Network Service Description for Virtual Network Deployment: A constraints based OVF extension proposal

Gladys Diaz
Paris 13 University, Sorbonne Paris Cité
93430 Villetaneuse, France
e-mail: gladys.diaz@univ-paris13.fr

Noëmie Simoni
Engineering School Telecom ParisTech
75013 Paris, France
e-mail: noemie.simoni@telecom-paristech.fr

***Abstract*—The Cloud has allowed an evolution in the design, deployment and management of services. Applying a Cloud model at the network level involves incorporating into the life cycle of network services the new processes that take into account the virtualization. To take advantage of advances such as SDN (*Software-Defined Networking*) and NFV (*Network Functions Virtualization*), we must not only take into account the virtualization of network equipment, but we must incorporate the network level virtualization with its control plane into the design of virtual networks to meet the demands of users. This leads us to distinguish the "network deployment phase" from the "placement" of virtual elements into a physical infrastructure.**

To help automate the deployment and management process, it is necessary to describe all of the information used by them. Our proposal defines an OVF (*Open Virtualization Format*) extension enabling to describe the behavior of elements implemented in the virtual network to deploy (nodes, links, network). We include in our description the QoS and placement constraints. Our description shows two viewpoints: requested service quality and quality of service offered.

***Keywords*—Network Service Description; Virtual Network deployment; Information Model; Virtual Network Management; QoS and placement constraints; Cloud Networking; NFV; SDN.**

I. Introduction

Incorporating the Cloud's viewpoint on current service architectures has allowed an evolution in the design, deployment and management of network architectures. SDN [1] and NFV [2] are the two major enablers of this evolution at network level. Cloud is based in virtualization technology which is applied at different levels. Applying virtualization at the network level changes the way networks are designed and deployed. Currently networks are not described as a service, but are regarded as part of the infrastructure.

Incorporate virtualization into the design of virtual networks (VNs) and theirs control plane involves the introduction of "a prior step" to the current placement process, enabling to identify the expected behavior of VN (the network service associated with it). The deployment of virtual networks becomes in this sense a key point in the life cycle of network services, this last affected by the introduction of virtualization that defines a VN building process.

The agility of deployment of services and networks is defined through the specification, installation and configuration of theirs constituent elements. To help automate the deployment and management process, it is necessary to describe all of the information used by these processes. One must describe the service requested, taking into account the related information about the behavior of services (the Quality of Service - QoS) as well as the description of resources for deployment. This is expected to fill the current gap between the information of the service network layer and those physical resources. Such an information model will allow the monitoring and if necessary (according with changes in SLA, user mobility, etc.) the rapid substitution of VN components. Currently, we work in the definition of a complete information model concerning the network deployment phase. This work is out of the scope of this paper.

This paper focuses on the "placement phase". We are motivated to consider the description of service's viewpoint at the network level allowing to introduce the on-demand VN building. We define the VN behavior through our QoS model [3] and we describe it by using the OVF language. Given the possibilities of the extension and its portability, OVF stands today as an appropriate candidate to describe new information about VNs in virtualized environments. In our knowledge, no extension has been proposed to introduce the VN description with a service viewpoint with OVF.

We present the related work about OVF language in (II) and our proposed OVF extension for VN description in (III). The conclusions and perspectives are presented in (IV).

II. Related work about OVF language

In a virtualized environment, the current deployment process is a placement process, and it is based on the use of images that include the services and the IT resources needed for their implementation. In the context of cloud applications, information is represented through Virtual Machines (VMs), in the context of cloudified networks the information represented is about the NFV, where services are *virtualized network functions* (VNFs). The deployment/placement in the case of NFV is guided by the resource requirements for the execution

978-3-901882-85-2

of the VM(s) that contains the network functions. As for the Cloud IT, the Telco Cloud deployment/placement is based on optimizing the use of IT resources. But how to consider and describe the "on-demand" of the user, which can change during the session? There is a lack in the description of the behavior of VNs to be deployed (the service network layer).

In addition, the "network deployment process" is concerned by the programming of network control plane. SDN enables control and adapt the network, regardless of the physical constraints of the underlying hardware, and through an abstract layer called a controller. Thus, virtualize the control plane will enable to consider the calcul of the routing tables in accordance with the QoS requirements of flows, and to describe the placement of their translation (the forwarding tables) by considering these same constraints (QoS, placement).

Standardization to allow interoperability in cloud addresses different categorizations of solutions with different objectives [4], [5]. The OVF language [6] is proposed for placement description at IaaS (*Infrastructure as a Service*) level.

A. OVF: a placement-oriented language

OVF is a standard published by the DMTF (*Distributed Management Task Force*) that describes a neutral specification (regarding the hypervisor) on how to package and distribute software to run in virtual machines.

The OVF package consists of several different files such as the OVF descriptor, OVF manifest, OVF certificate, disk image files, and additional resources files. The OVF descriptor is defined by an XML metadata format associated with VMs. The OVF descriptor focuses on placement aspects and the description of the hardware and software used in virtual machines. In the current proposal of the OVF descriptor, the section that describes the logical networks between VMs is called the <NetworkSection>.

Different versions of OVF have been proposed; the most recent being V2.0.1 (December 2013) [7], that introduces new concepts about network description: the *NetworkPortProfile* and the *EthernetPortItem*.

B. OVF extensions related works

Various researchers have studied the possibility of an OVF extension. The authors in [8] [9] propose the definition of new sections in the OVF package. One section provides KPI (*Key Performance Indicator*) tags to enable monitoring operations. The other section defines elasticity rules which making use of the KPI, and describes how the service should be scaled. Another proposal is the definition of macros in the <ProductSection> that can describe the entire IaaS setup required for a service.

In [10] the authors introduce new OVF extensions that define a *Manifest Language* which is used in the RESERVOIR project [11]. OVF is used to propose a service definition language that captures the functional requirements of the service. This work is focused in the mapping of high-level service requirements/metrics (e.g., response time) to infrastructure level requirements/metrics (e.g., CPU utilization).

Another extension called OVF ++ has been proposed as part of 4CaaSt [12] project. We use OVF ++ under Project OpenCloudware [13] to describe the complete architecture to be deployed. The specification of the architecture section is based on ADL (*Architecture Definition Language*) language. The OVF ++ proposal is focused on the description of the components of the architecture by using the Fractal component model. OVF ++ integrates a description of physical resources and the virtual machine architecture software components.

These proposals allow the representation of some network information, but again are limited on physical requirements (based on the placement of network elements into local area networks), such as reserved bandwidth and ports (used at level 2). Even with these proposals, there is still a lack of information regarding the description of the network service offered by the virtual network (the service description associated with the processing of data flow at VN, what we call the media delivery). To introduce the concept of "network as a service" we propose the description of QoS properties to characterize the expected behavior of VN. Our proposal joined the movement of the referenced works by using the extensibility properties of the OVF language.

III. OVF Extension for Virtual Network Description

The OVF extension proposal allows completing the current view of a VN, based just on virtualization of equipment (switches, routers, bridges, etc.) and resources (Ethernet interfaces, bandwidth link) with a view of the service offered (characterized by the QoS and placement constraints). The constraints including QoS and placement aspects (at nodes and links), as well as the description of the network-related services (VNFs) contained in the nodes of the VN characterize this view.

Our proposal enables to describe, at deployment phase, the VNs in accordance with the QoS requested by the application flows. During the operational phase, this information can be used for the management process. Thus, the monitoring process may use this information to verify that the current behavior of the network elements involved in the VN is consistent with the selected deployment configuration.

A. Virtual Network Section Description

We describe in this section the Meta-information for specifying the new section called <VirtualNetworkSection> that completes the current section proposed by the OVF language, the <NetworkSection>. By apply our NLN (Node, Link, Network) model [14] and QoS generic model [3] the VN description includes three parts: the *VirtualLink section*, the *Virtual Node section* and the *QoS Network Constraints section*. Several <VirtualNetworkSection> can be described inside one <NetworkSection>. An overall view about these new sections proposed is shown in Fig. 1.

For each proposed new section (node, link, network), we describe the placement and QoS constraints involved, according with the definition given in [15] (See Fig. 2).

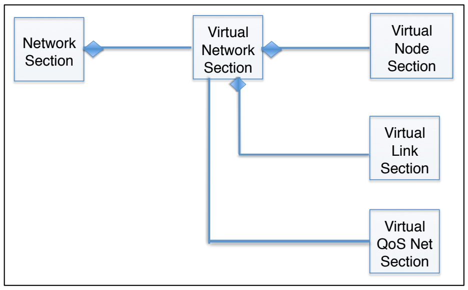

Fig. 1. *New sections proposed by the <VirtualNetworkSection>*

	QoS Criteria			
	Availability (A)	***Capacity (C)***	***Delay (D)***	***Reliability (R)***
Description	is the portion of time that a virtual network component makes the requested service without failure.	represents the ability of a virtual network component to fulfill the required service.	is the total time taken by a virtual network component to fulfill its functions.	is the compliance rate of the rendered service compared to the demanded one.
Example of parameters	MTBF, MTTR, etc.	Bandwidth, PDU number, transaction number, etc.	Latency, jitter, propagation delay, etc.	Loss rate of delivery messages, error rate, etc.

Fig. 2. *QoS criteria considering in our QoS model*

```
<ovf:QoSFlowSensitivity>
  <ovf: QoSSensitivity>
      <ovf:Availability ovf: Parameter= "Bandwidth" ovf: Value="80%" ovf: Degree="High" /ovf:Availability>
      <ovf:Delay ovf: Parameter= "One-way delay" ovf: Value="'150 ms" ovf: Degree="High" /ovf:Delay>
      <ovf:Reliability ovf: Parameter= "PacketLoss rate" ovf: Value="<2%" ovf: Degree="Low" /ovf:Reliability>
      <ovf:Capability ovf: Parameter= "Bandwidth" ovf: Value=">64 Kbps" ovf: Degree="Medium" /ovf:Capability>
  </ovf: QoSSensitivity>
  <ovf: Service> VoIP service with G711 codec </ovf: Service>
</ovf:QoSFlowSensitivity>
```

Fig. 3. *Example of QoSFlowSensitivity description for a VoIP flow*

To characterize the behavior of the VN, we take into account the QoS requirements of flow applications. We consider this description such as an input to the deployment process and be included with an OVF file that describes the flow application sensitivities [16]. Fig. 3 shows an application of an OVF++ file that represent the *QoSFlowSensitivities* of VoIP flow. In this example we consider the service of VoIP over UDP/IP/Ethernet, and we describe this service through our four QoS criterion. For Availability (high sensitivity) we consider the bandwidth availability equal to 80%, for Delay (high sensitivity) we consider a one-way delay with a value of 150 ms, for the Reliability (low sensitivity) we define a packet loss rate of <2% and for Capability criteria (medium sensitivity) we define a bandwidth of 64kbps or more. This description enables the NaaS (*Network as a Service*) provider to consider appropriate information to build the best VN solution in accordance with the flow requirements.

Such as is shown in Fig. 4, the first task (before the *Virtual Deployment process*) consists to translate the QoS request (SLA) into the QoS constraints to be guaranteed by the network services and equipment. This approach enables the recovery of available resources that can meet the required QoS.

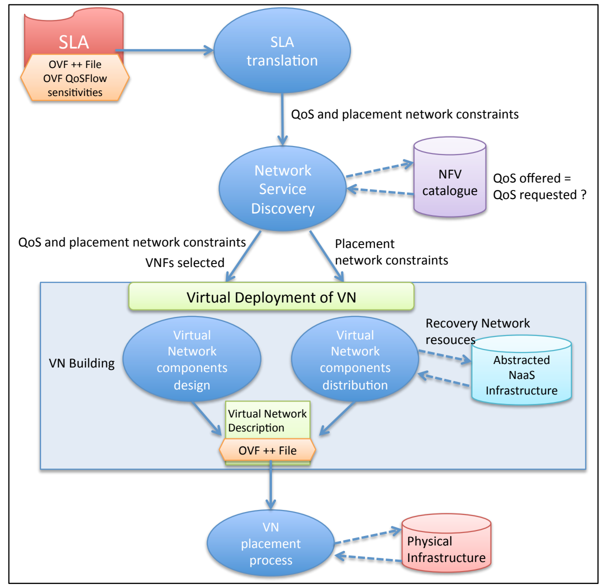

Fig. 4. *Virtual Deployment of VN process and the VN description*

Thus in accordance with the request and their QoS constraints, the NaaS provider selects the VNFs which corresponds with the requested service. This is made by comparing the QoS offered by the VNF descriptor (in the NFV catalogues) with the translated QoS constraints.

In the next step, which we refer to as the "Virtual Deployment of VN", the NaaS provider designs the components of the VN (virtual nodes containing the VNFs and virtual links), recovers the available network resources and distributes these components into a abstracted view of the infrastructure. We associated the OVF description with this last task to describe the behavior of VN components. Afterward, the OVF++ file is passed to the "Placement process" to finally map the VN description into the physical infrastructure.

The example in Fig. 5 shows some network elements involved in description of the VN offering the network service that supports the VoIP application:

- At <VirtualLinkSection>, *VL1* is characterized by QoS constraints. These constraints include but are not limited to a one-way delay which indicates the maximum expected time for transmission of data (150 ms), a capacity to indicate the minimal available bandwidth (64 Kbps), and placement constraints which indicates the number maximum of nodes to be traversed by the virtual link (2 transition nodes in our example).
- At <VirtualNodeSection>, we describe the constraints at the node level. In the example (at *VNode1*) a packet loss rate is associated to the expected treatment of the node (no more of 2% for packet loss rate).
- The <VirtualQoSNetSection> gives an overall view about the expected behavior of the VN. The network service expected to support the VoIP QoS must respect the threshold values of QoS criterion. In our example

the capacity is associated with the number of requested connections to be supported with this service and the delay refers to the end-to-end delay or response time expected.

```xml
<NetworkSection>
 <VirtualNetworkSection ovf:VirtualNetName="VirtualNetwork VoIP"
 ovf:VirtualNetDescription="Virtual Network to deploy the service of VoIP">
    <VirtualLinkSection ovf:VirtualLinkName="VL 1" >
       <VirtualNodeEdge> VNode 1   </VirtualNodeEdge>
       <VirtualNodeEdge> VNode 2 </VirtualNodeEdge>
       <PLinkConstraints>
          <TransitionNodeNumber> 2 </TransitionNodeNumber>
          <IntermediateNodeList> Vnode12, VNode13 </IntermediateNodeList>
       </PLinkConstraints>
       <QoSLinkConstraints>
          <LinkAvailability>
              <Rate>80%</Rate>
          </LinkAvailability>
           <MaxLinkDelay>
             <Parameter> Latency </Parameter>
             <Rate>maximun 100 ms</Rate>
          </MaxLinkDelay>
          <LinkReliability>
              <Parameter>Bit error rate</Parameter>
             <Rate>maximum 2%</Rate>
          </LinkReliability>
          <LinkCapacity>
            <Parameter> Bandwidth </Parameter>
             <Rate> minimum of 64Kbps</Rate>
          </LinkCapacity>
       </QoSLinkConstraints>
    </VirtualLinkSection >

    <VirtualNodeSection ovf:VirtualNodeName="VNode1"
   ovf:VirtualNodeFunctions="VNF1, VFN22">
       <NodeType> Edge </NodeType>
       <VNIC> VEth1 </VNIC>
       <VirtualLinkList> VL1, VL3 </VirtualLinkList>
        <PNodeConstraints>
          <IsolationNodeConstraint>   VNode1    </IsolationNodeConstraint>
          <AffinityNodeConstraint>         </AffinityNodeConstraint>
          <Anti-AffinityNodeConstraint>  VNode2  </Anti-AffinityNodeConstraint>
        </PNodeConstraints>
       <QoSNodeConstraints>
          <NodeAvailability>
             <Parameter>  Accessibility time </Parameter>
             <Value>  0.5 ms  </Value>
          </NodeAvailability>
          <NodeDelay>
             <Parameter> Delay of traitement for each request </Parameter>
             <Value>   [1ms, 3ms]  </Value>
          </NodeDelay>
          <NodeReliability>
             <Parameter>  Packet Loss rate </Parameter>
             <Value> [0%, 2%] </Value>
          </NodeReliability>
          <NodeCapacity>
             <Parameter> Maximum number of requests </Parameter>
             <Value> 100 </Value>
          </NodeCapacity>
       </QoSNodeConstraints>
    </VirtualNodeSection>

   <VirtualQoSNetSection>
       <NetAvailability>  Availability rate =80%  </NetAvailability>
       <NetDelay>  end-to-end delay = [150ms, 300 ms]   </NetDelay>
       <NetReliability>  Reliability rate ≤  2%</NetReliability>
       <NetCapacity>  100 simultaneous connections </NetCapacity>
   </VirtualQoSNetSection>
  </VirtualNetworkSection>
</NetworkSection>
```

Fig. 5. *Example of OVF VN description supporting a VoIP service*

IV. Conclusion and Future Work

To build network solutions in virtualized environment, operators must consider a generic information model that allows the provisioning, monitoring and management of shared resources. The information model gives the possibility to represent the overall vision of the virtual network to deploy, to place on infrastructure and to monitor.

Representing this information model is needed not only to allow exchange of business among providers but also to enable federated management of the environment. This includes the automation of deployment and management process concerning the virtual resources that are exploited, by using for example SDN and NFV approaches.

Our proposition describes the network elements (nodes, links, networks) involved in VN, as an extension of OVF file. The VN description includes three parts: the virtual link section, the virtual node section and the QoS network constraints section. Included in our description is the QoS and placement constraints that NaaS must to consider for the deployment at each level (network, link and node). In this way NaaS can describe the service offered in accordance with the QoS requirements for the application flows.

Currently, we work in describing the monitoring and orchestration processes that must be considered in order to enable complete automation supporting a dynamic and continous deployment.

References

[1] ONF, software-defined networking: The new norm for networks. White paper. https://www.opennetworking.org.

[2] Network Function Virtualization. White paper. http://portal.etsi.org/NFV/NFV_White_Paper.pdf

[3] ETSI TR 102 805-2 v1.1.1 (2009-11). User group. end-to-end qos management at the network interfaces. Part 2: Control and management planes solution - QoS continuity?

[4] G. A. Lewis, "Role of Standards in Cloud-Computing Interoperability," System Sciences (HICSS), 2013 46th Hawaii International Conference on, Wailea, Maui, HI, 2013, pp. 1652-1661

[5] M. Waschke, "Cloud-specific standards", Cloud Standards: Agreements That Hold Together Clouds. Apress, 2012, pp.289-331.

[6] OVF: https://www.dmtf.org/standards/ovf

[7] OVF v2: http://dmtf.org/sites/default/files/standards/documents/DSP0243_2.1.0.pdf

[8] F. Galan, A. Sampaio, L. Rodero-Merino, et al. "Service Specification in Cloud Environments Based on Extensions to Open Standards". In Proc. of the Fourth International Conference on Communication System software and middleware, june 2009.

[9] L. Rodero-Merino, L.M. Vaqueroa, V. Gilb, et al. "From infrastructure delivery to service management in clouds". Future Generation Computer Systems 26 (2010) 1226-1240.

[10] C. Chapman, W. Emmerich, F. G. Marquez, et al. "Software architecture definition for on-demand cloud provisioning". Cluster Comput (2012) 15:79

[11] Revervoir project: http://ercim-news.ercim.eu/en83/special/reservoir-a-european-cloud-computing-project

[12] S. Garcia-Gomez, M. Jimenez-Ganan, Y. Taher, et al. "Challenges for the comprehensive management of Cloud Services in a PaaS framework". Scalable Computing: Practice and Experience, 13(3), 201-213. 2012.

[13] OpenCloudware project : http://www.opencloudware.org/

[14] ETSI TR 102 805-3 V1.1.1 (2010-04). User Group. End-to-end QoS management at the Network Interfaces. Part 3: QoS informational structure.

[15] I. Ayadi, G. Diaz, N. Simoni. "QoS-based Network Virtualization to Future Networks: An approach based on QoS network constraints". The 4th International Conference Network of Future. 23-25 October, 2013. Pohang, South Korea.

[16] I. Ayadi, N. Simoni, and G. Diaz, "Naas: Qos-aware cloud networking services", in 12th IEEE International Symposium on Network Computing and Applications. 2013, pp. 97-100.