Inter-domain Monitoring and Software-Defined Network Connectivity for Federated Infrastructures Management

Jose Gonzalez, Federico Alvarez
Universidad Politecnica de Madrid (UPM)
Madrid, Spain
jge@gatv.ssr.upm.es, federico.alvarez@upm.es

Luis M. Contreras, Oscar Gonzalez
Telefonica Investigacion y Desarrollo (TID)
Madrid, Spain
lmcm@tid.es, ogondio@tid.es

***Abstract*—This paper presents a solution for the management of federated infrastructures, based on the monitoring of performance among different domains and a controlled connectivity which follows the Software Defined Networking paradigm. The advances proposed are combining two innovative solutions for enhancing the management and control of infrastructures that belong to multiple administrative domains, but work collaboratively in a common federation, and enhance the quality of the offered service: a dynamic management of network connectivity based on the software-defined paradigm over multiple locations and the control of the heterogeneous environment performance with a unified monitoring framework including cross-domain network active monitoring and passive measurement. The combined solution has been tested in a multi-domain federation of infrastructures.**

Keywords—Federation, Infrastructure, Multi-domain, Management, Monitoring, Connectivity, Software-Defined Networking, Adapter, Controller, API, Network, QoS

I. Introduction

The conventional approach for connecting computing and network resources normally involves the burden of requiring a manual reconfiguration every time that a new capability is deployed within an infrastructure, creating inefficiencies and extra costs in the process. This manual setup leads to an evident deviation between business-level requirements—in terms of service connectivity and associated Key Performance Indicators—and the provision of the capabilities.

Despite it may result approachable when considering a domain controlled by a single infrastructure administrator, the procedure turns unfeasible in the case of configuring, operating and monitoring a distributed environment, where multiple federated administrative entities provide cross-domain services.

This paper attempts to introduce two innovative solutions that, working in a complementary manner and integrated under a unique Federation Platform, will enhance the inter-domain management of infrastructures:

- A Software-Defined Network (SDN) framework able to arrange, coordinate and manage the automated deployment of the traffic forwarding rules and the configuration of additional network resources for the service composition. To that end, some abstraction capabilities are needed to facilitate the programmability of the network in an agnostic way regarding the specific underlying infrastructure, allowing a rapid and dynamic response to application needs, changes in network conditions, and business policy enforcement.
- A multi-domain monitoring system capable of standardizing the access to the performance metrics, both in terms of computing and network capabilities. Such dataset shall be directly collected from the private monitoring systems of each particular infrastructure, aggregated accordingly to a common data model and exposed through standard interfaces. This document will assess in detail the service which is able to monitor the status of inter-domain connectivity.

The rest of this paper is organized as follows. In Section II we will elaborate on the description in which the federation of infrastructures is established to make the reader aware of the context. Section III will introduce the dynamic network connectivity based on the software-defined paradigm, whereas Section IV will present the unified multi-domain monitoring framework to control the federated performance. Finally, in Section V we will draw our conclusions.

II. Context

The federation outlined in this paper is a compendium of computing infrastructures, also called *nodes* or *regions*, belonging to independent administrative organizations. Such nodes provide, in a collaborative manner, multi-domain cloud-based computing services through a common Federation Platform. The reader shall be aware that the description of such platform is out of the scope of this paper and it will be addressed in future references.

The initial community of infrastructures creates a fully functional federation so that other potential nodes can join and become part of it. The work ensures definition of operational and technical requirements to be met by a joining node as well as supportive procedures to aid this node in the integration and deployment within the existing federation. Hence, this is a flexible environment for potential nodes that have reached minimum level of compliance and are ready for a certain commitment to the main principles and objectives of the federation.

From a technical perspective, three types of infrastructure capacities shall be considered:

- Computing capacities: infrastructures that provide hosting capacities for provisioning software resources (e.g. Data Centers);
- Data capacities: infrastructures that provide data sources that can be connected to applications (e.g. Smart Cities or Sensor Networks);
- Transport capacities: infrastructures that provide connectivity to support service provisioning and access to/from data and users (e.g. via National Research and Education Networks).

III. Software-Defined Network Connectivity

Currently, cloud management systems like OpenStack [9] enable to create overlay networks in a single infrastructure with a variety of technologies (e.g. GRE tunnels, VxLAN and VLANs). These networking capabilities are restricted to the devices directly managed by a single instance of the system, which is typically located in a unique region. However, in the federation scenario the information resources are distributed in geographically dispersed locations which are administrated separately forming different infrastructure domains. The creation and provision of a service involving multiple regions result in a number of control and management actions in each of the involved domains, interfacing with the specific control mechanisms present in each of the administrative domains.

Such heterogeneous environments can be highly benefitted from the existence of automated control mechanisms which can coordinate autonomously the resources present in every node of the federation. This can allow performing unified service operation, control and management as it was operated by a single organization.

SDN is proposed as the control mechanism capable of doing that because of its programmability capabilities. Nevertheless, the deployment in the proposed federation of a single SDN controller responsible for managing all the required connections is an approach that presents relevant constraints. The existence of different administrative domains and a large number of devices complicates the operability and compromises the scalability of the controller. Nevertheless it is yet required a logically centralized entity maintaining a comprehensive view of all the available network resources to orchestrate them for providing the connectivity service.

The way of doing that is by means of performing orchestration capabilities on top of separated SDN control domains, one per node of the federation. The orchestration will allow conjugate resources from different infrastructures composing true end-to-end services while keeping the local complexities being handled by the SDN controller existing in each domain. The orchestrator will maintain the service awareness and will interact with each of the SDN controllers in the federation responding to the tenants' demands.

A. *Network Controller Architecture*

The Network Controller proposed in this article will be assigned to play the role of such network orchestrator. As it is depicted in Fig. 1, the SDN service provisioning design proposed combines the Application-based Network Operations (ABNO) [3] and OpenNaaS [8] architectures. The coordination between these two functional approaches shall result crucial for a suitable implementation of the overall connectivity service.

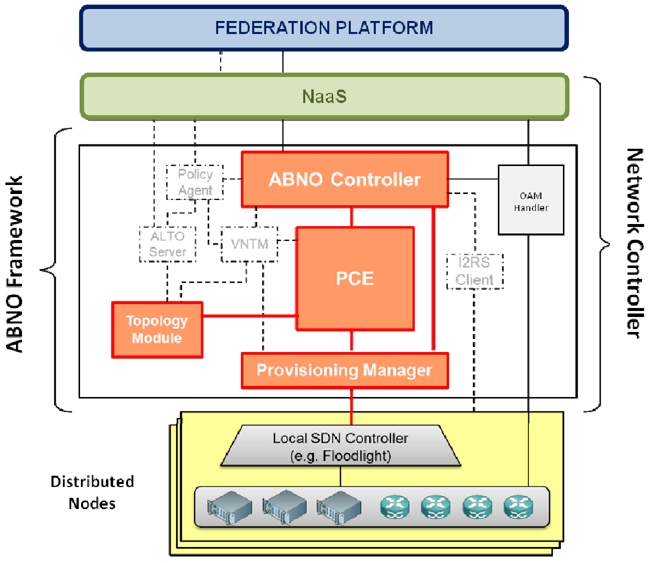

Fig. 1. Network Controller Architecture

1) The ABNO architecture has been proposed in IETF [3] as an SDN framework based on standard building blocks. From the whole set of functional components originally proposed, only some of them are meaningful in this particular case, and hence considered:

a) ABNO Controller, which is in charge of storing a repository of workflows for operations in the network (e.g. connectivity provisioning).

b) Path Computation Element (PCE) is the unit which handles the path computation across the network graph.

c) Topology Module (TM) handles databases with topology information for allowing advances services like traffic engineering.

d) Provisioning Manager (PM) is the module in charge of configuring the network elements by configuring the resources in each node. To do that, the PM will use the APIs of the local SDN controllers deployed in each node being a separate SDN domain.

2) The NaaS module which is required to provide connectivity service awareness across the federation. It is in charge of collecting and maintaining the information about the resources committed to the end user from the connectivity point of view, either if the resources are local to just one federation node or if they are spread in different nodes in the federation. One example of NaaS implementation is the *OpenNaaS framework* [8].

B. Multi-site Orchestration

The conventional approach for connecting computing resources leverages network segmentation based on VLANs to separate tenant services. This way of segmentation provides a limited number of configurable solutions per data center, leading to a careful planning of resources.

Multi-site services can be built under the concept of virtual patch-panels. In each location, ports on multiple switches (spread across the networks) can be programmatically connected among them to set up extended point-to-point connections, in a dynamic and automated way. It is also important to implement mechanisms capable of providing a dynamic on-demand scaling (up and down) of the resources offered to those services, and capable of automatically propagating any network change that could affect those services.

This logically centralized SDN framework suits the control of autonomous network environments, like the one formed by the service demarcation points for each data center. By offering a connectivity service API on top of the Network Controller, it is possible to enable the construction of such overlay networks to interconnect the data centers, independently of the virtualization technology used beneath.

The interconnection between regions can be accomplished by a mesh of overlay tunnels among the distinct locations (following either star or hub-and-spoke topologies), so different paths can be selected from a centralized controller to accommodate the end-user flows according to their needs, in the most efficient way.

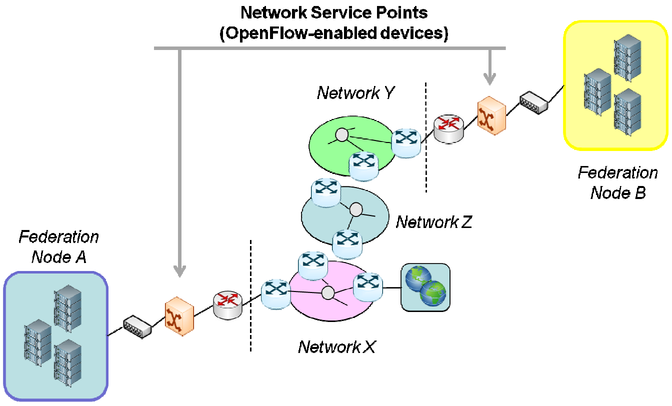

Fig. 2. End-to-end service from a point-to-point multi-site L2 connection

From the data plane perspective, OpenFlow-enabled switches [7] will act as demarcation points to forward the tenant traffic towards the network service provider router according to the rules installed in the switches. The capabilities for handling the traffic by the demarcation points are determined by the capabilities that such switch can provide. The Network Controller will dynamically instruct the OpenFlow switch by modifying the traffic flow tables to take the actions needed to prepare the frames before delivering them to the border router which delivers the traffic on top of the overlay network connecting the federation nodes.

IV. Unified Multi-Domain Monitoring Framework

Evaluating the heterogeneity of the context previously described, we assume that infrastructure administrators leverage on private Network Management Systems (NMS) [13] to control the performance of their specific nodes. Since these systems are designed to run within a single domain, they are generally packaged as integrated solutions.

In order to overcome a unified multi-domain monitoring framework, we propose an extended architecture where two additional layers are embedded in the traditional single-domain approach (**¡Error! No se encuentra el origen de la referencia.**). Attached to the monitoring systems already configured, we propound the inclusion of an adaptation mechanism denominated as Infrastructure Monitoring Middleware (IMM). Such abstraction layer will standardize the format and the accessibility—through common Application Programming Interfaces (APIs) —to the data collected from the different systems. On top of the several IMM instances to be deployed, the Federation Monitoring—which is included within the Federation Platform—will be the component in charge of aggregating and handling the distributed dataset along the federation, and finally publish the information through a Graphical User Interface.

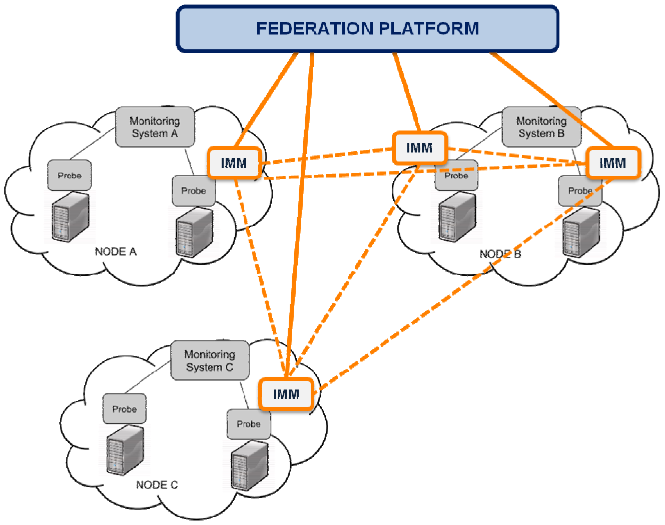

Fig. 3. High-level Overview Infrastructure Monitoring Middleware

Since a significant assessment of this architecture has been already accomplished by reference [14], this paper aims to elaborate more in detail the specific IMM service which provisions the monitoring of inter-domain connectivity within the federation.

A. *Background and Prior Art*

With the increase of distributed computing over multiple administrative domains came the requirement to measure network performance characteristics and share this information between the users and service providers. This was addressed by the Open Grid Forum Network Measurement and Network Measurement and Control working groups [6], which defined a set of protocols standards for sharing data between measurement and monitoring systems, often called the NMWG protocol. PerfSONAR (PERFormance focused Service Oriented Network monitoring Architecture) [12] is a framework that implements these protocols for both regular periodic observations, useful for forming historical records, and for making on-demand measurements to aid problem solving and resolution.

Each PerfSONAR service uses specific tools to perform the measurement of the network characteristic between the selected end-points. For instance, *Iperf* [5] is used by the *BWCTL* service [1] to fulfil TCP or UDP achievable throughput measurements; and the *owping* tool [11] is used to retrieve one-way delay, jitter and packet loss measurements by the *OWAMP* service [10].

B. *Inter-domain Monitoring Architecture*

The Infrastructure Monitoring Middleware represents the first abstraction mechanism in our unified monitoring framework. In order to assure multi-domain feasibility, this layer shall follow a joint design and obey common architectural principles. As it is depicted in Fig. 3, IMM will be enclosed between the distributed set of monitoring systems deployed in each node—more specifically, attached to the monitoring probes of such systems— and the upper Federation Layer of the architecture.

It is important to notice that just a limited bundle of computing resources will be dedicated from each region to the federated services. Hence IMM instances are only intended to be deployed on some specific hosts (either virtual or physical ones). Since these capabilities are required to support cross-domain federated services, the unified monitoring framework shall enable the establishment of end-to-end control tests along this distributed set of points of interest to check their connectivity performance.

Connectivity monitoring service relies on the capacity to inject test packets and following them to measure the service provided. The volume and other characteristics of the introduced traffic are fully adjustable, what implies testing what is required, and when is needed. This emulation of scenarios will enable to check if both Quality of Service (QoS) and Service Level Agreements (SLAs) are accomplished according to the real data obtained.

C. *Network Active Monitoring Adapter*

The Network Active Monitoring (NAM) Adapter is the software component in charge of active measurements among IMM instances, providing a mechanism able to handle latency and bandwidth-related tests. This implementation is the basis for monitoring inter-domain connectivity among regions.

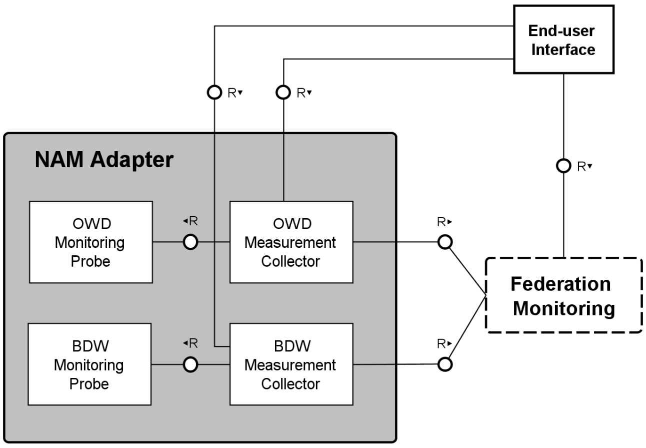

Fig. 4. Network Active Monitoring Adapter Architecture

Monitoring data is obtained by either live or periodic requests. Historical measurements represent results of regularly scheduled tests and shall cover one-way delay, jitter, one-way packet loss, and achievable throughput for a given path. Nevertheless, NAM Adapter also offers the possibility of requesting an on-demand measurement of achievable throughput or one-way latency measurement between certain endpoints.

Fig. 4 depicts the diagram of how NAM's main modules and its surrounding context interact with each other.

1) Monitoring Probes: which are the tools used to actually perform the measurement tests between given end-points of interest. These modules provide the Measurement Collectors with the raw network monitoring data. The interface to interact with them is command line-based.

To assure reachability, NAM implementation requires the inclusion of a pair of probes by default:

a) One-Way Delay (OWD) Monitoring Probe: in charge of managing one-way delay tests. Leveraging on PerfSONAR's *OWAMP* service [10], NAM's OWD Probe overcomes some existing functional requirements, which in terms of efficiency are not optimal, to enhance the operability.

b) Bandwidth (BDW) Monitoring Probe: following the Internet 2's PerfSONAR distribution with regards to bandwidth tests (*BWCTL* [1]), NAM's BDW Probe is based on the network throughput tool *Iperf* [5].

2) Measurement Collectors: the core modules within the adapter. They will be the responsible actors for collecting the data generated by the probes, processing and forwarding it to the upper layer via a REST-based Web Service API. There are also two types—OWD and BDW—according to the data they are required to handle.

A collector is composed of the following sub-modules (Fig. 5):

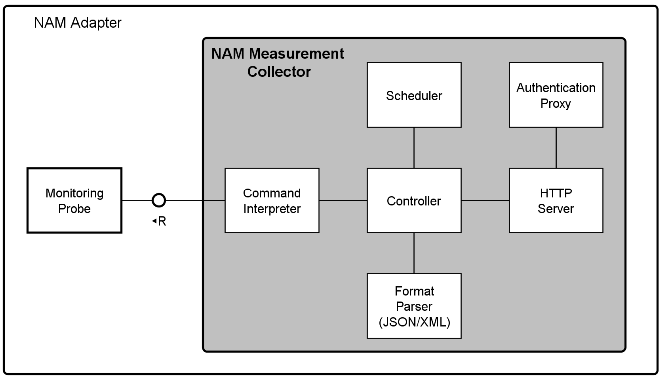

Fig. 5. NAM Measurement Collector - Internal Architecture

a) Command Interpreter is in charge of dealing with the probe through command line-based operations.

b) Format Parser, which adjusts the result obtained from the command to a standard response, e.g. JSON or XML format.

c) Scheduler is the element responsible for the timing in scheduled tests, triggering the process when the setup time slot is reached.

d) HTTP Server will handle the exchange of request/responses.

e) Authentication Proxy, which intercepts external requests to validate if they represent an authenticated access to the services through OAuth-based [2] mechanisms.

f) Controller is the central entity which manages the sub-modules.

Each node in the federation requires the presence of at least one instance of this component to be reachable by other nodes. Nevertheless, it is up to the infrastructure administrator to deploy more instances and provide a more fine-grained view to avoid single points of failures.

The reader shall be aware that the installation of such adapter is recommended to be performed in physical resources to assure stable conditions. Although the deployment in a virtual host is a feasible possibility, and perhaps a more suitable option to manage, the instance may carry inaccuracies with the obtained values.

V. CONCLUSIONS

In this paper we outlined two complementary solutions designed to enhance the management of federated infrastructures spread across multiple domains. Since a traditional single-domain approach cannot be contemplated in this context, and the current state of the art only considers separated paradigms such as SDN, we need to integrate solutions to operate a common Federation Platform. Therefore, these frameworks were proposed as abstraction layers to provide inter-domain services, considering that computing and network capabilities require to be controlled dynamically and monitored through standard means.

REFERENCES

[1] BWCTL, “Internet 2 Bandwidth Test Controller (BWCTL)”, Website, available online at http://software.internet2.edu/bwctl/, last visited on April 30, 2014. http://software.internet2.edu/bwctl/

[2] D. Hardt, Ed., “The OAuth 2.0 Authorization Framework”, IETF RFC 6749, October 2012. Available online at: http://tools.ietf.org/html/rfc6749

[3] D. King and A. Farrel, “A PCE-based Architecture for Application-based Network Operations”, IETF Internet-Draft, October 2013. Available online at http://tools.ietf.org/html/draft-farrkingel-pce-abno-architecture-06

[4] E. Haleplidis, S. Denazis, K. Pentikousis, J. Hadi Salim, D. Meyer and O. Koufopavlou, “SDN Layers and Architecture Terminology”, IETF Internet-Draft, March 2014. Available online at http://trac.tools.ietf.org/html/draft-haleplidis-sdnrg-layer-terminology-04

[5] Iperf, “iperf3: A TCP, UDP, and SCTP network bandwidth measurement tool”, GitHub repository, available online at http://code.google.com/p/iperf/, last visited on April 30, 2014.

[6] OGF, “Open Grid Forum”, Website, available online at http://www.ogf.org/, last visited on April 30, 2014.

[7] OpenFlow, “OpenFlow Switch Specification, ver. 1.3.3”, Open Networking Foundation, September 2013. Available online at https://www.opennetworking.org/images/stories/downloads/sdn-resources/onf-specifications/openflow/openflow-spec-v1.3.3.pdf

[8] OpenNaaS, “Open Platform for Networks as a Service”, Website, available online at http://opennaas.org/, last visited on April 30, 2014.

[9] OpenStack, “Open source software for building private and public clouds”, Website, available online at https://www.openstack.org/, last visited on April 30, 2014.

[10] OWAMP, “Internet2 One-Way Ping (OWAMP)”, Website, available online at http://software.internet2.edu/owamp/, last visited on April 30, 2014.

[11] OWPING, “owping: client application to request one-way latency tests”, Website, available online at http://software.internet2.edu/owamp/owping.man.html, last visited on April 30, 2014.

[12] PerfSONAR, “PERFormance focused Service Oriented Network monitoring Architecture”, Website, available online at http://www.PerfSONAR.net/, last visited on April 30, 2014.

[13] R. Khan, S. Ullah Khan, R. Zaheer, and M. Inayatullah Babar, “An Efficient Network Monitoring and Management System”, International Journal of Information and Electronics Engineering, Vol. 3, No. 1, January 2013. Available online at http://www.ijiee.org/papers/280-N011.pdf

[14] Y. Al-Hazmi, J. Gonzalez, P. Rodriguez-Archilla, F. Alvarez, T. Orphanoudakis, P. Karkazis, T. Magedanz, “Unified Representation of Monitoring Information Across Federated Cloud Infrastructures”, unpublishedpaper submitted to the Workshop on Federated Future Internet and Distributed Cloud Testbeds 2014.