Orchestration of IT/Cloud and Networks: From Inter-DC Interconnection to SDN/NFV 5G Services

(invited paper)

R. Casellas, R. Muñoz, R. Vilalta, R.Martínez
CTTC – Communication Networks Division
Av. Carl Friedrich Gauss, 7, 08860 Castelldefels, Spain
ramon.casellas@cttc.es

***Abstract*— The so-called 5G networks promise to be the foundations for the deployment of advanced services, conceived around the joint allocation and use of heterogeneous resources, including network, computing and storage. Resources are placed on remote locations constrained by the different service requirements, resulting in cloud infrastructures (as pool of resources) that need to be interconnected. The automation of the provisioning of such services relies on a generalized orchestration, defined as to the coherent coordination of heterogeneous systems, applied to common cases such as involving heterogeneous network domains in terms of control or data plane technologies, or cloud and network resources. Although cloud-computing platforms do take into account the need to interconnect remote virtual machine instances, mostly rely on managing L2 overlays over L3 (IP). The integration with transport networks is still not fully achieved, including leveraging the advances in software defined networks and transmission. We start with an overview of network orchestration, considering different models; we extend them to take into account cloud management while mentioning relevant existing initiatives and conclude with the NFV architecture.**

Keywords*— *Inter-DC, Cloud and Network Orchestration, Control Plane, Over-arching control, GMPLS / PCE.

I. Introduction

The so-called *5G networks* promise to be the foundations for the deployment of advanced services that go beyond basic data connectivity and that are increasingly conceived around the joint allocation and use of heterogeneous resources, combining networking functions (transmission, switching, forwarding) and IT functions (computing, storage, processing). The fact that new architectural solutions are needed to cope with the huge increase in traffic and address new service requirements in a cost-effective way has been known for years, but, until recently, the operations of cloud infrastructures and of data networks were decoupled to a large degree.

A direct consequence of the need to support such new services is that the existing, underlying communications infrastructure needs to evolve, grow and adapt. It combines heterogeneous technologies, both at the data and control planes (including wireless such as 4G / 5G / mmWave, etc. and wired as flexgrid, NG-PON, etc.), and deploying compute resources supporting data processing at distributed geographical locations (see Fig. 1). The automation of the service provisioning implies the dynamic allocation and management across different operative domains: the networking community is adopting cloud computing to answer to emerging requirements and use cases, whereas the cloud community is building geographically distributed computing infrastructure requiring inter-connection.

In the context of this paper, *orchestration* refers to the coherent coordination of heterogeneous systems, allocating diverse resources and composing functions to offer end-user services. It involves automating processes and using or invoking the programming interfaces of subordinate or external systems, platforms and infrastructures, typically with transactional semantics and using high-level frameworks, constructs and languages. Common uses of orchestration involve, for example, heterogeneous network domains (in terms of control and/or data plane technologies), known as *network orchestration,* or cloud and network resources (*joint orchestration*), exemplified by the emerging use case of inter-connection of segregated data centers, composed of multiple geographically disperse sites or locations.

A data-center (DC) could be defined as a centralized resource pool for the storage, management, processing and distribution of data and information organized pertaining to a particular business or administrative domain. It is commonly understood as a (large) group of networked computer servers. Availability and performance requirements have been constraining the topologies that are commonly deployed, considering inter- and intra-site traffic patterns, and power and cooling distribution, with or without redundant components. Until recently, cloud management frameworks have been built assuming, in most cases, a centralized location and servers within the DC are connected using mainly L 2 Ethernet technologies within a LAN. However, new trends have been emerging that challenge these somehow simplistic definitions: on the one hand, constraints and service requirements such as latency or response time are limiting the feasible locations for the deployment of computing and processing capabilities (e.g. near the end user), justifying the need to the so-called segregated or distributed cloud infrastructures (including DCs). On the other hand, the size of the *component* sites can also be arbitrarily small while still adopting the same cloud management approaches.

From the point of view of control and management aspects, there is a clear need to operate distributed infrastructures that are disperse in different geographic locations, and the dynamic provisioning of services implies more and more the integration with the WAN and long haul transport segment.

The paper is structured as follows: after an overview of emerging use cases, Section III introduces the concept of

978-1-4673-9988-3/16/$31.00 ©2016 IEEE

network orchestration. Section IV extends with the unified orchestration of networks and clouds and finally, Section V covers the link with Network Function Virtualization (NFV).

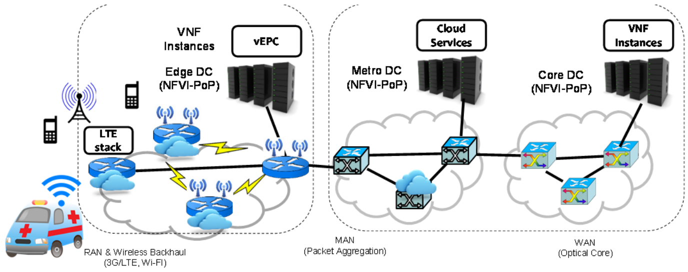

Fig. 1. Heterogeneous Infrastructure for the deployment of 5G services

II. EMERGING USE CASES, DRIVERS AND REQUIREMENTS

As mentioned, the canonical, representative use case is the interconnection of distributed DCs and the dynamic and constrained allocation of heterogeneous resources. This generic use case shares common aspects with the terms *Fog Computing* or *Mobile Edge Computing (MEC)*. The former refers to extending cloud computing to the edge of the network, where end-users or edge devices are provisioned with substantial resources emphasizing proximity and local resource pooling. The latter is an ETSI initiative [1] targeting the deployment of cloud-computing capabilities and an IT service environment at the edge of the mobile network (with ultra-low latency, high bandwidth and real-time access to radio network information).

Generically speaking, the interconnection of remote DC locations (Inter-DC service provisioning) involves different network segments (i.e., local, access, aggregation, metro and core networks) each having different traits. For example, within the DC, traffic is characterized by highly dynamic flows, relying on combined L2 and L3 technologies. As we will see in the next section, the centralized control of such flows along with the potential flexibility that it enables is one of the main drivers for the adoption of SDN principles. That said, there is a growth in the adoption of the optical technology within the DC (with optical interconnects and Software Defined Optical Transmission), constituting a flexible data plane supporting elastic, dynamic and reconfigurable optical networks with scalable, power/cost efficient, and reliable technologies, whose remote programmability is still not fully exploited.

The long haul core transport may depend on one or more service providers, commonly with peering agreements. Flows are less dynamic (although the increase of Inter-DC traffic is changing this). While this segment is more constrained by existing technology and return-on-investment of current deployments, it can benefit from a control plane supporting, notably, error-free multi-domain provisioning and distributed recovery. Transport networks within providers are multi-layer, based e.g., on IP/MPLS over Optical Transport Networks (OTN), fundamental in view of latency, jitter and bandwidth requirements. This multi-layer and multi-domain aspect of transport network is also requiring the coordinated control (e.g. dynamic IP link provisioning over optical connections).

Finally, in the most generic cases, the environment around the interconnection of DCs is characterized by having different stakeholders, and infrastructure providers. *Virtual Network Providers* aggregate infrastructures from different *Physical Infrastructure Providers*, so *Virtual Network Operators* can offer their services over a shared infrastructure. In this sense, a most important driver for the development of control and management architectures is the ability to support transport network slicing (including partitioning and aggregation) and virtualization, along with the concept of multi-tenancy.

III. OVERARCHING CONTROL AND NETWORK ORCHESTRATION

From the networking perspective, the inter-connection of DC geographic locations involves the overarching control and network orchestration of multiple heterogeneous network resources, arranged in technological layers and under different control and management approaches. In this section, we use the term domains, loosely defined as a collection of resources (i.e., links and nodes) for the transmission and switching of data flows and the transport of client signals, grouped into homogeneous sets with the same technological layer and control architecture.

A. ASON / GMPLS control plane

The ASON/GMPLS architecture has since long included support for multi-domain and multi-layer networking. While the ASON/GMPLS architecture works well in specific scenarios, it relies on a specific layered network model and targets a clear concrete service. It is limited into what kind of resources it can manage and does not address how a network domain can be integrated in a wider SDN-based control and management architecture (including, for example, a better integration with the provider operation and business support systems - OSS/BSS). Consequently, the ASON/GMPLS architecture remains as a viable, mature approach for the provisioning of data channels benefiting of mature protocols, existing implementations and well-understood procedures. The adoption of the Path Computation Element (PCE) [2] and the Application Based Network Operation (ABNO) [3] architectures is facilitating the adoption of SDN principles in the scope of transport networks while allowing a progressive migration.

B. SDN and OpenFlow control plane

On the other hand, the benefits of a more centralized approach, in which business and application logic can be easily integrated into a control layer in view of supporting verticals and services are also well known. The potential of deploying systems with modular, open and standard interfaces for remote device programmability is fundamental for the orchestration of heterogeneous systems. From the networking perspective, SDN and OpenFlow as a particular interface and protocol have been adopted in a large number of deployments, demonstrating their flexibility. In particular, one of the OpenFlow strengths is to have identified a basic common hardware model for a data switch (with emphasis in packet switching) and, by virtue of the formal specification of tables, flows, and actions providing a flexible and extensible mechanism to automate networking decisions. The advantage of adopting SDN/OpenFlow is not only to reproduce the basic behaviour of IP, Ethernet or MPLS networks, focusing mainly on the packet forwarding function, but also to enable and automate, from a centralized operation point, the management and configuration of advanced networking rules and policies. Such rules related to e.g., VLAN to tunnel mapping, traffic separation, security groups, dynamic instantiation of functions, etc., way more difficult to address in

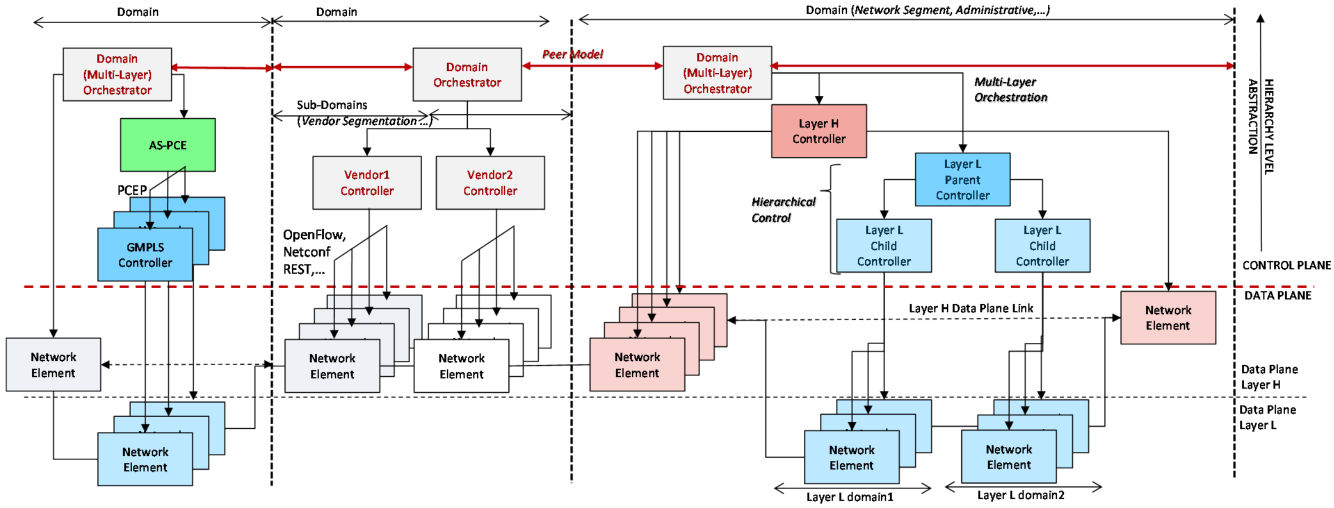

Fig. 2. Generalized view of Network Orchestration for multi-domain, multi-layer and multi-vendor scenarios, showing hierarchical and peer models

a purely distributed manner. This enables new models of network virtualization and multi-tenancy and exploiting software implementations of network functions (L2 switches, L3 routers, networking services, etc.) in order to offer overlay networks for tenants over an infrastructure of interconnected DC. In all, there is a need to provision end-to-end service across multiple technology domains, addressing vendor islands and segmentation. Scalable solutions will need to rely on abstraction and a hybrid combination of centralized and distributed entities. Abstraction refers to the selection of an entity relevant characteristics, based on targeted functionality and scalability, and referring to a network domain it involves selected network topological information, referring to the synthesizing of reported TE information for a set of elements or networks (up to providing a reachability and connectivity matrix).

C. *The unique controller approach*

A single SDN controller with full topology visibility can be designed to control multiple data plane technologies, but such an approach may have important shortcomings: larger domains need to be engineering carefully to overcome scalability issues. Having a single controller deployed for multiple data plane technologies (by means of software extensions, plugins, or an all-encompassing generalized protocol) is not straightforward. A single controller may be only possible if a common information model for all layers/technologies is designed, or its use scoped to a reduced number of mature technologies (e.g., packet layer such as Ethernet or IP/MPLS with an OTN circuit-switching layer). In general, the diversity and heterogeneity of the relevant involved technologies calls for a more segmented orchestration.

D. *Orchestration of multiple controllers*

In this deployment model, a (possibly redundant, high-available) SDN controller is deployed for a given (technology, layer or vendor) domain, the whole network system orchestrated by a "parent" controller. For example, the parent controller or *orchestrator* may be responsible for the selection of domains to be traversed for a new provisioned service [5]. Such domain selection is based on high-level, abstracted knowledge of intra- and inter-domain connectivity and topology. The topology abstraction, needed due to scalability and confidentiality reasons, is based on a selection of relevant attributes and represented as allowed by the domain internal policy. Per-domain controllers are responsible for the actual provisioning in their respective domains. This orchestration can be based on the use of consolidated architectures and protocols such as PCEP and BGP-LS [5][6]. The ONF is working on the specification of the transport API (T-API) functional requirements, and the new version of the architecture defines the interfaces A-CPI (unified between the control plane and the application plane and between the orchestrator and network controllers) and D-CPI (interfacing with the data plane) [7].

E. *Network Controller Peer models*

While a hierarchical approach naturally fits the concept of top-down network orchestration, it does come with its own drawbacks and shortcomings, such as the fact that ownership of the parent controller remains an issue in multi-operator scenarios. While adopting SDN principles locally, a given controller (or orchestrator) can cooperate with other controllers in a peer model, requiring the use of protocols for controlled topology and reachability dissemination and end-to-end service provisioning. A set of SDN controllers that are interconnected in a mesh cooperate to provision services. Commonly, the mesh is implicit by the actual (sub-)domains connectivity. The controllers synchronize state using the so-called East/West interfaces, which should support functions such as network topology abstraction, control adaptation, path computation and segment provisioning. Service provisioning may be driven by an upstream controller. In particular, a possibility is to re-use, for example, the GMPLS protocol suite as an inter-SDN controller protocol [8] but reusing such protocol suite is still limited due to the lack of aforementioned flexibility. Fig. 2 summarizes the different models.

F. *Protocol architecture and framework*

Strongly tied to the functional architecture, network orchestration also requires selecting a protocol architecture. The choice of a protocol framework should be based on a gap/requirements analysis (e.g., efficiency, extensibility...), but is often conditioned by other factors such as existing mature open-source frameworks, ease of maintenance and support, market and industry adoption, etc. While low-level protocols with binary encodings are most efficient, they are more errorprone (e.g., byte ordering issues) and often harder to extend or modify due to initial assumptions (e.g., fixed header sizes, field lengths, float number encodings). While these issues can be overcome, such low-level approaches cannot compete with the ease of development of high-level approaches (coming, for example, from the existing frameworks widely available, e.g. in open source with licenses with varying requirements). A rough guideline is that low-level efficient protocols are adapted to fast changing conditions while higher orchestration layers can use high-level interfaces based on REST/RESTConf architectures with the required protocol stacks (e.g. HTTP) and text encodings (e.g., JSON, XML) easier to debug and process.

Regardless of the encoding, to achieve the desired automation, data about services and network equipment must be organized and described in a common way, such as in terms of data models so that services can be programmatically provisioned across multi-vendor networks. The industry is settling on the IETF standard YANG as the data modelling language to describe and provision services and network devices. The Control Orchestration Protocol (COP) [9] is a pre-standard implementation of the Transport API concept, abstracting a control plane technology of a transport domain, relying on YANG and RESTConf.

For more complex uses and communication models, the use of other communication approaches is also increasing. For example, the Advanced Message Queueing Protocol (AMQP) [10] aims at being the platform agnostic, standard protocol for messaging middleware, supporting producer/consumer, work queues, publish/subscribe, routing and Remote Procedure Calls (RPC). It is expected that complex systems will combine different methods and protocols.

IV. Joint Cloud and Networking Orchestration

As discussed, network orchestration grossly deals with the provisioning of connectivity in a heterogeneous setting, translated into the dynamic configuration of flows and the provisioning of data channels in transport networks. With the increased use of virtualized servers and cloud computing (and the subsequent need to interconnect virtual machines, VMs), the service provisioning process no longer stops at the physical switch, router or transport node, but needs to interact with whatever mechanism the hosting nodes (and virtualization hypervisor) offers. Without configuration, the (virtual) interfaces of a VM interfaces are isolated. In general, providing connectivity requires instantiating one or multiple software bridges within the host and associate virtual and physical interfaces to software bridge instances such as the OpenFlow enabled OpenVSwitch (OVS) [11]. There are several projects and initiatives addressing this. Most deployed modes assume the need to configure the local intra-DC site network, and delegating inter-DC traffic to external networks mostly relying on IP connectivity. A common mechanism involves jointly managing L2 and L3 networks, segregating tenant traffic by means of e.g. VLANs and mapping them into tunnels. In particular, common technologies are using Generic Routing Encapsulation (GRE) or Virtual Extensible LAN (VXLAN). VXLAN is a L2 overlay over a L3 network, with a MAC-in-UDP encapsulation, extending L2 segments across the network, supporting large-scale multi-tenant environment over a shared common physical infrastructure. This results in having e.g. Ethernet broadcast domains that span multiple remote locations. It is worth noting that even if using SDN based control, given the potentially high number of VM instances, centrally managing point-to-point flows and replicating ARP behaviour may have scalability issues. This explains why some approaches combine centralized control with distributed elements similar to the transparent bridge learning mechanism.

A. *OpenStack Cloud Management*

As a Cloud Management System (CMS), OpenStack integrates networking and inter-VM connectivity aspects by means of its Neutron component and associated plugins, driving dynamically instantiated software switches within computing nodes (hosting VMs) and network nodes (allowing complex tenant network topologies). It allows the instantiation of both provider and tenant overlay networks, and supports the creation of (per tenant) networks & subnetworks, the attachment of ports to subnetworks and their interconnection via virtual routers, which can route traffic between internal and external networks. The Neutron design allows pluggable plugins for different types and drivers enabling, e.g. integration with SDN controlled networks or a hardware vendor to interface with network elements via existing management interfaces (e.g. Netconf).

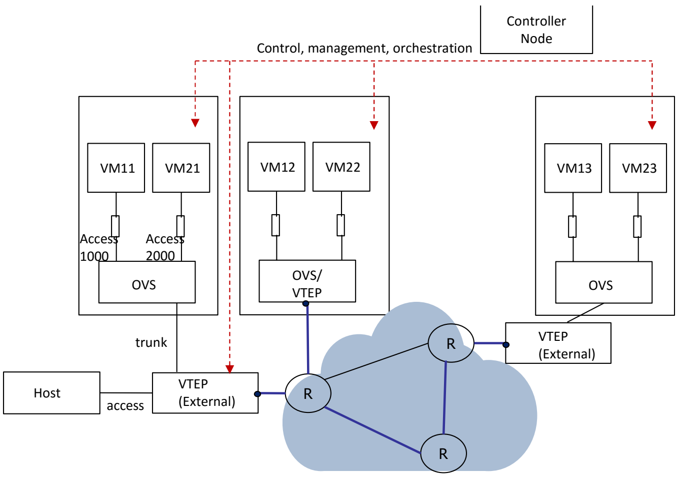

Fig. 3. Example of VXLAN overlay over an IP transport network.

Segregated DCs, with multiple geographically remote sites, are supported in OpenStack, relying on the aforementioned tunnelling. To illustrate this, consider Fig.3. A tenant VM MAC frame is tagged with its VXLAN Network Id (VNID) and encapsulated within a UDP datagram. The datagram is sent (e.g. via IP multicast or unicast) between endpoints (VTEP). VTEPs use learning methods similar to transparent bridges to add forwarding entries mapping VNIDs, remote VTEPs and tenant MAC addresses. From the control plane perspective, multiple locations can be interconnected, with a level of integration that depends on the number and type of endpoints with which interact. For example, OpenStack segregation methods supports *host aggregates* to schedule a group of hosts with common features, *availability zones* as a logical separation for physical isolation of redundancies or *regions* in which a shared infrastructure is used within a cloud with multiple sites. In the latter, there is a different compute API endpoint for each, although the identity service (keystone) is unique. Recently, OpenStack cells, a work in progress, are designed to allow running the cloud in a distributed approach: hosts in a cloud are partitioned into groups called cells, arranged in a hierarchical or tree-like setting. Each cell has its own message queue and database service and, with the top-level cell becoming a single entry point to control access to multiple cloud installations.

B. OpenVSwitch Virtual Networks

The OVN project, part of OVS [11] is an open source project building on top of the OpenVSwitch logical switch implementation providing L2/L3 virtual networking by controlling logical switches and routers and managing multiple tunnel overlays. It aims at complementing OpenStack by assuming the networking service (Fig. 4) and combines a centralized server/database with a high-level status and virtual network views with distributed controllers at each hypervisor and infrastructure nodes that implement the state.

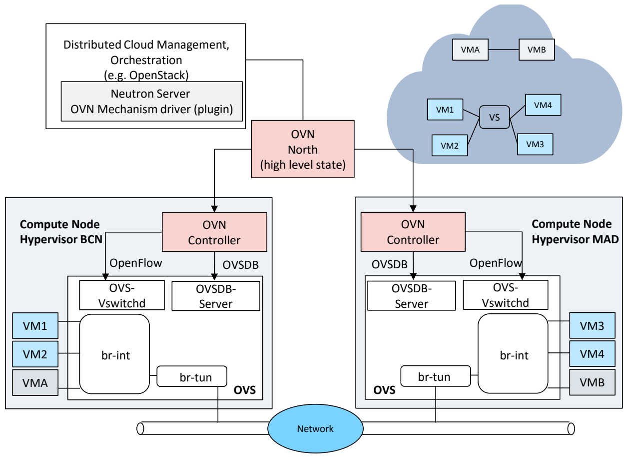

Fig. 4. OVN architecture for the provisioning of virtual overlay networks

C. Integration with Transport networks

In general, the integration of CMS with transport network segments is not (directly) considered; architectures are lacking the full integration of transport layers across different network segments and, in particular with OTN and their respective elements programmability. Consequently, the integration of a CMS (such as OpenStack) with a transport network can be envisioned and solved in multiple ways. For example, by means of an extended subsystem that includes the networking aspects to a larger degree and scope, or by orchestrating one or more CMS by means of a high-level component that coordinates the allocation of services in different locations, along with the configuration of network to interconnect them [12]. Either way, cloud management and network orchestration need to be further integrated, especially given the heterogeneity of technologies and the multiple vendors involved.

V. NETWORK FUNCTIONS VIRTUALIZATION

A. NFV and the relationship with Network Orchestration

The ETSI NFV ISG addresses the dynamic deployment and operation of common network functions in virtual computing instances running in commodity hardware, and defines the architecture and interfaces for the management and orchestration of Virtualized Network Functions (VNFs).

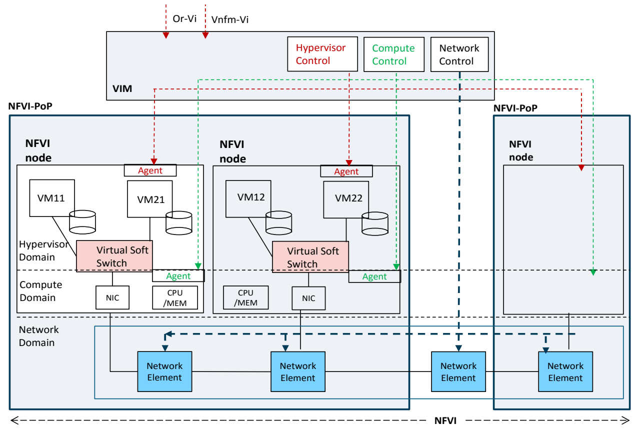

Fig. 5. Single NFVI domain with two NFVI-PoP and single VIM

The initial documents recognize the need for the arbitrary and flexible composition of such VNFs into graphs. The concept of domain within the NFV is manifold. The architecture defines, amongst others, the concepts of *VNF domain*, *infrastructure* domain and *tenant* domain, where multiple tenant domains can co-exist in a single infrastructure domain, separating domains associated with VNFs from domains associated with the NFV infrastructure (NFVI). Within the NFVI [13] given the current technology and industrial structure, compute, hypervisors, and infrastructure networking are already largely separate domains and are maintained as separate. Geographically speaking, a NFVI may have multiple points of presence (NFVI-PoP), defined as a single location with a set of deployed NFVI-Nodes (Fig. 5). A given NFVI can be administratively split into NFVI domains, thus managed by one or more Virtual Infrastructure Managers or VIMs (Fig.6). A VIM is thus a cloud and network orchestrator and multiple VIMs can be orchestrated by the NFV Management and Orchestration (MANO) orchestrator.

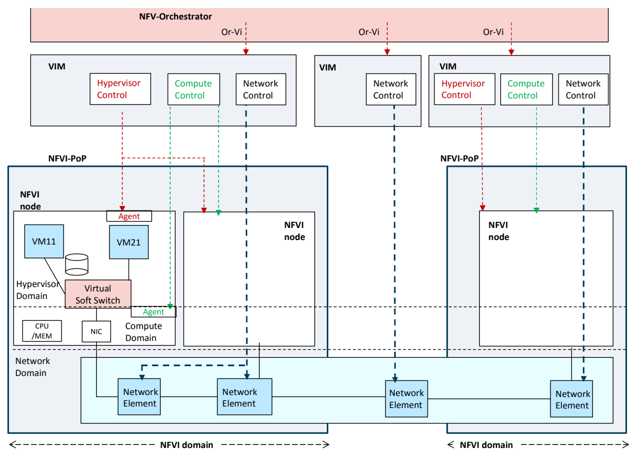

Fig. 6. Orchestration of multiple NFVI/VIM via the NFV-O (Or-Vi)

We are mostly concerned with a single VNF domain, potentially across multiple infrastructure domains. It is thus the role of the NFV Orchestrator to orchestrate NFVI resources across multiple VIMs (with parts of an NFVI may be physically dedicated for use by a given NFV domain and other resources, e.g. WAN or transport networks shared with other NFV or non-NFV domains), and deploy Network Services (NS) and their forwarding graphs. Whether there is a single VIM, abstracting

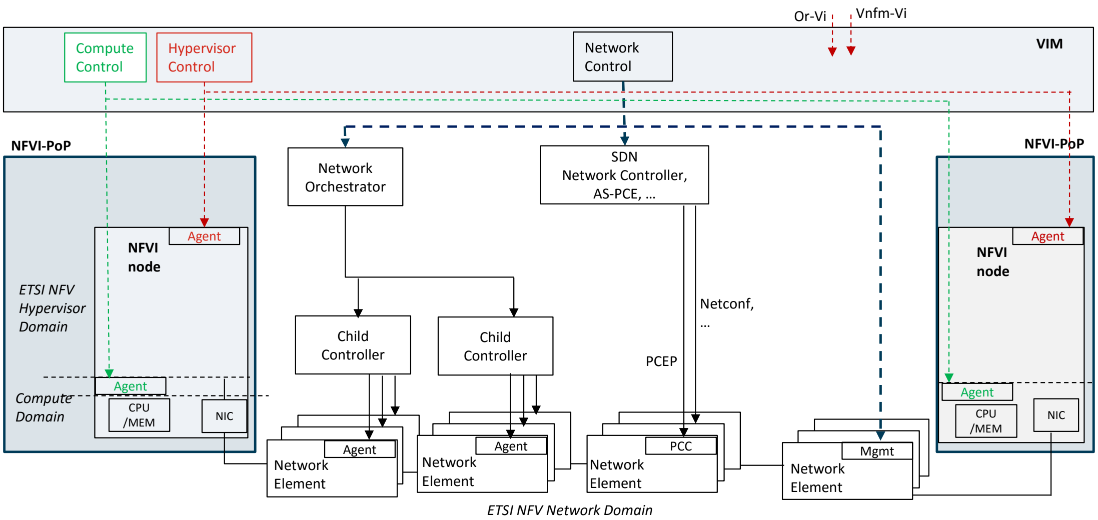

Fig. 7. Over-arching Network Orchestration as the Network control function of the ETSI NFV VIM (single NFVI domain)

the network resources and performing over-arching control and orchestration (Fig. 7), or there are multiple VIMs will depend on a particular administrative arrangement and relationship, and whether the NFVI is segmented into domains.

VI. Final Remarks and Conclusions

The joint orchestration of cloud and network resources, driven by 5G services that go beyond the established models of network connectivity or cloud computing services, requires new control and management architectures that combine centralized and distributed elements. There is a strong adoption of SDN and NFV technologies as the foundations, and the latter provides a comprehensive framework to drive this joint Cloud/IT orchestration, from research to market. The full relationship between the ETSI NFV architecture, the SDN principles and architecture, and the different most relevant (open source) projects are still not completely clear, especially considering their respective widest-senses or in the most flexible cases e.g. where the components of the ETSI MANO architecture are themselves instantiated as VNFs. While it seems clear that SDN can provide connectivity between VNFs, the SDN architecture can also see a VNF as another resource, a node function in a network graph with known connectivity points and known and controllable transfer function. Finally, open source projects may also follow roadmaps driven by service requirements not always with a clear mapping to a reference architecture or combining elements from many. For example, OpenStack cloud-computing software can fulfil the role of VIM but it has its own orchestration capabilities.

To conclude, this paper has given an overview of the drivers behind the adoption of joint cloud and network orchestration, such as segregated data centers or mobile edge computing, that require the allocation of cloud and IT resources in specific locations – e.g. constrained by 5G service requirements --. We have seen how a building block is the network orchestration, and how SDN in its widest sense can be leveraged for provisioning of connectivity between VMs. However, until now, little consideration has been given to the integration with WAN and transport, long-haul networks other than relying on IP connectivity and the overlay of tunnels for traffic isolation.

This trend, towards the full automation of provisioning processes along with a generalized orchestration of systems, is relying on diverse initiatives combining *de facto and de jure* open, modular and extensible standards, including the fast prototyping, open development, user-driven and frequent releases of open source projects and initiatives.

Acknowledgment

This paper has been funded by the Eur. Comm. through the H2020-ICT-2014-2 project 5G-Crosshaul (671598) and the Spanish Ministry MINECO project DESTELLO (TEC2015-69256-R).

References

[1] ETSI Mobile Edge Computing Introductory White paper

[2] O. González de Dios, et al, Multi-partner Demonstration of BGPLS enabled multi-domain EON control and instantiation with H-PCE , Journal of Opt. Communications and Networking, V. 7, N. 11, Nov. 2015

[3] D. King, A. Farrel, editors, “A PCE-Based Architecture for Application-Based Network Operations”, RFC7491, March 2015.

[4] R. Muñoz, et al., Transport Network Orchestration for End-to-End Multi-layer Provisioning Across Heterogeneous SDN/OpenFlow and GMPLS/PCE Control Domains, JLT, , Vol. 33, No. 8, April 2015

[5] R. Casellas, et al. “Overarching Control of Flexi Grid Optical Networks: Interworking of GMPLS and OpenFlow Domains,” IEEE/OSA Journal of Lightwave Technology, vol 33 No. 5, 2015.

[6] R. Casellas, et al., SDN Orchestration of OpenFlow and GMPLS Flexi-grid Networks with a Stateful Hierarchical PCE, Journal of Opt. Communications and Networking, V. 7, N. 1, Jan 2015.

[7] ONF Draft Technical Document (onf2015.420.12.pdf) "TR-521 SDN architecture issue 1.1", work in progress, final for publication

[8] R. Casellas, et al, Control and Orchestration of Multidomain Optical Networks With GMPLS as Inter-SDN Controller Communication”, Journal of Opt. Communications and Networking, V. 7, N. 11, Nov. 2015.

[9] R. Vilalta, et al., The Need for a Control Orchestration Protocol in Research Projects on Optical Networking , in Proceedings of EuCnC 2015, 29-2 July 2015, Paris (France).

[10] The Advanced Message Queueing Protocol (AMQP) specification, The International Standard (ISO/IEC 19464) [https://www.amqp.org/]

[11] The OpenVSwitch project [http://openvswitch.org]

[12] A. Mayoral et al, “Experimental Seamless Virtual Machine Migration Using an Integrated SDN IT and Network Orchestrator”, OFC2015, Los Angeles, CA, USA, March 2015.

[13] ETSI Group Specification “Network Functions Virtualisation (NFV); Infrastructure Overview” ETSI GS NFV-INF 001 V1.1.1 (2015-01)