radio Access Network
MANO	Management and Orchestration
PVMapping	Mapping between Physical and Virtual networks
RAN	Radio Access Network
SCTP	Stream Control Transmission Protocol
SDN	Software-Defined Networking
TCP	Transmission Control Protocol
TLS	Transport Layer Security
UDP	User Datagram Protocol
VM	Virtual Machine
VPC	Virtual Private Cloud
VPN	Virtual Private Network
VXLAN	Virtual extensible Local Area Network
XML	extensible Markup Language

5 Conventions

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

In the body of this Recommendation and its appendices, the words shall, shall not, should and may sometimes appear, in which case they are to be interpreted, respectively as, is required to, is prohibited from, is recommended and can optionally. The appearance of such phrases or keywords in an appendix or in material explicitly marked as informative are to be interpreted as having no normative intent.

{A}: indicates that the parameter A is mandatory;

[B]: indicates that the parameter B is optional;

*: indicates that the parameter may be multiple items.

6 Overview of PVMapping

6.1 Introduction

Network virtualization facilities multiple tenants to share the same physical network by decoupling the virtual network from the physical network, usually using tunnelling overlay technology.

NOTE – The well-known tunnelling overlay technologies include virtual private network (VPN) [b-IETF RFC 4364], virtual extensible local area network (VXLAN) [b-IETF RFC 7348], etc.

The virtual overlay networks are the logical abstractions of the physical underlay network. Usually, the physical underlay network is not aware of the packets carried by the virtual overlay networks.

The manipulation of the virtualized overlay network may affect the configuration of physical underlay networks. From the tenants' perspective, knowledge of the underlay network utilization is needed in order to obtain enough connection resources. From the operator's perspective, knowledge of the underlay network utilization is also needed to realize quick troubleshooting and protection switching when underlay physical links fail. Therefore, the mapping mechanism between the physical underlay network and its virtual overlay networks is required.

In many real world scenarios, with the maturity of SDN technology, network virtualization is always applied together with SDN in order to allow the operators to lower the management complexity of their networks and better meet their tenants' customized needs. Therefore, it is required to develop the mapping mechanism for the SDN-based underlay and overlay networks.

6.2 Functional architecture of PVMapping

The functional architecture of PVMapping is depicted to perform the SDN based underlay and overlay mapping functionality by architecturally adding dedicated components and the corresponding interfaces based on the SDN framework defined in [ITU-T Y.3302], as shown in Figure 6-1. The dotted block performs the mapping functionality, in which vNetwork Management functional components and global mapping functional components are defined.

Figure 6-1 shows a functional architecture of PVMapping. PVMapping is not restricted to the implementation of concrete functional components shown in Figure 6-1. It is feasible for a controller or other separate functional components to implement this functionality.

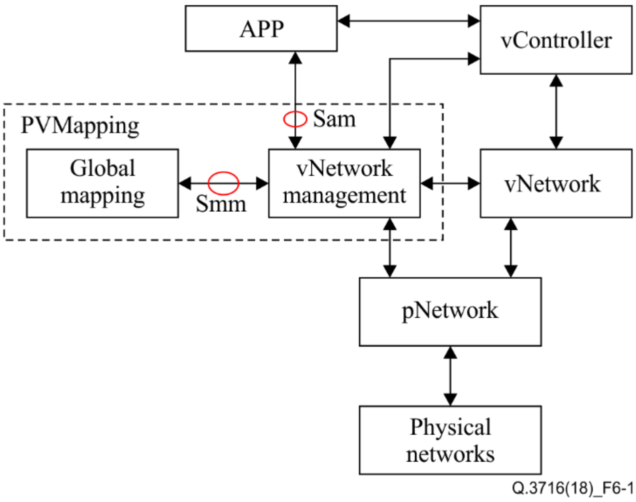

Figure 6-1 – Functional architecture of PVMapping

6.2.1 Global mapping functional components

Global mapping functional components enforce the mapping between virtual overlay networks and physical underlay networks and can be realized by proprietary algorithms.

NOTE – The mapping algorithms do not restrict the implementation technology of virtualization.

6.2.2 vNetwork management functional components

vNetwork management functional components enforce the validation check, creation, deletion, query, activation, shutoff and release of virtual networks based on the tenant's demands. The mapping pairs of physical network (pNetwork) and virtual network (vNetwork) calculated by global mapping functional components are also stored in the database of vNetwork management functional components, which can be accessed by vNetwork functional components and pNetwork functional components.

6.2.3 Other functional components

Apart from global mapping functional components and vNetwork management functional components, the following functional components are depicted in order to obtain a better understanding of complete service procedures:

– pNetwork functional components

  pNetwork functional components provide representatives of physical network elements, such as physical nodes, ports, switches, links, etc. and activate the virtual network in the physical networks.

– vNetwork functional components

  vNetwork functional components provide virtual network elements, such as virtual nodes, ports, switches, links, etc. Each tenant is represented by one virtual network and different tenants can only discover their own virtual network. All virtual network elements are mapped to at least one physical element and can be enabled, disabled, modified or reorganized at run time.

– vController functional components

  vController functional components control the virtual network dynamically based on the instruction of application (APP) functional components and resource allocation of vNetwork management functional components.

7 Data model description for PVMapping

7.1 Virtual network

7.1.1 Virtual network

The data model description for virtual networks is specified in Table 7-1.

Table 7-1 – Data model description for virtual networks

Element	Necessity	Description
VNID	Mandatory	Unique identification of the virtual network.
SDNController	Optional	Controller information of the virtual network.
NodeList	Mandatory	Nodes information of the virtual network. Clause 7.1.2 provides a virtual node data model description.
LinkList	Mandatory	Link information of the virtual network. Links are selected via nodes and interfaces. Clause 7.1.4 provides a virtual link data model description.
ForwardingPolicy	Optional	Forwarding policy running in the virtual network, attached when the virtual network is created.
State	Mandatory	State of the virtual network, including initiation, creation, running, stop, migration, etc.

7.1.2 Virtual node

The data model description for a virtual node, which is referred to as NodeList in the virtual network data model of Table 7-1, is specified in Table 7-2.

Table 7-2 – Data model description for a virtual node

Element	Necessity	Description
VirtualNodeID	Mandatory	Unique identification of virtual node.
VirtualNodeCapacity	Optional	Capacity of virtual node, including CPU, memory, forwarding table, etc.
DeviceType	Mandatory	Device type of virtual node, which is related with corresponding physical network device type, including vRouter, vSwitch, etc. Different device requires their respective forwarding information.
InterfaceList	Optional	The list of interfaces. Clause 7.1.3 provides a virtual interface data model description.
MappingRestriction	Optional	The constraints for physical network mapping, e.g., geographical information or node lists.

7.1.3 Virtual interface

The data model description for the virtual interface, which is referred to as InterfaceList in the virtual node data model of Table 7-2, is specified in Table 7-3.

Table 7-3 – Data model description for a virtual interface

Element	Necessity	Description
InterfaceID	Mandatory	Interface ID of the virtual network.
Rate	Optional	Interface rate of the virtual network.
AccessInfo	Optional	Interface information description, including IP address, mask, etc.
StatisticsInfo	Optional	Statistics information of the interface, which can be used for virtual network state monitoring.

7.1.4 Virtual link

The data model description for virtual link, which is referred to as LinkList in the virtual network data model of Table 7-1, is specified in Table 7-4.

Table 7-4 – Data model description for a virtual link

Element	Necessity	Description
Source	Mandatory	Source of virtual link, including virtual node and interface.
Destination	Mandatory	Destination of virtual link, including virtual node and interface.
Bandwidth	Optional	Link bandwidth.
Delay	Optional	Link delay.
PhysicalPath	Optional	Physical path information mapped to the corresponding virtual link.

7.2 Physical network

7.2.1 Physical network

The data model description for the physical network is specified in Table 7-5.

Table 7-5 – Data model description for physical network

Element	Necessity	Description
PhysicalNetworkID	Mandatory	Unique identification of physical network.
Devices	Mandatory	Device collection of current physical network topology.
Links	Mandatory	Directed link collection of current physical network topology.

7.2.2 Physical device

The data model description for a physical device, which is referred to as Devices in the physical network data model of Table 7-5, is specified in Table 7-6.

Table 7-6 – Data model description for a physical device

Element	Necessity	Description
DeviceID	Mandatory	Unique identification of physical device.
Type	Mandatory	Type of physical device, including router, switch, firewall, load balancer, etc.
Latency	Mandatory	Forwarding latency of physical device.
Availability	Mandatory	Availability of physical device.
Manufacturer	Optional	Manufacture of physical device.
Hardware	Optional	Hardware version of physical device.
Software	Optional	Software version of physical device.

7.2.3 Physical link

The data model description for physical link, which is regarded as Links in physical network data model of Table 7-5, is specified in Table 7-7.

Table 7-7 – Data model description for a physical link

Element	Necessity	Description
Source	Mandatory	Source of physical link, including physical device ID and interface.
Destination	Mandatory	Destination of physical link, including physical device ID and interface.
Type	Mandatory	Type of physical link, including DIRECT, INDIRECT, OPTICAL, TUNNEL, etc.
State	Mandatory	State of physical link, including ACTIVE and INACTIVE.
Latency	Mandatory	Forwarding latency of physical link.

8 Interfaces requirements for PVMapping

PVMapping functionality follows the functional architecture as shown in Figure 6-1. The interfaces presented in Figure 6-1 (Sam, Smm) among newly defined functional components in the dotted block and existing functional components in the SDN framework [ITU-T Y.3302] are used to exchange information to perform the PVMapping functionality. The definition and requirements of other interfaces presented in Figure 6-1 are out of the scope of this Recommendation.

8.1 Sam

This interface permits the interaction between APP functional components and vNetwork management functional components. vNetwork management functional components receive the virtual network creation request from the tenant, as described using the virtual network data model (see clause 7.1.1), and perform the validation check, including primarily the parameters' validation check of data model. If the validation fails, vNetwork management functional components send the failure message to APP functional components.

8.2 Smm

This interface permits the interaction between vNetwork management functional components and global mapping functional components. If the validation of the virtual network creation request is successful, vNetwork management functional components request global mapping functional components to calculate the physical mapping of this virtual network and resource allocation. The mapping pair of virtual network and physical network is sent back to vNetwork management functional components from global mapping functional components if there is a successful calculation result. The mapping pair and the instantiated virtual network data model (XML file) are stored in the database of vNetwork management functional components.

9 Signalling requirements for PVMapping

9.1 Overview

The complete PVMapping function is performed by global mapping functional components and vNetwork management functional components, whose messages are exchanged over the interfaces Smm and Sam.

NOTE – No transport protocol for the signalling messages is specified here. No message content format is specified here either.

The signalling messages may be extensible markup language (XML)-based messages over (or carried by) transmission control protocol (TCP), user datagram protocol (UDP), stream control transmission protocol (SCTP), transport layer security (TLS), etc. All of the messages consist of the message header and the message body.

The message format is described in Figure 9-1.

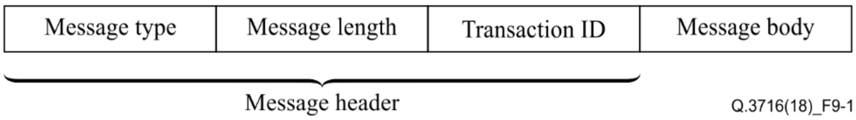

Figure 9-1 – Message composition

The message header field contains the following information:

– Message type: uniquely specifies the type of message;

– Message length: specifies the length of the message body;

– Message transaction ID: generated by the sender of the message. If there is a response message for the request message, the transaction IDs of the request and response messages are the same.

The message body field contains the message contents.

9.2 Signalling requirements of interface Sam

9.2.1 Virtual network creation message and response message

The virtual network creation message is defined as a VN-CR message.

The VN-CR message, indicated by the message type in the message header field, is sent by the APP functional components to vNetwork management functional components to request the creation of a virtual network.

Message format:

```
< VN-CR-Message > ::= < Message header >
               [ SDN-Controller-ID ]
               { Node-List }
               { Link-List }
               [ Forwarding-Policy ]
```

Meanings and explanations:

– `SDN-Controller-ID` is optional and specifies the identification of SDN controller if forwarding policy is required.

– `Node-List` uniquely specifies the nodes information of the virtual network. Each node in `Node-List` is described using the data model defined in clause 7.1.2.

– `Link-List` uniquely specifies the link information of the virtual network, which is described using the data model defined in clause 7.1.4.

– `Forwarding-Policy` is optional and specifies the forwarding policy running in the virtual network. `SDN-Controller-ID` and `Forwarding-Policy` are usually used together.

The response message to the VN-CR message is defined as a VN-CR-REP message.

The VN-CR-REP message, indicated by the message type in the message header field, is sent by vNetwork management functional components to APP functional components in response to the VN-CR message.

Message format:

```
< VN-CR-REP-Message > ::= < Message header >
              { VN-Creation-Result }
              [ VN-ID ]
```

Meanings and explanations:

– `VN-Creation-Result` specifies the creation result of virtual network. If both of the validation check and the mapping creation are successful, `VN-Creation-Result` is TRUE; if not, `VN-Creation-Result` is FALSE.

– `VN-ID` is optional and uniquely specifies the identification of virtual network if successfully created.

9.2.2 Virtual network activation message and response message

The virtual network activation message is defined as a VN-AC message.

The VN-AC message, indicated by the message type in the message header field, is sent by APP functional components to vNetwork management functional components to activate the already created virtual network.

Message format:

```
< VN-AC-Message > ::= < Message header >
              { VN-ID }
```

Meanings and explanations:

– `VN-ID` uniquely specifies the identification of a virtual network.

The response message to the VN-AC message is defined as a VN-AC-REP message.

The VN-AC-REP message, indicated by the message type in the message header field, is sent by vNetwork management functional components to APP functional components in response to the VN-AC message.

Message format:

```
< VN-AC-REP-Message > ::= < Message header >
              { VN-Activation-Result }
```

Meanings and explanations:

– VN-Activation-Result specifies the activation result of the virtual network. If the virtual network is activated successfully, VN-Activation-Result is TRUE; if not, VN-Activation-Result is FALSE.

9.2.3 Virtual network shutoff message and response message

The virtual network shutoff message is defined as a VN-SH message.

The VN-SH message, indicated by the message type in the message header field, is sent by APP functional components to vNetwork management functional components to shutoff the already activated virtual network.

Message format:

```
< VN-SH-Message > ::= < Message header >
              { VN-ID }
```

Meanings and explanations:

– VN-ID uniquely specifies the identification of virtual network.

The response message to the VN-SH message is defined as VN-SH-REP message.

The VN-SH-REP message, indicated by the message type in the message header field, is sent by vNetwork management functional components to APP functional components in response to the VN-SH message.

Message format:

```
< VN-SH-REP-Message > ::= < Message header >
              { VN-Shutoff-Result }
```

Meanings and explanations:

– VN-Shutoff-Result specifies the shutoff result of the virtual network. If the virtual network is shut off successfully, VN-Shutoff-Result is TRUE; if not, VN-Shutoff-Result is FALSE.

9.2.4 Virtual network release message and response message

The virtual network release message is defined as a VN-RE message.

The VN-RE message, indicated by the message type in the message header field, is sent by APP functional components to vNetwork management functional components to release the already created virtual network.

Message format:

```
< VN-RE-Message > ::= < Message header >
              { VN-ID }
```

Meanings and explanations:

– VN-ID specifies the identification of virtual network.

The response message to the VN-RE message is defined as a VN-RE-REP message.

The VN-RE-REP message, indicated by the message type in the message header field, is sent by vNetwork management functional components to APP functional components in response to the VN-RE message.

Message format:

```
< VN-RE-REP-Message > ::= < Message header >
              { VN-Release-Result }
```

Meanings and explanations:

– `VN-Release-Result` specifies the release result of the virtual network. If the virtual network is released successfully, `VN-Release-Result` is `TRUE`; if not, `VN-Release-Result` is `FALSE`.

9.2.5 Virtual network deletion message and response message

The virtual network deletion message is defined as a VN-DE message.

The VN-DE message, indicated by the message type in the message header field, is sent by APP functional components to vNetwork management functional components to delete the already created virtual network.

Message format:

```
< VN-DE-Message > ::= < Message header >
              { VN-ID }
```

Meanings and explanations:

– `VN-ID` uniquely specifies the identification of the virtual network.

The response message to the VN-DE message is defined as a VN-DE-REP message.

The VN-DE-REP message, indicated by the message type in the message header field, is sent by vNetwork management functional components to APP functional components in response to the VN-AC message.

Message format:

```
< VN-DE-REP-Message > ::= < Message header >
              { VN-Deletion-Result }
```

Meanings and explanations:

– `VN-Deletion-Result` specifies the deletion result of the virtual network. If the virtual network is deleted successfully, `VN-Deletion-Result` is `TRUE`; if not, `VN-Deletion-Result` is `FALSE`.

9.2.6 Query message and response message

9.2.6.1 Virtual network list query message and response message

The virtual network list query message is defined as a VN-LIST-QU message.

The VN-LIST-QU message, indicated by the message type in the message header field, is sent by APP functional components to vNetwork management functional components to query the list of already created virtual networks.

Message format:

```
< VN-LIST-QU-Message > ::= < Message header >
```

The response message to the VN-LIST-QU message is defined as a VN-LIST-QU-REP message.

The VN-LIST-QU-REP message, indicated by the message type in the message header field, is sent by vNetwork management functional components to APP functional components in response to the VN-LIST-QU message.

Message format:

```
< VN-LIST-QU-REP-Message > ::= < Message header >
              *[ VN-ID ]
```

Meanings and explanations:

– `VN-ID` is optional and uniquely specifies the identification of virtual network. There may be multiple virtual network ID. If there is no created virtual network in the database, there will be no virtual network ID as response.

9.2.6.2 Virtual network query message and response message

The virtual network list query message is defined as a VN-QU message.

The VN-QU message, indicated by the message type in the message header field, is sent by APP functional components to vNetwork management functional components to query the already created dedicated virtual network.

Message format:

```
< VN-QU-Message > ::= < Message header >
              { VN-ID }
```

Meanings and explanations:

– `VN-ID` uniquely specifies the identification of virtual network.

The response message to the VN-QU message is defined as a VN-QU-REP message.

The VN-QU-REP message, indicated by the message type in the message header field, is sent by vNetwork management functional components to APP functional components in response to the VN-QU message.

Message format:

```
< VN-LIST-QU-REP-Message > ::= < Message header >
              { VN-ID }
               [ SDN-Controller-ID ]
              { Node-List }
              { Link-List }
              [ Forwarding-Policy ]
              { State }
```

Meanings and explanations:

– `VN-ID` uniquely specifies the identification of the virtual network.

– `SDN-Controller-ID` is optional and specifies the identification of the SDN controller if forwarding policy is required.

– `Node-List` uniquely specifies the nodes information of the virtual network. Each node in `Node-List` is described using the data model defined in clause 7.1.2.

– `Link-List` uniquely specifies the link information of virtual network, which is described using the data model defined in clause 7.1.4.

– `Forwarding-Policy` is optional and specifies the forwarding policy running in the virtual network. `SDN-Controller-ID` and `Forwarding-Policy` are usually used together.

– `State` uniquely specifies the state of the virtual network.

9.3 Signalling requirements of interface Smm

9.3.1 Virtual network and physical network mapping message and response message

The virtual network and physical network mapping message is defined as a PV-MA message.

The PV-MA message, indicated by the message type in the message header field, is sent by vNetwork management functional components to global mapping functional components to request the mapping of physical and virtual networks and physical resource allocation.

Message format:

```
< PV-MA-Message > ::= < Message header >
                  { VN-ID }
```

Meanings and explanations:

– VN-ID uniquely specifies the identification of already created virtual network.

The response message to the PV-MA message is defined as a PV-MA-REP message.

The PV-MA-REP message, indicated by the message type in the message header field, is sent by global mapping functional components to vNetwork management functional components in response to the PV-MA message.

Message format:

```
< PV-MA-REP-Message > ::= < Message header >
                  { PV-Mapping-Result}
                  [ VN-ID ]
                  [ PV-Mapping-Pair ]
```

Meanings and explanations:

– PV-Mapping-Result specifies the mapping result of physical and virtual networks. If mapped successfully, PV-Mapping-Result is TRUE; if not, PV-Mapping-Result is FALSE.

– VN-ID is optional and uniquely specifies the identification of the virtual network if the mapping pair of physical and virtual networks is created.

– PV-Mapping-Pair is optional and specifies the mapping pair of physical and virtual networks described using data models defined in clauses 7.1 and 7.2.

9.3.2 Remaining physical resource query message and response message

The remaining physical resource query message is defined as a PR-QU message.

The PR-QU message, indicated by the message type in the message header field, is sent by vNetwork management functional components to global mapping functional components to query the remaining physical resource.

Message format:

```
< PR-QU-Message > ::= < Message header >
```

The response message to the PR-QU message is defined as a PR-QU-REP message.

The PR-QU-REP message, indicated by the message type in the message header field, is sent by global mapping functional components to vNetwork management functional components in response to the PR-QU message.

Message format:

```
< PR-QU-REP-Message > ::= < Message header >
                  [ Devices ]
                  [ Links ]
```

Meanings and explanations:

– `Devices` is optional and uniquely specifies the devices collection of current physical network topology, described using data model defined in clause 7.2.2.

– `Links` is optional and uniquely specifies the directed link collection of current physical network topology, described using data model defined in clause 7.2.3.

Appendix I

An example of SDN-based slice abstraction

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

I.1 ONF abstraction model

[b-ONF TR-526] introduced the relevance of 5G slicing with SDN networking based on the Next Generation Mobile Networks (NGMN) vision for 5G [b-NGMN Slice]. The Open Networking Foundation (ONF) anticipated that the SDN controller (especially via its northbound interface) would fulfil the 5G purpose-specific requirements and definitions for slices.

Figure I.1 is an ONF abstraction model for 5G slicing.

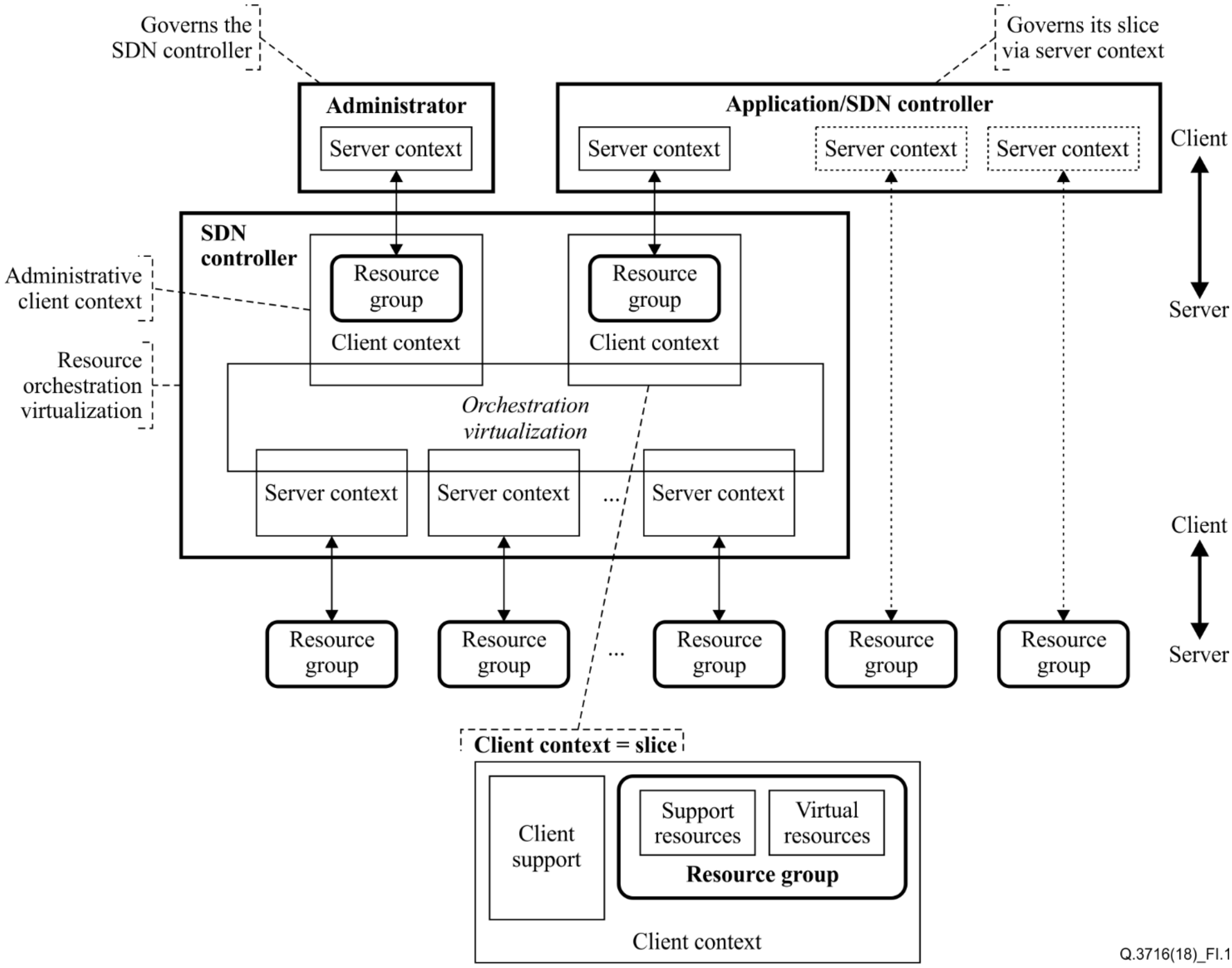

Figure I.1 – SDN-based slice abstraction

An SDN controller comprises the following functions and manageable contexts:

– Virtualization is the function of a controller to aggregate and abstract the underlying resources it manages-controls;

– Orchestration is the responsibility of the controller to dispatch resources in a way that simultaneously satisfies service demands from all of its clients as cost-effectively as possible;

- Client context represents the necessary and sufficient material in the SDN controller to support a given client, where a client may be a customer, partner, or even another functional component within the same administration that owns the controller. It includes all of the attributes of a service as requested by the client and may contain service-specific information necessary to map service attributes into the realization of the service;
- Server context is the symmetric counterpart to a client context. It contains everything necessary and sufficient to interact with a group of underlying resources, which could be, for example, a discrete network element or the virtual resources contracted from a partner domain.

The client context consists of resource groups and client support:

- Virtual resources represent infrastructure resources that are created from the SDN controller's underlying resources through the process of virtualization and that are exposed to the client by way of a mapping function;
- Support resources represent functions hosted in the SDN controller itself. Their purpose is to enable or facilitate interaction with the client, including attributes of services as requested by the client and information necessary to map these service attributes into the realization of the respective service;
- Client support may contain additional functions, resources and information needed, but not exposed to the client.

Each client context represents a set of resources managed-controlled by a controller and assumed to be directly applicable to 5G slicing. Resources may be statically predefined by the administrator, or allocated dynamically by the controller's orchestration function. A (higher-level) controller may slice resources further to support different clients, or combine resource groups to create more complex services.

I.2 Relations between PVMapping and ONF model

When Figure 6-1 and Figure I.1 are compared, it is found that i) the PVMapping mechanism closely relates to the creation/management of ONF's client and server context and ii) the PVMapping functions may be involved in the virtualization function of ONF's SDN controller model.

Appendix II

Restful API for PVMapping

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

II.1 Restful API for virtual network creation

URL：

post http:// 192.168.3.116:8181/opennh/v1/networkslices

Request Body:

{
"VNID": "9185804f-f910-46f5-8bfa-159fdc03df79",
"Name": "example_slice",
"NetworkSliceSLA": {
"HardReservation": {
"Reliability": "MTBF:100wh",
"Availability": "99.99%"
},
"ForwardingPolicy": "ospf"
},
"NodeList": [
{
"VirtualNodeID": "3ac7d5b0-3cef-4343-81d1-85e3030f9490",
"Name": "Node_1",
"DeviceType": {
"OFSwitch": {
"VirtualNodeCapacity": {
"OFTableCapacity":[
{
"TableID": 0,
"TableSize": 1000
}
]
}
}
},
"MappingRestriction": "BeiJing",
"InterfaceList": [
{

"ID": 1,
"Name": "1/0/1",
"Rate": "1 Gbps"
},
{
"ID": 2,
"Name": "1/0/2",
"Rate": "1 Gbps",
"AccessInfo": {
"Site": "office1"
}
}
]
},
{
" VirtualNodeID ": "1142027e-c03c-428d-b6bf-dd123248e45d",
"Name": "Node_2",
"DeviceType": {
"OFSwitch": {
"VirtualNodeCapacity":{
"OFTableCapacity": [
{
"TableID": 0,
"TableSize": 1000
}
]
}
}
},
"MappingRestriction": "GuangZhou",
"InterfaceList": [
{
"ID": 1,
"Name": "1/0/1",
"Rate": "1 Gbps"
},
{

"ID": 2,
"Name": "1/0/2",
"Rate": "1 Gbps",
"AccessInfo": {
"Site": "office2"
}
}
]
}
],
"LinkList": [
{
"Source": {
"VirtualNodeID": "3ac7d5b0-3cef-4343-81d1-85e3030f9490",
"Interface": 1
},
"Destination": {
" VirtualNodeID ": "1142027e-c03c-428d-b6bf-dd123248e45d",
"Interface": 1
},
"Type": "Ethernet",
"Durable": false,
"Annotations": {
"Latency": "500000",
"Bandwidth": "50"
}
},
{
"Source": {
" VirtualNodeID ": "1142027e-c03c-428d-b6bf-dd123248e45d",
"Interface": 1
},
"Destination": {
" VirtualNodeID ": "3ac7d5b0-3cef-4343-81d1-85e3030f9490",
"Interface": 1
},
"Type": "Ethernet",

"Durable": false,
"Annotations": {
"Latency": "500000",
"Bandwidth": "50"
}
}
],
"SDNController": {
"ActiveOFType": [
{
"ServerName": "guest1",
"IPAddress": "192.168.18.106",
"Port": 6653
}
]
}
}

Response Body：

```
Successful response result:
{
    "id": " 9185804f-f910-46f5-8bfa-159fdc03df79"
}
HTTP/1.1 201 OK
```

II.2 Restful API for virtual network activation

URL ：

Post http://192.168.3.116:8181/opennh/v1/networkslices/9185804f-f910-46f5-8bfa-159fdc03df79/active

Request Body: NULL

Response Body：

```
Successful response result:
HTTP/1.1 200 OK
```

II.3 Restful API for virtual network shutoff

URL：

Post http://192.168.3.116:8181/opennh/v1/networkslices/ac72e582-bbe4-4473-be8c-488d8911ec2b/inactive

Request Body: NULL

Response Body：

```
Successful response result:
HTTP/1.1 200 OK
```

II.4 Restful API for virtual network release

URL：

Post http://192.168.3.116:8181/opennh/v1/networkslices/ac72e582-bbe4-4473-be8c-488d8911ec2b/release

Request Body: NULL

Response Body：

```
Successful response result:
HTTP/1.1 200 OK
```

II.5 Restful API for virtual network deletion

URL：

Delete http://192.168.3.116:8181/opennh/v1/networkslices/9185804f-f910-46f5-8bfa-159fdc03df79

Request Body: NULL

Response Body：

```
Successful response result:
HTTP/1.1 200 OK
```

II.6 Restful API for query

II.6.1 Restful API for virtual network list query

URL：

Get http://192.168.3.116:8181/opennh/v1/networkslices

Request Body: NULL

Response Body：

```
Successful response result:
{
    "NetworkSlices": [
        {
            "VNID": "840eb319-0f82-4250-adc3-911c778526d4",
            "Name": "slice1",
            "State": "running"
        },
        {
            "VNID": "4dcd328d-bff0-48fb-b8df-65f6cfd53fae",
            "Name": "slice2",
            "State": "stop"
        }
    ]
}
```

II.6.2 Restful API for virtual network query

URL:

Get http://192.168.3.116:8181/opennh/v1/networkslices/9185804f-f910-46f5-8bfa-159fdc03df79

Request Body: NULL

Response Body：

```
Successful response result:
{
"VNID": "9185804f-f910-46f5-8bfa-159fdc03df79",
"Name": "example_slice",
"NetworkSliceSLA": {
"HardReservation": {
"Reliability": "MTBF:100wh",
"Availability": "99.99%"
},
"ForwardingPolicy": "ospf"
},
"NodeList": [
{
"VirtualNodeID": "3ac7d5b0-3cef-4343-81d1-85e3030f9490",
"Name": "Node_1",
"DeviceType": {
"OFSwitch": {
"VirtualNodeCapacity": {
"OFTableCapacity":[
{
"TableID": 0,
"TableSize": 1000
}
]
}
}
},
"MappingRestriction": "BeiJing",
"InterfaceList": [
{
"ID": 1,
"Name": "1/0/1",
"Rate": "1 Gbps"
},
{
"ID": 2,
"Name": "1/0/2",
"Rate": "1 Gbps",
"AccessInfo": {
"Site": "office1"
}
}
```

```
]
},
{
" VirtualNodeID ": "1142027e-c03c-428d-b6bf-dd123248e45d",
"Name": "Node_2",
"DeviceType": {
"OFSwitch": {
"VirtualNodeCapacity":{
"OFTableCapacity": [
{
"TableID": 0,
"TableSize": 1000
}
]
}
}
},
"MappingRestriction": "GuangZhou",
"InterfaceList": [
{
"ID": 1,
"Name": "1/0/1",
"Rate": "1 Gbps"
},
{
"ID": 2,
"Name": "1/0/2",
"Rate": "1 Gbps",
"AccessInfo": {
"Site": "office2"
}
}
]
}
],
"LinkList": [
{
"Source": {
"VirtualNodeID": "3ac7d5b0-3cef-4343-81d1-85e3030f9490",
"Interface": 1
},
"Destination": {
" VirtualNodeID ": "1142027e-c03c-428d-b6bf-dd123248e45d",
"Interface": 1
},
```

```
"Type": "Ethernet",
"Durable": false,
"Annotations": {
"Latency": "500000",
"Bandwidth": "50"
}
},
{
"Source": {
" VirtualNodeID ": "1142027e-c03c-428d-b6bf-dd123248e45d",
"Interface": 1
},
"Destination": {
" VirtualNodeID ": "3ac7d5b0-3cef-4343-81d1-85e3030f9490",
"Interface": 1
},
"Type": "Ethernet",
"Durable": false,
"Annotations": {
"Latency": "500000",
"Bandwidth": "50"
}
}
],
"SDNController": {
"ActiveOFType": [
{
"ServerName": "guest1",
"IPAddress": "192.168.18.106",
"Port": 6653
}
]
}
}
```

II.7 Restful API for virtual network and physical network mapping

URL:

Get http://192.168.3.116:8181/opennh/v1/networkslices/ac72e582-bbe4-4473-be8c-488d8911ec2b/mapping

Request Body: NULL

Response Body：

```
{
    "VNID": "ac72e582-bbe4-4473-be8c-488d8911ec2b",
    "Name": "example_slice",
```

```
    "State": "Started",
    "NetworkSliceSLA": {
        "HardReservation": {
            "Reliability": "",
            "Availability": "",
            "MaxDelay": "",
            "Jitter": ""
        },
"ForwardingPolicy": "ospf"
    },
    "NodeList": [
        {
            "VirtualNodeID": "1142027e-c03c-428d-b6bf-dd123248e45d",
            "Name": "BT1",
            "DeviceType": {
                "OFSwitch": {
  "VirtualNodeCapacity":{
                    "OFTableCapacity": [
                        {
                            "TableID": "0",
                            "TableSize": "1000"
                        }
                    ]
}
                }
            },
            "InterfaceList": [
                {
                    "ID": "1",
                    "Name": "1/0/1",
                    "Rate": "1 Gbps"
                },
                {
                    "ID": "2",
                    "Name": "1/0/2",
                    "Rate": "1 Gbps",
                    "AccessInfo": {
                        "Site": "office2"
                    }
                }
            ],
            "MappingRestriction": "GuangZhou"
        },
        {
            " VirtualNodeID ": "3ac7d5b0-3cef-4343-81d1-85e3030f9490",
```

```
            "Name": "BT2",
            "DeviceType": {
                "OFSwitch": {
                    "OFTableCapacity": [
"VirtualNodeCapacity":{
                        {
                            "TableID": "0",
                            "TableSize": "1000"
                        }
                    ]
}
                }
           },
            "InterfaceList": [
                {
                    "ID": "1",
                    "Name": "1/0/1",
                    "Rate": "1 Gbps"
                },
                {
                    "ID": "2",
                    "Name": "1/0/2",
                    "Rate": "1 Gbps",
                    "AccessInfo": {
                        "Site": "office1"
                    }
                }
            ],
            "MappingRestriction": "BeiJing"
        }
    ],
    "LinkList": [
        {
            "Source": {
                "VirtualNodeID": "1142027e-c03c-428d-b6bf-dd123248e45d",
                "Interface": "1"
            },
            "Destination": {
                " VirtualNodeID ": "3ac7d5b0-3cef-4343-81d1-85e3030f9490",
                "Interface": "1"
            },
            "Type": "Ethernet",
            "Durable": false,
            "State": "DOWN",
            "Annotations": {
```

```
                "Bandwidth": "30",
                "Latency": "20"
            },
            "PhysicalPath": {
                "Path": [
                    {
                        "BackupProperty": "ACTIVE",
                        "PathNodeList": {
                            "PathNode": [
                                {
                                    "DeviceID": "of:0000000000000003",
                                    "SupplimentaryNode": false,
                                    "EgressInterface": "DefaultPort{element
=of:0000000000000003, number=1, isEnabled=true, type=COPPER, portSpeed=1048
5}"
                                },
                                {
                                    "DeviceID": "of:0000000000000002",
                                    "SupplimentaryNode": true,
                                    "IngressInterface": "DefaultPort{elemen
t=of:0000000000000002, number=2, isEnabled=true, type=COPPER, portSpeed=104
85}",
                                    "EgressInterface": "DefaultPort{element
=of:0000000000000002, number=1, isEnabled=true, type=COPPER, portSpeed=1048
5}"
                                },
                                {
                                    "DeviceID": "of:0000000000000001",
                                    "SupplimentaryNode": false,
                                    "IngressInterface": "DefaultPort{elemen
t=of:0000000000000001, number=1, isEnabled=true, type=COPPER, portSpeed=104
85}"
                                }
                            ]
                        }
                    }
                ]
            }
        },
        {
            "Source": {
                " VirtualNodeID ": "3ac7d5b0-3cef-4343-81d1-85e3030f9490",
                "Interface": "1"
            },
            "Destination": {
                " VirtualNodeID ": "1142027e-c03c-428d-b6bf-dd123248e45d",
                "Interface": "1"
```

```
            },
            "Type": "Ethernet",
            "Durable": false,
            "State": "DOWN",
            "Annotations": {
                "Bandwidth": "30",
                "Latency": "20"
            },
            "PhysicalPath": {
                "Path": [
                    {
                        "BackupProperty": "ACTIVE",
                        "PathNodeList": {
                            "PathNode": [
                                {
                                    "DeviceID": "of:0000000000000001",
                                    "SupplimentaryNode": false,
                                    "EgressInterface": "DefaultPort{element=of:0000000000000001, number=1, isEnabled=true, type=COPPER, portSpeed=10485}"
                                },
                                {
                                    "DeviceID": "of:0000000000000002",
                                    "SupplimentaryNode": true,
                                    "IngressInterface": "DefaultPort{element=of:0000000000000002, number=1, isEnabled=true, type=COPPER, portSpeed=10485}",
                                    "EgressInterface": "DefaultPort{element=of:0000000000000002, number=2, isEnabled=true, type=COPPER, portSpeed=10485}"
                                },
                                {
                                    "DeviceID": "of:0000000000000003",
                                    "SupplimentaryNode": false,
                                    "IngressInterface": "DefaultPort{element=of:0000000000000003, number=1, isEnabled=true, type=COPPER, portSpeed=10485}"
                                }
                            ]
                        }
                    }
                ]
            }
        }
    ],
    "SDNController": {
        "ActiveOFType": [
```

```
            {
                "ServerName": "guest1",
                "IPAddress": "192.168.18.106"
            }
        ]
    }
}
```


II.8 Restful API for remaining physical resource query

URL:

Get http://192.168.3.116:8181/opennh/v1/topology

Request Body: NULL

Response Body：


```
{
"Time": 172121599689975,
"Devices": [
{
" D e v i c e I D " :  " o f : 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 3 " ,
"Type": "SWITCH",
"Availability": true,
"Manufacturer": "Stanford University, Ericsson Research and CPqD Research",
"Hardware": "OpenFlow 1.3 Reference Userspace Switch",
"Software": "Sep 18 2015 17:28:33",
"Annotations": {
"managementAddress": "192.168.18.99",
"protocol": "OF_13",
"channelId": "192.168.18.99:48204"
}
},
{
"DeviceID": "of:0000000000000004",
"Type": "SWITCH",
"Availability": true,
"Manufacturer": "Stanford University, Ericsson Research and CPqD Research",
"Hardware": "OpenFlow 1.3 Reference Userspace Switch",
"Software": "Sep 18 2015 17:28:33",
"Annotations": {
"managementAddress": "192.168.18.99",
"protocol": "OF_13",
"channelId": "192.168.18.99:48200"
}
},
{
"DeviceID": "of:0000000000000001",
```

```
"Type": "SWITCH",
"Availability": true,
"Manufacturer": "Stanford University, Ericsson Research and CPqD Research",
"Hardware": "OpenFlow 1.3 Reference Userspace Switch",
"Software": "Sep 18 2015 17:28:33",
"Annotations": {
"managementAddress": "192.168.18.99",
"protocol": "OF_13",
"channelId": "192.168.18.99:48202"
}
},
{
"DeviceID": "of:0000000000000002",
"Type": "SWITCH",
"Availability": true,
"Manufacturer": "Stanford University, Ericsson Research and CPqD Research",
"Hardware": "OpenFlow 1.3 Reference Userspace Switch",
"Software": "Sep 18 2015 17:28:33",
"Annotations": {
"managementAddress": "192.168.18.99",
"protocol": "OF_13",
"channelId": "192.168.18.99:48201"
}
},
{
"DeviceID": "of:0000000000000005",
"Type": "SWITCH",
"Availability": true,
"Manufacturer": "Stanford University, Ericsson Research and CPqD Research",
"Hardware": "OpenFlow 1.3 Reference Userspace Switch",
"Software": "Sep 18 2015 17:28:33",
"Annotations": {
"managementAddress": "192.168.18.99",
"protocol": "OF_13",
"channelId": "192.168.18.99:48203"
}
}
],
"Links": [
{
"Source": {
"Interface": "2",
"DeviceID": "of:0000000000000002"
},
"Destination": {
```

```
"Interface": "1",
"DeviceID": "of:0000000000000003"
},
"Type": "DIRECT",
"State": "ACTIVE"
},
{
"Source": {
"Interface": "2",
"DeviceID": "of:0000000000000003"
},
"Destination": {
"Interface": "1",
"DeviceID": "of:0000000000000004"
},
"Type": "DIRECT",
"State": "ACTIVE"
},
{
"Source": {
"Interface": "2",
"DeviceID": "of:0000000000000004"
},
"Destination": {
"Interface": "1",
"DeviceID": "of:0000000000000005"
},
"Type": "DIRECT",
"State": "ACTIVE"
},
{
"Source": {
"Interface": "2",
"DeviceID": "of:0000000000000001"
},
"Destination": {
"Interface": "2",
"DeviceID": "of:0000000000000005"
},
"Type": "DIRECT",
"State": "ACTIVE"
},
{
"Source": {
"Interface": "2",
```

```
"DeviceID": "of:0000000000000005"
},
"Destination": {
"Interface": "2",
"DeviceID": "of:0000000000000001"
},
"Type": "DIRECT",
"State": "ACTIVE"
},
{
"Source": {
"Interface": "1",
"DeviceID": "of:0000000000000002"
},
"Destination": {
"Interface": "1",
"DeviceID": "of:0000000000000001"
},
"Type": "DIRECT",
"State": "ACTIVE"
},
{
"Source": {
"Interface": "1",
"DeviceID": "of:0000000000000003"
},
"Destination": {
"Interface": "2",
"DeviceID": "of:0000000000000002"
},
"Type": "DIRECT",
"State": "ACTIVE"
},
{
"Source": {
"Interface": "1",
"DeviceID": "of:0000000000000004"
},
"Destination": {
"Interface": "2",
"DeviceID": "of:0000000000000003"
},
"Type": "DIRECT",
"State": "ACTIVE"
},
```

```
{
"Source": {
"Interface": "1",
"DeviceID": "of:0000000000000005"
},
"Destination": {
"Interface": "2",
"DeviceID": "of:0000000000000004"
},
"Type": "DIRECT",
"State": "ACTIVE"
},
{
"Source": {
"Interface": "1",
"DeviceID": "of:0000000000000001"
},
"Destination": {
"Interface": "1",
"DeviceID": "of:0000000000000002"
},
"Type": "DIRECT",
"State": "ACTIVE"
}
],
"Clusters": [
{
"NodeID": "192.168.18.66",
"IPAddress": "192.168.18.66",
"TcpPort": 9876,
"Status": "ACTIVE"
}
]
}
```

Appendix III

Typical scenarios and procedures

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

III.1 Network slicing

Network slicing is a concept for multiple logical networks as virtually independent business operations on a common physical infrastructure. Taking a 5G scenario as an example, its complete slicing is composed of not only virtual machine based network functions at cloud-RAN (C-RAN) and cloud core (C-Core) level, but also transport network resources which can be sliced to a logical network slicing based on a radio access/transport network and satisfy the customized throughput, delay and mobility.

Although traditional VPN technology can provide physical network resource isolation, there is a lack of rigid QoS isolation, independent and flexible operation and management. Network slicing allows network operators to deploy their new service on a dedicated slice, also known as a logical virtual network, on a shared physical infrastructure and does not affect the existing services. Each slice has its independent lifecycle management, including slice creation/deletion, bandwidth/topology adjustment for a slice, dynamic slice migration, etc. In addition, the global network utilization can be improved with SDN.

A PVMapping framework is well adapted to archive the network slicing requirement described in this Recommendation. Each slice can be regarded as a real physical network topology including nodes and links, and slices are not affected by each other. The flow of each slice can be directed by the corresponding tenant's controller.

The status and statistic information about the network slicing and the corresponding physical and virtualized nodes and links can be managed and orchestrated by management and orchestration (MANO), whose functional components need to be instantiated, and the corresponding interfaces between PVMapping and MANO need to be defined.

The corresponding signalling procedure for the creation of network slicing is shown in Figure III.1

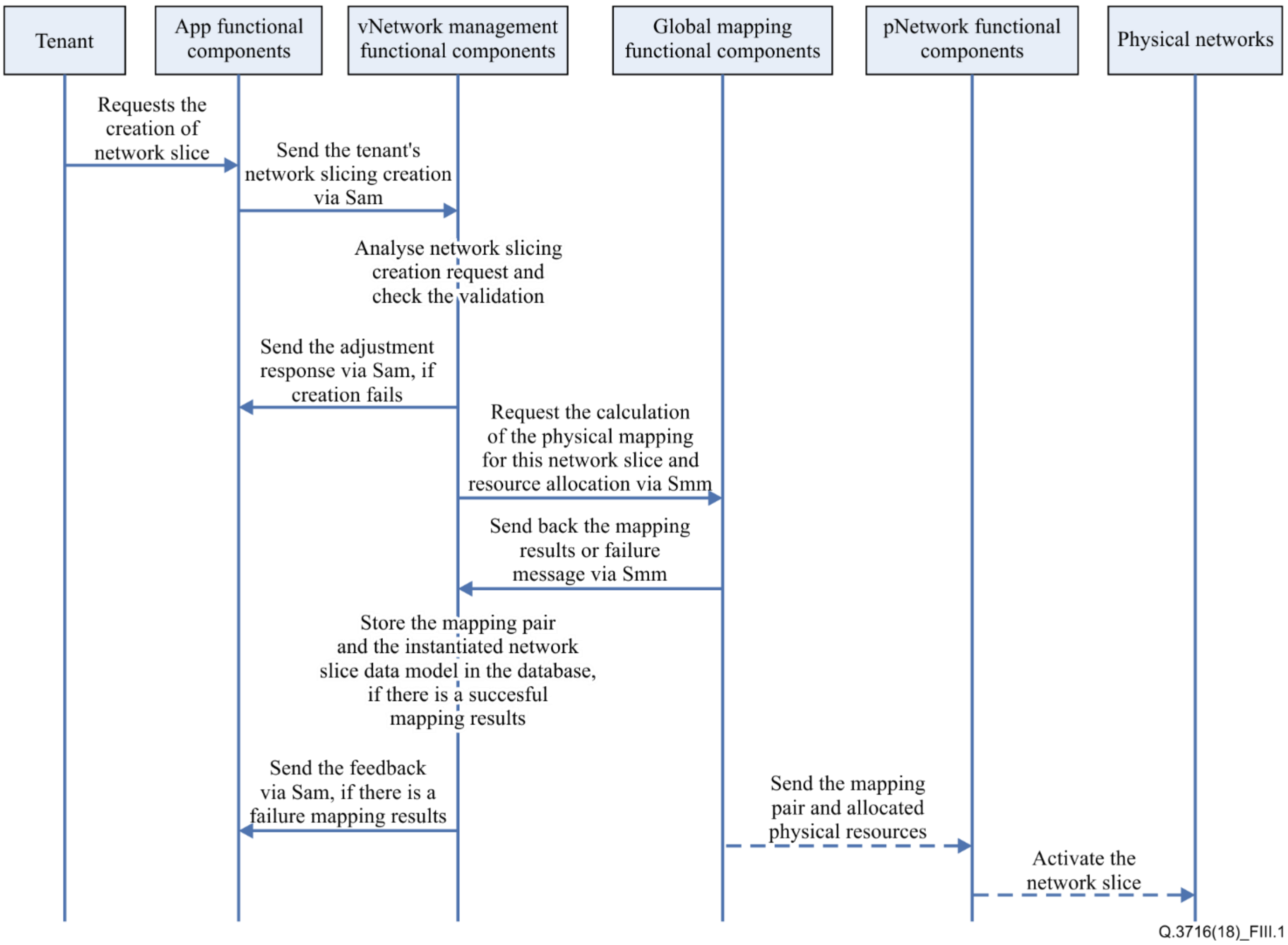

Figure III.1 – Signalling procedure for creation of network slicing

1) Tenant requests the creation of a network slice via App functional components, including required nodes, ports, links, etc. and sends it to vNetwork management functional components via Sam.

2) vNetwork management functional components analyse the requests on network slice creation and check the validation. If the creation fails, the tenant will receive the adjustment response via Sam; if the creation succeeds, vNetwork management functional components request global mapping functional components to calculate the physical mapping of this network slice and resource allocation via Smm.

3) The mapping results will be sent back to the vNetwork management functional components from global mapping functional components via Smm, if there is a successful calculation result. The failure message will be sent back to the vNetwork management functional components from global mapping functional components via Smm, if the calculation fails, e.g., lack of physical resource.

4) If vNetwork management functional components receive the failure message of mapping via Smm, it will send feedback to APP functional components via Sam, if not, the mapping pair and the instantiated network slice data model (XML file) are stored in the database of vNetwork management functional components.

NOTE – After step 4, pNetwork functional components activate the network slice in the physical network, including the creation and issue of a flow table, based on the mapping pair and allocated physical resources sent by vNetwork management functional components.

III.2 Virtual private cloud

Network operators can virtualize cloud and networking resources into multiple isolated virtual private clouds (VPCs) and provide them to their customers. A customer can establish and manage the network easily in a typical VPC, for example by deploying or removing virtualized network devices (e.g., vRouter and vSwitch), adjusting the topology of VPC networks, specifying packet forwarding policies and deploying or removing virtualized network services (e.g., load balancer, firewalls, databases, DNS, etc.). VPCs are actually performed by virtual machines (VMs) located on compute servers, which may be located in different geographically distributed data centres, connected through physical or overlay networks.

The manipulation of the VPC network may also affect the configuration of physical networks. For example, when two new VMs associated with a given VPC are deployed in two different data centres, the VPC control mechanism needs to generate a VPN between these two data centres for the internal VPC communications. Therefore, the control mechanism for a VPC should be able to adjust the underlying network at run time when the tenant requests change to the VPC network or service deployment. Considering VPC itself cannot be aware of the physical nodes, when the tenant's controller sends an OpenFlow flow table to the specific virtual nodes of VPC. A translation function needs to be provided to convert the VPC based flow table to a physical nodes table in order that underlying physical elements can understand the instruction.

A PVMapping framework is well suited to address the problem above. Each VPC can be regarded as a real physical network topology including nodes and links and VPCs are not affected by each other. The flow of each VPC can be directed by its tenant's controller. The corresponding signalling procedure for creation of VPC is shown in Figure III.2.

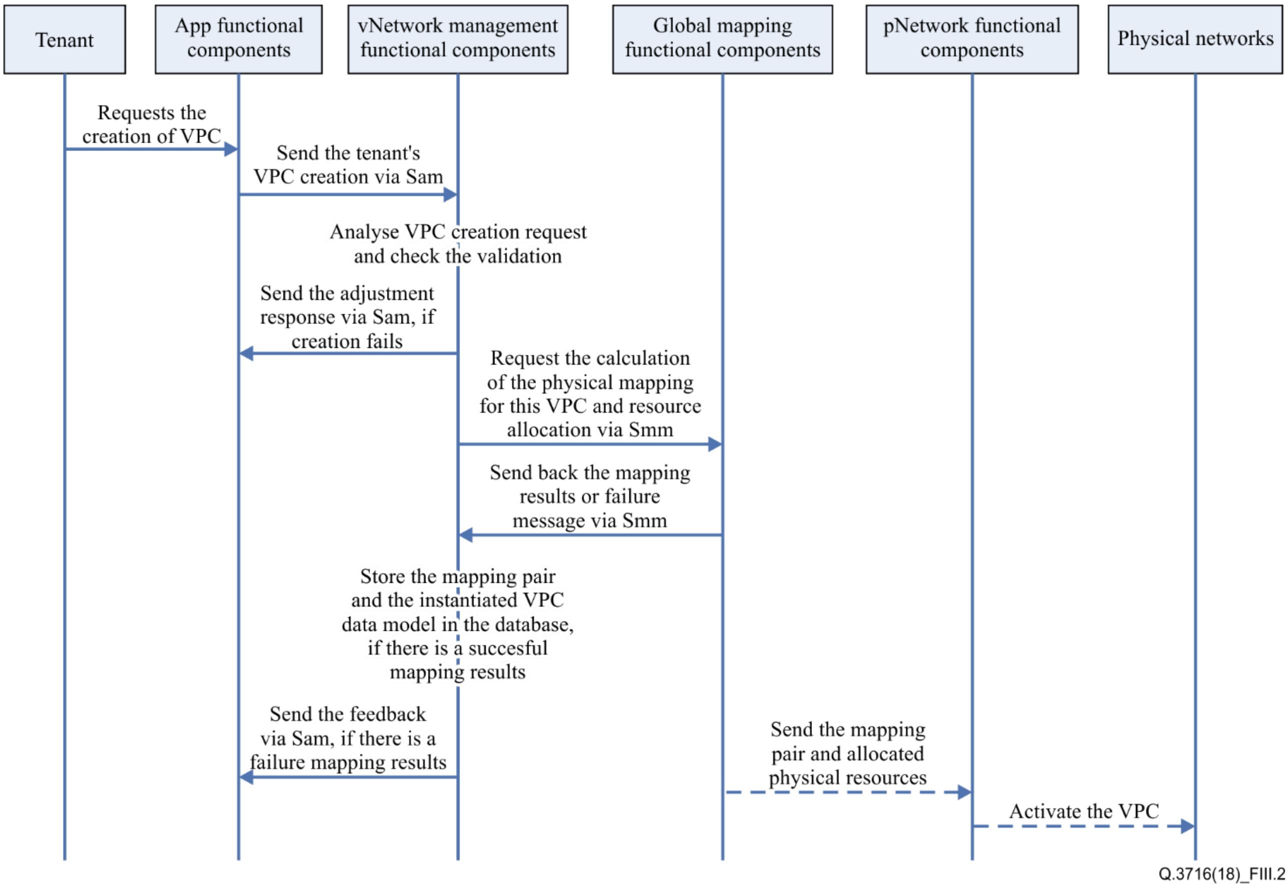

Figure III.2 – Signalling procedure for creation of VPC

1) Tenant requests the creation of VPC via App functional components, including required nodes, ports, links, etc. and sends it to vNetwork management functional components via Sam.

2) vNetwork management functional components analyse the requests on VPC creation and check the validation. If the creation fails, the tenant will receive the adjustment response via Sam; if the creation succeeds, vNetwork management functional components request global mapping functional components to calculate the physical mapping of this VPC and resource allocation via Smm.

3) The mapping results will be sent back to the vNetwork management functional components from global mapping functional components via Smm, if there is a successful calculation result. The failure message will be sent back to the vNetwork management functional components from global mapping functional components via Smm, if the calculation fails, e.g., lack of physical resource.

4) If vNetwork management functional components receive the failure message of mapping via Smm, it will send feedback to APP functional components via Sam, if not, the mapping pair and the instantiated VPC data model (XML file) are stored in the database of vNetwork management functional components.

NOTE – After step 4, pNetwork functional components activate the VPC in the physical network, including the creation and issue of a flow table, based on the mapping pair and allocated physical resources sent by vNetwork management functional components.

Bibliography

[b-IETF RFC 4364] IETF RFC 4364 (2006), *BGP/MPLS IP Virtual Private Networks (VPNs).*
https://tools.ietf.org/html/rfc4364

[b-IETF RFC 7348] IETF RFC 7348 (2014), *Virtual eXtensible Local Area Network (VXLAN): A Framework for Overlaying Virtualized Layer 2 Networks over Layer 3 Networks.*
https://tools.ietf.org/html/rfc7348

[b-NGMN Slice] NGMN Alliance document, (2016), *Description of Network Slicing Concept*.
https://www.ngmn.org/fileadmin/user_upload/160113_Network_Slicing_v1_0.pdf

[b-ONF TR-526] ONF TR-526 (2016), *Applying SDN Architecture to 5G Slicing*.

SERIES OF ITU-T RECOMMENDATIONS

Series A	Organization of the work of ITU-T
Series D	Tariff and accounting principles and international telecommunication/ICT economic and policy issues
Series E	Overall network operation, telephone service, service operation and human factors
Series F	Non-telephone telecommunication services
Series G	Transmission systems and media, digital systems and networks
Series H	Audiovisual and multimedia systems
Series I	Integrated services digital network
Series J	Cable networks and transmission of television, sound programme and other multimedia signals
Series K	Protection against interference
Series L	Environment and ICTs, climate change, e-waste, energy efficiency; construction, installation and protection of cables and other elements of outside plant
Series M	Telecommunication management, including TMN and network maintenance
Series N	Maintenance: international sound programme and television transmission circuits
Series O	Specifications of measuring equipment
Series P	Telephone transmission quality, telephone installations, local line networks
Series Q	**Switching and signalling, and associated measurements and tests**
Series R	Telegraph transmission
Series S	Telegraph services terminal equipment
Series T	Terminals for telematic services
Series U	Telegraph switching
Series V	Data communication over the telephone network
Series X	Data networks, open system communications and security
Series Y	Global information infrastructure, Internet protocol aspects, next-generation networks, Internet of Things and smart cities
Series Z	Languages and general software aspects for telecommunication systems

Printed in Switzerland
Geneva, 2018