

17-08-17

Deliverable D7.1 Overview of SDN Pilots Description and Findings: Part A

Deliverable D7.1

Contractual Date:	30-04-2017
Actual Date:	17-08-2017
Grant Agreement No.:	731122
Work Package/Activity:	7/JRA1
Task Item:	Task 2
Nature of Deliverable:	R (Report)
Dissemination Level:	PU (Public)
Lead Partner:	GRNET
Document ID:	GN4-2-17-5EA76
Authors:	Adam Pavlidis (GRNET), Alaitz Mendiola (EHU), Aleksandr Kurbatov (GEANT), Andreas Papazois (GRNET), Artur Juszczak (PSNC), Carolina Fernandez (i2CAT), Guy Roberts (GEANT), Jordi Ortiz (UMU), José Soler (DTU), Jovana Vuleta (AMRES), Kostas Stamos (GRNET), Lukasz Makowski (PSNC), Matteo Gerola (CREATE-NET), Pankaj Sharma (RENATER), Paolo Lungaroni (CNIT), Pier Luigi Ventre (CREATE-NET), Sebastiano Buscaglione (GEANT), Sharique Javaid (LRZ), Victor Olifer (JANET), Afrodite Sevasti (GRNET)

© GEANT Limited on behalf of the GN4-2 project.

The research leading to these results has received funding from the European Union's Horizon 2020 research and innovation programme under Grant Agreement No. 731122 (GN4-2).

Abstract

This document reports on SDN technology pilots utilising infrastructure deployed across the production GÉANT backbone in parallel to the infrastructure carrying production traffic. Technology pilots aim to verify the functionality and stability of novel, integrated software and hardware modules in a holistic way in an out-of-the-lab environment, while at the same time assessing the operational readiness of the SDN solutions.

Table of Contents

1	Introduction	3
2	Pilot Assumptions	5
2.1	Infrastructure	5
2.2	Stakeholders	6
3	SDX L2 Pilot	7
3.1	Introduction - The SDX L2 Concept	7
3.2	Description	8
3.3	Functional/Operational Overview	8
3.4	Pilot Overview	10
3.5	Detailed Infrastructure Setup	12
3.6	Detailed Software Architecture	13
3.7	SDX L2 Pilot - Detailed Results	14
3.8	Findings and Future Work	17
4	SDN-Based Circuit on Demand	19
4.1	Description	19
4.2	Functional/Operational Overview	20
4.3	Pilot Overview	20
4.4	Findings and Future Work	24
5	SDX L3 Status	25
5.1	Description	25
5.2	Current Status of SDX L3	26
5.3	Future Work	29
6	Transport SDN Status	30
6.1	Current Status of Transport SDN	31
7	Working with Corsa	33
8	Working with Hewlett-Packard Hardware: Janet Testbed	34
8.1	Findings	34
9	Working with ONOS	35
10	OF Conformance Scripts	37
11	Monitoring	39

12	Conclusion	41
Appendix A	Details of DynPaC Testing in Janet Testbed	42
A.1	Janet SDN Testbed	42
A.2	Janet SDN Lab	43
A.3	ONOS Support for HP3800 Switches	44
	References	47
	Glossary	49

Table of Figures

Figure 2.1: Physical infrastructure	6
Figure 3.1: SDX-L2 application and its abstractions	7
Figure 3.2: SDX-L2 pilot	12
Figure 3.3: SDX-L2 architectural components	14
Figure 4.1: Using the DynPaC framework as a domain manager in the NSI architecture	19
Figure 4.2: SDN-based circuit-on-demand pilot slice overview	21
Figure 5.1: Sample IXP with a single ONOS instance with SDX L3 combined with a BGP Route Server	25
Figure 6.1: SDN architecture for the Infinera OTS	30
Figure 9.1: Software stack under development by the Deployment Brigade	35
Figure 10.1: OF conformance setup	37
Figure 11.1: Monitoring architecture	40
Figure A.1: Janet SDN testbed	42
Figure A.2: Janet SDN lab	43

Table of Tables

Table 3.1: SDX-L2 test results summary	16
Table 5.1: SDX-L3 test results summary	27
Table 6.1: Transport SDN requirements	31

1 Introduction

Task 2 of the Joint Research Activity 1 (JRA1) of GN4-2 will take the SDN Use Cases prototyped during the previous GN4-1 [\[GN4-1 POC\]](#) project, and transition them to pilot status. The time frame of this transformation lies within GN4-2.

This document reports on SDN technology early pilot phases, utilising infrastructure deployed across the production GÉANT backbone in parallel to the infrastructure carrying production traffic. Technology pilots aim to verify the functionality and stability of novel, integrated software and hardware modules in a holistic way in an out-of-the-lab environment, while at the same time assessing the operational readiness of the SDN solutions.

The upcoming deliverable D7.3 Overview of SDN Pilots: Description and Findings – Part B, reports on outward-facing pilots of the developed SDN solutions, as well as pre-production activities, focusing on user-perceived functionality, usability and maturity of the solutions. Intermediate iterations, in progress at the moment of writing this deliverable, focus on eliminating issues and adding functionality following utilised software and vendor hardware updates. Pilot results are constantly updated, based on rapid codebase/hardware feature evolution.

The SDN use cases include:

- **SDX L2:** This use case provides software-controlled configuration of point-to-point Layer 2 services between access (Ethernet) interfaces of interconnected parties, by introducing SDN capabilities to GÉANT Open [\[GOPEN\]](#) for R&E parties or equivalent NREN facilities, thus removing the need for the operators to manually configure the fabric. Exchange points can thus become programmatically configurable by the interconnected entities, under certain isolation principles.
- **SDX L3 / SDN-IP:** This use case deals with Layer 3 capabilities at SDX context, where an SDN controller acts as a route server on behalf of the member autonomous systems (ASes) at the Exchange Point. It also deals with the SDN IP application allowing IP transport over an OpenFlow-capable network for connected external BGP routers and their ASes.
- **SDN-Circuit on demand:** The provisioning of circuit on demand services utilising SDN-based flow handling, combined with NSI protocol support for multidomain discovery and signalling is the aim of this use case. It also adds advanced path computation, enhanced resiliency and agile flow-handling capabilities to existing services of GÉANT/NRENs [\[NSI\]](#).
- **Transport SDN:** The aim is to achieve software-controlled capacity allocation at the optical layer of GÉANT (based on the Infinera platform). This can fit the needs of elephant flows and route bypass as/when needed to selectively reduce costs of routing traffic.

The SDN controller ONOS has served as a basis for the implementation of the use cases [\[ONOS\]](#). Its modular architecture allows for the pluggable extension with respective applications that enable the

ONOS core to communicate with extended functionality at both the northbound (applications) and southbound (drivers) layer. For this purpose, the JRA1 Task2 team continued its close cooperation with the ONOS community and the Open Networking Lab (now known as the Open Networking Foundation [\[ONF\]](#)).

The use cases were initially prototyped in a lab environment, using a variety of OpenFlow-enabled equipment, including Corsa 6400, Pica8 and Dell switches. During the past year, the previous generation Corsa switches were gradually replaced with a newer Corsa DP2000 SDN data plane, which supports switch virtualisation, and is the platform deployed for the SDN technology pilot environment.

The Corsa DP2000 platform allows for full virtualisation of the hardware, where all the hardware resources of the switch can be exposed as independent virtual SDN switches or routers. Such an approach enables division into Virtual Forwarding Contexts (VFCs) that appear as independent switches to the OpenFlow controller with variable pipeline characteristics. Thus, advanced OpenFlow characteristics are maintained in this version of the hardware, with different multi-table pipelines now made available on a VFC basis rather than a switch basis. For the proper operation of the use-case applications, the appropriate ONOS driver had to be used per VFC, in order to manage flows on a per-VFC basis, as best practice for multi-table pipeline switches supported by OpenFlow 1.3 denotes. The Corsa platform supports several VFC types, including the "openflow" VFC type, which presents a single table and does not therefore require a specialised ONOS driver.

The first two planned technology pilots have been SDX L2 and SDN-Circuit on demand, with the latter delayed due to the lack of metering support in the hardware platform used in the pilot environment, and specifically, at the VFC layer visible to the ONOS SDN controller and the applications by extension. This functionality was delivered later than planned and thus the pilot has been delayed by four months. However, the novelty of flow-based handling at VFC level and the additional functionality delivered by the vendor provide the foundation of a solid pilot.

In close collaboration with SA1, the SDX L2 pilot was initiated and carried out, while preparations for the SDN-based circuit on demand pilot have also been conducted with the pilot itself in close succession.

2 Pilot Assumptions

An important, overarching objective of the pilots is to assess the resiliency of the control plane. Because SDN brings critical functionality out of the network equipment and into software components, the SDN controller needs to be robust enough in response to various types of network, equipment and software failures. ONOS promises to deliver pioneering clustering functionality, which the pilots are designed to verify. Another critical and relevant aspect is the assessment of monitoring operations and capabilities that need to be provided in the SDN use-cases, at least on par with the current operational practices.

2.1 Infrastructure

SDN technology pilots have been implemented on an overlay network that GÉANT has deployed using Corsa DP2100 switches. This network is shared with JRA2 and the GÉANT Testbeds Service (GTS), which takes advantage of the virtual switch capability and has access to the administrative underlay of the switches. JRA1 access is limited to the overlay (VFC – virtual instances), which are provisioned manually.

The SDN pilot infrastructure based on Corsa white box switches is interconnected by 10Gbps lambdas and deployed in London, Paris, Amsterdam, Prague and Milan. Each location includes one Corsa DP2100 box, which is shared with the GTS infrastructure. Figure 2.1 below shows the underlying physical infrastructure of GÉANT that has been used for the deployment of SDN pilot slices.

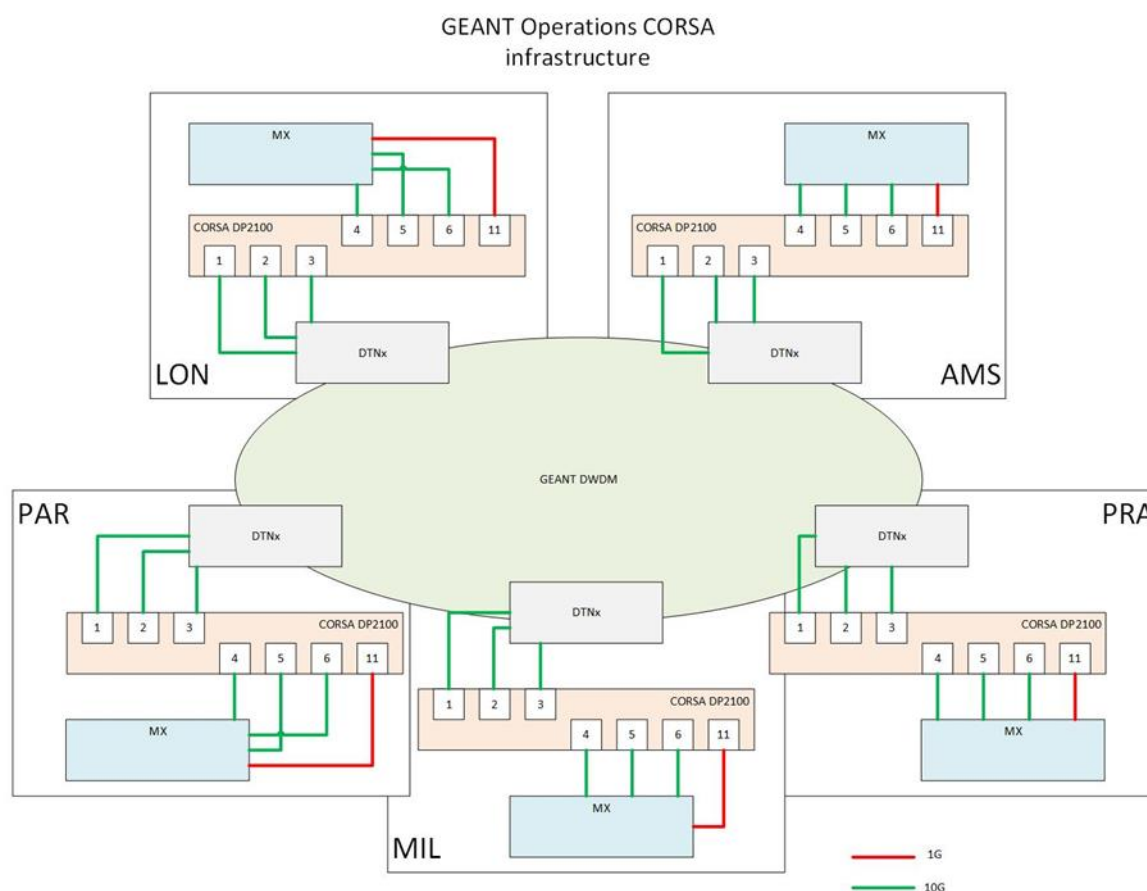


Figure 2.1: Physical infrastructure

Virtual Forwarding Contexts (VFCs) on Corsa boxes are OpenFlow enabled, and a dedicated L3VPN is used for connectivity with the controller. Users gain access to the data plane of the SDN pilot infrastructure via their normal interconnect at their local GÉANT MX router.

2.2 Stakeholders

Pilot outcomes are important for GÉANT and participating NRENs. During the first 12 months of GN4-2, consultation with participating members from NRENs took place in order to identify the overall characteristics that are considered important by NREN operators in this context. Detailed feedback from this process is provided in the following sections.

3 SDX L2 Pilot

3.1 Introduction - the SDX L2 Concept

GÉANT provides GÉANT Open [[GOPEN](#)], a connectivity service for NRENs to connect with external (non-GÉANT) networks through Open eXchange Points (OXPs). Inside an OXP, the users (NRENs or external participants) request the establishment of Layer 2 circuits between end-points, which are manually provisioned through VLAN tunnels, according to the relevant GÉANT Open Exchange Policy rules. Several NRENs and regional networks are currently operating similar services.

In order to set-up a GÉANT Open service between two access points (ports or VLANs) inside an OXP, the user has to contact the operators to manually configure the connection. These operations (creation of virtual interfaces, VLAN ID selection on both endpoints, VLAN ID rewriting) are error prone, and require coordination between the interested parties. Any arising issue also requires further manual intervention from the operators. A typical target for the provisioning time of these services is five days. In addition, as some of those exchanges are in the process of being interconnected, the waiting time to establish a service between two end-points increases to at least about ten days. In general, this results in a lengthy, cumbersome and manual process for setting up new services.

This process could be streamlined by the use of the SDX-L2 service, the SDN-isation of the GÉANT Open service. Using SDX-L2, the provisioning process is reduced from days to minutes. Moreover, by leveraging features provided by SDN controllers, most failure cases are automatically resolved without any manual intervention: a failure of a controller within a cluster is solved using redundancy of SDN controller instances, and a data-plane failure is solved with automatic re-computation of data plane paths around the faulty network element, pending capacity and adequate protection foreseen by the service.

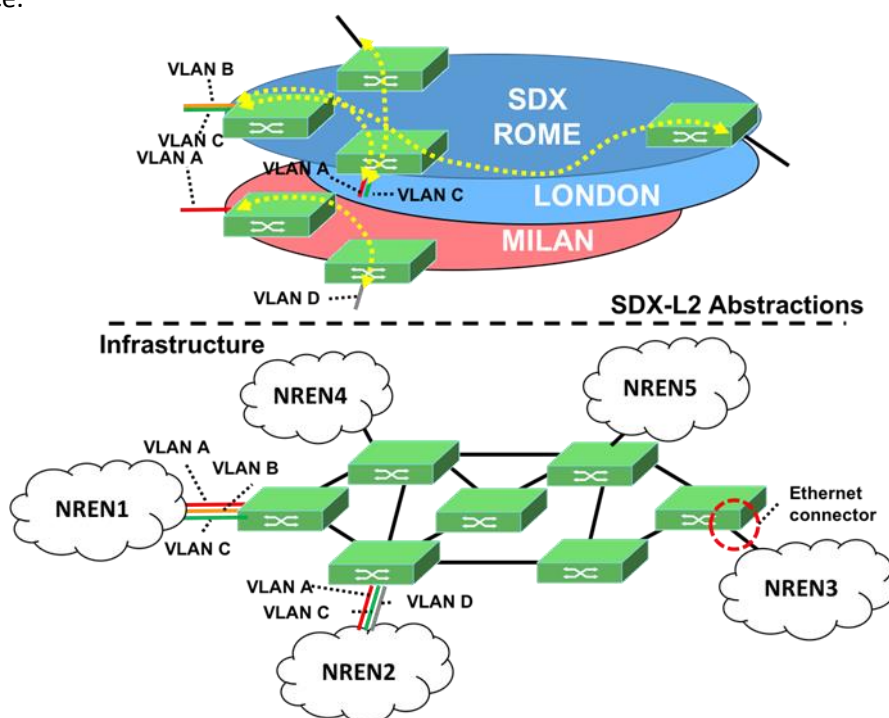


Figure 3.1: SDX-L2 application and its abstractions

SDX-L2 provides users with powerful abstractions, as shown in Figure 3.1. A virtual SDX (e.g. Rome, London or Milan) contains a number of edge ports or VLANs (untagged or tagged ports from physical devices), which are internally modelled as edge connectors. Network operators can establish virtual circuits between these ports, as required, to serve each individual customer. Alternatively, customers that operate only on the edge of the SDN network are controlled by SDX-L2. This procedure eases service management and provisioning by reducing provisioning times, providing easier abstractions and operations to network operators, enforcing isolation and avoiding several types of conflicts during services creation.

3.2 Description

Following development and lab-testing activities on the SDX-L2 use case during the first period of GN4-2, GÉANT Operations has assessed the technology pilot on the GÉANT operational network. A specific version for the ONOS codebase and Corsa software has been selected for use in the testing. The features to be tested have been verified to work correctly in the lab before they are handed over for the pilot deployment. In particular, ONOS master (1.10 at the time of testing) has been used, as it contains a thin driver developed by the JRA1 Task 2 team and required to mitigate a 'clear_actions' instruction, originally sent by ONOS to support the Corsa DP2100 devices (as detailed in Section 3.7.1). Regarding Corsa software, Corsa DP2100 - 2.1.1 build 10 has initially been used, and subsequently upgraded to 2.1.6.

The set of features currently available for the technology pilot in the GÉANT operational environment is a subset of the set of features related to the overall requirements for the use case. However, in order to maximise efficiency, it has been agreed that a first phase of piloting in the operational network will take place on the functional subset of features in parallel to development and lab testing work.

The features of the technology pilot and their limitations are clearly described in the following sections. The detailed set-up of the pilot infrastructure and supported features, as well as the SDX-L2 application architecture can be consulted in Section 3.6

3.3 Functional/Operational Overview

The functional/operational features of the pilot include both the existing capabilities of the operational exchange points and the new functionality introduced by the proposed solutions.

Below is a summary of basic capabilities:

- Definition of L2 connections between two edge ports or VLANs.
- Definition of multi-point L2 connections between edge ports or VLANs.
- VLAN and Stacked-VLANs (802.1ad) encapsulation for the circuits.
- [*] MPLS encapsulation for the L2 connections.
- IPv6 support.
- Control plane resiliency.

- Control plane failure recovery.
- Network status after control plane failure.
- Traffic re-routing after data plane failure.
- Allow collecting and storing statistics related to traffic and errors and polling by using SNMP from other systems.
- Allow sending SNMP traps to central monitoring systems advising on the status of interfaces and L2 connections across virtual switches in a single domain. Ideally, this should also provide status of overall service across multiple domains.
- Provide logging facilities, with the ability to distinguish system-related and network-related events. Logging facilities should be easily accessible for support teams, also logging on separate remote server should be possible.
- Information about services configured across multiple domains should be accessible in details (i.e. connection points, capacity, VLANs, start/end times) in a standard format, such as REST, to allow external recording or further processing. A specific use case is a CMDB and incident reporting.
- The controller should allow for easy access to information about interfaces and services. This should be presented in a format that can be read by NOC staff. It should not take a number of commands to obtain traffic, errors, interface state, speed, configured VLANs and interface descriptions information for one or multiple interfaces on the virtual or physical switches. This information could be collated and presented using troubleshooting facilities via the GUI if not available in this format through CLI.
- Allow diagnosis of root cause, or as a minimum, locate the sub-system(s) where the fault originates – Controller, orchestrator, switch.

[*] ONOS controller and SDX-L2 support this, Corsa devices do not at the time of writing.

Further desired functionalities include:

- Isolation of virtual switches and related controllers. Issues on a virtual switch and its controller should not have any adverse impact on any other virtual switches, their controllers or the physical box itself.
- Allow part of a virtual switch to control a specific range of VLANs on a port to allow a single physical connection to be part of several virtual switches.
- Allow rate limiting VLANs on interfaces for defined bandwidth services.
- Add and remove ports and/or VLAN ranges from a virtual switch without impact on other ports, services or VLANs on the same switch.
- Provide an API for integration of the domain controller or multidomain controller with other systems as well as support for NSIv2.
- Allow, at the controller side, for the increase of the number of switches under a controller without fully re-configuring the whole domain and associated services. A serviceable interruption during the expansion of the fabric is acceptable.
- Allow for a hierarchy of users and admin accounts, including fine-grained authorisation for access to particular ports or VLANs per user or user group, as well as definition of rules for resource management by multiple owners

- Support of LAG with LACP. Functionality of the LAG interfaces must be identical to the physical interfaces in terms of configuration options and the report options (show commands, statistics, etc.)

The latter group of requirements extends beyond the core SDX L2 concept, and is subject to an integrated SDN-ised L2 service offering.

3.4 Pilot Overview

In the context of JRA1, two ONOS applications have been selected as suitable to fill most of the described requirements, SDX-L2 [\[SDX-L2\]](#) and VPLS [\[VPLS\]](#). SDX-L2 was developed within JRA1 with the aim of offering an implementation of the SDX L2 use case concept and offered to the ONOS community, whereas VPLS has been developed by ONOS and JRA1, who are contributing to its further development and testing. The main difference between SDX-L2 and VPLS is that VPLS offers the capability for creation of multi-point connections, whereas SDX-L2 is limited to point-to-point connections only.

At the time of testing with the available releases (the Corsa firmware, the ONOS codebase and the two ONOS applications), some features, such as multi-point connections, are not yet available via the VPLS application. Therefore, SDX-L2 was chosen to be deployed in the technology pilot over the GÉANT network.

The following list contains the functional requirements that have been included in the technology pilot so far:

- Definition of L2 connections between two edge ports or VLANs.
- [*] VLAN and Stacked-VLANs (802.1ad) encapsulation for the circuits.
- Control plane resiliency.
- Control plane failure recovery.
- Network status after control plane failure.
- Traffic re-routing after data plane failure.
- The controller should provide an open API to access management and service information, such as traffic, errors, interface state, speed, configured VLANs and interface descriptions information for one or multiple interfaces on the virtual or physical switches. It is expected that the controller will eventually be complemented by an application tailored for NOC use.
- Allow diagnosing root cause, or as a minimum, locate the sub-system(s) where the fault originates – Controller, orchestrator, switch.

[*] Only VLAN circuits were tested, Stacked-VLANs are currently not available.

Details about the pilot functionality are reported in the following sections, along with a short summary of functionality testing results. During the testing of the SDX-L2 application, the GÉANT operations team encountered a number of limitations. The most serious relate to the operational perspective:

1. ONOS cluster:

- Procedure to access a controller instance varies between the deployer of the cluster and the rest of the instances, due to the ability to use the development scripts provided at /onos/tools/test/bin/onos source folder only from the master instance of the cluster [[ONOS-CLUSTER](#)].
 - The communication between cluster members demonstrated random errors during the pilot duration, even when VMs are in the same subnet, same hypervisor, etc.
 - Lack of simple tool or set of commands to check the status of the cluster.
 - The availability of remote monitoring of the cluster via SNMP or other means is unknown.
2. Random VLAN tag selection on the trunk with no tracking and no checking. This will eventually result in a race condition, assigning same tag to different circuits.
 3. De-synchronisation between controller and switches: flows that were not specified in the request were seen in the switches. Furthermore, these flows were not removed by the removal of the circuit and connection points -- a wipe operation was required.

3.4.1 Limitations Imposed by the Pilot

Some technical limitations exist and are imposed by the nature of the pilot slice. Such limitations are mainly affecting hosts-to-VFC connectivity as hosts are deployed on the GÉANT IT infrastructure.

The main limitations of the slice are:

- Due to the VM infrastructure used to instantiate the end hosts, they cannot use VLANs when connecting to Corsa VFCs.
- Due to the connections to the VFCs, hosts can send less than 100Mb to the VFC.
- Only 50% of the table space can be assigned per pipeline type to VFCs to be used for testing, due to simultaneous usage of the VFC resources in GÉANT.
- Links between the Corsa DP2100 in different sites need to support traffic for multiple testbeds and therefore cannot have the capacity fully dedicated to a specific slice. This means that a limit of 1G is to be considered for all links to VFCs.

The limitations are imposed by the nature of the IT infrastructure where the VMs are deployed, as such infrastructure is built for supporting web services available through the global R&E internet and not a virtualised slice for testing. As a workaround, MX routers' sub-interfaces can be used to test using tagged traffic, however, the number of VLANs that can be configured on MXs for simultaneous use are limited to 20. In addition, only simple, non-volumetric tests can be run between MXs sub-interfaces.

The limitation on VFCs and Corsa DP2100 connectivity is due to the shared pilot facilities usage, i.e. the fact that the same physical switch has to support multiple use cases and GTS needs, therefore, the switch and its links cannot be fully dedicated to testing for a single use case.

3.4.2 Features and Functionalities Available for Next Phase

There are a few functionalities that, at the time of writing, were not yet ready for pilot evaluation, but are in the list of features developed by JRA1 and are expected to conclude in the next phase:

- SDX-L2 GUI.
- Multipoint connections/VPLS application.
- Monitoring systems and operational tools.

3.4.3 Features and Functionalities Not Available

The MPLS encapsulation on trunks is supported by ONOS and SDX-L2, but there are not known future plans from Corsa to add support for MPLS labels.

3.5 Detailed Infrastructure Setup

A network slice has been provisioned in the production GÉANT network to provide an operational environment for the SDX-L2 Pilot. Figure 3.2 illustrates the environment.

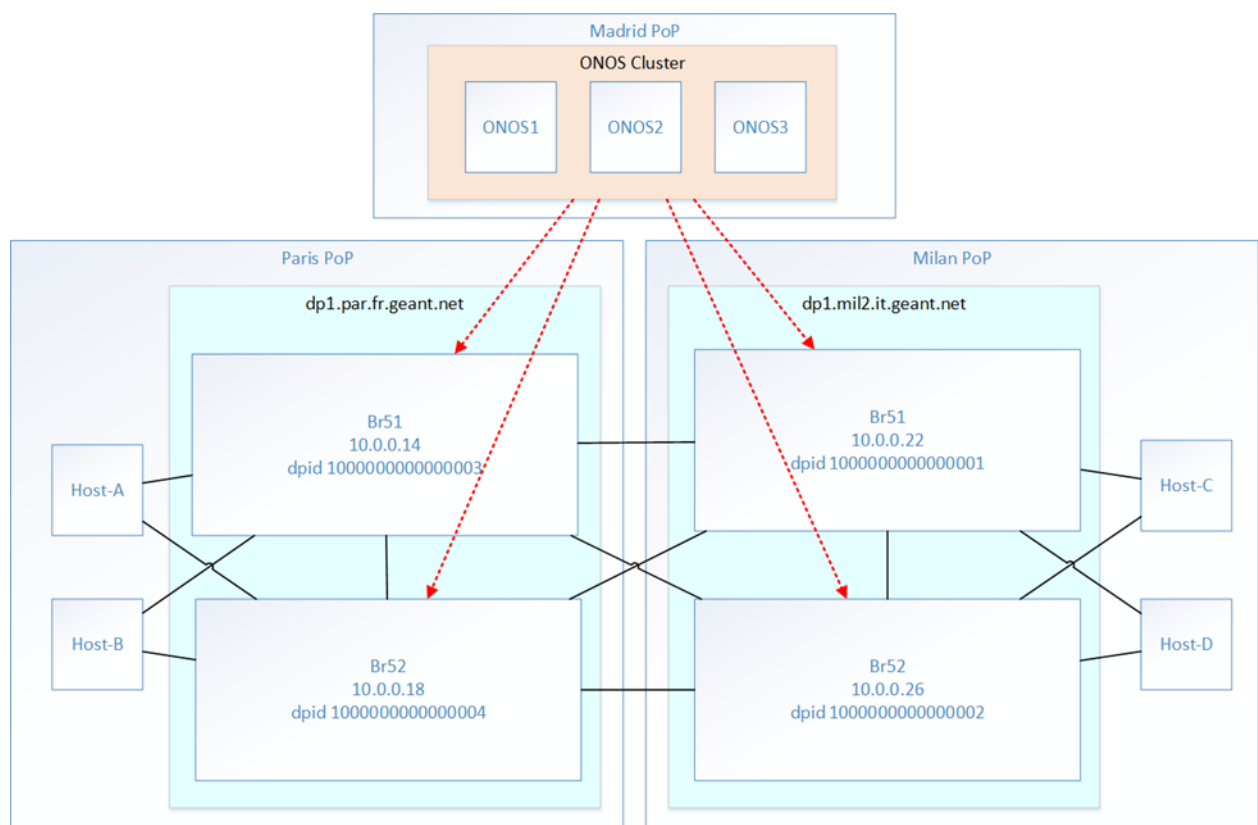


Figure 3.2: SDX-L2 pilot

A set of four VFCs have been configured on two Corsa DP2100, deployed in the GÉANT PoPs of Milan and Paris. VFCs are interconnected using a mix of Infinera and Juniper MXs-based point-to-point links.

Seven VMs have also been provisioned: Three to function as a set of ONOS instances in a cluster (ONOS1, ONOS2, ONOS3) and four to be used as end hosts (HOST-A, HOST-B, HOST-C, HOST-D). Hosts are deployed in the GÉANT IT VM infrastructure local to the Corsa switch, where hosts are connected.

Connectivity between the IT VM infrastructure and Corsa DP2100 is provisioned across local LAN switches and MX routers. An L3VPN has also been configured in the GÉANT IP/MPLS network to provide connectivity between the ONOS cluster and VFCs while maintaining isolation from the rest of GÉANT IP network.

3.6 Detailed Software Architecture

SDX-L2 provides operators with high-level APIs:

- Operators see the abstraction of managing virtual SDXs.
- An SDX contains edge connectors; either physical or virtual. These can be tagged (VLAN-tagged interfaces) or not (device edge-ports).
- Connectors can be interconnected through virtual circuits (VCs).

Different sub-components are in use:

- SDXL2Service: allows creating the SDX, which can later on contain VCs.
- SDXL2VCService: allows creating the VC, coupling two different connectors.
- SDXL2MonitoringService: performs a background check on failed intent installations and on failure regarding edge-ports.

The rest of subcomponents provide access to the operator (through CLI and GUI), register information, or interact with the core to manage network resources (IntentService, PacketService, EdgePortService).

ARP and NDP packets are not forwarded in the data-plane, but instead are relayed through the control plane by using the SDXL2ARPNDPHandler sub-component.

The current architecture of SDX-L2 until this phase is depicted in Figure 3.3.

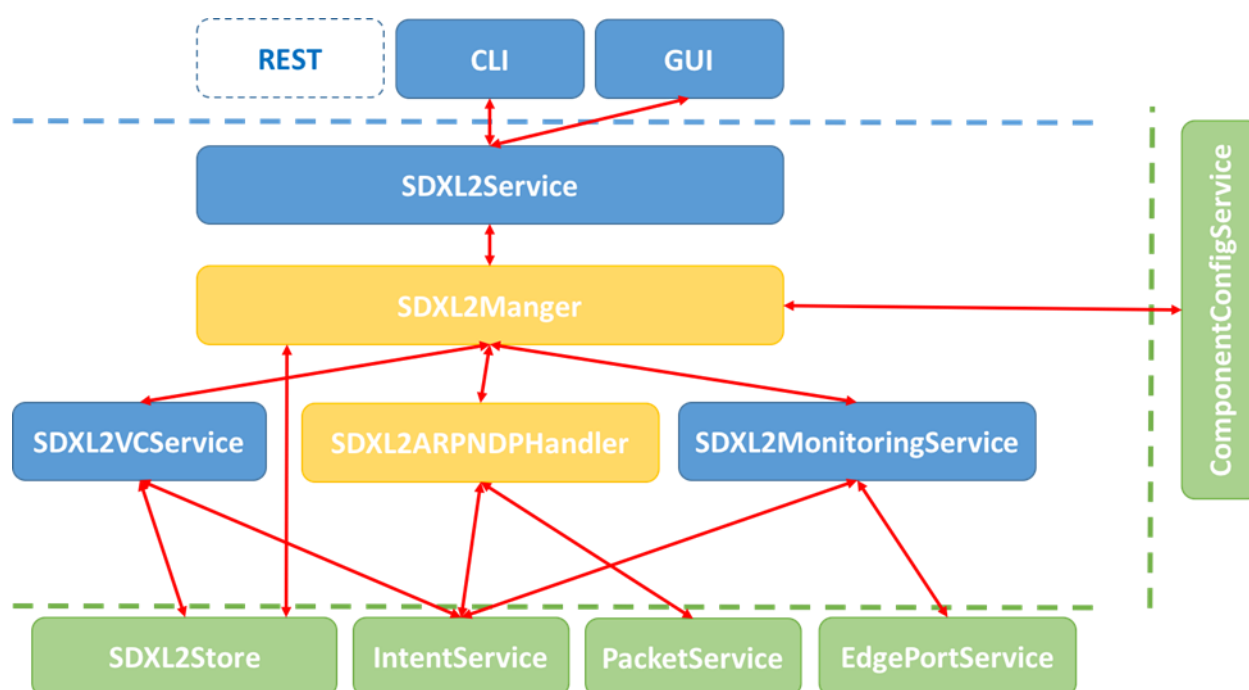


Figure 3.3: SDX-L2 architectural components

Initially, the architecture allowed the creation of the following types of virtual circuits:

- MAC-based: Point-to-point communication would be achieved by registering the MAC addresses of the endpoints.
- VLAN-encapsulated: a VLAN tag is defined per connector. If different VLANs are used per connector, VLAN translation will be performed.
- MPLS-encapsulated: An MPLS label is defined per connector.

During this phase, the MAC-based virtual circuits were dropped in order to avoid MAC learning and registration. The pilot includes only on VLAN-based, VLAN-encapsulated and MPLS-encapsulated VCs.

3.7 SDX L2 Pilot - Detailed Results

Detailed results obtained from SDX L2 pilot are detailed in the SDX-L2 pilot results appendix located on the GÉANT project Wiki. Table 3.1 below summarises the result per feature tested, and the expected reason for the failing conditions, where applicable.

Functionality group	ID	Requirement	Result	Remarks
SDX-L2: definition of SDX data	1	Creation of an SDX instance	Success	
	2	Deletion of an SDX instance	Success	
	3	Addition of connection points	Success	
SDX-L2:	4	Creation of a circuit – no access	Success	

Functionality group	ID	Requirement	Result	Remarks
creation of circuits		VLAN, no encap, no transit		
	6	Creation of a circuit – access VLAN hostA to no VLAN hostC, no encap, no transit	Success	
	7	Creation of a circuit – access VLAN both end, NONE encap, no transit	Success	
	9	Creation of a circuit – diverse access VLANs, no encap, no transit	Success	
	11	Creation of a circuit – diverse access VLAN, NONE encap, transit	Success	
SDX-L2: creation of circuits with encapsulation	5	Creation of a circuit – no access VLAN, VLAN encap, no transit	Fail	Expected: first available VLAN on trunk Obtained: random VLAN tag on trunk
	8	Creation of a circuit – access VLAN, VLAN encap, no transit	Fail	Expected: preserve VLAN tag on egress Obtained: VLAN tag is popped on the output interface (flows at Corsa device). The issue was related to misalignment of SDX L2 intents with the updated Intent Framework implementation in ONOS, and has been fixed in subsequent SDX L2 releases (to be demonstrated at subsequent pilot phases).
	10	Creation of a circuit – diverse access VLANs, VLAN encap, no transit	Fail	See 8
	12	Creation of a circuit – access VLAN, VLAN encap, transit	Fail	See 8
	13	Creation of a circuit – diverse access VLANs, VLAN encap, transit	Fail	See 8
SDX-L2: removal of data	14	Deletion of circuit	Success	
	15	Removal of SDX and all circuit are removed	Success	
	16	Removal of host and all associated circuit are removed	Success	
	17	Clean state	Success	

Functionality group	ID	Requirement	Result	Remarks
Controller & cluster behaviour	18	ONOS host reboot	Partial	Circuit is still working but ONOS didn't start automatically in the rebooted machine
	19	Check ONOS cluster status	Partial	Cluster status is obtained with a number of commands. Highly desirable to simplify the procedure.
	20	VC removal from different ONOS VMs	Fail	Expected: Remove VC from ONOS2, check the result in Corsa bridges Obtained: Improper behaviour of SDX-L2 in cluster mode
	21	Connect to ONOS application from ONOS2 and ONOS3 VMs	Partial	ONOS is reachable with a different set of commands

Table 3.1: SDX-L2 test results summary

Three main error conditions are observed:

- The VLAN assigned to use encapsulation on a VC is assigned randomly by ONOS. In order to avoid possible collisions or race conditions, it is expected to assign the first available VLAN to the VC (ID 5).
- The VLAN assigned to use encapsulation on a VC is popped on the switches. This is not expected, and occurs during the translation of the Intents provided by SDX-L2 into the flows installed at the Corsa devices (IDs 8, 10, 12, 13).
- The data managed by SDX-L2 is not being properly synchronised when running in a clustered environment (ID 20).

The steps to address such conditions deal with:

1. Configuring ONOS to provide the first available VLAN for VCs.
2. Avoid the translation step that introduces the popping of the outer VLAN in the switches.
3. Investigate the failure condition occurred in clustered mode for SDX-L2.

3.7.1 Blocker Conditions and Proposed Solutions

In addition to the results of the functional testing described above, there have also been some conditions on the ONOS applications and the Corsa devices that have introduced issues in the technology pilot. Descriptions of the issues faced as well as the proposed solutions for this phase of the pilot are detailed below.

3.7.1.1 *Issue with ONOS VPLS App*

VPLS initially translated multi-point connections into multiple output ports for a specific flow. However, this behaviour was not supported in the Corsa switches, which led to an ongoing development effort in order for VPLS to make use of the OpenFlow-defined group type “ALL” in the group table supported by the Corsa switches. In parallel, Corsa will support multiple output actions in its future firmware releases. Despite these developments, during the execution of the pilot, only SDX-L2 was a viable choice, therefore, it was deployed and evaluated.

3.7.1.2 *Issue with ONOS interaction with Corsa Devices*

A further issue before being able to proceed with the pilot turned out to be a behaviour introduced in ONOS versions, starting with v1.9, with the intention to properly support multi-table pipelines. In order to avoid deferred actions when packets are matched in more than one table, a behaviour clearing deferred actions was added in the relevant ONOS module (PacketService). This, however, led to the switch refusing the flow installation. In order to overcome this problem, a thin Corsa driver was implemented for ONOS, which inherits the behaviour of the standard single table pipeline driver, and filters the clear deferred actions. This driver was used in order to successfully execute all ONOS apps (SDX L2, SDN-circuit on demand, SDX L3) over the Corsa single-table VFC.

3.8 Findings and Future Work

The main users of the technology pilot, the GÉANT Operations engineers, have concluded that the SDX-L2 application and the ONOS platform, including its implementation of the HA cluster, are not yet ready for the use in production network. Addressing the limitations reported in the development/lab environment and further iterations of technology pilots are required. Therefore, the JRA1 T2 team is currently working on SDX-L2 development plans to address these limitations (mentioned in Section 3.4) before the second iteration of the technology pilot. New requirements may be added, as needed.

The SDX-L2 GUI has not been available for the current pilot. In the next iteration, JRA1 will deliver the GUI component and it will be available for evaluation by GÉANT Operations engineers. In parallel, new features, such as the L2 circuit GUI, will also be developed. Moreover, it is necessary to provide better error handling, and the creation of resources should make use of panels instead of tables in order to be more manageable

In this phase, JRA1 T2 team started the development of a new Corsa driver for ONOS in order to make Corsa switches compatible with last versions of ONOS. In the next phase, the T2 team will continue this work in order to implement some of the missing functionality, such as the support for the OpenFlow “ALL” group. The latter will unblock the piloting of the VPLS application, which provides multipoint circuits. Drivers also need to be aligned with the new release of ONOS and the new Intent Framework introduced by ONOS 1.10.

There will be development work devoted to update SDX-L2 with the changes within the SDN controller. This includes the alignment of the SDX-L2 definition of intents or flows to cope with the changes introduced within the new Intent Framework 1.7.0. Other efforts will be spent to align SDX-

L2 with ONOS 1.10, which introduces a new intent framework with the full support of OpenFlow 1.3.X.

Further work also includes:

- GÉANT OF conformance scripts to be run at each release of new Corsa firmware against the hardware switches.
- Automatic testing solutions based on the TestON framework [[TESTON](#)].

Finally, during the latest technology pilot evaluation, new requirements were specified by the GÉANT engineers. An evaluation of these new required capabilities is necessary in order to evaluate feasibility and appropriateness. Capabilities deemed important will be implemented, tested and moved from the lab testing phase to the next iteration of the technology pilot.

4 SDN-Based Circuit on Demand

4.1 Description

Circuit-oriented services are already a reality in several NRENs, including GÉANT, where AutoBAHN allows an end-user to request multi-domain services with a guaranteed bandwidth during a set period of time [AutoBAHN]. AutoBAHN provides the Bandwidth on Demand (BoD) service across multiple domains through the NSI-CS v2 protocol and custom made per-domain agents. It currently does not support SDN domains. The SDN-based circuit on demand use case of JRA1 relies on the DynPaC application to deliver SDN-based circuit services. DynPaC is a solution for advance reservation of circuits with guaranteed capacity that allows provision of resilient VLAN-based L2 services taking into consideration bandwidth constraints,¹ while at the same time leveraging an OpenFlow-based interaction with the data plane [DynPaC]. Therefore, an on demand SDN-based circuit alleviates the need for custom domain proxies and is deployable in any network that supports standards-based protocols, such as OpenFlow.

With the introduction of DynPaC, support of the NSI-CS protocol is maintained. Support for OpenFlow domains is added by means of utilisation of the DynPaC framework as the Domain Manager for domains, as depicted in Figure 4.1.

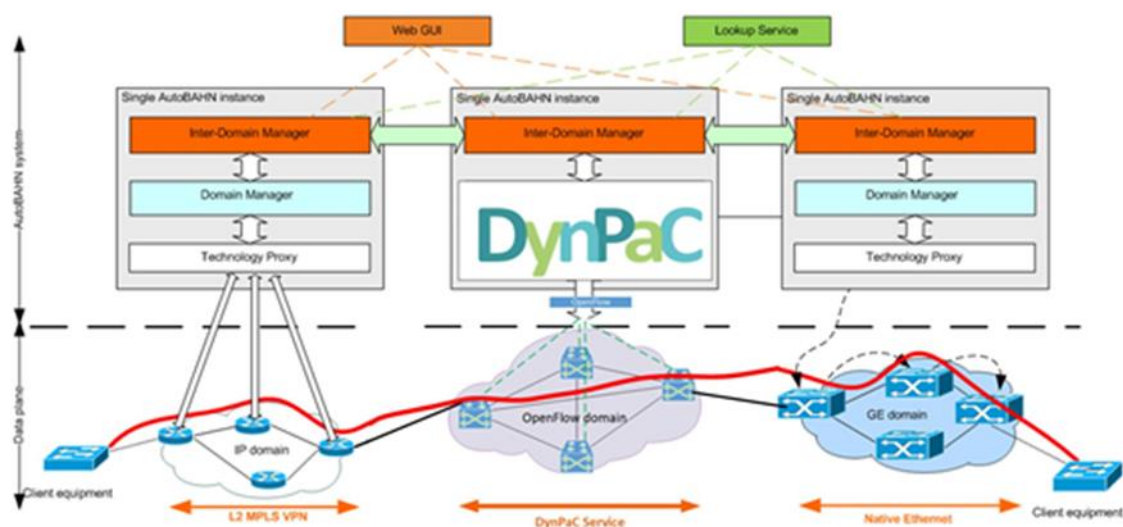


Figure 4.1: Using the DynPaC framework as a domain manager in the NSI architecture

The following list summarises the features provided by the DynPaC framework, some of which are additional to what AutoBAHN/other existing circuit on demand solutions offer:

- Compute the best possible path between two end-points, taking into account:

¹ Similarly, the US NREN, ESnet, offers OSCARS to its users, which adopts the PCE-based architecture in order to compute the paths.

- Available bandwidth.
- Minimum hop count.
- Path stability.
- Optimise network resource utilisation.
 - Flow relocation mechanism.
 - Flow disaggregation mechanism.
- Resiliency.
- Rate limiting.
- Automated network programming.

In the context of the GN4-2 JRA1, DynPaC development focussed on implementing the missing required functionalities and meeting the required readiness level. For this reason, it was deemed necessary to redesign and refactor part of the application in order to:

- Expose the Service Manager component of DynPaC as an ONOS service.
- Improve the path computation element (PCE) and ultimately, make it stateless.
- Move resiliency handling to Service Manager.
- Improve topological changes event handling.

At the moment of writing, the technology pilot evaluation is still under preparation due to the late arrival of required features from the vendor. However, a progress update and planned next steps have been included in this report.

4.2 Functional/Operational Overview

The technology pilot of the SDN-based circuit on demand use-case aims to demonstrate the ability to rate-limit flows at the level requested and agreed, to verify the capability of the software to discover network topology automatically and react in a timely fashion to topology changes, make efficient decisions on admission requests, optimise the reallocation of flows, and act effectively on network disruptions for protected services.

A detailed list of desired requirements as defined during the use case specification and planned to be available during the technology pilot, is provided on the JRA1 wiki to internal participants.

4.3 Pilot Overview

Figure 4.2 shows the location of the Corsa switches in GÉANT POPs to be used for the SDN-based circuit-on-demand pilot.

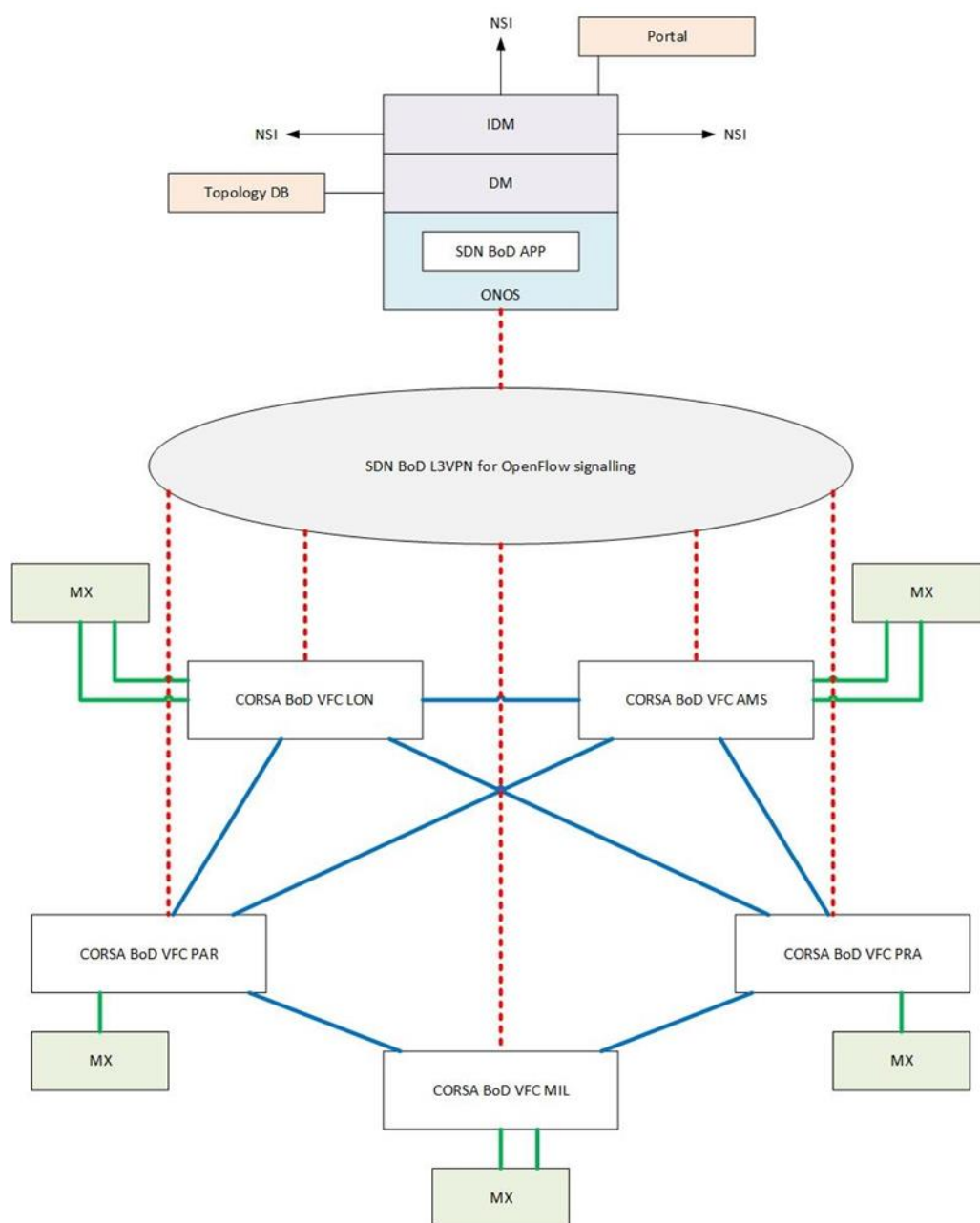


Figure 4.2: SDN-based circuit-on-demand pilot slice overview

The technology pilot will be focused on two fundamental aspects.

- The features provided by the DynPaC application to provide the circuits on demand.
- The behaviour of both the application and the ONOS controller to handle failures.

The DynPaC application works on a continuous refactoring process to improve the performance of the algorithms and add new functionalities, so its pilot evaluation has been divided into four different parts, each of which is associated to a different release of the DynPaC framework. The releases and the set of features included in each of them are listed below:

- DynPaC v1.0 (basic features):

- Bugs detected in previous demonstrators solved.
- Basic VLAN translation.
- No improved algorithms for network snapshots management and the path computational element (PCE).
- Synchronous scheduling.
- Meter handling.
- DynPaC v2.0 (improved performance):
 - New network snapshots management algorithm.
 - New path computation algorithm.
 - Asynchronous scheduling.
- DynPaC v2.1 (clustering support):
 - Refactorisation to expose the service manager as a service.
 - Clustering support.
 - Automate the driver discovery.
- DynPaC v2.2:
 - Resolves problems in the Janet testbed.
 - Propose further features for inclusion in this release.

Regarding the suitability of the DynPaC framework to provide the guaranteed bandwidth, circuit-on-demand service, and taking into account topology discovery features and resilience, the following list summarises the evaluation elements of the technology pilot. For this evaluation, the first stable release of the DynPaC application running as an application of the ONOS controller will be used:

- Check that an unprotected circuit is established correctly, using the shortest path available to connect the specified source and destination nodes.
This will also be checked in case of a link failure (the service is provided through the affected link), the backup path is installed using a disjoint path and that the primary path is deleted from the flow entries of the OpenFlow devices.
- Check that an unprotected circuit is established correctly, using the second-shortest path available to connect the specified source and destination nodes as a result of having any of the links of the first shortest path fully reserved.
- Check that a protected circuit is correctly established, using the shortest path available to connect the specified source and destination nodes. Notwithstanding, the backup path will not be the best disjoint path, but the second-best disjoint path, as a result of having any of the links in the best disjoint path fully reserved. In addition, the establishment of the backup path in case of a link failure and the proper programming of the OpenFlow devices will be tested.
- Check that when the network is fully booked, a new service reservation is rejected.
- Check if the peak bandwidth specified for an unprotected service is enforced correctly at the Corsa devices. In order to test that the bandwidth is correctly enforced, traffic will be injected and the meter tables of the Corsa devices and the received traffic will be checked.
- Check if the peak bandwidth specified for a protected service is correctly enforced at the Corsa devices. Once it is, check that the traffic is correctly shaped. The interface utilised by the primary

path at the Corsa device will be disabled to see if the devices are correctly updated to install the backup path, while the rate limiting is still enforced.

- Check if a protected service is still being provisioned after the failure of a node used by that service.
- Check if the network can be over-subscribed by the sequential request of multiple service reservations, and check if under such circumstances, the services experience any packet loss as a result of having the network operating to its fullest capacity.
- Check if the VLAN translation is correctly enforced for a service where different VLANs are requested for the ingress and the egress.
- Check if the circuit for a given service reservation is correctly established, and that the traffic is exchanged without any problem when Q-in-Q is used [\[Q-in-Q\]](#).

In addition, the evaluation will analyse the impact of failures on the control channel or at the controller side.

- Test the behaviour of the switches when the connection between all the switches in the network and the ONOS controller, and therefore with the DynPaC application, fails. In this regard, the behaviour at both the controller side and switch side will be tested, to see the logs that are generated and if the flow entries associated with the service reservation are still active or not.
- Test the behaviour of the switches when the connection between all the switches in the network and the ONOS controller, and therefore with the DynPaC application, is restored after a disconnection. In this regard, the logs generated at the controller side will be tested, as will the flow entries of the devices.
- Test the behaviour of a given switch when the connection with the ONOS controller, and therefore with the DynPaC application, fails.
- Test the behaviour of a given switch when the connection with the ONOS controller, and therefore with the DynPaC application, is restored.
- Furthermore, given the fact that the circuits will be provided in a multi-domain fashion, the following evaluation will be also conducted:
 - Check if DynPaC is able to retrieve the topology of the network correctly, perform the necessary abstractions to ease the path computation and ensure the STPs that will be advertised to the NSA are correctly identified.
 - Check if DynPaC correctly advertises the topology information related to the domain through the REST interface to the NSA.
 - Check if a multi-domain circuit is established correctly when requested from the GÉANT's Bandwidth-on-Demand portal.
 - Check if a multi-domain circuit is removed correctly when it is requested from the GÉANT Bandwidth-on-Demand portal.

In addition, taking into account the relevance of monitoring the status of the circuits and the network elements, a set of tests related to operations and management will be also conducted:

- Check that it is possible to retrieve the number of packets transmitted on a given switch on a per-port basis.
- Check that it is possible to retrieve the number of packets transmitted on a given switch on a per-flow basis.

- Check if it is possible to retrieve the list of services currently being provisioned in the network.

It is worth remarking that once the subsequent releases of the DynPaC applications are ready, the next iteration of the pilot will be conducted, including the performance of the application to handle service requests (release v2.0) and the clustering capabilities of the solution (release v2.1). Furthermore, apart from piloting with the Corsa devices, other network devices, such as the HP switches available in the Janet testbed (release v2.2) will also be involved.

At present, the technology pilot evaluation is under preparation and is to be executed as soon as the pilot infrastructure has been set up, given that the Corsa devices have been recently upgraded in order for the VFCs to support the required metering features.

4.4 Findings and Future Work

The DynPaC framework has undergone a deep refactoring process to improve the algorithms used for path computation, advance reservations, topology handling and resilience. The necessity for such refactoring was identified during the previous phase of the project in order to increase the scalability of the solution. As such, several improvements have been applied to the application, while others are still pending.

Overall, the resilience and topology handling of the framework have been improved. In the previous version of DynPaC, the topology was retrieved at the activation of the application and remained unchanged. Only the failures of nodes already retrieved in that initial topology were handled, which imposed some limitations to the solution. The new version of the application is able to detect the addition of new nodes to the network, as well as the removal of nodes, and react to the failures by installing the backup paths and modifying future reservations, taking network changes into account.

The advance reservation system has been improved by the introduction of a new database of network snapshots (recording the use of network resources at different points in time) that arranges the snapshot information in a hierarchical manner. This way, the solution is able to perform the admission control process in a much more efficient way, avoiding the linear behaviour of the previous version.

There are still many pending features to be included in the DynPaC framework, mostly related to the resilience capabilities of the application. Future work can be summarised, as follows:

- Execution and thorough evaluation of the technology pilot.
- Add clustering support to make the circuit on demand provisioning resilient to ONOS instance failures.
- Add a mechanism to retrieve the list of service reservations when all the ONOS instances fail. This is a very important feature, since the total disconnection of the switches and the ONOS controller results in the removal of all the flow entries installed in the network devices once the connection with the ONOS controller is back online. As such, and given the fact that all states of the service reservations are kept within the DynPaC application and not on an external element, all the information related to the reservations will be lost, and the services that were already running in the network would not be appropriately restored. As such, an external database to store the reservations is necessary, as well as the means to restore the switches to their previous state.

5 SDX L3 Status

5.1 Description

Functionality for the SDX L3 use case is based on an ONOS northbound application developed by GN4-1 and GN4-2 projects, namely the SDX L3 application. This application has been designed as a variant of ONOS SDN-IP [SDN-IP], a native ONOS application distributed with ONOS. SDX L3 supports all the features that SDN-IP provides, which can be summarised as follows:

- Interconnection of the SDN network with external ASes using the BGP protocol.
- Addition of the ability to the SDN network to act as a routable network, that can also transit traffic between external ASes.
- Moving the routing logic from the SDN network routers to the SDN fabric switches.

Consequently, the SDN-IP application allows the whole SDN network to act as a router by translating the routing information into appropriate rules for the programmable switches. The SDN network routers are still necessary, but their responsibilities are restricted to the BGP message exchange, thus router offloading is achieved and the routing performance of the SDN network is increased with significant cost reduction.

SDX L3 is targeted at control of an SDX fabric, therefore SDX L3 can support the case where all routers of the peering networks have interfaces in the IXP subnet. Thus, peering networks are able to directly exchange IP traffic without needing an additional routing hop inside the SDN network. SDX L3 can be combined with a BGP route server, which is a common type of router in current IXPs. Therefore, SDX L3 remains compatible with legacy deployments, and also takes advantage of router server features for simple IXP configuration and router efficiency. Figure 5.1 provides an overview of an IXP using ONOS with SDX L3 application. For simplicity, a single instance deployment instead of an entire cluster is depicted.

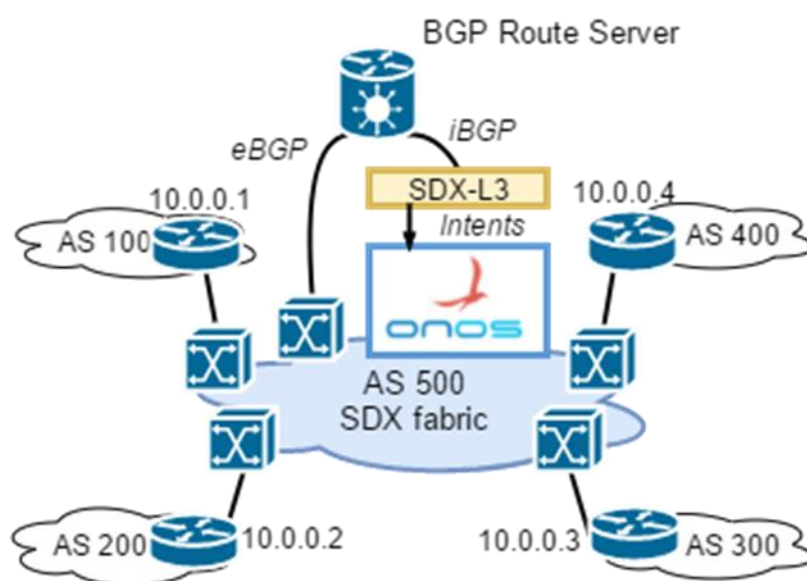


Figure 5.1: Sample IXP with a single ONOS instance with SDX L3 combined with a BGP Route Server

So far, the GN4-2 project work on SDX L3 has focused on the following:

- Adaptation of the SDX L3 application to the ONOS evolution. Since ONOS is a very active project, there are frequent updates at the APIs as well as introduction of new features. SDX L3 software has been regularly adapted to stay compatible with ONOS recent releases and take advantage of the new or improved ONOS features. In order to make the software ready for deployment, the use case also participated in the ONOS community “Deployment Brigade” [\[ONOS BRIGADE\]](#) activities.
- Integration of the ONOS SDX L3 specific ‘corsa-l3-overlay’ driver developed by Corsa for Corsa DP2000 series switches.
- Extensive evaluation of SDX L3, the Corsa ‘corsa-l3-overlay’ and ONOS in the laboratory testbed setup on the GÉANT premises in Cambridge UK. This is the most important step, and its results form a tollgate, since they validate the software and evaluate its state prior to pilot phase. Such evaluation is important, especially in terms of system scalability, which is a major concern for L3 production networks. SDX L3 testing examines whether the needs of GÉANT and NRENs can be adequately met.

5.2 Current Status of SDX L3

The current status is summarised in the SDXL3 traceability matrix [\[SDX-L3\]](#) in the form of lab testing, as this use case has not matured enough to be considered suitable for a technology pilot.

A simplified version, with the status at the time of writing, is provided in Table 5.1.

Functionality group	Requirement / Test title	Result	Remarks
L3 Functionality	IPv4 Support - BGP Transport Between BGP Peers	Success	
	IPv6 Support - BGP Transport Between BGP Peers	Fail	
	VLAN Support - BGP Transport and IP Routing for Different VLAN Setups	Fail	
Redundancy, High Availability and Failures	Network Status After a Complete CP Failure (Single Controller Scenario Only)	Fail	
	SDN Control Plane Failure Recovery (Single Controller Scenario Only)	Pass	Partial results
	Resiliency Against Link Failures	n/a	
BGP Functionality	Add, Remove or Shutdown BGP Peerings Without Impact on Other Peerings	Success	
	Possibility of IBGP and EBGP Peering	Pass	Partial results
	MD5 Support	Success	

Functionality group	Requirement / Test title	Result	Remarks
Switch-Related	LACP Support for Bundling the Links Between Switches	n/a	Test was not defined
Monitoring and Operations	Correctness and Intuition of ONOS SDX-L3 Commands	Success	
	Export of the Flow Statistics From the Controller	Fail	
	Coexistence with Other Applications	n/a	Prerequisite tests were failed
Scaling	IPv4/IPv6 Prefixes Supported in the order of hundreds of thousands/1 million	Fail	

Table 5.1: SDX-L3 test results summary

Not all of the tests have been successful and it has not been possible to perform all tests, due to limitations in the given test setup. The main issues related to the failed requirements are IPv6 testing (Test IPv6 support - BGP Transport Between BGP peers) and mixed tagged/non-tagged traffic (VLAN Support - BGP Transport and IP Routing for Different VLAN Setups test) for the general Layer 3 functionality testing.

By the time most of the tests were completed, a new driver for ONOS was provided by Corsa and selected tests had to be performed with this new driver. The driver 'corsa-l3-overlay' has been designed to efficiently support SDX L3 functionality on Corsa DP2000 series switches. The following tests were performed with the new Corsa driver:

- IPv4 Support – BGP Transport Between BGP Peers.
- Export of the Flow Statistics From the Controller.
- Number of IPv4/IPv6 Prefixes Supported.

Regarding redundancy and high-availability features:

- The Network Status After a Complete CP Failure (Single Controller Scenario Only) test failed, but only in the case where the route server feature was activated. The test was successful without that feature.
- Due to the fail result of test IPv6 Support - BGP Transport Between BGP Peers, the IPv6 protocol was not tested in SDN Control Plane Failure Recovery (Single Controller Scenario Only) or SDN Control Plane Failure Recovery (Single Controller Scenario Only).
- The Resiliency Against Link Failures test was not performed, since the lab topology does not allow for it.
- Regarding the specific BGP functionality testing, all tests were successful, however the following should be noted:

- The Possibility of iBGP and eBGP Peering test was not fully performed, due to missing topological elements.
- For the same reasons, the LACP Support for Bundling the Links Between Switches test was not performed.

Regarding monitoring and operations testing:

- The Coexistence with Other Applications test has not been performed, since the previous generic tests were not a complete success.
- The scalability test, Number of IPv4/IPv6 Prefixes Supported failed, indicating poor performance beyond thousands of BGP routes (max 6000 intents were installed). The results were even worse after the new driver (corsa-overlay) was used – as there were only just over one thousand intents installed on the Corsa switches.

During the testing process, instabilities with the setup and the base ONOS controller have seriously affected the execution of the tests. The main problem is related to the ONOS in the lab setup and appears to have a number of causes:

- When ONOS process is restarted with “service onos restart” command, the corsa driver often does not load properly, so it has to be reloaded after every restart.
- It appears crucial to isolate ONOS from SDN switches and to deactivate the SDX-L3 application prior to following the process.

It is not clear whether some of the results (e.g. in the Network Status After a Complete CP Failure test) have been affected by this. A more streamlined process to verify and secure the validity of the setup before any test is performed will be pursued in future phases.

It should be noted that the Corsa driver for the multi-table pipeline is considered to be far from finished, as scalability issues within ONOS have been identified. Thus testing with the driver was restricted to understanding the limitations ONOS presented and considering options for improvement. The main outstanding driver issues include the following:

- Each multipoint to single point (MP2SP) intent is completely independent, which causes each advertised prefix to allocate a separate group. This causes a scalability issue due to the inefficient management of available groups (the pipeline supports 1022 groups, which is, in fact, quite a large number of next hops for any router). According to ONOS developers, this is the current state-of-the-art in the intents framework, and significant work would be needed to overcome this. One option for the driver would be to try to flatten common next hops, however, this is considered a heavy burden and leaks application knowledge into the driver. In order to get to a significant scaling capability this approach may be required, although it has some architectural drawbacks without clear resolution. Initial work on the Corsa driver used a single, advertised prefix per router.
- As a side-effect of the above, any common flows that are generated as a result of installing flow objectives need to be reference counted by the driver in order to avoid the situation where a withdrawal of an intent affects a flow used by other users. In a single-table design, this is avoidable due to sufficient selector differences that make each flow unique. However, in a multi-table design, common flows may fall out (e.g. coming in on port x in table n should transition to table y). The driver is currently lacking this reference counting logic.

- The objectives coming out of a MP2SP intent cause a great number of ‘ripple effects’. Whereas ideally, the next hop path from ingress switches to egress would be tacked up once, and adding/removing prefixes should merely change the FIBs on the ingress routers. At present, there is much churn triggered by flow changes.
- Withdrawal of intents is problematic due to unresolved sequencing/race issues where next/forward objectives are occasionally not sent in the right order.
- Flow objective store clean-up is not fully implemented.
- The driver is taking advantage of device-specific group IDs. With Corsa using group range 0-1021, it is important to ensure that groups are mapped into this range. As the group IDs in ONOS are by default global, it is possible to cross outside the valid group range.

The above issues are known to Corsa, which has undertaken the development of the ONOS driver. Further work on these issues is pending overall use-case evaluation, and potential architectural redesign.

5.3 Future Work

The laboratory evaluation of the SDX L3 use case showed that, apart from the known weaknesses and deficiencies of the ONOS platform, there are major issues regarding system scalability compared to a realistic (set) number of routing prefixes. Known ONOS platform problems include:

- Problematic system recovery after failures, to an extent that ‘system-clean installation’ is required to achieve proper operation.
- Immature driver subsystem makes driver configuration laborious and error-prone.

However, the most important SDX L3 deficiency is its poor scalability, which cannot be improved by the deployment of the specialised Corsa SDX L3 driver due to the inefficient management by the controller. This makes the system scale far less than the desired levels (magnitude of thousands against hundreds of thousands or millions of IPv4/IPv6 prefixes supported). A flow rule compression is imperative for the controller to adequately support the SDX L3 use case. Since ONOS is a general purpose SDN controller, it is difficult to achieve the desired flow rule compression, therefore a custom ONOS intent framework that is tailored for SDX L3 could be a possible solution to this direction.

One possible future direction of the use case is a more sophisticated route selection mechanism that does not rely only on BGP primitives. When faced with the problem of the congested edge links, it is important for the backbone network to make a decision that takes into account the capacity of the links and the packet loss. To this direction, there are several different sources of information that can be used by the control plane exclusively or collectively, such as full visibility of the peer prefixes via BMP protocol, flow monitoring on interfaces via NetFlow protocol, as well as link utilisation via SNMP.

In addition, an investigation of how P4’s support for hardware acceleration could also improve scalability for SDX L3 cases is also possible [\[P4\]](#).

Clearly, the existing issues and limitations prevent the SDX L3 solution from being deployed in a pilot environment for now.

6 Transport SDN Status

The Software Defined Network (SDN) paradigm could be used to dynamically provision and manage the capacity of optical connections in transport networks. Considering that the current GÉANT optical network is based on Infinera's equipment, GÉANT decided to test Infinera's implementation of the Open Transport Switch (OTS) as part of its piloting activities.

The Infinera OTS is a lightweight virtual transport switch that provides an SDN interface to the DTN-X equipment. The OTS adds transport extensions to the SDN interface that can be used to control OTN transport circuits. However, the interface is not OpenFlow-compliant. Communication between OTSv and ONOS is through a JSON-based proprietary REST API by Infinera. OTS could also be used to offer deterministic Layer 2/circuit services over the optical layer.

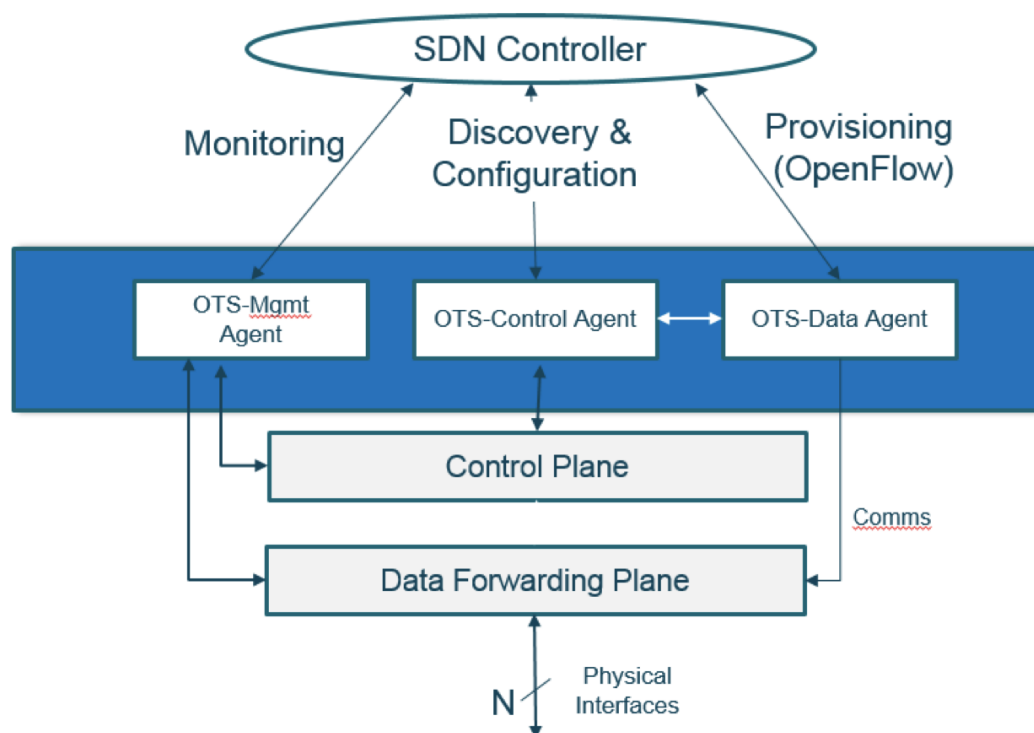


Figure 6.1: SDN architecture for the Infinera OTS

Table 6.1 summarises the major requirements defined for this pilot.

Requirement	Description
EVPL on demand	The Infinera equipment used in this trial must support Ethernet Virtual Private Line (EVPL) services (referred as L1-EVPL).
Infinera OTS enabled	The Infinera equipment used in this trial must be at software version 11.03.
DTN-X in Cambridge lab with 2 PXM units	The Infinera equipment used in this trial will be installed in the Cambridge lab. This requires 2 PXM units and two OTM2 units.

Requirement	Description
OTS compatible	The controller must be compatible with the Infinera OTS.
REST support	The controller used must support the REST interface used by the OTS.

Table 6.1: Transport SDN requirements

For consistency with the other use cases, and to allow future integrations of the Transport SDN use case with upper layer services (e.g. to exploit multi domain scenarios, such as L1, L2 and L3, under the same control), we have chosen to leverage on the Open Networking Operating System (ONOS).

6.1 Current Status of Transport SDN

While ONOS already offers some abstractions to manage optical devices, it completely lacked two main requirements of our use case (Row 4 and 5 in the requirements listed in Table 6.1):

- ONOS is intended to act as single control plane software and is designed to be directly connected to the devices (both optical and IP) that ONOS manages. However, Infinera does not allow such direct interaction with the physical devices, and offers a central management software (OTS) that is in charge of the configuration. This is common behaviour for transport vendors that usually prefer to hide the complexity of the optical feasibility estimation and to expose a simplified northbound interface to collect information and require for services. To overcome this limitation, in collaboration with the ONOS engineers, we had to design and develop a new ONOS Adapter as an extension of the existing REST provider, to support multiple devices under the same REST connection. This modification required the improvement of several ONOS modules to offer this new functionality, while being still compliant with the standard behaviour. The developed code has been included in the ONOS codebase, starting from ONOS 1.9 release.
- The specific REST API and workflow implemented by the Infinera OTS has to be supported by the SDN controller. The ONOS southbound interface is designed to be modular and easily extensible. Following this approach, the REST adapter, composed of a provider and a protocol module, is generally enough to offer an abstraction for all the use cases and solutions that leverage the REST protocol to communicate with the Infinera DTN-X devices. However, to support vendor-specific extensions and interfaces, ONOS allows definition of drivers that are associated with specific equipment and implement the detailed calls and workflows supported by the devices. Following this approach, we have designed, developed and tested a new driver to manage Infinera DTN-X devices through the northbound REST API exposed by OTS. This interface is covered by an NDA, so the code is securely stored in the GÉANT repository and is not shared with the ONOS community.

After the development of the ONOS driver for the support of Infinera OTSv node and the appropriate REST protocol extensions for the communication at the southbound interface (SBI), a major challenge since the beginning of the GN4-2 project has been the complete integration of OTS with ONOS core, i.e. also with the ONOS Intent Framework. This integration would allow any ONOS northbound application to use the ONOS' intent abstractions to control the optical transport network devices and would pave the road for multilayer support, that would allow combination of

GN4-2 L2, SDX-L2 and SDN circuit on demand use cases to be combined with SDN transport under a common control plane.

The major challenge of this complete integration is the fact that the ONOS interface assumes that it communicates directly with the switching devices at its southbound, as previously mentioned. However, OTS adds another level of abstraction and acts as a “proxy” or “child-controller” of the optical transport network. As a solution, the “Domain Intent” concept emerged via the collaboration of GN4-2 engineers, together with people from other R&E projects, namely “ACINO” [[ACINO](#)] and “ICONA” [[ICONA](#)], as well as ON.Lab experts. The Domain Intent solution addresses the case of a child-controller, sitting at the ONOS southbound, which is responsible for an external administrative domain, thus forming a hierarchy of controllers. When a request for policy application, e.g., for some connectivity at a given rate, concerns devices that belong to an external administrative domain, ONOS is able to identify this particularity and requests the installation of any policies, not directly, but via the child-controller. Domain intent is the abstraction that is used for the communication of the policies of an external administrative domain. The proposed implementation of the “Domain Intent” solution was recently approved by ON.Lab and is now included in the ONOS codebase starting from ONOS 1.10 release. However, the solution is not yet fully integrated with OTS since the OTS ONOS driver should be adapted to support the Domain Intent API.

The future steps for this use case depend on the network evolution planning and priorities, and will be determined in collaboration with SA1.

7 Working with Corsa

Corsa Technology solutions have been tested in the JRA1 lab environment, starting with the DP6400 switching platform. During the course of this testing, the task has closely collaborated with the vendor in order to obtain support for the multi-table pipeline driver needed by the SDN controller in order to install flows.

Subsequently, Corsa made its newer platform of DP2000 switches available, which was obtained by GÉANT both for the lab and the pilot environments. A major feature of the DP2000 platform is the support for Virtual Forwarding Contexts (VFCs), and the resulting capability to present various types of virtual switches to the SDN controller. These include, among others, a generic, single table OpenFlow VFC type and a multi-table Layer 3 VPN VFC type designed for IP Routing. (All of the VFC types are expected to be multi-table after a Corsa software upgrade during Q3 2017.) The former VFC type was selected for SDX L2 and SDN circuit on demand use cases, while the latter was selected for SDX L3/SDN-IP, due to its focus on scalable IP Routing.

During the development and experimentation with the new switches, JRA1 was in constant communication with Corsa engineers that provided the support and development assistance into getting the software operational with the hardware platform. GÉANT and JRA1, on the other hand, provided feedback to Corsa for bug fixes and feature suggestions, as realised during the testing.

The most important issues that had to be resolved in collaboration with Corsa over the course of the project and pilot preparations included:

- Support for metering at VFC layer of Corsa DP2000 in order for meters to be visible by the SDN controller when controlling a virtual switch.
- Implementation of an ONOS driver for the multi-table pipeline of the L3VPN-type VFC instantiated within Corsa DP2000.
- Bug fixes observed during testing of the use cases over the Corsa equipment.

All of the above issues were handled and ultimately resolved in coordination with Corsa.

Furthermore, a number of issues were investigated in collaboration with Corsa, and were deemed to be issues related to software layers (ONOS controller and applications), thus resolved after proper controller adjustment. The most important issues were:

- Support of a group table for VPLS multi-point operations instead of multiple-output action generation.
- Filtering of clear deferred actions in the context of a Corsa-specific extension of ONOS default single table pipeline driver.

8 Working with Hewlett-Packard Hardware: Janet Testbed

The HP3800 switch provides OF 1.3 support in the hybrid mode. The model supports all the OF functionality required by DynPaC, including:

- Forwarding rules based on VLAN IDs and ingress port numbers.
- VLAN translation.
- Drop meters.

It was decided that it would be useful to test DynPaC on HP3800 switches to see how the software works on the switches of a popular vendor, and whether it has any vendor-specific features. Ideally, it shouldn't have any, as the corresponding vendor driver should mask them.

Another reason to test DynPaC on the Janet testbed was for its suitability along with the main GÉANT pilot infrastructure deployment in the multi-domain scenario.

More details of testing DynPaC in the JANET lab are provided in Appendix A.

8.1 Findings

The problems discovered during DynPaC testing in Janet SDN testbed have been:

- ONOS (up to version 1.10) does not support HP3800 switches: it does not have a specific driver for HP3800 switches and the ONOS default driver uses table 0 as default table instead of table 100 used by HP3800.
- DynPaC generated flow rules using an action order unsupported by HP3800: the first action uses the output port number, the second one - the VLAN ID, but the order should be the opposite.
- ONOS can't detect links using BDDP when switch ports are not assigned to the first VLAN ID of a OF instance (VLAN 150 in the testbed configuration).

Appendix A provides further insight to how these issues were addressed by the JRA1 team.

All the works inside the Janet testbed created a great opportunity for ONOS and DynPaC software to be tested in another close-to-production environment. As a result of this work, the driver for the HP3800 was developed. At the end of research and development activities, the new software has been tested in the ONOS 1.10.0-SNAPSHOT and ONOS 1.11.0-SNAPSHOT environments, with the two HP3800 switches. After testing in the Janet lab testbed, the driver code was submitted for the ONOS code review, as it was important to share the outcome with the ONOS community. New commits were pushed to the DynPaC repository for VLAN range configuration and priority value configuration for flows using DynPaC.

9 Working with ONOS

GÉANT has recently joined ONOS core members and other affiliated organisations in a working group called Deployment Brigade [\[BRIGADE\]](#). Brigades are teams focused on developing specific features to ship in the upcoming versions of ONOS. Once formed, anyone else in the community with interest in such features is able to join the group and contribute towards it. The Deployment Brigade was the first attempt to use the model in the ONOS community, which turned out to be an effective way to recruit community members and ship new features.

The goal of the Brigade is to build a software stack running on top of ONOS, able to facilitate the adoption of the platform in field trials and production deployments by RENs and operators. Basic requirements include: providing Layer 2 and Layer 3 functionality to network users, convergence of packet and optical resources, and compatibility with major standards. After only a few months of work, the Brigade released non-trivial software components, already part of the SDX L2 Pilot within GÉANT. Figure 9.1 shows the converging stack the brigade members agreed to work on. Using the same applications allows the whole group to benefit from better test coverage, to obtain more significant feedback from a larger community and receive better support from the core developers while still allowing for minor deployment-specific customisations.

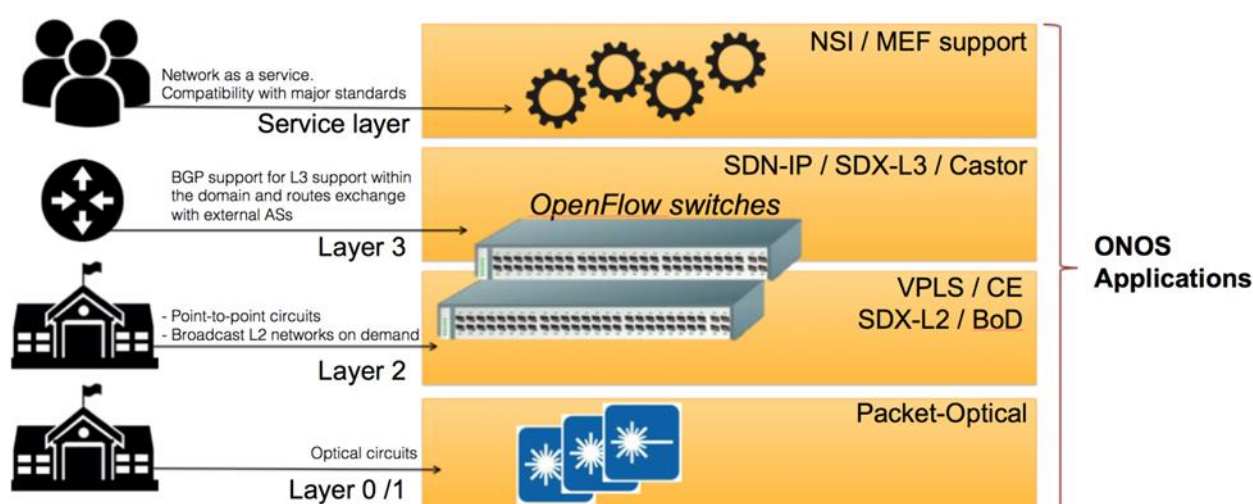


Figure 9.1: Software stack under development by the Deployment Brigade

The stack that the group decided to share and improve includes applications to help operators manage different network layers. Specifically, VPLS [\[VPLS\]](#) is used to create on-demand Layer 2 broadcast networks, Carrier Ethernet [\[CE\]](#) supports MEF standards for the creation of Layer 2 services such as E-Lines (point-to-point circuits) and E-LANs (multi-point broadcast networks), SDN-IP [\[SDN-IP\]](#) transforms SDN networks into IP transit networks, translating BGP messages exchanged with external ASes to OpenFlow entries on the switches inside the domain, and packet-optical [\[PO\]](#) is able to manage the packet and the optical layers, both during normal operations and after network failures.

The activities of the Deployment Brigade include the design, the development of new software components, and their integration; Quality Assurance and deployment activities. Members usually

take part in different sub-projects and activities, depending on their background and interests. The Deployment Brigade has produced a two-fold contribution: first, it worked with GÉANT to evaluate and take advantage of new SDN solutions, thereby encouraging a wider community to be part of this work and improve its chances for success. Second, it gave back to the ONOS community with significant artefacts (including new applications, features, improvements and bug-fixes), made available in the ONOS repository and to other network operators.

The Brigade initiative has been so successful that the ONOS project has expanded the number of active Brigades in 2017. The Brigades initiative also created the opportunity for GÉANT team member internships at ON.Lab, enhancing the communication between the team members and shortening the learning curve of the ONOS architecture. This way of working allows GN4-2 project teams to produce outcomes directly to the open-source code base and make the project visible to a part of the industry that is starting to see SDN as something more tangible. Also, participating in the Brigades has produced an unexpected benefit to the team in the form of advice and patches.

In the spirit of open source, all the developed applications have been made available to the broader community. The work of the Brigade and the results of the tests have been showcased and demonstrated at different international meetings and conferences. The Brigade is now extending the ONOS intent framework to introduce the notions of queues, meters and bandwidth, and is evaluating how the actual applications can run over the framework once all the functionalities are in place. As opposed to working as stand-alone applications, both DynPaC and CE will be integrated with the stack, bringing compatibility with the NSI and MEF standards while reusing the primitives offered by SDX-L2 and VPLS. SDX-L3 and SDN-IP will be the first integrated and then merged with the rest of the stack; tested and deployed. Finally, additional tests will be conducted before moving the platform to production, in order to ensure the robustness of the solution before it will be offered as an official service to the broader community.

10 OF Conformance Scripts

In order to verify the progress of the Corsa firmware and support piloting activities, a suite of tests has been developed. The tests are realised using OFTest [\[OFTest\]](#), a community project started by Big Switch Networks [\[BIG SWITCH\]](#). It is a python based framework originally developed to test the compliance of the switches with the OpenFlow (OF) protocol. Ideally, these tests should be run at each release of Corsa firmware before proceeding with upgrades to the Corsa devices in the lab and in the pilot infrastructure. The objective is to identify in time regressions or eventual problems and avoid unnecessary downgrades.

The test library has a minimum set of requirements, it needs a host or a virtual machine equipped with three interfaces (two interfaces are connected with the data-plane in a closed loop and one is used for the control-plane). The switch to be tested needs to have two ports and a management interface. If the VM can be equipped with more than two interfaces, it is possible to create several VFCs on the switch and test different pipelines/configurations in parallel without the need to re-configure everything. Figure 10.1 shows the setup for executing the GÉANT OF conformance tests considering a multi-VFC scenario.

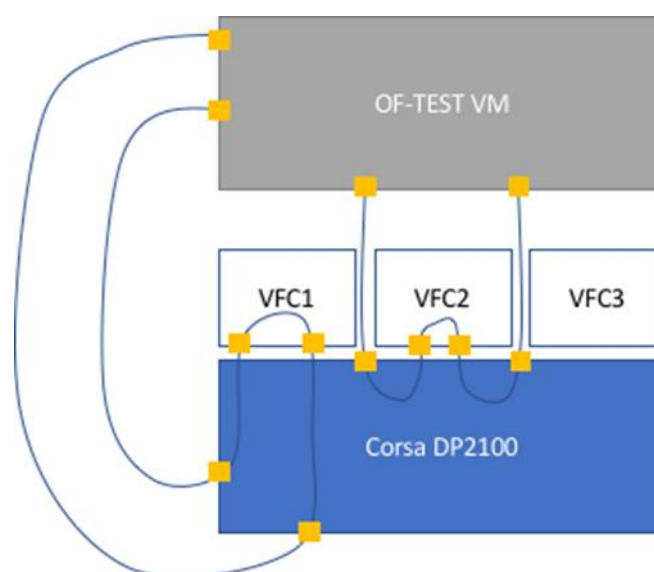


Figure 10.1: OF conformance setup

The workflow of tests is very simple and straightforward, it requires:

- Programming the switch according to the functionality to be tested.
- Sending a packet in the data-plane, using one of the available interfaces.
- Verifying that the packet received on the remaining interface matches the expected packet.

In order to facilitate test development, a virtual environment has been developed which does not require the presence of a real hardware switch as it emulates the Corsa DP2100 devices using a software switch.

At the time of writing, several tests have been developed. The tests are organised into three main categories, and a complete list is available for internal participants on the JRA1 Wiki. In the forwarding functionality tests, all procedures are meant to verify the basic functionality supporting our use cases, such as MAC address forwarding, VLAN switching, and so on. The BUG group contains all the tests developed to verify the progress of the Corsa software against the identified bugs. An example of this is the Clear Deferred action, which is currently unsupported by Corsa switches, and will be useful to try out the future firmware updates. Finally, Feature Requests contains the procedures meant to verify the support for the new functionalities introduced by Corsa. At present there is only one test to verify the progress of the Q-in-Q encapsulation with EtherType 0x8100 and 0x88a8 [\[Q-in-Q\]](#).

Testing results are not yet available, so they will be presented in the Part B Deliverable (D7.3). The source code is publicly available to project participants on the GÉANT Stash repository, under the section related to the GN4_1 JRA2 T2 project.

11 Monitoring

During JRA1's development efforts in the testbed environment (GÉANT's Cambridge Lab), troubleshooting and monitoring ONOS Deployments were found to be tedious tasks involving manual operations such as:

- Connecting to different nodes.
- Inspecting and filtering log files.
- Gathering metrics related to the nodes (status, resource consumption, etc.)
- Preserving information and metrics gathered for future reference.

In order to overcome the aforementioned problems, a solution was designed and implemented to facilitate and automate tasks. Typical examples include, but are not limited to:

- Collecting and storing data (e.g. important events, system metrics, logs).
- Visualising collected data and highlighting potential problems.
- Performing queries, as needed.

Figure 11.1 depicts a generic architecture designed and implemented as an indicative working prototype within the Cambridge lab environment. The prototype mainly utilises components from the Elastic Stack. Namely, we use Beats (FileBeat, MetricBeat, PacketBeat) as lightweight shippers for Log Files, System/Network related metrics. Additional information can be collected from the network elements using traditional methods such as SYSLOG, and SNMP. The aforementioned lightweight shippers send all gathered data to a processing Node running Logstash software.

The focus has been mainly on SDN controller log files, CPU utilisation, RAM consumption, disk status, per-flow byte and packet counters (grouped by their L2-L4 headers). In turn, the input data stream is structured and enriched based on a predefined pipeline processing and the output is exported to a datastore. Elasticsearch suited the particular needs of JRA1, and access to the data stored on instances is either via a RESTful API or by using specific tools for performing queries and creating visualisations (e.g. Kibana, Grafana).

Within the SDN-based circuit on demand Use Case the platform analysed above has been used more as a debugging and less as a monitoring tool. Specifically, queries are performed on the datastore based on instances, log timestamp, and severity of the message. In essence, any information stored in Elasticsearch may be visualised (e.g. CPU/RAM usage as a time series data). Note that the logs shipping procedure to the Portal Webhost instance, and IDM instance has also been extended.

Pending proper configuration, the platform can be tailored to a number of setups, depending on the operational needs/constraints and providing considerable operational benefits (e.g. troubleshooting a circuit service instance, monitoring the deployment of an SDN-based circuit on demand instance), when used in a production environment.

Next steps within the project lifetime include: a) linking specific logs with tenants/their respective service requests, b) multi-source data correlation towards identification of strange events – system problems.

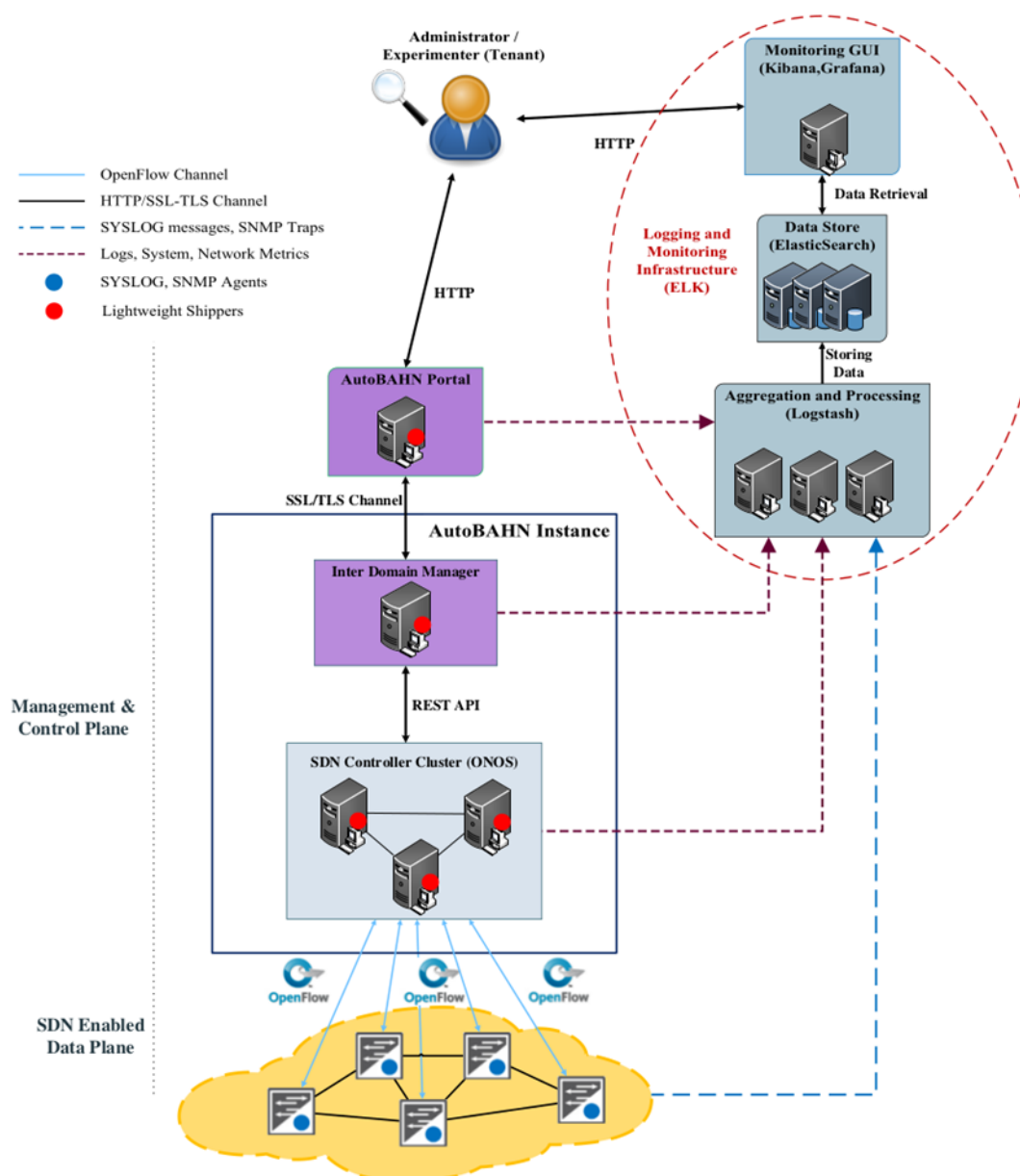


Figure 11.1: Monitoring architecture

Additional information on accessing and using the infrastructure is available to project participants at the JRA1 Monitoring and Infrastructure Wiki.

12 Conclusion

Technology pilots are very important as early steps of bringing use cases to production, as they provide the necessary operational feedback and real-world insight that cannot be obtained in a development/lab environment. This has been the case with the SDX L2 pilot, and the same is expected to occur with the upcoming SDN-based circuit on demand pilot. Operational teams have had hands-on experience with the pilot functionality, trialled features and the operational readiness of the SDX L2 application to a great extent and provided valuable feedback for the evolution of the solution and its maturity. Additionally, the constantly changing codebase of ONOS and related applications provide a basis for evolution and further testing of the pilot functionality. The experiences and findings will be used to further iterations of the technology pilots so that the solutions reach the maturity required for user-driven pilots and later on pre-production deployments. The team has also made significant contributions to open source software, both via commits to the core ONOS codebase and the SDN applications developed within the Task.

In addition, since monitoring is vital for the operation of a service in production, a monitoring solution has been designed and prototyped, which can be extended in a straightforward manner to more than one use case.

Aside from the technology pilots themselves, emphasis has been placed on verification of the use case functionality and operational readiness upon OpenFlow-enabled equipment from different vendors. The main goal has been to avoid implicit hardware-specific assumptions and subsequent vendor lock-in. Furthermore, OpenFlow conformance scripts have been developed in order to automate the verification of the required functionality for new switches and controller software upgrades.

Regarding use cases that have not transitioned to technology pilots (SDX L3 and Transport SDN) progress has been made in the lab environment and are currently under evaluation for planning the further steps in Period 2 of GN4-2.

Appendix A Details of DynPaC Testing in Janet Testbed

A.1 Janet SDN Testbed

The Janet SDN testbed consists of four HP3800 switches located in London, Manchester, Leeds, and Bradley Stoke PoPs, as shown in Figure A.1.

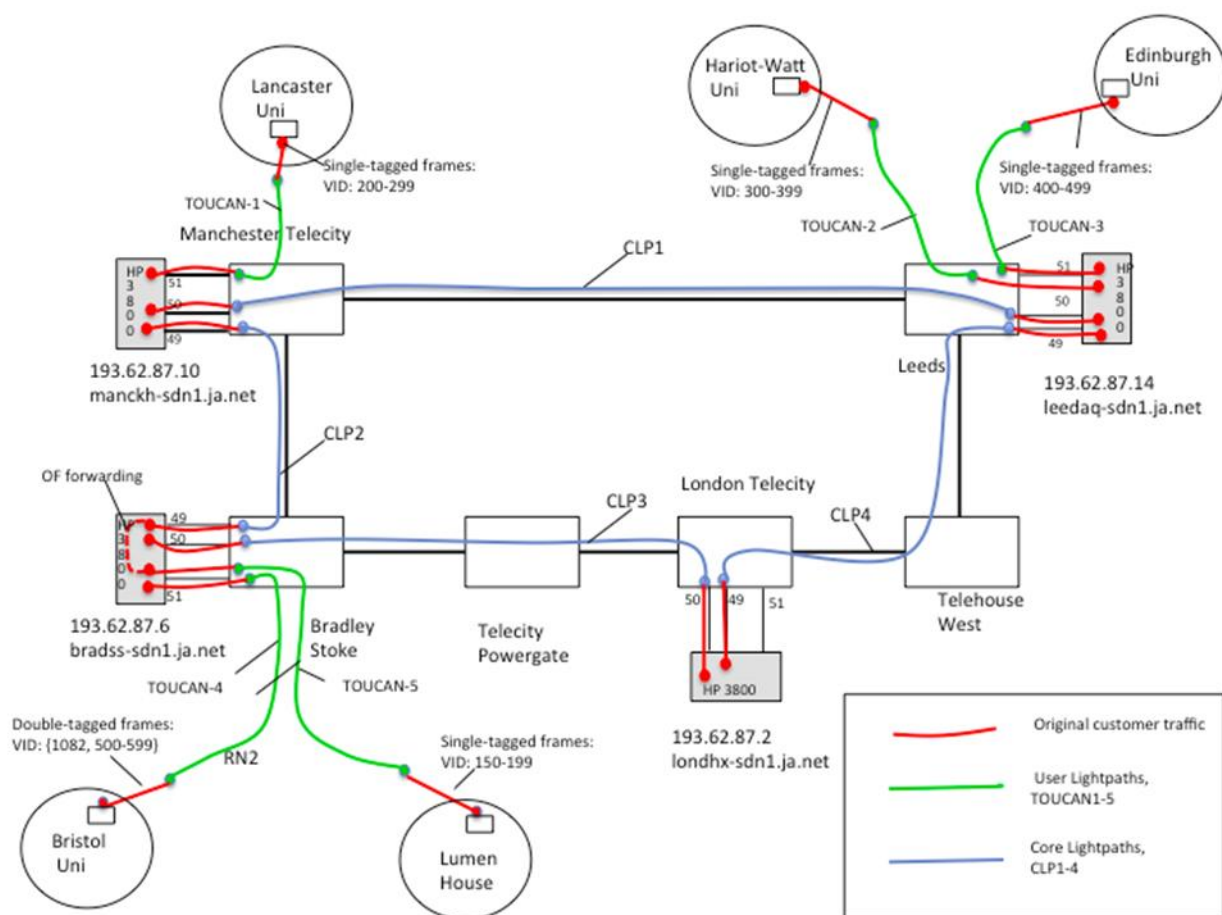


Figure A.1: Janet SDN testbed

The testbed is an overlay network built over the Janet Lightpath infrastructure. It uses EoMPLS circuits of the Janet Lightpath infrastructure to connect switches to user sites (1G circuits to ports 51) and to each other (10G circuits between ports 49 and 50).

The Lightpath circuits are transparent for the testbed traffic, so the HP3800 switches receive frames exactly in the form users send them.

User traffic must be tagged. In order to differentiate users on the shared port 51, each user site is assigned a unique range of VLAN IDs.

The switches are configured to support several virtual OF instances, and each OF instance is bound to a unique set of VLAN IDs and can be controlled by several SDN Controllers.

The testbed virtual instances (slices) are created on a per-test basis. For example, if test A involves two user testbeds, say Lancaster and Lumen House, then instance A should include sets of VLAN IDs of the both users.

To prevent loops and an unwanted LAN-style connectivity, it is recommended to assign ports to VLANs in a non-contiguous way, e.g. if port 51 of London Telecity switch is assigned to VLAN 151, then ports 49 and 50 of that switch should not be assigned to VLAN 151.

A.2 Janet SDN Lab

To verify the fact that DynPaC can work with HP3800 switches, on which the Janet distributed testbed is built, Janet SDN lab testbed was used. It consists of two HP3800 switches located at Lumen House, Oxfordshire (Janet/JISC headquarters).

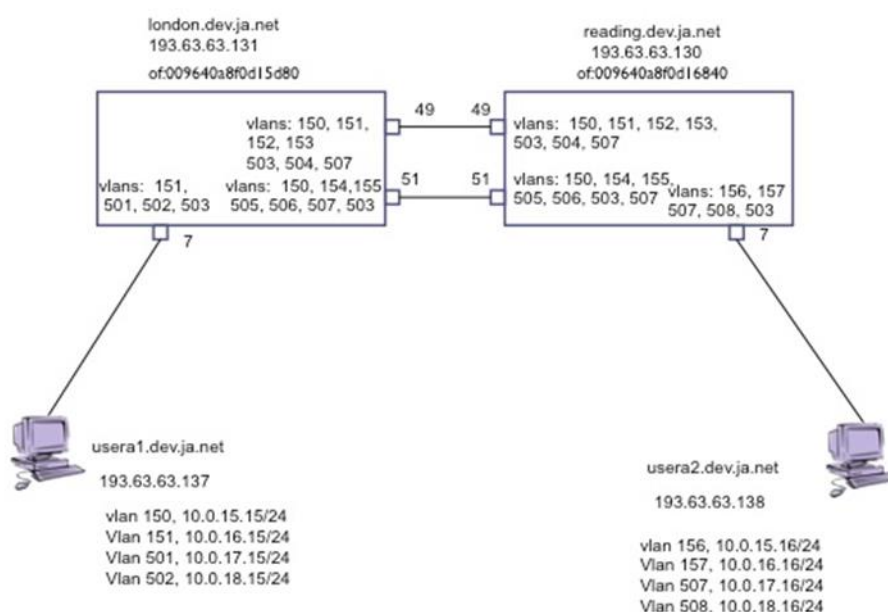


Figure A.2: Janet SDN lab

The switches are interconnected by the two 10G links (ports 49 and 51) to create a physical loop, and hence, emulate the ring topology of the Janet SDN testbed as close as possible. There are two hosts connected to port 7 of the switches that emulate user sites (Figure A.2).

The SDN lab configuration has been tested and the tests objectives were:

- To provide a connectivity between the hosts interfaces sitting on different VLANs.
- To check whether a drop meter limits traffic according its rate parameter.

Both tests brought the positive result but only after some changes in both ONOS and DynPaC have been done. Some problems still have to be solved, so we had to change the configurations of the HP3800 switches to make them to accept some DynPaC rules and to support ONOS link auto-detection functionality. These configuration changes are reflected in the lab testbed diagram above:

- VLAN 150 is assigned to the all ports on both switches to support ONOS BDDP auto-detection.
- VLAN 503 is assigned to the all ports on both switches to test the scenario without the DynPaC VLAN translation function.
- VLANs 501 and 507 are assigned to the ports in a way that corresponds to the current DynPaC limited VLAN translation technique.

When the remaining problems (described along with the solved problems in detail below) are solved the testbed config will be changed to be in line with the VLAN assigning rule (non-contiguous).

A.3 ONOS Support for HP3800 Switches

Up to ONOS 1.8.x the ONOS default driver was patched with the changes developed by Lancaster University.

In ONOS 1.9.0, PacketService was significantly changed and all APPLY_ACTIONS were accompanied with CLEAR_ACTIONS instruction in FLOW_MOD messages to prevent sending a packet to the controller and generating its copy to dataplane in TTP scenario. For example, assume the packet has a match in table i-1 and apply_actions are performed. An action_group is set in the write_actions and there is goto_table to i instruction. In table i the packet matches again, output_to_controller is performed and the write_actions in i-1 are performed, which also generates a copy of the packet in the data plane.

HP3800 switches do not support the CLEAR_ACTIONS type of instruction described, instead, they reject all the handshake FLOW_MOD messages. As a result, it was decided to develop a full ONOS HP3800 driver instead of patches to solve CLEAR_ACTIONS and some other issues.

Adhering to the ONOS principle that a driver should handle special device characteristics, the JRA1 team developed an ONOS driver for HP3800 switches based on the Lancaster University patches. The driver does not add CLEAR_ACTIONS (clear deferred in ONOS terminology) instruction in FLOW_MOD messages and also solves some other issues, such as:

- Filtering ETH_TYPE=VLAN criterion (not supported by HP3800).

- Filtering SET_QUEUE actions (not supported by HP3800).
- Correcting the order of actions in DynPaC FlowObjectives.

Another problem solved was the flow priorities installed by DynPaC. DynPaC generated flow rules with lower priorities than ONOS default controller apps (ARP, LLDP, BDDP). Due to the lower priorities, user packets did not match DynPaC rules and were instead sent to the ONOS controller, according to the default rules. Modification was made to DynPaC to prioritise DynPaC flows.

A.3.1 Actions Order in DynPaC FlowObjectives

Up until ONOS 1.8.0, when the patched default driver was used, DynPaC had to take care of the order of actions, i.e. to specify the VLAN ID first and then the output port. The developed HP3800 driver solved this problem with correct FlowObjective to FlowRule translation, resulting in the proper actions order.

A.3.1.1 BDDP Link Discovery

ONOS uses LLDP and BDDP packets to detect network infrastructure links. The tests showed that ONOS could not detect links between HP3800 switches using BDDP when switch ports were not assigned to the first VLAN ID of a OF instance (VLAN 150 in the testbed config). This happened because ONOS injects untagged BDDP packets to packet_out messages, and then HP3800 switches forward BDDP packets tagged with the first VLAN from the instance range. As for LLDP packets, they are not propagated by HP3800 switches, even when a port supports the first VLAN from the instance range.

There are a number of solutions to improve link detection:

- Short-term solution: all ports of two switches are assigned to VLAN 150. As there is a logical loop on VLAN 150, this VLAN is not used for user reservations. This solution was tested and proved successful.
- Long-term solution 1: to generate BDDP packets tagged with VLAN IDs supported by switch ports as LLDP-based detection does not work at all on HP3800 switches. To implement this solution, new ONOS CLI and REST API commands will be implemented. It will generate (during the specified period of time or specified number of packets) BDDP packets with the specified VLAN ID tag and send these packets in packet out messages through specified port or ports, as different ports might be bound to different VLANs.
- Long-term solution 2: to edit the topology manually, correcting the auto-detected topology. This solution will be implemented in a separate ONOS application. It will be enabled to read the topology information from a JSON file. However, the remaining problem is how to automatically track topology changes.

A combination of the two, long-term solutions looks the most promising.

A.3.1.2 VLAN Translation

At present, the DynPaC translation technique only takes place at the switch egress. DynPaC cannot take into account whether intermediate ports (in the same DynPaC domain) along the path support some VLANs or not. The short-term solution is to configure VLANs on switches in a way that fits the

DynPaC translation technique. The long-term solution will be to improve this technique. Works have been started and part of the DynPaC code (executive part of code responsible for flows formatting) modified to support more flexible translation functionality. The next step will be to find the way to enter VLAN/port bindings data into ONOS (ticket 374).

References

[ACINO]	ACINO project http://www.acino.eu/
[AutoBAHN]	https://geant3.archive.geant.org/service/autobahn/pages/home.aspx
[BIG_SWITCH]	http://www.bigswitch.com/
[BRIGADE]	https://wiki.onosproject.org/display/ONOS/Deployment+brigade
[CE]	Carrier Ethernet ONOS application, https://wiki.onosproject.org/display/ONOS/Carrier+Ethernet+Application
[DynPaC]	DynPaC ONOS application, https://wiki.onosproject.org/display/ONOS/BoD+DynPac+application
[GN4-1 POC]	Deliverable D14.1 Future Services PoCs - Implementation and Dissemination, GN4-1, April 2016
[GOPEN]	GÉANT Open Service, https://www.geant.org/Services/Connectivity_and_network/Pages/GEANT_Open.aspx
[GTS]	GÉANT Testbeds Service, https://www.geant.org/Services/Connectivity_and_network/GTS
[ICONA]	ICONA project, https://wiki.onosproject.org/pages/viewpage.action?pageId=4162523
[NSI]	NSI Connection Services v2.0 - Open Grid Forum https://www.ogf.org/documents/GFD.212.pdf
[OFTTEST]	OFTtest homepage, http://www.projectfloodlight.org/oftest/
[ONF]	Open Networking Lab, now known as the Open Networking Foundation following a recent merger https://www.opennetworking.org/
[ONOS]	Open Network Operating System, http://onosproject.org/
[ONOS-BRIGADE]	ONOS Deployment Brigade, https://wiki.onosproject.org/display/ONOS/Deployment+brigade
[ONOS-CLUSTER]	https://wiki.onosproject.org/display/ONOS/Forming+a+cluster
[P4]	The language for programming the data plane of network devices: http://p4.org/
[PO]	Packet Optical ONOS use case, https://wiki.onosproject.org/display/ONOS/Packet+Optical+Convergence
[Q-in-Q]	http://www.cisco.com/en/US/docs/ios/lanswitch/configuration/guide/lsw_ieee_802.1q.html
[SDN-IP]	SDN-IP ONOS application, https://wiki.onosproject.org/display/ONOS/SDN-IP

- [SDXL2] SDX-L2 ONOS application,
<https://wiki.onosproject.org/display/ONOS/SDX-L2+application>
- SDX L3 ONOS application,
<https://wiki.onosproject.org/display/ONOS/SDX-L3+application>
- [TESTON] <https://wiki.onosproject.org/display/ONOS/System+Testing+Guide>
- [VPLS] VPLS ONOS application,
<https://wiki.onosproject.org/display/ONOS/Virtual+Private+LAN+Service+-+VPLS>

Glossary

ARP	Address Resolution Protocol
AS	Autonomous System
BDDP	Broadcast Domain Discovery Protocol
BGP	Border Gateway Protocol
BoD	Bandwidth on Demand
CE	Carrier Ethernet
CLI	Command Line Interface
CP	Cooperation Protocol
CMDB	Configuration Management Database
DTN-X	Infinera's next-generation multi-terabit transport network platforms
DynPaC	Dynamic Path Computational Framework
EBGP	Exterior Border Gateway Protocol
EoMPLS	Ethernet over MPLS
FIB	Forwarding Information Base
GUI	Graphical User Interface
IBGP	Internal Border Gateway Protocol
ID	Identity
L2	Layer 2
L3	Layer 3
LACP	Link Aggregation Control Protocol
LLDP	Link Layer Discovery Protocol
MAC	Media Access Control
MEF	Metro Ethernet Forum
MP2SP	MultiPoint to Single Point
MPLS	Multiprotocol Label Switching
MX	Juniper series of Ethernet routers and switches
NDP	Neighbour Discovery Protocol
NOC	Network Operations Centre
NSI	Network Service Interface
OF	OpenFlow
ONF	Open Networking Foundation
ONOS	Open Network Operating System
OTS	Open Transport Switch
OMP	Open eXchange Points
PCE	Path Computational Element
PXM	Packet Switching Module
PO	Packet Optical
POC	Proof of Concept

PoP	Point of Presence
REST	Representational State Transfer
SA	Service Activity
SBI	SouthBound Interface
SDN	Software Defined Networking
SDX	SDN eXchange point
SNMP	Simple Network Management Protocol
STP	Spanning Tree Protocol
VC	Virtual Circuit
VFC	Virtual Forwarding Contexts
VLAN	Virtual Local Area Network
VM	Virtual Machine
VPLS	Virtual Private LAN Service
VSC	Virtual Switch Context
Y1	Year 1