Service Description in the NFV Revolution: Trends, Challenges and a Way Forward

The authors review the proposals of service description in the main initiatives related to the NFV arena. They elaborate on key novel challenges and evaluate how the different proposals solve them. They propose a straw man model of service and resource description addressing these challenges and defining the features that would serve as design directions for future initiatives and updates in this topic.

Jokin Garay, Jon Matias, Juanjo Unzilla, and Eduardo Jacob

Abstract

The telecommunications landscape has been undergoing a major shift in recent years. Initially Software Defined Networking (SDN), and then Network Function Virtualization (NFV) have opened up new ways of looking at the increasingly demanding service provider scenario. The description of the service to be provided will be a key point determining the success in the integration and interoperability of the different proposals. However, the refined understanding of the future scenario and its requirements has recently introduced unique challenges in the path to fully achieve the benefits of the NFV vision. In this paper we review the proposals of service description in the main initiatives related to the NFV arena. Then we elaborate on key novel challenges and evaluate how the different proposals solve them. Finally, we propose a straw man model of service and resource description addressing these challenges and defining the features that would serve as design directions for future initiatives and updates in this topic.

Introduction

Network Function Virtualization (NFV) [1] offers the promise of flexible and efficient service delivery to network operators [2], leveraging the benefits of virtualization technologies to break the strong coupling in current networks between the services offered and the resources supporting them. From the traditional approach, mainly built around hardware appliances, NFV proposes an evolution to consolidate the required functions into industry standard high volume servers, switches, and storage. In the NFV vision, virtualized network functions (VNFs) are dynamically deployed over the infrastructure to create and manage network services.

There are currently different trends pushing for this vision, either specifically launched in the wake of NFV or converging from related areas. One of the foundations underlying the different proposals is the description of the service to be provided, and its evolution will be a key point determining success in the integration and interoperability of the different proposals.

In this paper we review the proposals of service description in the main initiatives related to the NFV arena. Then we elaborate on novel key challenges, appearing as understanding of the future scenario and its requirements refines and evolves, and evaluate how the different proposals solve these challenges. Finally, we propose a straw man model of service and resource description, addressing these challenges and providing design directions for future initiatives and updates in this topic.

The rest of the paper is organized as follows. We cover the approaches for service description. We detail the challenges and how the initiatives meet them. Finally, we describe the proposed model and end the paper with our conclusions

Related Work

In order to provide a short yet comprehensive view of the alternatives in service description, we have focused on the following:

- The definition from the European Telecommunications Standards Institute (ETSI) as initial proponents of NFV.
- The work carried out in the Internet Engineering Task Force (IETF) for the impact of the RFCs from this organization.
- The approach in OpenStack as the de facto open source standard in cloud computing.
- The standards from the Organization for the Advancement of Structured Information Standards (OASIS) for its orientation to interoperability.

Each of these alternatives targets somewhat different problem spaces and defines its own set of requirements, yet the service description is a key part of all of them.

ETSI Network Functions Virtualization

The ETSI NFV Industry Specification Group (ISG) has been leading the way since the publication of the seminal white paper[1] that launched the NFV idea and called for action. Since then the ISG has published the architecture [3] defining the main components and the Management and Orchestration (MANO) framework [4], among many other documentation.

According to ETSI, network service (NS) is the "composition of network functions and defined by its functional and behavioral specification." Following this approach, the NS can be defined as a set of VNFs and/or physical network functions (PNFs), with virtual links (VLs) interconnecting them and one or more virtualized network function forwarding graphs (VNFFGs) describing the topology of the NS. The VNFFG in turn contains network forwarding paths (NFPs) that describe a traffic flow in the NS based on policy decisions. Figure 1(a) shows the elements included in a NS; Fig. 1(b) represents a single NS with multiple VNFFGs and NFPs defined.

The processes of service deployment and overall lifecycle management rely on the information elements describing the NS and its components, both as templates in a service catalog and as records of running instances. Both the

The authors are with the University of the Basque Country UPV/EHU.

[1]Network Functions Virtualisation: Introductory White Paper (2012): http://portal:etsi:org/nfv/nfv white paper:pdf

VNFs and VLs contain the resource requirements that will be used in the orchestration of the NS, together with the VNF-FGs to have the complete connectivity information. In contrast, the PNFs contain the requirements for the attached VLs, as the PNF, by definition, covers its own resource requirements and cannot be deployed in locations other than its own. The VNF also describes its operational behavior requirements for life cycle management.

IETF Service Function Chaining

The IETF Service Function Chaining (SFC) Work Group (WG) focuses on the definition of a new approach to service delivery and operation, built around the idea of an abstract view of the required service functions and the order in which they are to be applied. Currently they have published the problem statement, and the architecture as RFCs [5, 6].

The architecture is defined around three main components that are deployed in an SFC domain and which compose the service, as depicted in Fig. 2 and detailed below. It relies on the SFC encapsulation, which includes metadata to be carried between service functions and the identification of the path to be followed for the service function forwarder.

- Service functions (SFs): Functions responsible for specific treatment of received packets.
- Service function forwarders (SFFs): Responsible for forwarding traffic to/from one or more connected SFs, as well as to other SFFs.
- Service classification functions (SCFs): Used to select which traffic enters an SFC domain. The initial classification determines the SFC that must process the traffic, and subsequent classification can be used to alter the sequence of SFs applied.

As per the WG charter, the focus of SFC is oriented to the operation and composition of the service itself, aiming more for interoperability of the SFs from different vendors than for defining a detailed model of the service components or the processes to manage the service deployment and life cycle.

Inside the parallel organization Internet Research Task Force (IRTF), there is also the Network Function Virtualization Research Group (NFVRG). Currently it is focused on near-term work items that do not have in their scope the definition of a service description.

OpenStack

OpenStack is a major player in the cloud computing technology field. The project aims for simple implementation, massive scalability, and a rich set of features. Initially oriented toward the Infrastructure-as-a-Service model, it was a natural alternative to the infrastructure layer in NFV. In recent years, OpenStack has been extending its features to address several challenges, core to NFV as well as other cloud computing use cases, such as orchestration and advanced networking capabilities.

Heat is the OpenStack component for orchestration and defines the Heat orchestration template (HOT)[2] to describe the infrastructure for a service, called a cloud application in OpenStack. A HOT has three main components.

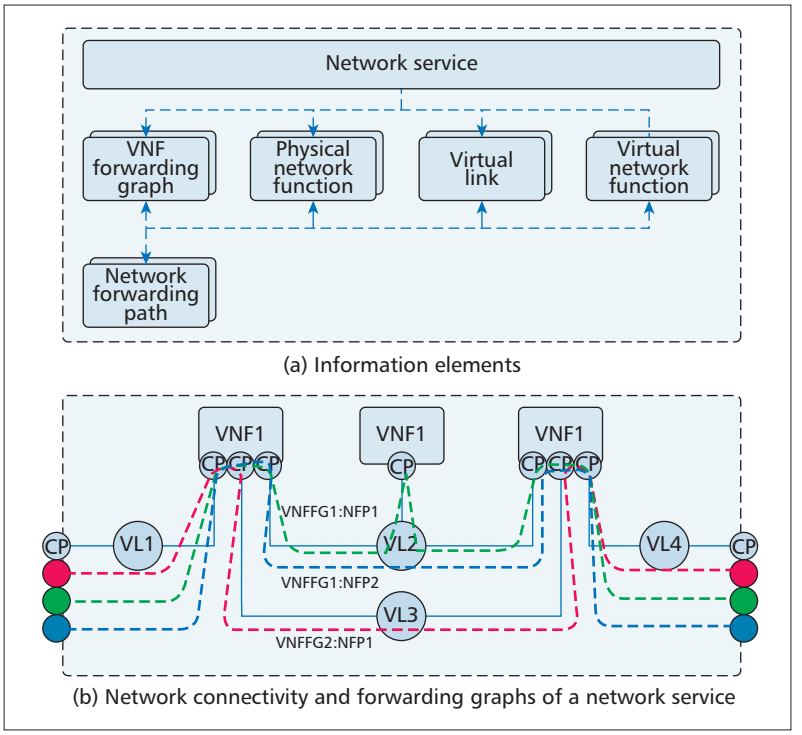

Figure 1. ETSI NFV network service.

- Input parameters to specify the information that has to be provided when the template is instantiated, thus allowing the customization of the instances to be deployed.
- Resources to define the actual resources (in OpenStack compute instances, networks or storage volumes) that will have to be instantiated, allowing also the definition of dependencies between them so the deployment sequence can be controlled.
- Output parameters to define which values from the instantiation process will be fed back to the requester of the deployment.

The application life cycle is also managed from Heat, which will keep track of the resources assigned to the deployed template, although this information is not explicitly included in the information model.

OASIS TOSCA

Topology and Orchestration Specification for Cloud Applications (TOSCA)[3] is an standard from OASIS that targets interoperable deployment and life cycle management of cloud services when the applications are ported over alternative cloud environments.

The core TOSCA specification defines a language and metamodel to describe services, its components, relationships and management procedures. The major elements defining a service are depicted in Fig. 3 and detailed as follows:

- A topology template defines the structure of a service as a set of node templates and relationship templates that together define the topology model as a (not necessarily connected) directed graph.
- Node and relationship templates specify the properties and the operations (via interfaces) available to manipulate the component. Relationships link different nodes and can

[2]Heat orchestration template (HOT) specification: http://docs:openstack:org/developer/heat/template guide/hot spec:html

[3]Topology and Orchestration Specification for Cloud Applications (2013): http://docs:oasis-open:org/tosca/TOSCA/v1:0/os/

have diverse meanings (e.g. a relationship between a "process engine" node and an "application server" node could mean "hosted by").
- Plans define the process models that are used to create and terminate a service as well as to manage a service during its entire lifetime.

The TOSCA NFV profile[4] defines an NFV data model using TOSCA language aligned with the ETSI definition. ETSI network services are represented as TOSCA service templates, whereas node templates are used for the rest of the information elements (including virtual links) and relationships to describe which elements are connected through each link.

Challenges

One of the challenges emerging in the NFV scenario is the consideration of hierarchical orchestration, as there are different aspects leading to this approach. First, we must consider the scalability of the orchestration process. The services will most likely be deployed over infrastructures covering significant geographical scopes, mixing resources deployed over access, aggregation, and core networks, and even reaching resources managed by third parties. Relying on a single orchestrator to handle such a wide variance of resources would significantly hinder the scalability of the process. Also, the different capabilities of the involved infrastructure domains (e.g. operator points of presence, transit networks, data centers, etc.) would be more efficiently used by specialized orchestration processes rather than a common, global orchestration.

A related challenge is the intrinsic multidomain nature of service deployment. Taking the entire, end to end service, few use cases will be confined within the boundaries of a single domain. All the services available for end users outside the domain must consider how the users will access the service. As for the hierarchical orchestration scenarios discussed before, they need to take care of the interconnection of the segments of the service deployed across the different domains. Finally, any infrastructure involving distinct business or administrative domains will also face a similar situation.

Another challenge to be considered is the expected variability of the service across its life cycle, and in the deployment process itself. Network services will be built as a mix and match of the different available network functions, but this process will probably be carried out in multiple steps by different actors. To the original definition from the user, the service provider could transparently add accounting or security related functions, which would ultimately be expanded by functions supporting resiliency or scalability in the infrastructure. Multidomain scenarios, as presented before, could also require interdomain adaptation functions. Finally, internal re-optimization processes, triggered for example by policies, service modification requests, or external changes (e.g. infrastructure updates, auto-scaling) would also require updating the service.

Finally, there is the importance of the associated resource model. The embedding process [7] is one of the hot topics in NFV, and relies on the alignment of the resource requirement information from the service with the resource description of the infrastructure. This premise must be kept even in the face of the aforementioned challenges. Hierarchical orchestration would also imply the necessity of carrying over certain resource assignment information if embedding decisions are taken at the different levels.

ETSI NFV MANO details resource requirements in the NS, but does not extend to the resource model. It is more oriented to a single-layer approach with resource requirements described at the lowest level of detail (CPU, PCIe parameters, etc.). The defined approach for PNFs slightly disrupts the homogeneity of this information, as it also contains resource requirements for the links. The interdomain scenario is not specifically addressed in the NS, which references endpoints that have no information element defined. The different elements detailing connectivity information (VLs, VNF-FGs, and NFPs) add complexity to the additional operations needed when considering hierarchical scenarios or service variability. Finally, resource assignment information is not fully covered. The ETSI model includes resource reservation for the overall NS and the reference of the virtual infrastructure managers that will manage each VL.

IETF SFC focuses mainly on the operation of the service, not how it is described, and treats the SFs as black boxes, considering the chain-

[4]TOSCA Simple Profile for Network Functions Virtualization (2015): http://docs:oasis-open:org/tosca/tosca-nfv/v1:0/csd01/tosca-nfv-v1:0-csd01:pdf

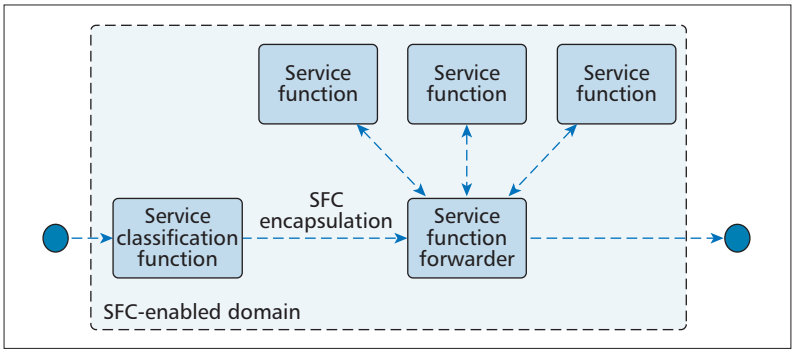

Figure 2. Service function chain architecture.

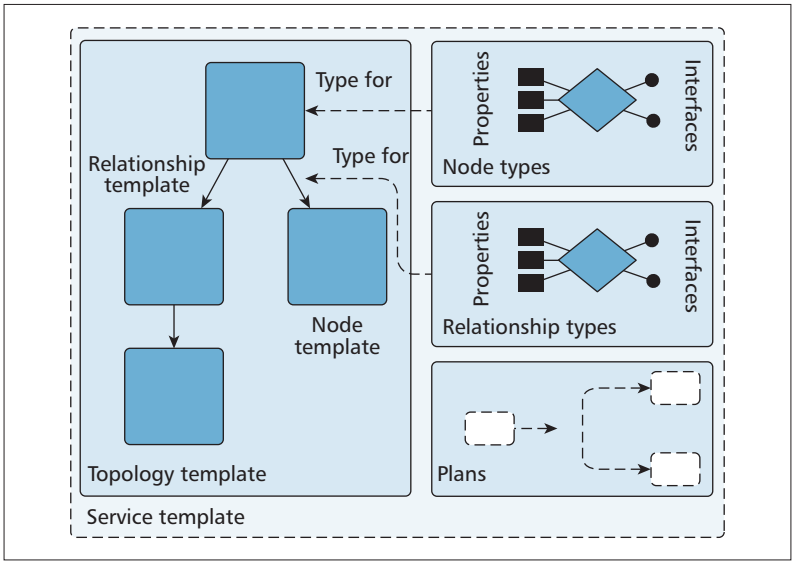

Figure 3. TOSCA service template elements and relations.

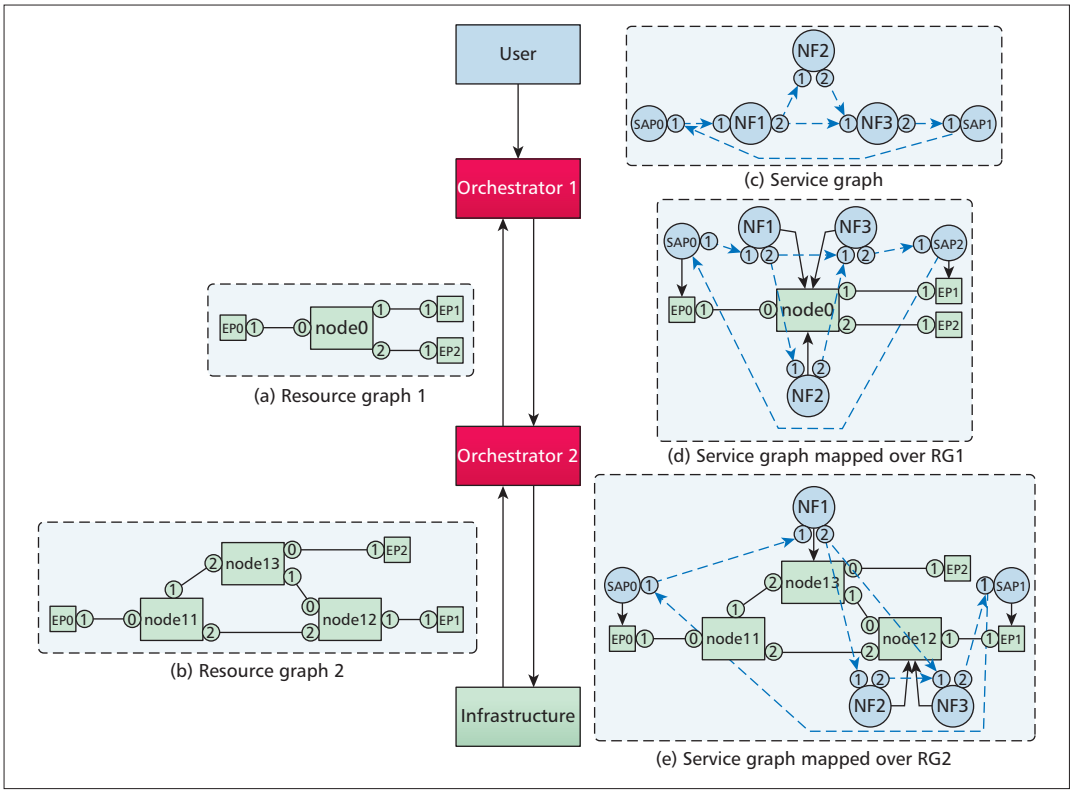

Figure 4. Examples of service graph, resource graph and mapped service graph.

IETF SFC focuses mainly on the operation of the service, not how it is described, and treats the SFs as black boxes, considering the chaining of the SFs and the criteria to invoke them as specific to each domain. One of the targets is for it to be topologically independent from the underlying forwarding topology, thus it does not target resource description.

ing of the SFs and the criteria to invoke them as specific to each domain. One of the targets is for it to be topologically independent from the underlying forwarding topology, thus it does not target resource description. The role of the SFC addresses the requirements of interdomain connections, as it rightly defines the first step of the service is to decide what, in fact, must be processed by the service.

The first consideration about OpenStack stems from its trajectory. Originated in the cloud scenario, it was at its inception more compute-oriented, so its networking capabilities are still not as advanced and lack the required detail (e.g. the connectivity description is more oriented toward networks rather than links). Also, the focus has been more on pure deployment and operation rather than orchestration as defined in the NFV architecture, so its capabilities in this area are still evolving.

Finally, TOSCA, on the one hand, shares with OpenStack its cloud oriented origins, as well as its focus on deployment and operation. On the other hand, the core specification is a metamodel, whereas the detailed specification maps the information elements of ETSI over the TOSCA model, so the same considerations as for ETSI NFV would apply.

Straw-Man Service and Resource Graph Models

One conclusion of the described challenges is that the description of the service and the resources are strongly coupled and will be processed multiple times, so their structure should be closely aligned. Also, this would simplify an explicit mapping of services to resource elements as a result of the embedding. Moreover, the necessity of considering jointly the network function placement and the path calculation in the embedding process [8], as well as the deployment of services including both compute and networking elements [9], points towards a common representation of both. Multiple and variate processing also calls for a careful placement of the information according to its uses, concentrated in the affected elements.

Following these ideas, we propose a service graph (SG) and corresponding resource graph (RG) model described in Fig. 4 and detailed next. It has been designed to provide support for functionalities such as resource orchestration or service deaggregation, on the one hand, and features such as interdomain, scalability, and dynamicity, on the other.

Service Graph

The SG is a directed graph representing the service. It is depicted in Figs. 4(c), 4(d), and 4(e), with round blue nodes joined by dashed arrows, and they contain the following elements:

- Network functions (NFs) are nodes in the SG representing the functions composing the service, including any specification and deployment information, as well as support for life cycle management operations. NFs contain ports for detailing connectivity descriptions (i.e. links connecting the node can be related to a port).
- Service access points (SAPs) are nodes in the SG representing the attachment of the SG to other elements outside the domain. Examples could be "Company branch A, office 1," "All users with Gold service," "Internet," "Service 147685 from Domain x." Optionally,

Information about connectivity or traffic handling policies is associated with SAPs and SLs. Resource requirements are associated with each of the elements, as well as possible placement constraints. Once the embedding process has been carried out, each element will also include the resources assigned from the RG.

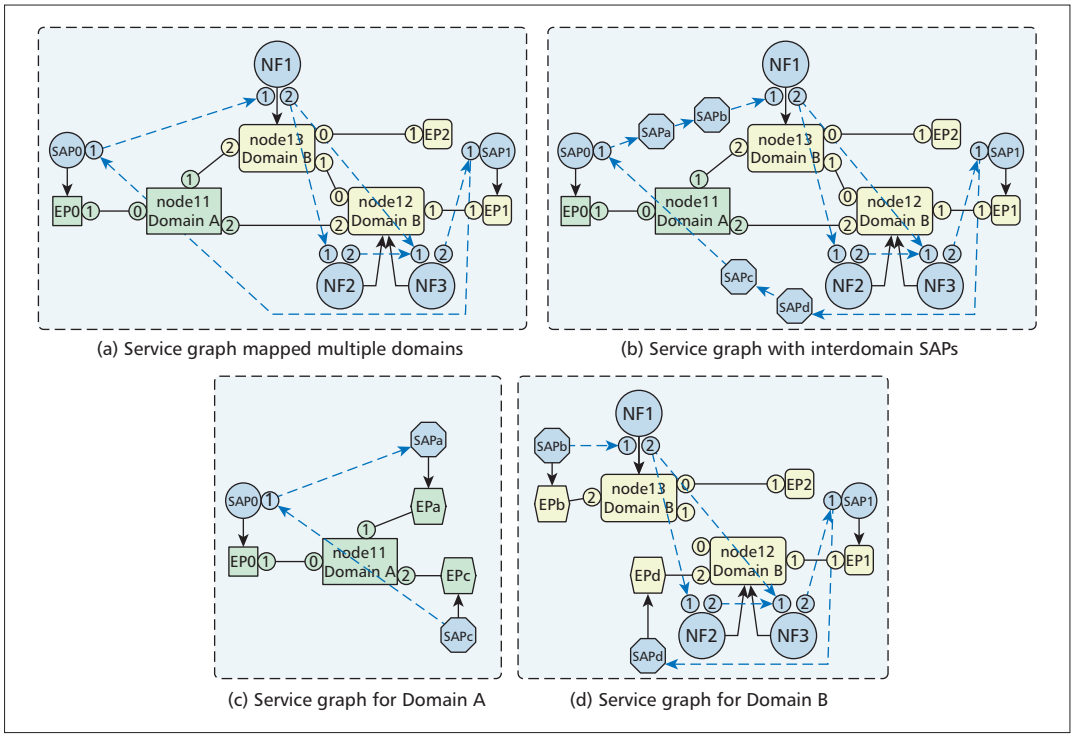

Figure 5. Examples of SG deployed over multiple domains.

SAPs can perform classification functions to select the traffic entering the service.

- Service links (SLs) are edges in the SG representing logical connectivity. Optionally, SLs can perform classification functions to select the traffic entering the link.

Information is attached to the SG or its components based on its scope (i.e. applicable to the whole service or specific elements). Examples of information pertaining the SG would be placement constraints, e.g. privacy related (so resources are not shared among SGs or isolation is guaranteed), geographical, etc., and service level agreements (SLAs) applying to the service, such as service availability, end to end latency, etc. This allows an efficient access to the information, either for retrieving it during the embedding process, updating it during the service lifecycle or splitting it when the service is deployed across several domains.

Resource requirements are associated with each of the elements, as well as possible placement constraints (e.g. geographic or legislative restrictions). Scalability and resiliency requirements are also attached to the corresponding elements (or the graph as a whole), whereas information about connectivity or traffic handling policies is associated only to SAPs and SLs. Once the embedding process has been carried out, each element will also include the resources assigned from the RG, as represented in the mapped SGs in Figs. 4(d) and 4(e) for orchestrators 1 and 2 respectively. This placement of the information follows the approach outlined in the previous paragraph, so any process that must handle information from the service at any point of the lifecycle, can efficiently access it based on the affected elements. Progressive refinement of the service is carried out adding or deaggregating NFs (i.e. substituting single NFs for SGs) and reconfiguring the VLs connecting them.

In hierarchical or multidomain scenarios, the SG is split into subgraphs according to the orchestrator or domain responsible for the assigned resources, as represented in Fig. 5: NFs and SAPs are assigned to the corresponding subgraph and VLs connecting elements in different subgraphs are split in two, terminating now in newly introduced SAPs representing the interdomain connectivity.

These SAPs contain all the required information to configure the interdomain connection in each of the corresponding domain endpoints. As each element contains all the related information, the only processing required in the graph splitting is for the information related to the whole graph, on the one hand, and for the SLs traversing interdomain links, on the other hand.

Resource Graph

The RG describes the resources that will be used to deploy the requested services (potentially a directed graph, but could be simplified as undirected in infrastructures with symmetrical links) It provides a homogeneous representation of the infrastructure, in terms of both capacities and capabilities, and is composed of the following elements, represented in Fig. 4 (isolated in (a) and (b), with a mapped SG in (d) and (e)) with green square nodes joined by lines:

- Infrastructure nodes (INs): Nodes in the RG that, depending on their capabilities, can have NFs deployed on them, provide network connectivity or general traffic processing capabilities.
- End Points (EPs): Nodes in the RG that represent a reference point that defines the attachment of the RG to other elements outside the domain in the context of the infrastructure.

- Infrastructure links (ILs): Edges in the RG that represent the connectivity available between the INs.

The resource model included in the RG mirrors that of the SG and is built around three main abstractions, compute, networking, and storage, described in terms of capacities (which are finite and consumed by the requests) and capabilities (which further characterize the resources and are not consumed by the requests). Examples of capacities are the number of vCPUs an IN can handle or the bandwidth of an IL. Examples of capabilities are redundancy for a link, the delay matrix for a node abstracting a network, or the presence of a hardware accelerator for SSL. Support for PNFs, for example, is included in this model based on a capability describing the type of NFs the IN can handle. The elements in the RG are assigned to domains, which group all the elements managed by the same orchestrator entity and determine the splitting to be performed.

Mirroring the top-down process of the SG, which can be extended and split into several subgraphs, we envision the RG being subject to several aggregations and abstractions in a bottom-up process.

In a hierarchical orchestration scenario, each orchestrator would construct a composed RG based on the input of the different infrastructure domains as input for its embedding process. This same aggregation could, in turn, be exposed to the next level, or be aggregated and presented to the orchestrator above, hiding the details of the inter-domain connections, as shown between orchestrators 1 and 2 in Figs. 4(a) and 4(b). In the top-down process, the details would be added back to support splitting the graph between the different domains, as exemplified in Figs. 5(b), 5(c), and 5(d).

Moreover, in these scenarios we envision the resource description to be different in each level, increasing the abstraction in the higher layers, aligned with the service description. In this vision, the two extremes would be a descriptive/quantitative view where the resource description would be oriented to what the resources are (more meaningful in the lower layers, e.g. one IN can offer x vCPUs) and a functional/qualitative view where the resource description would be oriented to what the resources can do (more meaningful in the upper layers, e.g. one IN can support service X for up to 1,000 users).

In this way, in domains with hierarchical orchestration processes, the RG in the higher-level orchestrators would have a wider scope and abstract away the finer grain details of the underlying resources, whereas the RG in the lower-level orchestrators would have a narrower scope and fine grain detail of the resources, thus fostering scalability. Current models target resource description at the lowest level, but in upper orchestrator levels the RG should follow the level of abstraction in the corresponding SG embedding process.

Deployment Process

The mapping between SG and RG elements is stored in the SG elements (both nodes and edges) with the following considerations:

- NFs are mapped to INs. This assignment is maintained during deaggregation of NFs. For shared NFs, the mapping could be extended to specify an instance of NF running in that IN.
- SAPs are mapped to one or multiple EPs.
- SLs are mapped to an IN internal connection (if both ends of the SL are mapped to the same IN), a single IL, or a sequence of them forming a path. A single SL can be mapped into several ILs (or paths) simultaneously to provide multipath links and offer more possibilities for embedding. In such a case, the embedding process must also define the flow space corresponding to each of the available paths that will be active at the same time (as opposed to resiliency, where only a single link/path would be active).

The necessity of SAPs and EPs to also be part of the embedding process can be seen as excessive. The mapping could be more straightforward and require a lighter embedding process than for NFs or SLs, just picking one element from a list of available EPs or even a direct assignment, such as selecting the EP corresponding to the requesting user. Nevertheless, the replication of the SAP and EP elements allows for a coherent and complete description of the SG and RG and prevents changes in the infrastructure (e.g. adding one more endpoint) impacting the service definition. If an SAP is defined so it is mapped to all endpoints of a certain type (for example, all Wi-Fi hotspots in one area) that would just be a change in the infrastructure that would be handled by the orchestration process (not having to modify the service to follow infrastructure changes and vice versa). Also, depending on the scenario, the SAP to EP mapping could be impacted by the embedding process. For example, if the infrastructure on which the service will be deployed has several possible EPs offering connection to the Internet (as an SAP example), the one selected must be reachable (optimally) from the INs in which the NFs are deployed.

In all the aggregation scenarios, the explicit declaration of the mapping between SG and RG allows for clearly identifying the scope for the re-orchestration based on the mapping already done by the layer above, and the relation between the RG exposed to the layer above and the RG received from the layer below. In the example in Fig. 4, if node12 and node13 represent an aggregation of resources (same as node0 represents the aggregation of node11, node12 and node13), the responsible orchestrator would need to refine the placement, choosing a IN among those aggregated in node13 for NF1, and among those aggregated in node12 for NF2 and NF3 (not necessarily the same IN for both but from the same set). Placement of NF1 in a IN aggregated in node12 would not be allowed as it contravenes the placement done by the upper level orchestrator. Following the proposed model, the boundaries within which each orchestrator can function independently are clearly set, thus reducing the need for interaction between the orchestrators in the management of the service lifecycle.

Finally, the SG and RG model allows for two different ways of supporting resiliency: in the SG

In all the aggregation scenarios, the explicit declaration of the mapping between SG and RG allows clearly identifying the scope for re-orchestration based on the mapping already done by the layer above, and the relation between the RG exposed to the layer above and the RG received from the layer below.

NFV is increasingly being recognized as the future direction in service provider scenarios. Multiple efforts are working to bring all the necessary components to the required level of maturity, so the expected benefits can begin to be reaped from actual deployments.

for resiliency managed by the orchestrator or the RG for resiliency managed by the infrastructure. In the first case, the output of the embedding process determines primary and secondary resources for deployment of the SG, and all of them will be assigned in the SG. In the second case, the elements included in the RG offer resiliency and the specific details are hidden from the orchestrator (e.g. a single link in the RG represent multiple different links in the infrastructure, so if any one of them fails, the infrastructure switches to a backup link without this change being propagated to the orchestrator). The orchestrator would select those resources offering the capability and signal the layer below that such capacity must be used and to what extent. Service elasticity, in a similar way, could be achieved in two different ways: either requesting updates to the SG or embedding the elasticity requirements in the SG. In the former, the SG would be updated to modify the resource requirements of NFs or SLs (for scaling up and down) or to add/remove additional NFs and SLs (for scaling out and in). In the latter, the elements of the SG would contain the triggers for the scalability (e.g. in the form of SLAs or specific metrics) and the sequence of actions to perform, so the orchestrator could handle the elasticity. As the proposed model attaches the information to the corresponding element, in both cases the orchestrator would find all the relevant information grouped and would not need to parse any other element but those added/removed in the first case, and those for which a trigger has been met in the second.

Conclusion

NFV is increasingly being recognized as the future direction in service provider scenarios. Multiple efforts are working to bring all the necessary components to the required level of maturity, so the expected benefits can begin to be reaped from actual deployments. A key piece of the puzzle is the service description, required to allow for the different components to interoperate and address the upcoming challenges, and it is still an open topic. This paper presented some novel challenges to be addressed and contributed with a strawman model addressing them, thus fostering the refinement of the service description models toward their successful completion.

Acknowledgment

This research was partly funded by the Spanish Ministry of Economy and Competitiveness under the "Secure Deployment of Services over SDN and NFV Based Networks" project S&NSEC TEC2013-47960-C4-3-P, and by the European Commission under the FP7 UNIFY (Unifying Cloud and Carrier Networks) project CNECT-ICT-619609. This work has been produced within the Training and Research Unit UFI11/16 supported by the UPV/EHU.

References

[1] B. Han *et al.*, "Network Function Virtualization: Challenges and Opportunities for Innovations," *IEEE Commun. Mag.*, vol. 53, no. 2, Feb 2015, pp. 90–97.
[2] E. Hernandez-Valencia, S. Izzo, and B. Polonsky, "How will NFV/SDN Transform Service Provider OpEx?," *IEEE Network*, vol. 29, no. 3, May/June 2015, pp. 60–67.
[3] ETSI, "Network Functions Virtualisation (NFV); Architectural Framework," ETSI, Tech. Rep. GS NFV 002 v1.2.1, Dec. 2014, accessed: 2015/09/01, available: http://www.etsi.org/deliver/etsi_gs/NFV/001_099/002/01.02.01_60/gs_NFV002v010201p.pdf.
[4] —, "Network Functions Virtualisation (NFV); Management and Orchestration," ETSI, Tech. Rep. GS NFV-MAN 001 V1.1.1, Dec. 2014, accessed: 2015/09/01, available: http://www.etsi.org/deliver/etsi_gs/NFV-MAN/001_099/001/01.01.01_60/gs_NFV-MAN001v010101p.pdf.
[5] P. Quinn and T. Nadeau, "Problem Statement for Service Function Chaining," Internet Requests for Comments, RFC Editor, RFC 7498, Apr. 2015, available: http://www.rfc-editor.org/rfc/rfc7498.txt.
[6] J. Halpern and C. Pignataro, "Service Function Chaining (SFC) Architecture," Internet Requests for Comments, RFC Editor, RFC 7665, Oct. 2015, available: http://www.rfc-editor.org/rfc/rfc7665.txt.
[7] A. Fischer *et al.*, "Virtual Network Embedding: A Survey," *IEEE Commun. Surveys & Tutorials*, vol. 15, no. 4, Feb. 2013, pp. 1888–906.
[8] R. Cohen *et al.*, "Near Optimal Placement of Virtual Network Functions," *2015 IEEE Conf. Comp. Commun. (INFOCOM)*, Apr. 2015, pp. 1346–54.
[9] J. Matias *et al.*, "Toward an SDN-Enabled NFV architecture," *IEEE Commun. Mag.*, vol. 53, no. 4, Apr. 2015, pp. 187–93.

Biographies

Jokin Garay (jokin.garay@ehu.eus) received his B.S. and M.S. degrees in telecommunication engineering in 2003 from UPV/EHU. After a period in the private sector, he returned to the university to work as a researcher and pursue a Ph.D. His research interests include software defined networking, network functions virtualisation, and cloud computing.

Jon Matias (jon.matias@ehu.eus) received his B.S. and M.S. degrees in telecommunication engineering from the University of the Basque Country (UPV/EHU) in 2003, and his Ph.D. from the same university in 2016. He currently works as a researcher in the Communications Engineering Department of the same university. His research interests include software defined networking, network functions virtualization, broadband access networks, and security.

Juanjo Unzilla (juanjo.unzilla@ehu.eus) holds B.S. and M.S. degrees in electrical engineering (1990), and Ph.D. in communications engineering (1999), and is a professor in the Communications Engineering Department at UPV/EHU, where he teaches subjects related to telecommunications networks and services. He is a member of the I2T Research Group, where he participates in several national and European R&D projects. His research interests include SDN and NFV, network security, and techno-economic models for access networks. Among his other interests are models and metrics for knowledge transfer from universities to enterprises.

Eduardo Jacob [SM] (eduardo.jacob@ehu.eus), after spending a few years in the private sector, first as a network manager and later as an R&D project leader, returned to UPV/EHU, where is a professor and leads a research group at the university that is participating in several national and European R&D projects. Among his other interests are industrial applications of SDN and NFV for resiliency, experimental network infrastructures, and cyberphysical systems.