# GÉANT World Testbed Facility

## Federated and distributed Testbeds as a Service facility of GÉANT

**Fabio Farina**
Consortium GARR – The Italian Academic and Research Network
Rome, Italy
fabio.farina@garr.it

**Peter Szegedi**
Trans-European Research and Education Networking Association (TERENA) – Secretariat
Amsterdam, Netherlands
szegedi@terena.org

**Jerry Sobieski**
Nordic Infrastructure for Research & Education (NORDUnet)
Copenhagen, Denmark
jerry@nordu.net

***Abstract*—Global network innovation requires large-scale distributed test facilities that are similar to the typically multi-domain real-world environments in order to ensure the agile adaptation of new concepts, architectures, technologies and protocols from prototyping through testing into production. Virtualization, in general, allows network researchers to create insulated autonomous slices of production environments that have the required scale and at the same time provide a safe and secure playground to carry out disruptive research by simultaneous user groups without affecting each other's experiments. The GÉANT World Testbed Facility is leveraging a distributed infrastructure with well-defined domain boundaries and federated authentication, authorization and access control on the underlying resources. The highly programmable nature of the facility allows researchers to create a complex hierarchy of atomic resources and composite testbeds consisting of computing, storage and dynamic networking elements instantiated on demand.**

***Keywords—Software Defined Networking, OpenFlow, Network Function Virtualization, Testbeds as a Service, GÉANT Word Testbed Facility, GN3plus Project***

## I. INTRODUCTION

Virtualization, in computing, traces its roots back to the 1960s when mainframes were logically partitioned for different applications to increase efficiency and spare space and power in the datacenters. By the beginning of the 2000s virtualization had reached the desktop and later the application layer. Network virtualization is the process of combining hardware and software network resources and network functionality into a single, software-based administrative entity. Network virtualization is the vital component for application development and testing environments.

PlanetLab [1], initiated in 2002, was the first open platform for developing, deploying and accessing planetary-scale services. Although PlanetLab is still a popular tool for networked services research, it may not be used for research on lower layer network protocols and architectures. The GENI concept [2], started in 2007 in the US, addressed these shortcomings of PlanetLab and extends its virtualization principle to the whole diameter of the infrastructure with the notions of substrate, slicing and federation. The FEDERICA project [3] in the EU (2008-10) followed an architecture similar to GENI's and, as there was an emerging user requirement, ensured the reproducibility of the complete testing environment and the repeatability of the experiments at a different location or time. FEDERICA leveraged the tools and techniques previously developed by other European projects such as MANTICORE, GEYSERS or UCLP [3]. Eventually, the infrastructure was adopted by the GN3 project in 2011 as the GÉANT Network Factory [4].

By that time, programmability of computer networking resources had become a major user requirement, which led to the development of the Software Defined Networking (SDN) concept [5] evolved from academic research in the US. SDN allows network administrators to manage network services through abstraction of lower level functionality. This is done by de-coupling the system that makes decisions about where traffic is sent (the control plane) and the underlying systems that forward traffic to the selected destination (the data plane). SDN requires some method for the control plane to communicate with the data plane. One such mechanism, OpenFlow [6], gives access to the forwarding plane of a network switch or router over the network. Version 1.1 of the OpenFlow protocol was released in 2011 and adopted by many testbed projects around the world. The OFELIA project [7] in the EU (2010-13) developed a Control Framework (OCF) based on OpenFlow that is, used by the GÉANT OpenFlow Facility [8], among other projects. The FED4FIRE


978-0-9836283-9-2 © 2014 ITC

integrating project [9] (2012-16) applies the federation concept to the various purpose-built OpenFlow-capable testbeds supporting Future Internet Research and Experimentation (FIRE) in Europe.

Having first appeared in 2012, the Network Functions Virtualization (NFV) concept [10] addresses the users' needs for virtualizing entire classes of network node functions into building blocks that may be connected, or chained, together to create communication services. While SDN is focused on the separation of the network control layer from its forwarding layer, NFV is focused on porting network functions to virtual environments on top of industry-standard high-volume servers, switches and storage, or even cloud computing infrastructure, instead of having custom hardware appliances for each network function. One of the unique features of the GÉANT World Testbed Facility – designed, developed and deployed by the GN3plus project's SA2 [11] Testbeds as a Service activity – is the ability to create a complex hierarchy of atomic resources and composite testbeds in a highly programmable manner. The GÉANT World Testbed Facility combines the emerging concepts of SDN and NFV and follows the path paved by landmark initiatives such as GENI, FEDERICA, OFELIA and other FED4FIRE testbeds.

The paper is organized as follows: after this brief overview on the emerging network research ecosystem, the high-level service description of the Testbeds as a Service is given, then some details of the architecture design and engineering plan of the GÉANT World Testbed Facility is discussed highlighting its unique features. The paper, finally, elaborates on the service development roadmap and summarizes the major conclusions and challenges ahead.

## II. Testbeds as a Service - Description

The GN3plus Project's SA2 - Testbeds as a Service (TaaS) activity constitutes a new GÉANT production service intended to offer customized experimental network facilities to the network research community for the express purpose of testing novel networking and telecommunications concepts, at scale and across a geographically realistic European footprint.

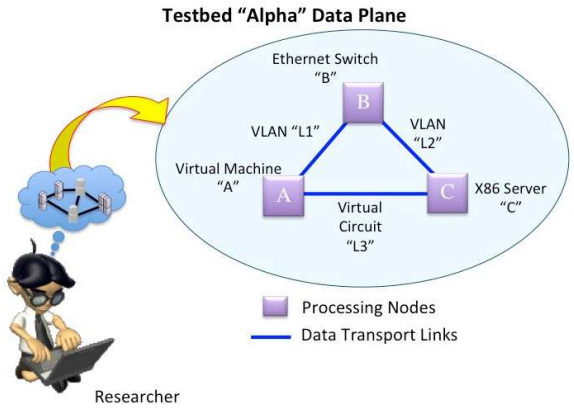


*Fig. 1 Researcher describes his testbed*

The TaaS builds dynamic, virtual networks constituting the GÉANT World Testbed Facility. These networks are constructed from components of the shared infrastructure that have been delegated to the TaaS. These base components consist of data transport elements, switching and/or forwarding elements, computational and storage elements. These components are virtualized or partitioned to provide a pool of tightly prescribed resources that can be allocated to a testbed, and whose control can be further delegated to the research activity associated with the testbed. These packet testbeds are insulated from each other and the experiments do not unduly affect each other or the core GÉANT production services. Some of the key TaaS characteristics are highlighted as follows:

### A. *Programmability*

The network research community has access to the testbeds via a Command Line Interface (CLI) and an interactive web-based Graphical User Interface (GUI) that presents a more human interface to the service. The testbeds, containing a list of resources and connectivity relations that are allocated to the given testbed, must be described via an electronic document using a TaaS Domain Specific Language (DSL) descriptor. Using the power of the state-of-the-art object-oriented programming languages, the DSL allows the researchers to define resource classes, objects, iterators, etc. in order to efficiently describe their testbeds (Fig. 1). The testbed service engine takes the network testbed description, reserves the necessary resources, and then instantiates the testbed. The user is then given control of the resources within the testbed in order to initialize the resource elements and then to perform experiments within the context of that testbed without fear of causing harm to other aspects of the network infrastructure. The users have the ability to override any activity in their testbeds via the meta-control interface in order to manage the testbeds properly. And the testbeds operation staff will also have the ability to shut down or freeze any testbed experiment that might go awry.

### B. *Reproducibility*

Each testbed's DSL description should include all performance characteristics that the testbed resources are expected to meet. Thus, given a particular testbed description, the TaaS agent creates a testbed instance whose resources meet the specified characteristics. Where resource descriptions specify minimum or maximum values for particular resource characteristics, conforming resources selected for use within a testbed instance may vary from one instance to the next. While this provides similar testbeds in terms of their specification, it does not per se guarantee identical testbeds simply by merit of a common testbed description document. So it is important that the client specifies resource characteristics that will result in consistent resource selection where experimental reproducibility is desired. Further, since reproducibility is a goal, the initial TaaS endeavors to offer resources that can provide consistent performance in different instantiations as part of Phase 1, and more fine-grained and configurable resources will be forthcoming over time.


978-0-9836283-9-2 © 2014 ITC

### C. Composable Testbeds and Resources

In order to simplify large-scale testbed specifications and to hide complexity and minimize errors, the TaaS allows users to construct "Composable Testbeds" using object-oriented constructs. A testbed description document can be inserted into the Resource Database to act as a resource template itself. Presumably, the testbed has been previously instantiated and tested and is known to function as intended. But then, having done this, it is extremely useful to be able to replicate that testbed descriptor for use in other more sophisticated testbed environments. This recursive object-oriented process provides an important capability for the researcher – it allows the researcher to construct very complex testbeds using customized and tested libraries.

### D. Multi-domain Interoperability

It is important that the TaaS be able to construct testbeds that extend beyond the scope of the GÉANT service domain. This means that the GÉANT TaaS must ultimately be able to construct testbed instances that are comprised partially (or wholly) of resources from other domains. These domains may be NRENs (National Research and Education Networks) from within the GÉANT project or other service domains beyond the GÉANT core partners. And likewise, services based outside the GÉANT domain should be able to similarly construct testbeds or other similar virtual networks that include resources that are part of the GÉANT domain.

### E. Structured Resource Database

A resource in the TaaS is a unit to be allocated. All resources are virtualized – i.e. there is a software layer that defines the capabilities and manages the resource. The Resource Database is a structured repository containing information about each resource available to the GÉANT World Testbed Facility. Resources may be physical objects presented as a virtualized resource, or may be objects dynamically created as necessary. In either case, the resource type and its capabilities are described for each resource type and/or instance in the Resource Database. Examples of resources might be "bare metal" PC server blades virtualized as "X86 PC", or a physical Ethernet switch virtualized as an "Ethernet switch". Examples of dynamically instantiated resource might be virtual machines or network connections between other resources (e.g. VLANs).

### F. User Registration and Authorization Policy

The TaaS GUI allows new users to register to use the service via access federations. Initially, we anticipate a low bar to registration allowing the target research community to easily engage and to begin learning how to use the service. The service management team is responsible for initializing access policy and approving more advanced user authorization requests. In general, users will be assigned an "allocation" to their account. When this allocation is used up, the user must apply for additional allocations. Each resource will be assigned a "cost" or a rate at which the allocations are consumed within the TaaS. This initial accounting process will be reviewed and may be modified as necessary to fit the accounting requirements for GÉANT and to meet the community's needs.

### G. Service Operations and Support

The GÉANT TaaS is a production service. This means that the ability to define and instantiate testbed instances is treated as a 24 by 7 by 365 operational service to the GÉANT community. The resources and infrastructure required to field the service must be reliable and secure and able to deliver the service capabilities described in this document. The GÉANT Network Operations Centre is tasked with monitoring the availability of the service infrastructure and normal operational status of the service. The broader GÉANT Core and Partners are offered technical training in the service deployment engineering and operations. This training helps these organizations plan for the deployment of this service within their domain. User training on how to use the TaaS is offered to the research community.

## III. Architecture and Engineering

The most fundamental high-level concept in the TaaS is the notion of "testbed". A testbed is a collection of processing resources ("nodes") that are interconnected by a set of data transport resources ("links") to form a topology. The processing nodes operate upon data from inbound transport links and a set of internal information/rules. These processing nodes generate outbound data stream(s), which deliver those streams to other nodes on other transport links. The data transport links simply transport data - transparently and unmodified - from one node to another. These processing nodes and transport links constituting the testbed data plane are treated in the same way, meaning that both nodes and links have ports and their logical relations are represented by the port adjacencies (Fig. 2).

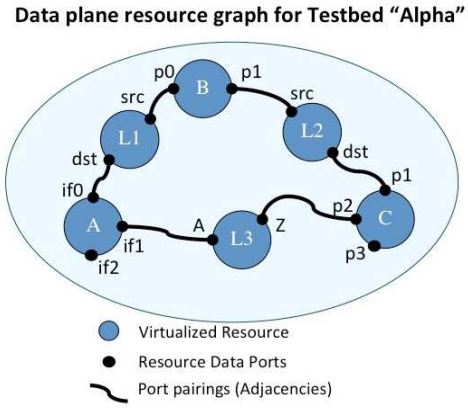


*Fig. 2 Resource graph with ports and adjacencies*

The functional set of agents and protocols that allow the user to acquire those resources, to schedule and to manage the state of those resources constitute the testbed control plane (Fig. 3). The key functional components are:

- The Testbed Control Agent (TCA)
- The Resource Manager (RM) and the Resource Database (RDB)


978-0-9836283-9-2 © 2014 ITC

- The Resource Control Agents (RCA)

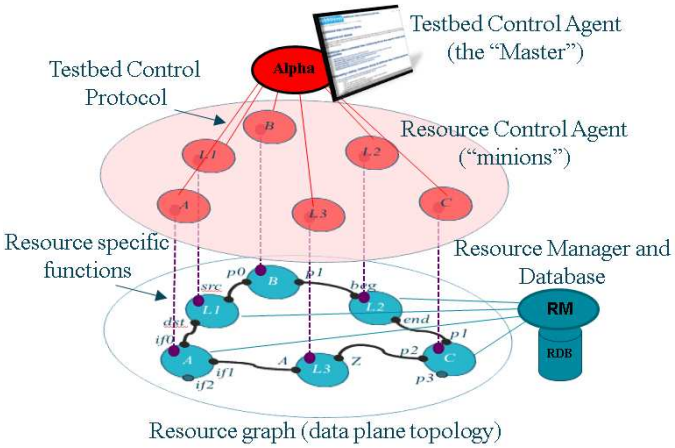


*Fig. 3 Functional components of the control plane*

The architecture classifies resources as "atomic" or as "composite" resources. Every resource is described by a resource descriptor and permits a set of control primitives:

- Reserve() – This method is invoked when the resource is reserved.
- Activate() – This primitive is invoked when the resource is to be placed in service.
- Query() – Returns a resource-specific data structure with information about the state of the resource instance;
- Deactivate() – Removes a resource instance from active service. The resource remains allocated to the user/project, but in a reserved state;
- Release() – releases the resource back to the available pool and terminates the reservation.

At reservation start time, the RM will activate the resource. The RCA will issue type- and instance-specific commands to the service infrastructure to initialize the resource instance in question. Upon activation, control plane connectivity is verified between the RCA and both the user TCA agent and the RM via a hello primitive. The control plane allows either the TCA or the RM agents to monitor and/or control the resource instance. The master TCA is a user agent that manages and/or orchestrates the overall user testbed network. A general purpose TCA provides a GUI front-end facing the user that allows the human to define the testbed elements, the attributes of those testbed elements, and their topology. The user also indicates how these testbed elements will be arranged by indicating which interface ports on each resource are connected to which other interface ports on other resources. From this interaction, the GUI constructs an inventory of resources required for the testbed and is able to resolve many unspecified or defaulted attributes of those resources.

### *A. Resources lifecycle and resource database*

Once the dependencies are known, the agent, processing the description, can begin reserving the resources. For each resource defined in the testbed description the TCA issues a Reserve() primitive to the RM. The RM is the agent representing the GÉANT World Testbed Facility. The RM searches the RDB, and finds and allocates a conforming resource. The RM will confirm the reservation to the TCA and returns a resource instance descriptor for each allocated resource. The TCA updates the testbed description to reflect the state of the resource (Fig. 4).

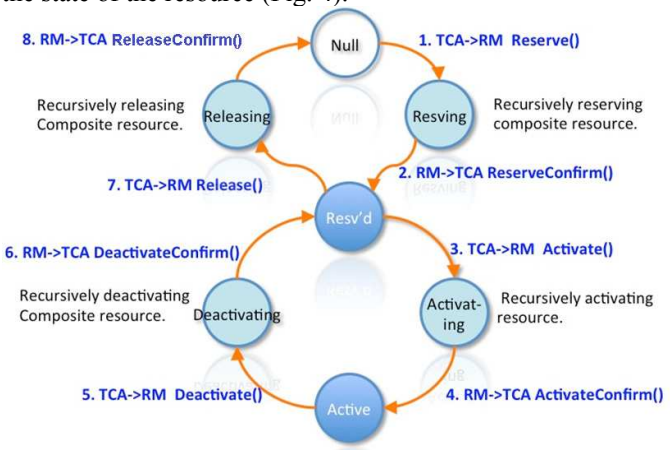


*Fig. 4 Resource lifecycle state machine*

All control primitives are specified in the resource type descriptor in the provider's RDB, including the common control primitives that all resources must recognize. When a control primitive is sent to the RM, the RM performs the common actions associated with that control primitive, and then invokes the associated resource specific primitive via the RCA.

In the architecture proposed, an RCA is an independent process (or agent) that is separate from the RM. The architecture allows the user TCA to interact directly with the RCA associated with a particular resource instance. The separation of the general service management functions of the RM from the resource specific control functions of the RCAs associated with each resource instance allows the user's overall testbed instance to function independently of the provider's RM. The separation of RM and RCA allows for a distributed reservation and control model for the testbed architecture.

Moreover, each resource is scheduled. This means that a time window is associated with the allocation of that resource, and the resource is only available during that window, allowing the provider to manage resources and bind their availability to very specific policy.

### *B. Infrastructure and virtualization concept*

The infrastructure design of the GÉANT World Testbed Facility's substrate is an engineering exercise. Given the service specification, and the service architecture, it will be necessary to identify appropriate hardware and software resources that will comprise the initial infrastructure components for TaaS. This includes planning and acquisition of co-location facilities at selected core GÉANT PoPs, and in the PoPs of participating NRENs. It also includes hardware and software specifications, installation, and operational configuration for the processing nodes that will be available.


978-0-9836283-9-2 © 2014 ITC

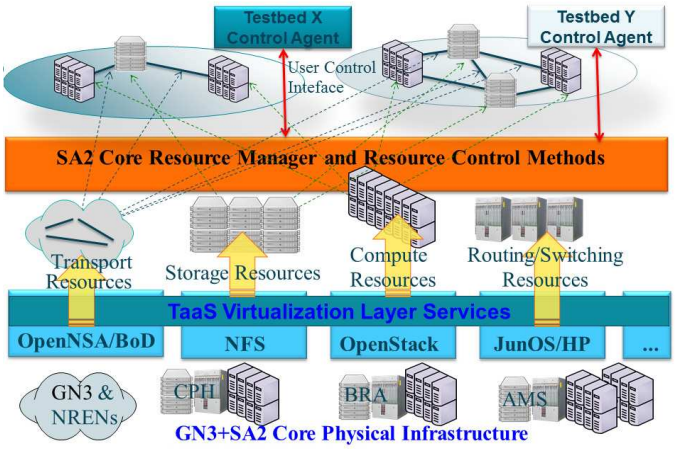


*Fig. 5 Infrastructure virtualization concept*

In the GÉANT domain, OpenStack cloud software components [12] have been selected for compute node virtualization. As OpenStack is widely deployed in various public and private cloud infrastructures it allows TaaS to extend its federated model beyond GÉANT to other cloud facilities powered by OpenStack. The OpenNSA implementation [13] of the protocol agnostic Network Service Interface (NSI) standard has also been chosen for provisioning network links. Since many dynamic network provisioning systems around the world comply with NSI – including the GÉANT Bandwidth on Demand (BoD) service – it also facilitates multi-domain interoperability. The initially available Juniper routers and OpenFlow capable HP switches provide their proprietary virtualization software integrated to TaaS (Fig. 5). The physical substrate may feature other types of resources as more and more PoPs are deployed at various NRENs. The core PoPs that are going to be deployed in the initial phase of the project are located in Cambridge (UK), Copenhagen (Denmark), Bratislava (Slovakia), Amsterdam (Netherlands) and Ljubljana (Slovenia).

## IV. Development Roadmap

The BETA Version 1.0 of the GÉANT TaaS has been operational since May 2014 and available for early adopters. It features a pre-defined set of resources in the RDB, including Linux based virtual machines, Ethernet framed virtual circuits, and virtualized hardware OpenFlow switch fabrics. Other resources such as storage capacity will initially be available as part of the virtual machines and may be separated out as individually configurable resources as the TaaS evolves.

Later in the Fall of 2014, more flavors of the virtual machine resources will be introduced (e.g., software-based OpenVSwitch fabrics) and dynamic virtual port allocation will be enabled on the servers. The TaaS will allow resources (such as block storage devices or bare metal servers) to be incorporated from multiple resource providers such that a testbed instance may span multiple similar service providers. NSI compliant Bandwidth-on-Demand virtual circuits will also be enabled ensuring inter-domain interoperability, possibly with GENI in the US and other similar facilities. The geographical footprint of the facility will gradually be extended inside the GÉANT core and also penetrating into the NRENs' networks in Europe.

## V. Conclusions and Challenges

The federated and distributed GÉANT World Testbed Facility, architected and deployed by the GN3plus project's SA2 Testbeds as a Service activity, combines the emerging concepts of SDN and NFV and follows the path paved by other testbed projects worldwide. It responds to the request of the Future Internet community for a European-scale, production-quality service where innovative ideas can be assessed against distributed virtual infrastructures. The GÉANT TaaS has the goal of being able to construct testbeds that comprise resources from other domains, and symmetrically it must be able to act as a resource provider for other federated testbeds focusing on the simplicity of use and the expressiveness of its object-oriented hierarchical testbed concept.

As the TaaS concept turns into concrete service implementation, challenges obviously arise. Today, OpenStack and OpenFlow (primarily designed for datacenter applications) do not support the creation of point-to-point, multi-hop, multi-domain bit pipes that the users can configure in their testbeds, choosing for example arbitrary or stacked VLANs and full control of IP networks. Maintaining consistent topology information is one of the key challenges in a global testbed environment. In the second year of the GN3plus project, the SA2 TaaS activity will work with other virtual network service architectures to develop a common global inter-operability standard.

## Acknowledgment

The FP7 Integrated Infrastructure Initiative project GN3plus is partially supported by the European Commission under the Grant Agreement No.: RI-605243. The authors acknowledge the contribution of all project partners and specifically for their input Błażej Pietrzak (PSNC) and Michal Hažlinský (CESNET).

## References

[1] L. Peterson, T. Anderson, D. Culler, and T. Roscoe, "A Blueprint for Introducing Disruptive Technology into the Internet" (HotNets-I '02), October 2002.

[2] J. B. Evans, D. E. Ackers, "Overview of GENI Overview of GENI and Future Internet in the US", 22 May 2007.

[3] P. Szegedi, M. Campanella, V. Maglaris, S. Figuerola, C. Cervello-Pastor, "With Evolution for Revolution: Managing FEDERICA for Future Internet Research", IEEE Communications Magazine, Vol.47 No.7 pp. 34-39, July 2009

[4] V. Ajanovski, K. Baumann, A. Chuang , F. Farina, et.al. "Network Factory Footprint", GN3 JRA2 Project Deliverable DJ2.5.1, November 2011.

[5] ONF White Paper, "Software-Defined Networking: The New Norm for Networks", April 13, 2012.

[6] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, J. Turner, "OpenFlow: Enabling Innovation in Campus Networks", March 14, 2008.

[7] A. Köpsel, H. Woesner, "OFELIA - Pan-European Test Facility for OpenFlow Experimentation", November 22, 2011


978-0-9836283-9-2 © 2014 ITC

[8] G. Androulidakis, Ch. Argyropoulos, K. Baumann, J. Jofre, A. Sevasti, "GÉANT OpenFlow Facility Design", November 2012.

[9] Piet Demeester, "FED4FIRE General Introduction", FIRE Engineering Workshop, November 2012.

[10] M. Chiosi, D. Clarke, P. Willis, A. Reid, et. al., "Network Functions Virtualisation – Introductory White Paper", SDN and OpenFlow World Congress, Darmstadt-Germany, October 22-24, 2012.

[11] J. Sobieski, "Testbeds as a Service - Building Future Networks, A view into a new GEANT Service", GLIF Tech, Atlanta, Mar 18, 2014.

[12] Frank J. Ohlhorst, "OpenStack: An Overview", November 2012.

[13] R. Krzywania, J. A. García -Espín, C. Guok, I. Monga, J. vd Ham, T. Kudoh, J. MacAuley, J. Mambretti, G. Roberts, J. Sobieski, A. Willner, "Network Service Interface - gateway for future network services", TNC2012, May 2012.


978-0-9836283-9-2 © 2014 ITC