This paper was accepted for publication in the IEEE Cloud Computing. The copyright was transferred to IEEE.
The final version of the paper will be made available on IEEE Xplore® via http://dx.doi.org/10.1109/MCC.2017.28

Cross-Site Virtual Network Provisioning in Cloud and Fog Computing

R. Moreno-Vozmediano[1], R.S. Montero[1], E. Huedo[1], I.M. Llorente[1, 2]
[1] *School of Computer Science, Complutense University, SPAIN.*
[2]*FAS Research Computing, Harvard University, USA.*

The interconnection of the different geographically dispersed cloud and fog infrastructures is a key issue for the development of the fog technology. Although most existing cloud providers and platforms offer some kind of connectivity services to allow the interconnection with external networks, these services exhibit many limitations and they are not suitable for fog computing environments. In this work we present a hybrid fog and cloud interconnection framework, which allows the automatic provision of cross-site virtual networks to interconnect geographically distributed cloud and fog infrastructures. This framework provides a scalable and multi-tenant solution, and a simple and generic interface for instantiating, configuring and deploying Layer 2 and Layer 3 overlay networks across heterogeneous fog and cloud platforms, with abstraction from the underlying cloud/fog technologies and network virtualization technologies.

Keywords: *Heterogeneous Fog and Cloud Interconnection; Virtual Network Provisioning; Layer 2 and Layer 3 Overlay Networks; Scalability and Multi-Tenancy Support.*

Fog computing paradigm [1] is emerging as a key enabling technology for the Internet of Things (IoT), by broadening the scope of cloud platforms and services to the edge of the network, and allowing the efficient and agile deployment of mobile applications with strict geo-distribution, location awareness, and low latency requirements. Other similar paradigms to fog computing are ETSI Mobile Edge Computing [2] and Berkeley Cloudlets [3]. As shown in Figure 1, fog computing extends the cloud computing model by including an additional layer between the cloud and the mobile devices. In this three layer architecture (device-fog-cloud), a fog computing node is a small to medium size computing infrastructure that includes compute, storage and networking elements and is usually located at the premises of the end mobile users (e.g. shopping centers, airports, tourist attractions, etc.), and fog instances are physical or virtualized resources, deployed on top of the fog node infrastructure, that run the applications consumed by these end users, and can be accessed by mobile devices at one-hop distance over the wireless network. In addition, various fog nodes can also be connected to a central cloud, that could provide coordination between the different fog infrastructures, and some other additional services, such as extra computing capacity, large database management, off-line data processing for business intelligence, etc. This central cloud can be implemented as a commercial cloud provider (e.g. Amazon EC2, Microsoft Azure, Google Cloud, etc.) or as a public or private cloud managed by a cloud management platform [4] (e.g. OpenNebula, OpenStack, etc.)

Some important features that must be considered in the design of a fog computing architecture are the following: a) *multi-tenancy and isolation*, since fog nodes must support the coexistence of several applications belonging to different tenants, and guarantee isolation among them; c) *scalability* regarding the number of fog resources serving a given application in a fog nodes, which must be dynamically adapted the number of mobile devices demanding this application; d) *heterogeneity*, since different applications can have different hardware and software requirements; e) *agile and on-demand provisioning* of application resources (fog instances and interconnection networks), to guarantee fast response time and low service latencies. To support all these features, fog computing relies on virtualization technologies, so that fog nodes are usually managed as virtualized platforms, also called micro-clouds, where fog resources can be implemented as virtual machines or software containers.

A key component in the deployment of a fog computing environment is the virtual network that interconnects the different components of a fog application [5], which can be geographically dispersed over various fog nodes and the central cloud. In this context, it is necessary to provide the fog computing architecture with tools for the simple, efficient, and automated instantiation, configuration, and isolation of virtual networks spanning different fog infrastructures and clouds. Furthermore, the features of each virtual network must be adapted to the applications needs, by supporting both Layer 2 (L2) and Layer 3 (L3) networks, different types of traffic (IPv4, IPv6 or non-IP traffic), or different addressing schemes (e.g. public or private addressing). It is important to note that these network provisioning methods for fog computing environments must be independent on the fog applications or the communication patterns, so classical peer-to-peer application overlays that to some extent address these problems cannot be applied in this scenario. Most existing cloud providers and platforms offer some kind of connectivity services, usually based on Virtual Private Networks (VPNs), to allow the interconnection with external networks from other clouds or user premises. However, these VPN services offers several drawbacks, such as different APIs, instantiation, and configuration methods, limited scalability, and lack of L2 support.

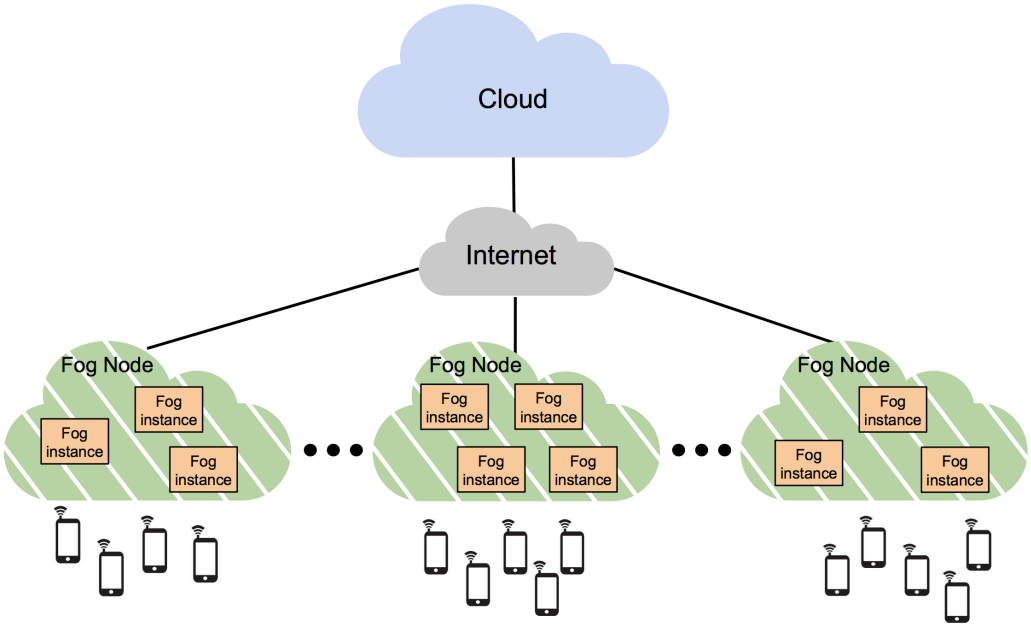

Figure 1. Fog computing environment.

These limitations are being tackled to some extent by BEACON [6], an European project aimed to define and implement a federated cloud network framework. In this article, we propose to extend the work being done in the field of cloud network federation to fog computing infrastructures, in order to provide this environment with the capacity of provisioning virtual networks spanning various heterogeneous fog and cloud sites.

Challenges in Fog and Cloud Interconnection

The goal of this article is to define a *Hybrid Fog and Cloud* (HFC) *Interconnection Framework* to enable the simple, efficient, and automated provision and configuration of virtual networks to interconnect different geographically distributed fog and cloud sites. We first analyze the main challenges that should be addressed in the design of this interconnection environment.

Abstraction and Simplicity
The configuration and instantiation methods of virtual networks spanning different fog and cloud sites should be independent of the technologies offered by the different cloud providers and fog infrastructures. The users should not be aware of how these virtual networks are configured in terms of network elements (e.g. switches, routers or VPN gateways) and technologies (e.g. tunnels or encapsulation techniques). These configuration and instantiation methods should be simple, to allow the deployment and operation of complex networking scenarios.

Heterogeneity
The HFC interconnection framework should enable application tenants to deploy a virtual network spanning heterogeneous fog infrastructures and clouds, which use different technologies and are managed by different fog/cloud

management platforms. The framework should be built upon basic and common functionalities, both in terms of APIs and the networking functionality provided by each cloud or fog infrastructure.

Scalability and Elasticity
The HFC interconnection framework should provide scalability from the point of view of number of networks from different fog infrastructures and clouds that constitute each virtual network. It must also support different interconnection topologies (e.g. tree or mesh) according with the service needs. This involves that the creation and configuration of interconnection channels between remote networks (e.g. VPN tunnels), the management of network devices, and the management of routes and routing tables, must be fully automatic and transparent for the users. In addition, the framework should also support elasticity, enabling the service provider to extend the network to a new fog or cloud location or to remove a location from the network.

Support for L2 and L3 Virtual Networks
The HFC interconnection framework should support the deployment of both L2 and L3 virtual networks across different fog platforms and clouds. Although most cloud providers only offer services for building L3 virtual networks (mostly based on VPN technology), many services require the creation of L2 overlays networks to interconnect the different networks in a fog computing scenario. The configuration of these L2 overlay networks is more challenging, considering the lack of support from the different providers, but they present multiple advantages: first they are independent of the network protocol, so they allow to encapsulate not only IPv4 traffic, but also IPv6 traffic, and non-IP traffic; second, they are suitable for deploying applications requiring L2-adjacency (e.g. high availability clusters, database clusters, network storage systems, etc.) or applications based on broadcast/multicast (e.g. multicast video streaming, LAN games, etc.); and third, they natively support mobility and migration, since hosts in different sites share a common overlay addressing scheme, so these hosts can be moved from one cloud to another with minimal reconfiguration requirements.

Security
Virtual networks spanning different sites in a hybrid fog and cloud computing scenario should provide the same security guarantees from external threats than a single-site network. So, in a hybrid environment where the interconnection of the different cloud and fog sites runs over public networks, if the user demands strict security requirements, it is necessary to guarantee data privacy and integrity by constructing the (L2 or L3) overlay virtual networks over secure communication channels.

Multi-tenant Support
The HFC interconnection framework should serve multiple application providers or tenants, each one deploying its own applications and virtual networks providing service to multiple end-users. To avoid very complex network configurations, saturation and bottlenecks of network devices, and performance interferences between tenants, the framework will be designed using a per-tenant solution: for each tenant, a HFC interconnection agent (see next section) is deployed on each fog infrastructure and cloud platform, which is responsible for managing and configuring the network devices and interconnection channels needed to create the virtual networks of that tenant.

Based on Existing Virtualization Technologies
The HFC interconnection framework will leverage on existing network virtualization technologies, such as overlay virtual networks [7], Software Defined Networks (SDN) [8] and Network Function Virtualization (NFV) [9].

The HFC Interconnection Framework

This section describes the architecture of the HFC interconnection framework and introduces the main components of this framework and their basic functionality. It also describes how the HFC framework operates, and provides the technological details about the implementation of the virtual overlay networks.

HFC Framework Architecture and Components

Figure 2 shows the architecture and basic components of the HFC interconnection framework for hybrid fog/cloud scenarios. In this scenario, we assume that fog nodes are implemented as micro-clouds, which are managed by some cloud-like management platform (e.g. OpenNebula, OpenStack, etc.). This management platform is responsible for deploying, monitoring, and removing the fog instances, which are implemented as Virtual Machines (VMs), and also for instantiating the different network segments to interconnect fog instances inside a fog node, which are implemented as isolated Virtual LANs (VLANs).

A *HFC Virtual Network* is an interconnection of different network segments deployed on different fog and cloud sites. The different network segments that constitute a HFC Virtual Network are interconnected through a L2 or L3 overlay network (data plane) on top of the physical network.

This overlay is built by special network elements in each network segment termed *HFC Agents*, which drive the control plane of the HFC Virtual Network using SDN technology (e.g. using OpenFlow). HFC Agents are deployed, on a per-tenant basis, in each one of the fog and cloud sites involved. They implement the required Virtual Network Functions (VNFs), such as virtual switches (e.g. based on Open vSwitch) or virtual gateways, and establish the necessary L2 or L3 tunnels over the Internet connection. HFC Agents can have different implementations depending on the management platforms used on each cloud and fog site. In addition, they can be deployed in a high-availability cluster configuration to increase the reliability of the network overlay, in this case all the HFC Agents will share a common routing public IP to create the overlay.

The *HFC Manager* is responsible for instantiating the different network segments on each fog and cloud site, and also for deploying, coordinating, and controlling the behaviour of HFC Agents (management plane). For this purpose, the HFC Manager interacts with the different cloud and fog management platforms and HFC Agents using the southbound interface. This component also exhibits a northbound API to enable application tenants to interact with the framework. This is a simple REST interface that allows to instantiate, configure or remove a HFC Virtual Network.

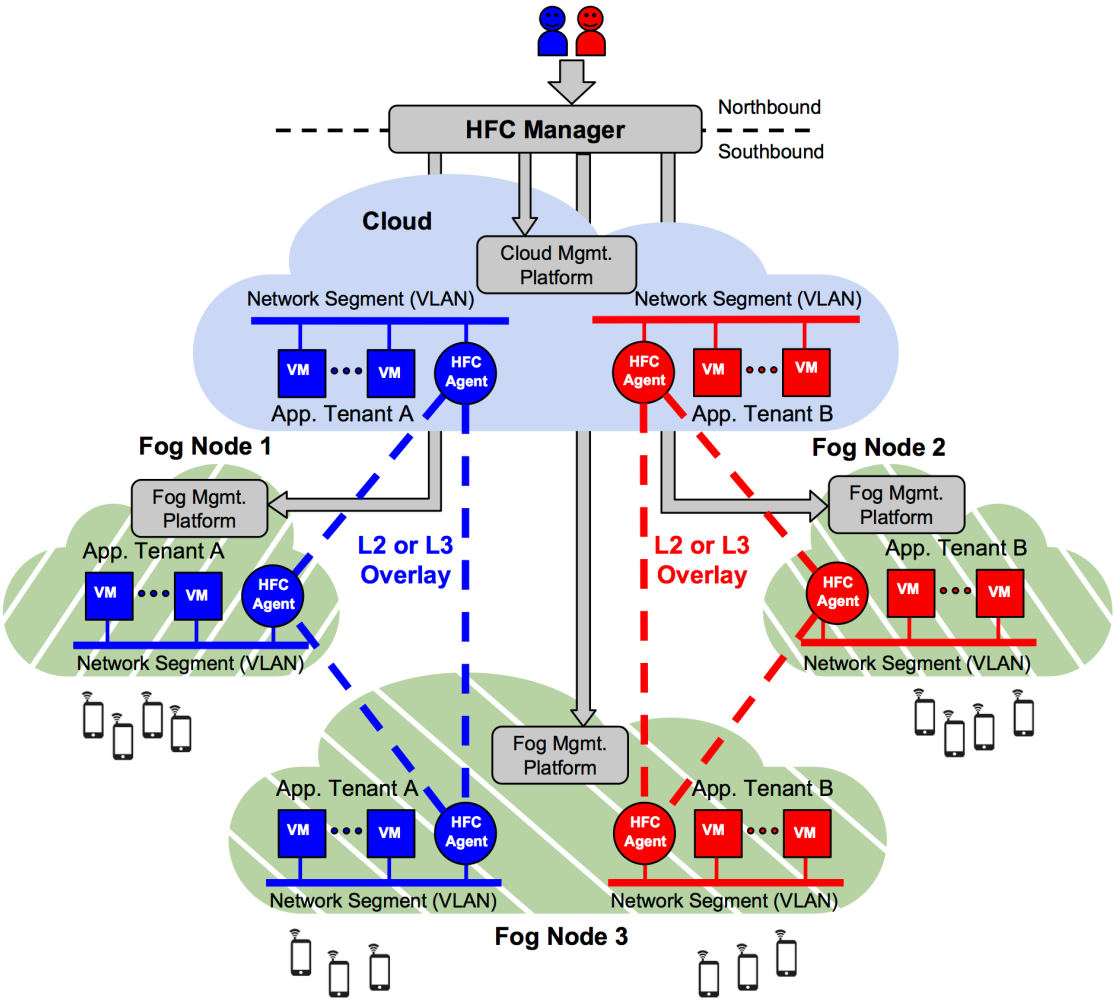

Figure 2. HFC interconnection framework architecture.

This HFC interconnection framework addresses all the aforementioned challenges. The HFC Manager provides *abstraction and simplicity,* since the northbound API provides a simple and generic method to instantiate an HFC Virtual Network, which is independent of the implementation and the type of the fog and cloud sites involved. *Heterogeneity* is also guaranteed, since the HFC Manager southbound interface can interact with different cloud and fog management platforms. The use of per-tenant HFC Agents provides *scalability*, since each virtual network is managed independently from other tenant's networks; *elasticity*, since virtual networks can be easily extended to new sites; and

multi-tenancy support. *Security* is provided by means of isolation between the different tenant's networks, and the possibility of implementing secure communication channels, if required. Finally, the overall framework has been conceived to natively support *L2 and L3 virtual networks*, and based on existing network *virtualization standards*.

HFC Framework Operation

When a tenant needs to deploy a HFC Virtual Network to support a given distributed application spanning different cloud and fog sites, it interacts with the HFC Manager, using the northbound interface, and instantiates the new network by specifying the main network attributes, such as the type of overlay network (L2 or L3); the security requirements (plain or encrypted communications); and the locations (clouds and fog sites) where the virtual network must be deployed. The HFC Manager, using the southbound interface, communicates with the different fog and cloud management platforms to instantiate a new network segment on each site, and deploys a HFC Agent per site attached to each network segment. Each HFC Agent receives the information of the other existing agent endpoints, the type of overlay, and the network security requirements. Then, the different HFC Agents communicate each other to construct the overlay network topology specified by the user by implementing the appropriate virtual network devices (e.g. switches, gateways, VPN endpoints, etc.) and configuring the appropriate L2 or L3 tunnels. Once the HFC Virtual Network is deployed and operational, the tenant can deploy the different cloud and fog instances that will run the different application components on each site.

Virtual Network Implementation

Regarding the implementation details of the virtual networks spanning the different cloud and fog sites, there are many different encapsulation and tunneling technologies that can be used to construct the L2 and L3 overlay networks. In the case of L3 overlay networks, if no secure communications are required, it is possible to implement a simple L3 over L3 (L3oL3) tunnel using GRE (*Generic Routing Encapsulation*) protocol [10]. However, if security is a prerequisite, one of the most common secure L3oL3 tunneling methods is a site-to-site IPsec VPN. IPsec is a standard protocol suite for secure IP communications, that provides several security features such as data origin authentication, data integrity, data encryption, and replay protection. To implement this configuration in the HFC interconnection framework, each HFC Agent implements a virtual VPN gateway, and establishes the necessary IPsec tunnels between the different VPN gateways, following a given topology (star, partial mesh, or full mesh), according to the application needs.

On the other hand, to implement L2 overlay networks, it is necessary to encapsulate L2 (Ethernet) traffic over public L3 (IP) networks, using some kind of L2 over L3 (L2oL3) tunneling protocol [11]. Some of the most outstanding technologies to encapsulate Ethernet traffic over TCP, UDP or IP packets are VXLAN, STT, NVGRE, or GRETAP. In this case, each HFC Agent implements a virtual switch, and establishes the necessary L2oL3 tunnels between these switches, according to the topology required by the application. It is important to notice that these L2oL3 encapsulation techniques do not guarantee data integrity and privacy, as L2 frames are encapsulated as plain data. So, if the user demands secure communications, the L2oL3 tunnels should be established over encrypted L3 tunnels based, for example, on IPsec.

Evaluation

To demonstrate the viability of the proposed HFC interconnection framework, we have deployed a simple proof-of-concept scenario to interconnect two remote private network segments, one located in a fog node located at the Universidad Complutense de Madrid (UCM) premises, and the other one located in a public cloud provider, implemented as a Virtual Private Cloud (VPC) on Amazon AWS (EU West Region). The UCM fog node is structured as a low-to-medium size cloud, consisting of hypervisor nodes and dedicated storage for VM disk images. This infrastructure is managed with OpenNebula that provides the needed orchestration and multi-tenancy features. The fog node uses KVM instances; images are stored in a central repository and used from the local hypervisor disks; and network isolation is implemented through 802.1Q VLAN tagging. The goal of this proof-of-concept testbed is to analyze the throughput of L2 and L3 overlay networks deployed and configured over the physical infrastructure, using the HFC interconnection framework. The HFC agents used in this testbed have been deployed using *virtio* network interfaces and without any other additional optimizations or tuning.

For this purpose, we have measured the bandwidth of three different network configurations:

- The first configuration (Fig 3.a), which represents the baseline case, is a direct connection (no tunnel) between a VM attached to a private network deployed in the fog node, and an AWS instance (with a public IPv4 address assigned) deployed in Amazon. In this case, a NAT router configured in the private fog network allows the communications between the fog instances and the outside world.

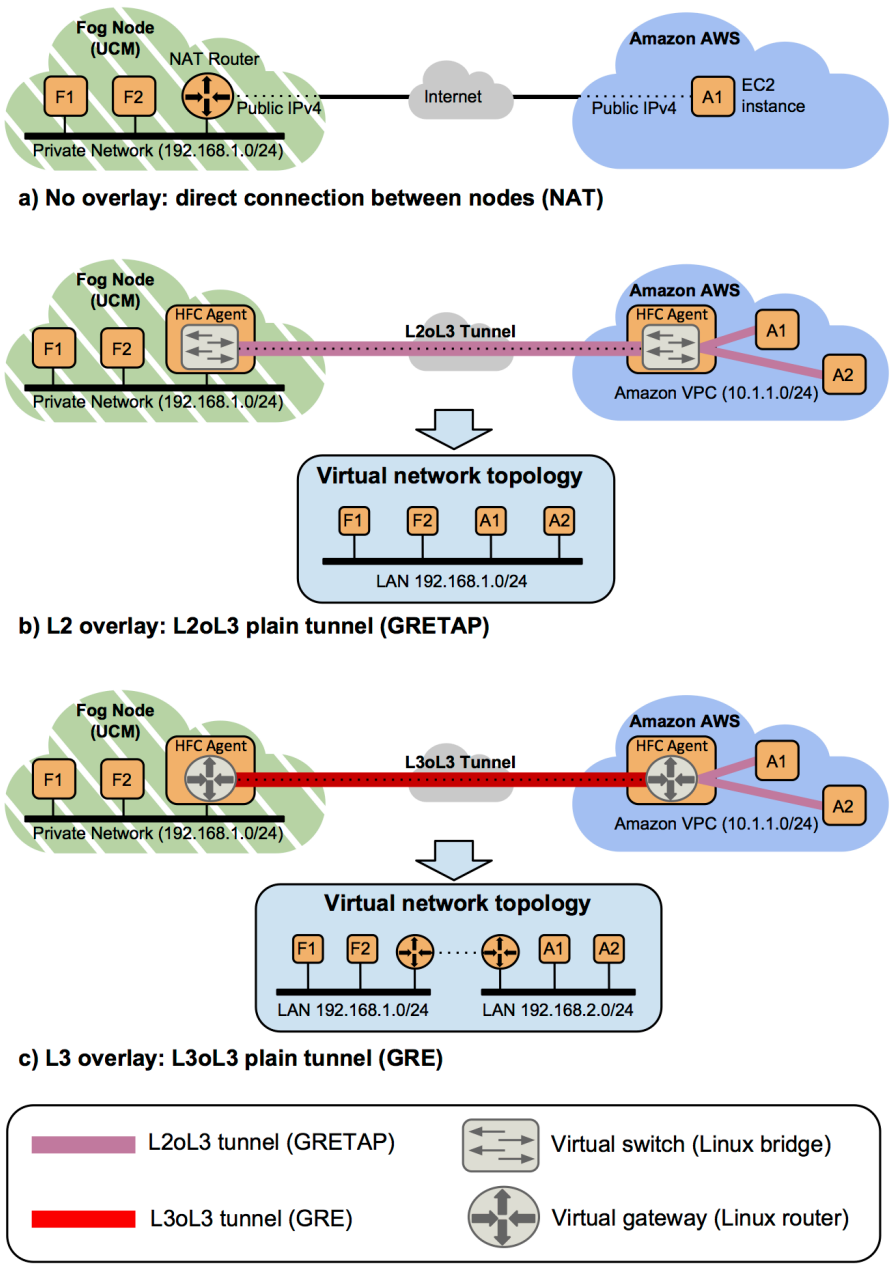

Figure 3. Network configurations.

- The second configuration (Fig. 3.b) is a L2 overlay network between the fog private network (address 192.168.1.0/24) and an Amazon VPC subnet (address 10.1.1.0/24). In this case, the HFC Agents deployed on each private network implement two virtual switches, based on Linux bridges, and configure a plain L2oL3 tunnel based on GRETAP to interconnect both network segments. In the Amazon side, due to the address filtering policies inside the Amazon VPC, the different instances must be connected to the HFC Agent using also GRETAP tunnels, making up a private overlay LAN on top of the VPC subnet. This enables to configure the Amazon instances with overlay IPv4 addresses belonging to the same address space than the fog private VMs (i.e. 192.168.1.0/24 network). With this configuration, VMs deployed on both sites (fog node and Amazon AWS) can communicate each other as if they were in the same L2 segment, as shown in the virtual network topology of Fig. 3.b.
- The third configuration (Fig. 3.c) is a L3 overlay network between the fog private network and the Amazon VPC subnet. In this case, the HFC agents implement two virtual gateways, based on Linux routers, and configure a plain L3oL3 tunnel based on GRE to interconnect both network segments. As in the previous case, in the Amazon side it is necessary to configure a private overlay LAN on top of the VPC subnet (using GRETAP tunnels between the HFC Agent and the cloud instances), to overcome the address filtering issue. This Amazon overlay network is configured with a different address space (network 192.168.2.0/24) than the

fog private network (network 192.168.1.0/24), so that VMs deployed on both sites (fog node and Amazon AWS) can communicate each other using virtual gateways as default routers, as shown in the virtual network topology of Fig. 3.b.

To compare the throughput of different network configurations, we have measured the real bandwidth for a 24-hour period by transferring a 100 Mbyte file every 10 minutes between a fog instance and an Amazon AWS instance (F1 and A1 in Fig. 3, respectively) using *Secure Copy Protocol* (SCP). Throughput results are displayed in Fig. 4, which shows the hourly average bandwidth (in Mbps) for each network configuration. This figure also shown the maximum achievable bandwidth for SCP transfers between a bare-metal host located at UCM and a public AWS instance located in Amazon EU West Region. In summary, the average throughput values and standard deviations measured for a 24-hour period for each network configuration are: 52.5 (± 6.5) Mbps for the baseline configuration; 42.1 (± 3.7) Mbps for the L2 overlay; and 48.3 (± 5.4) Mbps for the L3 overlay.

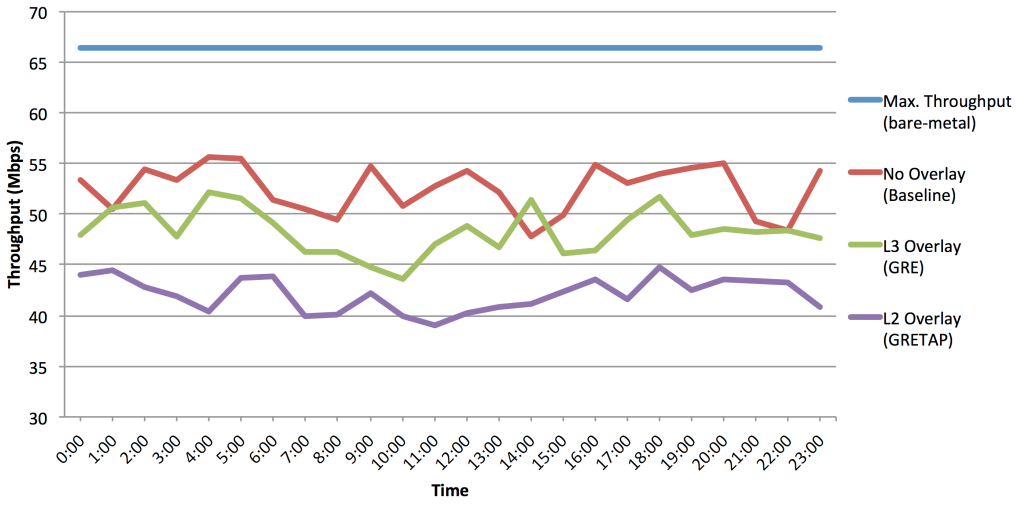

Figure 4. Throughput (hourly average) for different network configurations

As we can observe, the bandwidth reduction due to the overhead introduced by the virtual network devices (virtual switches and gateways) and the L2oL3 (GRETAP) and L3oL3 (GRE) tunnels is lower than 20% and 8%, respectively, which can be considered a reasonable overhead, and probes the viability of the proposed solution based on the HFC interconnection framework. Regarding the scalability issues of this framework, we must remember that it is based on a per-tenant approach, where each single tenant deploys its own HFC agents (one per site) to manage its own virtual networks, so the number of tunnels that each HFC agent has to configure and manage is limited, and it is not expected to observe any significant performance degradation when the number of fog nodes increases. On the other hand, as different tenants deploy and use different and independent HFC agents, these agents do not become a bottleneck when the number of tenants increases.

Conclusions

Most current public cloud providers offer different types of connectivity services, mainly based on L3 VPN technology, which allow the interconnection between the cloud network and remote networks deployed in a remote cloud or in the user premises. However, these services exhibit many limitations, such as heterogeneity, complex configuration, limited scalability, and lack of L2 support. In this paper we have presented a HFC interconnection framework that enables the agile, efficient and automated provision and management of virtual networks to interconnect different fog infrastructures and clouds, and allow both fog-to-cloud and fog-to-fog communications. This framework uses an agent-based solution that allows the interaction with different cloud providers and fog infrastructures, and provides advanced application demanding features, such as scalability, elasticity, security, multi-tenancy, and support for L2 and L3 connectivity.

Acknowledgments

This research was supported by the European Union's Research and Innovation Programme Horizon 2020 under the Grant Agreement No. 644048 (BEACON), and by Ministerio de Economía y Competitividad of Spain through research grant TIN2015-65469-P.

References

[1] F. Bonomi et al., "Fog Computing and Its Role in the Internet of Things," *Proceedings of 1st Workshop on Mobile Cloud Computing*, ACM, 2012, pp. 13-16.

[2] Y.C. Hu et al., "Mobile Edge Computing: A Key Technology Towards 5G," *ETSI White Paper,* No. 11, 2015.

[3] M. Satyanarayanan et al., "The Case for VM-Based Cloudlets in Mobile Computing," *IEEE Pervasive Computing* vol. 8, no. 4, 2009, pp. 14-23.

[4] R. Moreno-Vozmediano et al., "IaaS Cloud Architecture: From Virtualized Data Centers to Federated Cloud Infrastructures," *Computer,* Vol. 45, no. 12, 2012, pp. 65-72.

[5] F. Bonomi et al., "Fog Computing: A Platform for Internet of Things and Analytics," *Studies in Computational Intelligence*, Vol. 546, 2014, pp. 169-186.

[6] R. Moreno-Vozmediano et al., "BEACON: A Cloud Network Federation Framework," *Advances in Service-Oriented and Cloud Computing*, pp. 325-337, Springer, 2016.

[7] T. Narten et al., "Problem Statement: Overlays for Network Virtualization," *Internet Engineering Task Force*, RFC 7364, 2014.

[8] R. Jain et al., "Network Virtualization and Software Defined Networking for Cloud Computing: A Survey," *IEEE Communications Magazine*, vol. 51, no. 11, 2013, pp. 24-31.

[9] R. Mijumbi et al. "Network Function Virtualization: State-of-the-art and Research Challenges," *IEEE Communications Surveys & Tutorials,* vol 18(1), 2016, pp. 236-262.

[10] D. Farinacci et al. "Generic Routing Encapsulation (GRE)," *Internet Engineering Task Force (Network Working Group)*, RFC 2784, 2000.

[11] R. Kawashima et al., "SCLP: Segment-Oriented Connection-Less Protocol for High-Performance Software Tunneling in Datacenter Networks," *IEEE Conf. on Network Softwarization*, 2015, pp. 1-8.