none
L. Geng
Internet-Draft
China Mobile
Intended status: Informational
L. Qiang
Expires: September 6, 2018
Huawei
J. Ordonez
O. Adamuz-Hinojosa
P. Ameigeiras
University of Granada
D. Lopez
Telefonica I+D
L. Contreras
Telefonica

March 5, 2018

COMS Architecture
draft-geng-coms-architecture-02

Abstract

This document defines the overall architecture of a COMS based network slicing system. COMS works on the top level network slice orchestrator which directly communicates with the network slice provider and enables the technology-independent network slice management.

Table of Contents

1. Introduction

Network slicing itself is a new concept triggered by vertical industry, but that doesn't mean new forwarding technology is needed. As an example given by [draft-arkko-arch-virtualization] shows, there are multiple existing technologies could be used for network slicing - VLAN tags are used in an ethernet segment, MPLS or VPNs across the domain. If the storage and computing resources are considered, there will be more available technologies (e.g., SFC).

Let's follow IETF's routine and image what will happen from the bottom-up view. At first, existing technologies evolve toward network slicing at forwarding plane in their own scopes. Then slice management related functions will be patched at management/control planes. When a network slice is going to be deployed inside a domain, one of implementation technology will be selected, and the

NS
provider directly operates on the management plane of this selected technology. For example, If VPN is selected as the implementation technology, then a network slice is a VPN for the NS provider in this
domain. While if SFC is selected in other domain, then a network slice is a SFC for NS provider. What will happen if a network slice across both VPN and SFC domains? There is no uniform management manner in this case.

Then try to consider from the top-down view. There is no doubt that slicing requirement is generated from NS tenant. When a NS tenant request for NS service, normally he will not specify which implementation technology should be used. Similarly, when the tenant operates/manages his purchased slice, he doesn't want to care about the technical details.

We can easily observe that bottom-up and top-down approaches will eventually converge on a technology-independent common management plane, that is exactly what COMS (Common Operation and Management on network Slices) doing.

This document will explain how COMS works, and define the architecture of COMS. Architecture discussed in this document is assumed to be used only inside Transport Network region, and the end-to-end network slice/slicing also just refers to the slice/slicing across multiple TN domains in this document.

2. Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

Other network slicing related words used in this document are interpreted as description in [COMS-PS].

Notations used in this document are interpreted as follows:

T(x->y): end-to-end delay from x to y;

B(x->y): bandwidth from x to y;

S(x): storage space of x.

3. Overall Architecture

This section provides the overall architecture for a COMS based network slicing system as shown in Figure 1. If multiple such kind of systems deployed in different domains, these systems may stitches together through the method discussed in [Stitching-Management] and [Stitching-Data]. COMS works on the top network orchestrator inside Transport Network region, which directly receives the network slice service profile, operation and management requests for network slices. Based on received information, the network orchestrator will select the most appropriate implementation technologies, and map the technology independent requests into the technology specific

configuration information that will be sent to the corresponding network slice controller/orchestrator downwards.

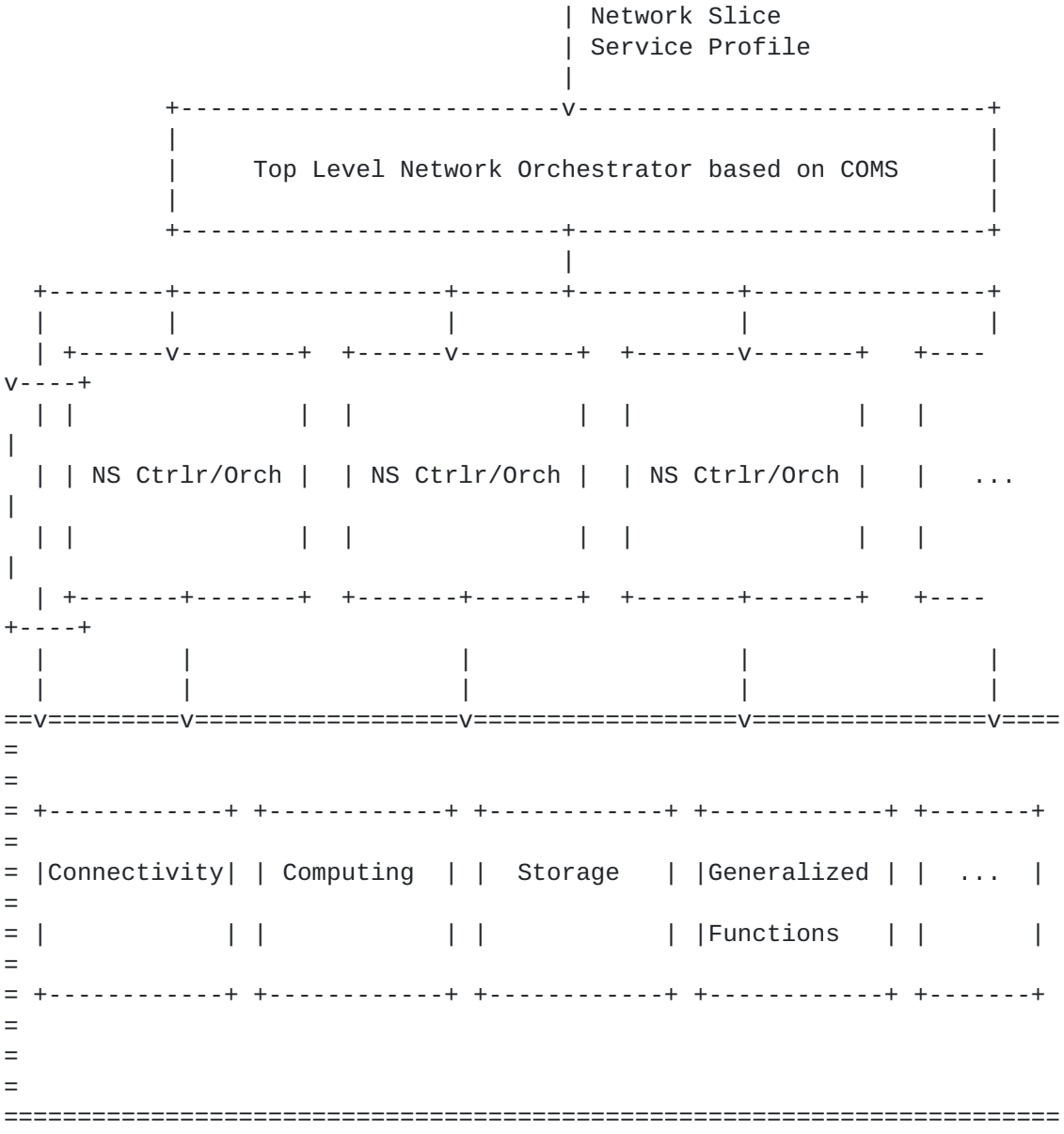


Figure 1: Overall Architecture of COMS

4. Advanced Architecture

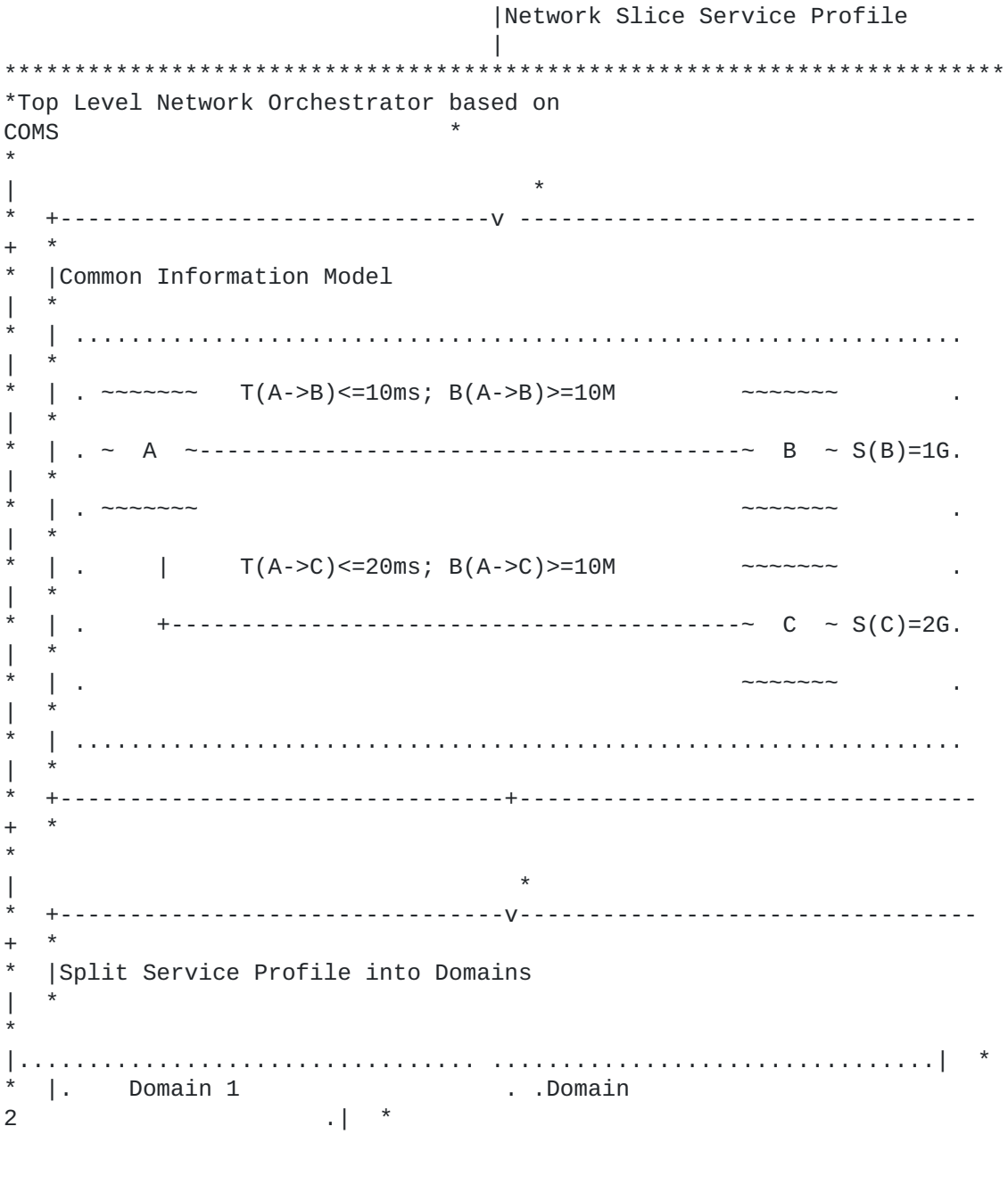
This section discusses the detailed architecture of a COMS based network slicing system through an example shown in Figure 2. We do not intend to design the inner framework of the top level network slicing orchestrator but to explain how COMS works. Four components insides the top level network orchestrator are logical components that could be converged sometimes.

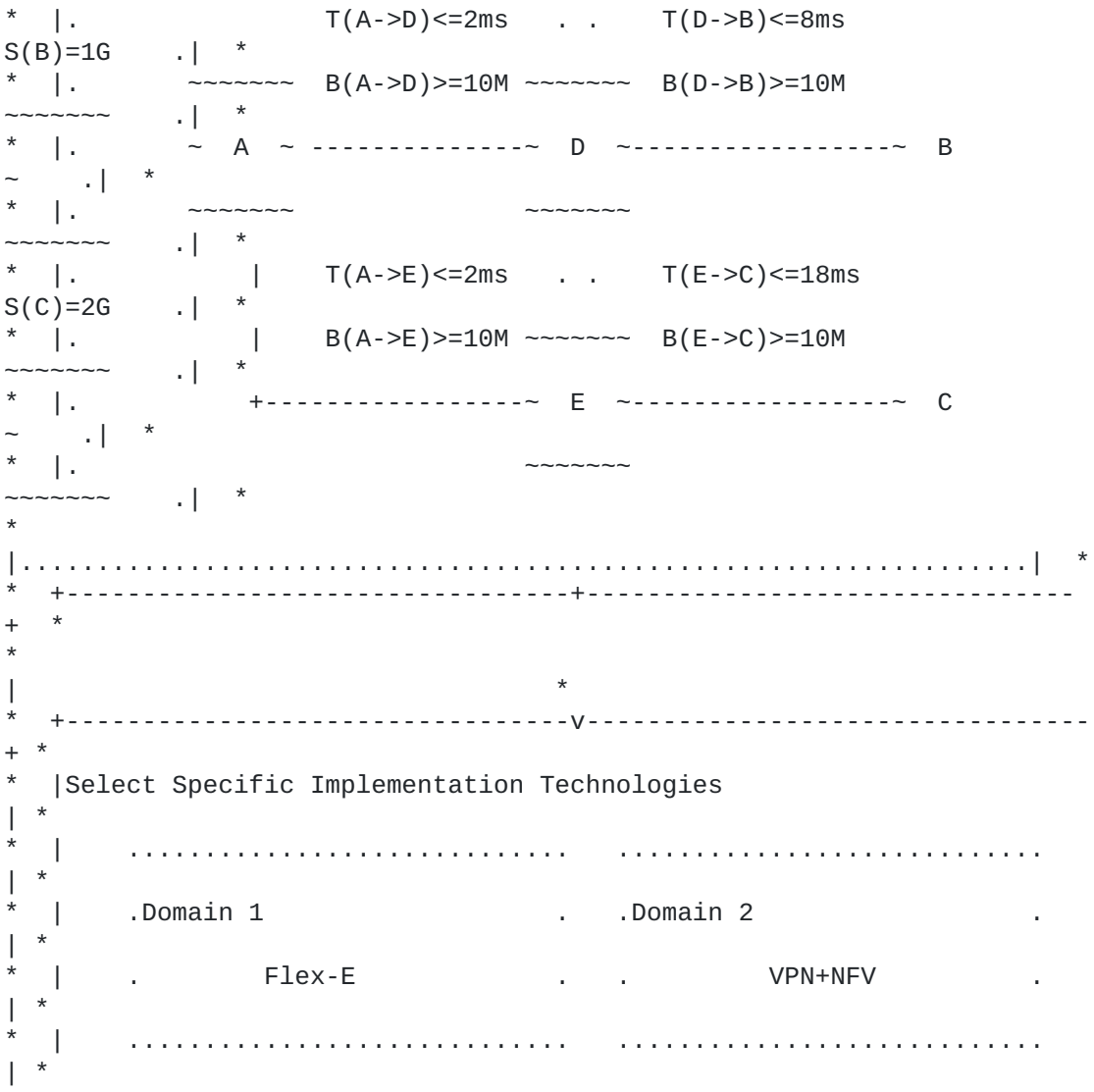
- Common Information Model: can be understood as the template, according to which the received network slice service profile is translated.

- Split Service Profile into Domains: the end-to-end service profile is split into the service profiles inside different domains.

o Select Specific Implementation Technologies: there may be multiple available implementation technologies inside a domain, select the most appropriate one according to the service profile. As Figure 2's example shows, since the end-to-end delay in Domain 1 is very small, the Flex-E will be selected. While in Domain 2 two storage units are required, the NFV technology will be selected.

o Map to Selected Technologies: necessary mapping to the controller/ orchestrator of selected technologies.




T(A->D)<=2ms
T(D->B)<=8ms
S(B)=1G
B(A->D)>=10M
B(D->B)>=10M
A
D
B
T(A->E)<=2ms
T(E->C)<=18ms
S(C)=2G
B(A->E)>=10M
B(E->C)>=10M
E
C
Select Specific Implementation Technologies
Domain 1
Flex-E
Domain 2
VPN+NFV

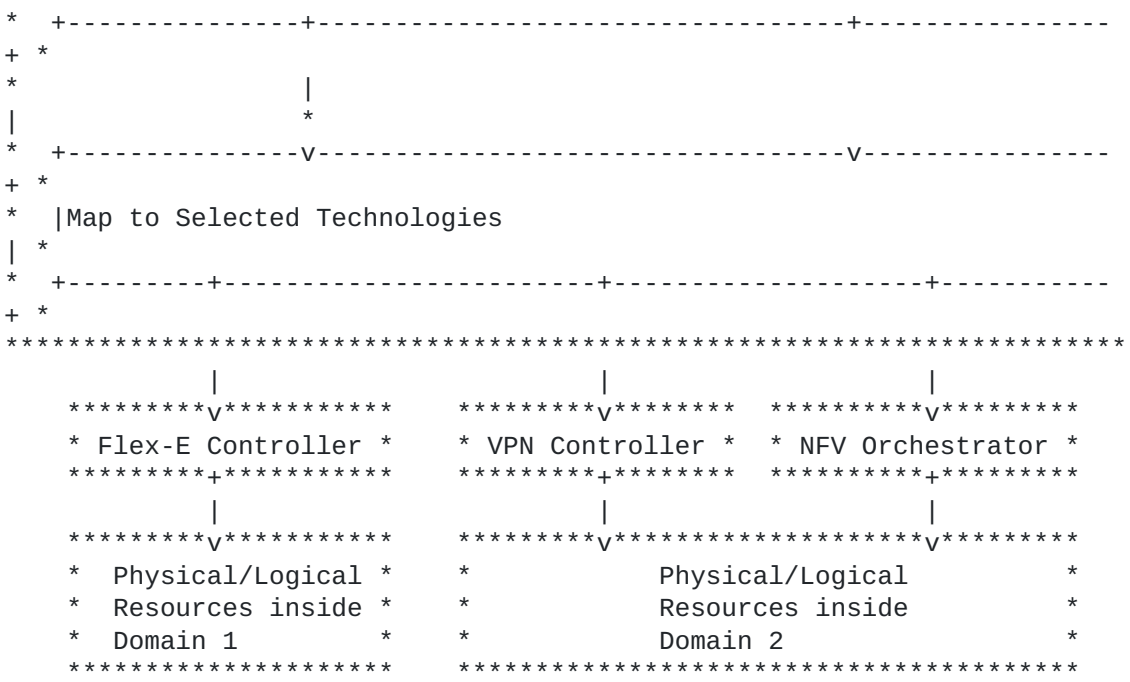


Figure 2: Advanced Architecture of COMS

5. Integration with NFV

This section details the integration of the NFV framework [NFV-MANO] in the COMS architecture.

Network slice providers aim to accommodate a myriad of use cases and application scenarios from multiple tenants over a common network infrastructure. To that end, network slice providers build up multiple network slice instances (NSIs), each customized to serve the specific service demands of a particular tenant. An NSI is a logical self-contained network instance that network slice providers offer to a tenant, and that a tenant can consume. Although NSIs may span across multiple network segments (e.g., RAN, transport, and core network), this document only considers the transport network domain.

NFV may play a key role in network slicing, enabling its realization in a cost-efficient manner. Using the flexibility and virtualization capabilities that the NFV framework brings, a network slice provider can create and operate multiple NSIs over a common shared network infrastructure with isolation guarantees in terms of performance, management, security, and privacy [Ordonez-Network-Slicing]. To provide the tenant with the required performance and functionality, an NSI includes one or more network services, each consisting of a chained set of compassable atomic units called virtualized network

functions (VNFs). These VNFs are software-based implementations of network functions that rely on computing, storage, and connectivity resources for their execution and communication. To simultaneously serve the requirements of multiple NSIs, the network slice provider makes use of the resources that are at its disposal, and efficiently

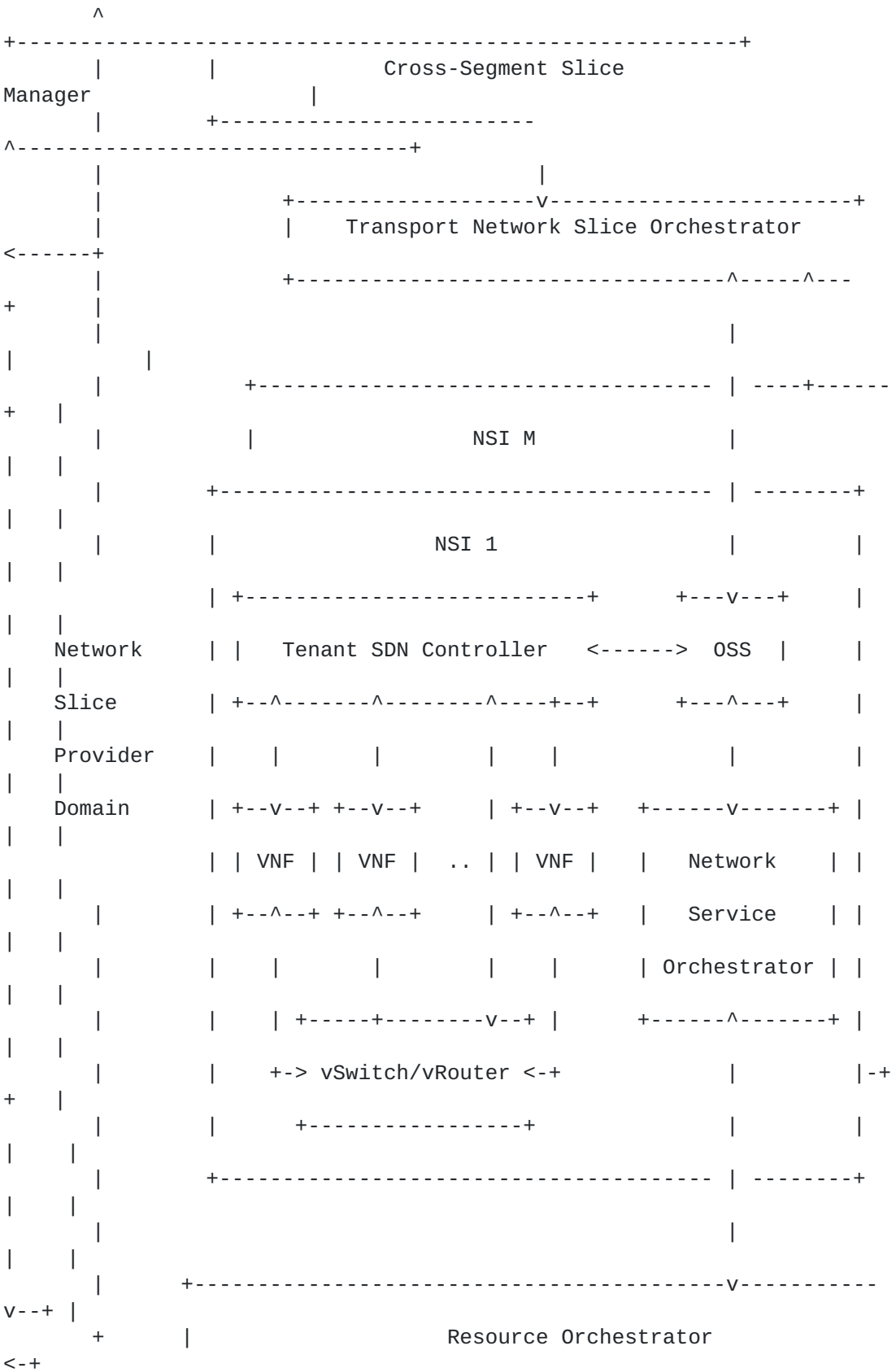
orchestrate them across NSIs. Although the network slice provider can own these resources, we consider it rents them from one or more infrastructure owners following the Infrastructure-as-a-Service (IaaS) paradigm. In this case, the network slice provider takes the role of a network infrastructure tenant. Note that each of the three actors presented here (network infrastructure owner, network slice provider, and network slice tenant) defines a different administrative domain.

The NSIs shown in Figure 3 run parallel on a common shared transport network infrastructure. The transport network infrastructure consists of connectivity resources that may span across multiple administrative domains (i.e., different network infrastructure owners). These resources include WAN nodes and links providing reachability across geographically remote data centers, where the VNFs from different NSIs run. In particular, they connect together the network connectivity endpoints (e.g., gateways) of those data centers.

To simultaneously serve the connectivity needs of the NSIs using resources within its administrative domain, each network infrastructure owner has a WAN Infrastructure Manager (WIM). The WIM is a NFV functional block that performs control-management actions over the underlying connectivity resources to deploy and operate a number of L2/L3 virtual topologies with different levels of abstractions. To enforces the connectivity required by an NSI, the WIM abstracts the resources under its management, and creates a customized virtual topology that logically connects the data centers hosting the NSI's VNFs. The resources of each data center are managed with a Virtual Infrastructure Manager (VIM). This NFV functional block play a similar role to the WIM, but extending their management domain to computing and storage resources.

The transport network resources, managed by the underlying network infrastructure owners using their WIMs/VIMs are delivered to the network slice provider logically placed on top of them. The network slice provider makes use of these resources to deploy and operate the NSIs that are under its management. For this end, it may rely on the NFV Orchestrator (NFVO) functionality. According to the NFV framework, NFVO is a functional block with two well-defined functionalities: resource orchestration and network service orchestration. The former focuses on orchestrating network infrastructure resources across multiple VIMs/WIMs, while the latter performs lifecycle management operations (e.g., instantiation, scaling, updating, termination, etc.) over the network service(s) built using those resources. Due to the different scope of these two set of functions, the NFVO may be logically split into two

functional
   blocks: Resource Orchestrator and Network Service Orchestrator.


Cross-Segment Slice Manager
Transport Network Slice Orchestrator
NSI M
NSI 1
Tenant SDN Controller
OSS
Network Slice Provider Domain
VNF
VNF
VNF
Network Service Orchestrator
vSwitch/vRouter
Resource Orchestrator

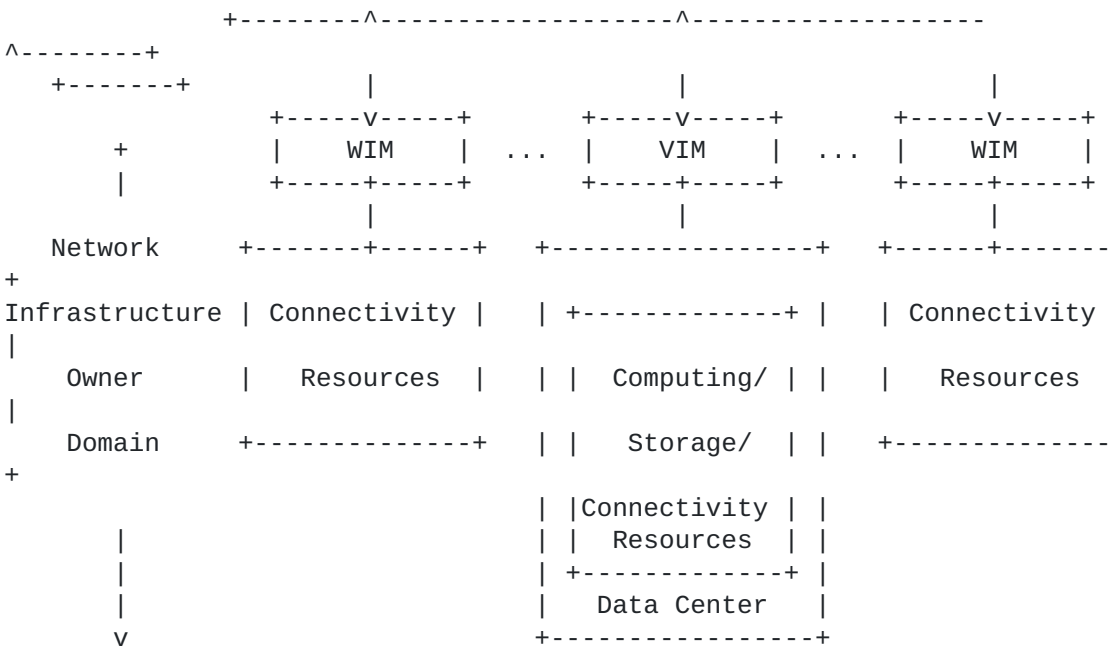


Figure 3: Integration of NFV framework in COMS architecture

To orchestrate the resources that are at its disposal (those provided by the underlying network infrastructure owners), the network slice provider has a single Resource Orchestrator. The main role of the

Resource Orchestrator is to dispatch this finite set of resources across the operative NSIs in an optimal way, with the aim of simultaneously satisfying their (potentially diverging) performance requirements. To bring multiplexing gains and cost savings in this task, the Resource Orchestrator may take advantage of resource sharing. Resource sharing introduces flexibility and efficiency in slice provisioning, as network slice provider's resources can be dynamically allocated and released across NSIs according to the time-varying resource requirements that their tenants impose. This approach requires an adequate resource management framework for the Resource Orchestrator that carefully finds an optimal solution, enabling resource sharing among NSIs when necessary, while preserving their performance isolation.

As shown in Figure 3, each of the operative NSIs serving a network slice tenant comprises a tenant SDN controller, a Network Service Orchestrator, and an Operation Support System (OSS). On the one hand, the tenant SDN controller configures the VNFs at application level, and chains them to dynamically build up the network service(s) that are required in the NSI. For VNF configuration management, the tenant SDN controller uses southbound configuration protocols such as NETCONF. For VNF chaining management, it leverages the networking capabilities provided by virtual switches/routers, sending them appropriate forwarding instructions using southbound control protocols such as OpenFlow. On the other hand, the Network Service Orchestrator manages the lifecycle of the network service(s). Finally, the OSS performs the intra-NSI management, bridging the gap between the Network Service Orchestrator and the tenant SDN controller, and coordinating their operations and management data. The OSS is also the entry point of the NSI, providing management capability exposure to external blocks. By way of example, the network slice tenant can use the OSS to gain access to the NSI and operate it at its convenience.

The description given above focuses on run-time phase, assuming the NSIs are operative, and omitting the deployment steps referred in Section 1. To trigger the deployment of a network slice, the network slice provider needs other functional blocks. These functional blocks include a Cross-Segment Slice Manager, and one or more Network Slice Domain Orchestrators. The Cross-Segment Slice Manager receives a network slice service profile from the tenant. This profile contains the (end-to-end) slice requirements. The Cross-Segment Slice Manager decompose these requirements into one or more network slice domain slice requirements, and send them to the respective Network Slice Domain Orchestrators (e.g., RAN Slice Orchestrator,

Transport Network Slice Orchestrator, Core Network Slice Orchestrator). Since the architecture discussed in this document is assumed to be inside the transport network domain, we only consider

the Network Slice Transport Orchestrator. The Network Slice Transport Orchestrator uses the network slice transport requirements to determine which VNFs and network service(s) are required, and what are their resource requirements. Once checked the Resource Orchestrator can provision them, the steps for deploying the slice begin. First, the Resource Orchestrator creates the resource slice. Then, the OSS takes over the resource slice and configures it, resulting in a networking slice. Finally, the OSS (assisted by the Network Service Orchestrator and the Tenant SDN controller), instantiates one or more network services (and their constituent VNFs) over this networking slice to realize a service slice, making it usable for the network slice tenant.

6. Security Considerations

There is no security problems introduced by this document.

7. IANA Considerations

There is no IANA action required by this document.

8. Acknowledgements

TBD

9. Informative References

[COMS-PS] "Problem Statement of Supervised Heterogeneous Network Slicing", <https://datatracker.ietf.org/doc/draft-geng-coms-problem-statement/>.

[draft-arkko-arch-virtualization] "Considerations on Network Virtualization and Slicing", <https://tools.ietf.org/html/draft-arkko-arch-virtualization-00>.

[I-D.boucadair-connectivity-provisioning-protocol] Boucadair, M., Jacquenet, C., Zhang, D., and P. Georgatsos, "Connectivity Provisioning Negotiation Protocol (CPNP)", draft-boucadair-connectivity-provisioning-protocol-15 (work in progress), December 2017.

[NFV-MANO] ETSI GS NFV-MAN 001, "Network Functions Virtualisation (NFV); Virtual Network Functions Architecture", V1.1.1, December 2014.

[Ordonez-Network-Slicing]
Ordonez-Lucena, J., Ameigeiras, P., Lopez, D., Ramos-Munoz, J., Lorca, J., and J. Folgueira, "Network Slicing for 5G with SDN/NFV: Concepts, Architectures, and Challenges", IEEE Communications Magazine, vol. 55, no. 5, pp. 80-87, May 2017.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/info/rfc2119>.

[RFC5440] Vasseur, JP., Ed. and JL. Le Roux, Ed., "Path Computation Element (PCE) Communication Protocol (PCEP)", RFC 5440, DOI 10.17487/RFC5440, March 2009, <https://www.rfc-editor.org/info/rfc5440>.

[Stitching-Data]
"Gateway Function for Network Slicing", <https://datatracker.ietf.org/doc/draft-homma-coms-slice-gateway/>.

[Stitching-Management]
"Interconnecting (or Stitching) Network Slice Subnets", <https://datatracker.ietf.org/doc/draft-defoy-coms-subnet-interconnection/>.

Authors' Addresses

Liang Geng
China Mobile

Email: gengliang@chinamobile.com

Li Qiang
Huawei

Email: qiangli3@huawei.com

Jose Ordonez Lucena
University of Granada

Email: jordonez@ugr.es

Oscar Adamuz Hinojosa
University of Granada

Email: oadamuz@ugr.es

Pablo Ameigeiras
University of Granada

Email: pameigeiras@ugr.es

Diego Lopez
Telefonica I+D

Email: diego.r.lopez@telefonica.com

Luis Miguel Contreras Murillo
Telefonica

Email: luismiguel.contrerasmurillo@telefonica.com