Distributing connectivity management in Cloud-Edge infrastructures : Challenges and approaches

David Espinel Sarmiento, Adrien Lebre, Lucas Nussbaum, Abdelhadi Chari

Compas'2019 : Parallélisme / Architecture / Système
LIUPPA - IUT de Bayonne, France, du 24 au 28 juin 2019

Distributing connectivity management in Cloud-Edge infrastructures : Challenges and approaches

David Espinel Sarmiento *, Adrien Lebre †, Lucas Nussbaum ‡, and Abdelhadi Chari *

Orange Labs Network, 22300 Lannion - France
davidfernando.espinelsarmiento@orange.com, abdelhadi.chari@orange.com*

IMT-Atlantique, Inria, 44000 Nantes, France
adrien.lebre@inria.fr†

Universite de Lorraine, CNRS, Inria, LORIA, F-54000 Nancy, France
lucas.nussbaum@loria.fr‡

Abstract
The classic approach of deploying large data centers to provide Cloud services is being challenged by the emerging needs of Internet of Things applications, Network Function Virtualization services or Mobile edge computing. A massively distributed Cloud-Edge architecture could better fit the requirements and constraints of these new trends by deploying on-demand Infrastructure as a Service in different locations of the Internet backbone (*i.e,* network point of presences). A key requirement in this context is the establishment of connectivity among several virtual infrastructure managers in charge of operating each site. In this paper, we analyze the requirements and challenges raised by the inter-site connectivity management in a Cloud-Edge infrastructure. We also aim at initiating the discussion about the research directions on this field providing some interesting points to promote future work.

Mots-clés : VIM, Cloud, virtualization, IaaS, overlay network

---

1. Introduction

Internet of Things (IoT) applications, Network Function Virtualization (NFV) services, and Mobile Edge Computing will require to deploy IaaS services within the backbone in a distributed fashion to respect delay or legal requirements [1]. One way to deploy such a Cloud-Edge infrastructure/Distributed Cloud Infrastructure (DCI) is enabling existing network points of presence (PoPs) to be used as micro DCs (potentially reaching several thousands of deployments) using Virtual Infrastructure Managers (VIM) like OpenStack, as proposed in the DISCOVERY initiative [7]. While this approach could better fit the requirements and constraints of the emerging needs, it raises new questions from a management point of view [4]. In order to provide the capacity of managing connectivity in a native way among several VIMs, a solution should answer the following Edge-Cloud challenges :

— **Scalability** : The Cloud-Edge infrastructure must allow an increasing number of instances without affecting performance.

— **Resiliency** : A Cloud-Edge infrastructure should be able to maintain control logic in case of network partitions without affecting the general behavior of the non-disconnected part and also being capable of providing local services in the isolated sites.
— **Locality awareness** : VIMs should have autonomy for local domain management. This implies that locally created data should remain local as much as possible, and only shared with other instances if needed, thus avoiding global knowledge. This also implies the capacity to peer with other instances to establish networking services and send information only when demanded as the sharing of global network information could be restricted or forbidden, and only allowed for certain cases.
— **Abstraction** : Configuration and instantiation of inter-site networking services should be kept as simple as possible to allow the deployment and operation of complex networking scenarios. The management of the involved implementations must be fully automatic and transparent for the users.

Among the required features, the capacity to interconnect virtual networking constructions belonging to several independent VIMs is an important one. As VIMs have been conceived to work in a pretty stand-alone way managing a single deployment, it is necessary to have a networking solution that enables the control of both intra-PoP and inter-PoP infrastructure services connectivity. In the case of OpenStack, if having one OpenStack instance by site is considered, the main issue will rely on the networking module (*i.e.,* Neutron), that has not been designed to interact with other instances.

This paper aims at initiating the debate on how inter-site connectivity challenges can be addressed. In that sense, our work contributes (i) to provide an analysis of the requirements and challenges raised by the connectivity management in a Cloud-Edge infrastructure managed by several VIMs, ii) to present an overview of the shortcomings of the current solutions aiming to addressing those challenges, and (iii) to provide a discussion of the research directions in this field. The rest of this paper is organized as follows. Section II gathers a list of networking services and features expected in a Cloud-Edge infrastructure. Section III describes the challenges that need to be addressed to provide those services. Section IV presents ongoing works. Possible approaches to overcome the challenges are discussed in Section V. Finally, Section VI concludes and discusses future works.

2. Inter-Site Connectivity Requirements

There are several requirements and constraints that a solution should guarantee or provide in a Cloud-Edge infrastructure. Among the required networking services that need to be deployed in such infrastructure, the following four could be considered as the cornerstones for the architecture [3] :

— **Layer 2 network extension** : be able to have a Virtual Network (VN) that spans several VIMs. This is the ability to plug into the same VN, virtual machines (VMs) that are deployed in different VIMs.
— **Routing function** : be able to route traffic between a user A's VN on VIM 1 and a user B's VN on VIM 2.
— **Traffic filtering, policy and Quality of Service (QoS)** : be able to enforce traffic policies and QoS rules for traffic between several VIMs.
— **Service Chaining** : Service Function Chaining is the ability to specify a different path for traffic in replacement of the one provided by default. A user needs to be able to deploy a service chaining spanning several VIMs, that means the possibility to have parts of the service VMs placed in different VIMs.

In the rest of this paper, we focus on the *Layer 2 Network Extension* requirement (Figure 1) To implement and deploy *Layer 2 Network Extension*, we will need :

— A way to request the Layer 2 extension across several VIMs : this means informing each of the VIMs that it must extend a local virtual network to another site.
— A mechanism to learn MAC/IP VMs addresses between the interested VIMs (in our example VIM1, VIM2, and VIM3) : e.g. Allows VIM1 to learn VIM2's and VIM3's MAC/IPs and know how to reach them and vice versa.
— Some switching instances located at every VIM that will use this learned MAC/IPs to forward L2 packets between the interconnected VIMs.

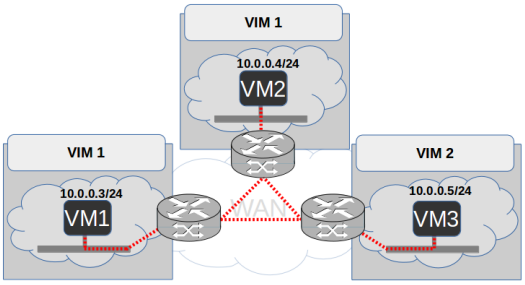

FIGURE 1 – Layer 2 extension feature

3. Inter-Site Connectivity Challenges for a Cloud-Edge Infrastructure

Several questions arise in order to provide the aforementioned Layer 2 extension feature in a Cloud-Edge infrastructure. This feature should be provided while satisfying the general requirements of scalability, locality awareness, resiliency, and abstraction. This section aims to explain some of the challenges.

3.1. Challenge 1 : Standard automatized and Distributed Interfaces

Due to the centralization, VIMs only present a user-oriented interface to provide local services. This is a management integration challenge which implies that the interface which faces the user (user side or north-side as traffic flows in a vertical way) and the interface which faces other VIMs (VIMs-side or east-west-side as traffic flows in a horizontal way) have to be smoothly bridged among them. This integration needs to be done in order to provide the necessary user abstraction and the automation of the VIMs communication process. Consequently, this necessitates the specification and development of well-defined north- and east-west-bound interfaces. The initial information exchanges among VIMs should take into account the identification of the mechanism to use (could be a pair of VLANs on an interconnection box, BGPVPN RT identifiers, VXLAN ids, etc.). This later, due to the fact that the user should not be aware of how these networking constructions are configured at the low-level implementation. Since a Cloud-Edge infrastructure could scale up to hundreds of sites, manual networking stitch techniques like [10][11] will be simply not enough. Thus, how to design an efficient interface for bot north-bound and east-west-bound communication is an important problem in the research of inter-site connectivity management tools.
The use of a centralized DataBase (DB) by all participants can be leveraged to avoid the need to request the Layer 2 extension service. In that case, the information of a Layer 2 segment created at VIM1 and the local MAC/IP addresses attached to it will be available to other VIMs. The problem with this approach lies in the locality awareness as we need that locally created data remains local as far as possible and the use of a centralized DB does not respect this point. Even more, in network partitioning cases, if the DB is not accessible by a site, the VIM will not be capable of providing even local services, thus impacting the desired resiliency of the system.

3.2. Challenge 2 : On-Demand States Sharing

VIMs need to be able to share states among them only when requested. The MAC/IP addresses and network identifiers of a local network in VIM2 do not need to be shared with VIM1 and VIM3 unless the user requests the L2 extension of this network. How to deal with requests demanding to extend Layer 2 segments with already deployed VMs presenting address overlapping presents as an important problem that need to be studied. Overlapping addresses could also be presented in network partitioning cases and should be treated as well. If a Layer 2 segment is shared among the three VIMs of Figure 1 and because of a network failure VIM2 is isolated from the others, the user may do the provisioning of a new VM in VIM2 with an IP address already granted to a VM in VIM1. Once the network failure is restored, VIM2 will notify the other VIMs about the VM information creating a conflict with the overlapping addresses. Such class of conflict resolution must be addressed minimizing the incoherence states and preserving local autonomy.

3.3. Challenge 3 : Dynamic Notification and Reconfiguration

Dynamic reconfiguration of networking services on each VIM needs to be done automatically according to (1) the requested service and (2) taking into account the explicit and implicit operations. For the first point, if a Layer 2 extension is requested between VIM1 and VIM2 and does not concern VIM3, there is no reason to notify or to reconfigure VIM3 network services. For the latter, for an already provisioned Layer 2 extension present at the three sites, if a new VM is attached to the segment in VIM3, it must execute implicit operations to notify the other VIMs about the reachability information of the new VM. Depending on the implementation, the solution needs be able to do this reconfiguration at the underlay level which implies the ability to talk to some physical equipment like the Edge site gateway ; or to do it at the overlay level which implies the ability to reconfigure virtual forwarding elements like GoBGP instances [14], Open vSwitch switches [6] or Linux bridges [5].

4. Ongoing works for Cloud-Edge infrastructures

The management of Cloud-Edge infrastructures has already been studied in the literature at different levels but without addressing the totality of the aforementioned challenges. We can distinguish two categories : the distributed cloud management solutions that do not consider the inter-site networking services, and the solutions proposing networking services.

4.1. Distributed Cloud solutions without networking services

A few works [1, 4, 7, 17] have investigated distributed approaches to deal with Cloud-Edge resource management challenges. However, they did not take into account the inter-site connectivity features. In [7] the authors make a complete analysis of the main limitations for OpenStack scalability, showing that the most critical points are the SQL Database and the messaging queue. Although the connectivity issue is mentioned, it is not studied and it is left as future work. In [1], the author experiments with several deployment configurations of OpenStack to provide a fully native DCI. The study concludes that using the regions feature, a deployment having an isolated communication bus and a shared database for all services could be a good start for future works. However, it presented network connectivity issues because the shared database did not respect the data locality and autonomy constraints, allowing remote instances to access private network information and trying to make non desired local changes. The challenge is left as future work.

4.2. Solutions proposing networking services

This section gathers a set of solutions differing in their architecture designs but that provide some kind of multi-site networking services.

4.2.1. Cloud-enabled networking solutions without distribution

Several works have proposed ways to manage a Cloud-Edge infrastructure connectivity services adopting a centralized or hierarchical architecture [2, 8, 9, 13]. The major drawback of these kinds of solutions is the fragility exposed by the root or central element controlling the entire architecture, as it can experiment troubles that can make the entire infrastructure inaccessible. In [13] Tricircle project is proposed, it achieves network automation in a multi-region deployment of OpenStack using a Central Neutron Service (the only exposed Neutron API to the user) and a series of modified Neutron Core plug-ins. Tricircle does not fit well in network partitioning cases as isolated sites are completely useless without communication with the Central Neutron.

4.2.2. Distributed networking solutions without Cloud-awareness

In this category we gather networking solutions like DISCO [16] that presents a distributed architecture but that are unable to manage a Cloud-Edge infrastructure. DISCO (stands for DIstributed SDN COntrol Plane) is an SDN controller for overlay networks. It relies on a per-domain organization where each controller is in charge of an SDN domain. Every controller has an intra-domain part that gathers the main functionalities like managing virtual switches, and an inter-domain part that manages communication with other DISCO controllers to make reservations, topology state modifications or monitoring tasks. DISCO provides an inter-controller channel using a message-oriented communication bus implemented using AMQP in federation mode, every controller has at the same time an AMQP server and a client.

4.3. Summary

As might be seen, the capability of establishing inter-site networking services among several VIMs instances has been left aside in the literature. Although it is still an important element that needs to be studied and addressed in order to deploy a DCI architecture where users resources can effectively communicate among them.

5. Research Directions to Decentralize Connectivity Management

As we stated in the last section, our community should propose effective connectivity management mechanisms among VIMs. Knowing the challenges that need to be addressed, it is possible to present some approaches that will allow us to overcome the limitations and propose a new way to interconnect Cloud-Edge infrastructures.

Inspired by the aforementioned DISCO architecture, we propose to leverage a physically and logically distributed inter-site connectivity solution. This solution will be by itself the inter-domain part of the VIMs main networking component, analogous to the inter-domain and intra-domain parts of DISCO. In case of network partition this architecture will provide full autonomy to isolated sites since only the inter-site part will be unavailable while local services are still provided. For the rest of the network, multi-site operations will remain available without affecting the normal behavior of the system. Such kind of autonomy for VIMs functionalities will allow to create a resilient system without the need for heavy synchronization.

The solution should leverage the use of E-VPNs and IP-VPNS based on BGPVPN [15] technologies to grant virtual routes exchanges among VIMs. In consequence, an easy extension of

Layer 2 local segments in several sites will be allowed. In this sense, the work presented in [12] is a promising start since it allows to create a two sites-related "interconnection" resources. The first resource references a local resource (e.g. network A in VIM1) and a remote resource (e.g. network B in VIM2) having the semantic informing that connectivity is desired between the two. The proposition then leverages the use of BGPVPNs at both sides to create an overlay network connecting the two local segments. In all cases, user has to define the interconnection at each VIM.
The future work should focus in a solution combining the distributed architecture from DISCO and using BGPVPN technologies. DISCO-based architecture will allow VIMs networking components to peer among them for communication purposes ; at the same time, BGPVPN will allow to span overlay networks among the requested sites involving the possibility to create Layer 2 extensions. The use of a physically and logically distributed architecture for inter-site networking services among several VIMs while providing full autonomy to each one, presents itself as a good approach for the DCI case. Such approach will allow to achieve a distribution of connectivity management accomplishing a scalable, locality-aware, resilient, and easy to use infrastructure.

6. Conclusions

Leveraging each PoPs in Telcos's architecture as part of a Cloud-Edge infrastructure is an approach that gives answers to the many constraints posed by services like NFV, IoT, and MEC. Although there are some proposals for the management of such infrastructure, provide inter-domain networking services in a distributed fashion is still a challenge to be addressed.
In this paper, we explained some possible ways to allow peering among several VIMs in order to provide networking services among sites. Then, we reviewed some related works aiming at leveraging a management system for a DCI and explained the problems of its designs. We then explained different approaches to tackle down those limitations in order to provide a distributed connectivity management. We conclude that there is an enormous potential using a physically and logically distributed architecture composed by an inter-site connectivity module leveraging some already existing overlay network technologies in order to achieve a distributed network automation among several VIMs.

Bibliographie

1. Bousselmi (A.), Peltier (J. F.) et Chari (A.). – Towards a massively distributed iaas operating system : Composition and evaluation of openstack. *IEEE Conference on Standards for Communications and Networking*, 2016.
2. Brasileiro (F.), Silva (G.), Arajo (F.), Nbrega (M.), Silva (I.) et Rocha (G.). – Fogbow : A middleware for the federation of iaas clouds. *16th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing*, 2016.
3. Chari (A.), Morin (T.), Sol (D.) et Sevilla (K.). – *Approaches for on-demand multi-VIM infrastructure services interconnection*. – Rapport technique, Orange Labs Networks, 2018.
4. Cherrueau (R.-A.), Lebre (A.), Pertin (D.), Wuhib (F.) et Soares (J. M.). – Edge computing resource management system : a critical building block ! *HotEdge*, 2018.
5. Foundation (T. L.). – Linux bridges, 2018. – https ://wiki.linuxfoundation.org/.
6. Foundation (T. L.). – Openvswitch, 2018. – https ://www.openvswitch.org/.
7. Lebre (A.), Pastor (J.), Simonet (A.) et Desprez (F.). – Revising openstack to operate

fog/edge computing infrastructures. *IEEE International Conference on Cloud Engineering*, 2017.

8. Moreno-Vozmediano (R.), Montero (R. S.), Huedo (E.) et Llorente (I.). – Cross-site virtual network in cloud and fog computing. *IEEE Computer Society*, 2017.
9. OpenStack. – Kingbird project, 2018. – https ://wiki.openstack.org/wiki/Kingbird.
10. OpenStack. – Neutron bgpvpn interconnection, 2018. – https ://docs.openstack.org/networking-bgpvpn/latest/.
11. OpenStack. – Neutron networking-l2gw, 2018. – https ://docs.openstack.org/networking-l2gw/latest/readme.html.
12. OpenStack. – Neutron-neutron interconnections, 2018. – https ://specs.openstack.org/openstack/neutron-specs/specs/rocky/neutron-inter.html.
13. OpenStack. – Tricircle project, 2018. – https ://wiki.openstack.org/wiki/Tricircle.
14. OSRG. – Gobgp, 2018. – https ://osrg.github.io/gobgp/.
15. Ould-Brahim (H.), Rosen (E.) et Rekhter (Y.). – Using bgp as an auto-discovery mechanism for layer-3 and layer-2 vpnss. *IETF*, 2004.
16. Phemius (K.), Bouet (M.) et Leguay (J.). – Disco : Distributed multi-domain sdn controllers. *Network Operations and Management Symposium*, 2014.
17. Soares (J.), Wuhib (F.), Yadhav (V.), Han (X.) et Joseph (R.). – Re-designing cloud platforms for massive scale using a p2p architecture. *IEEE 9th International Conference on Cloud Computing Technology and Science*, 2017.