Open Networking Foundation

5th Wireless Transport SDN Proof of Concept White Paper & Detailed Report

Version 1.0

2019-03-19

Content

1 Introduction

The 5th Open Networking Foundation Proof of Concept (ONF PoC) was organized by the Wireless Transport Project (WTP) and has been hosted by Telefonica in Munich during the week starting with 26th of November 2018. It was integrated into the Casablanca ONAP PoC, which has been executed by AT&T in parallel.

The Wireless Transport Project aims at enabling non-proprietary open source controllers to manage multi-vendor transport networks. This innovative approach leads to a modular, flexible and heterogeneous SDN architecture with open interfaces, non-proprietary controllers and free choice of applications.

ONF Proof of Concepts are executed roughly every nine months to demonstrate the progress in interface development and to verify the functionality and completeness of implemented enhancements in a multi-vendor environment using commercially available devices. Findings made during the PoCs are determining the standardization activities during the months following the event. In general, such reference implementation driven development is seen as an effective way towards industry broad adoption of the resulting interface definitions.

The 5th PoC emphasized the maturity of the Microwave Information Model (ONF TR-532 [2]) by showing a broad variety of commercially available applications and tools supporting the standard interface. At the same time, a broadening of the activities has been shown by testing new technology specific interface definitions for Ethernet PHY and optical transport for the first time. Further on, definitions for unifying the Netconf interface’s behavior and automated commissioning (both ONF TR-545 [4]) have been tested.

ADVA, Ceragon, Congiv, Infinera, Enghouse, Ericsson, highstreet technologies, Huawei, Indra Sistemas, Intracom Telecom, Mentopolis, Mycom OSI, NEC, Nokia, SIAE Microelettronica, SM Optics, Tech Mahindra, Wipro and ZTE contributed to the 5th PoC. In summary, more than 56 representatives of 21 companies were on site to perform tests and to present their results.

Many software codes, which have been developed during the 5th PoC, are stored and publically available as open source at the CENTENNIAL GitHub [7].

2 Objectives of 5th ONF PoC

Adoption of TR-532 Microwave Information Model

Eight applications and tools were able to manage the hardware of eight microwave vendors via the Microwave Information Model defined in ONF TR-532 [2]. Six of these applications and tools are commercially available and can be immediately deployed in any transport network.

Version 1.1 of TR-532

The modifications, which are planned to be introduced with version 1.1 of ONF TR-532, have been successfully tested.

Ethernet PHY Model

Already in 2017, a model covering most of the Ethernet PHY definitions according to IEEE 802.3 [6] had been elaborated, but never been discussed in a wider audience. The components of the Ethernet PHY model are seamlessly integrating with the ONF Microwave Information Model and have also been tested in this PoC.

Optical Transponder Model

The ONF Transport API (TAPI) definitions in their latest version 2.1 [5] was amended by a technology specific extension for optical equipment. During the 5th ONF PoC, this optical model has been tested as an extension of the ONF Core IM TR-512 [1] allowing basic management of partially disaggregated optical networks.

Unified Management Protocol Layer

During last year’s trial and PoC, it has been found that even interface implementations, which are based on a complete and well defined Yang data model, can behave very differently from vendor to vendor. The ONF Device Management Interface Profile (DMIP) sub-project had been founded unify the behavior on the southbound interface of the controller for ensuring unambiguous states and responses of different vendors' devices.

The DMIP group published TR-545 [5] in October 2018. During the 5th PoC several core elements of TR-545 have been successfully tested.

Automated Commissioning

There are hundreds of different ways of automating the commissioning of new devices. Operators as well as vendors would very much benefit from a unification of the process, which always starts with the device requesting for an IP address and announcing its presence to the controller. These first steps have been described in ONF TR-545 and also been tested in the 5th PoC.

Architecture

All aforementioned models are based on the same ONF Core IM v1.2 [1]. They were all running on the same SDN controller (OpenDaylight release Oxygen SR3), which was also used in the ONAP PoC. Some of the applications support even both versions of the microwave model (1.0 and 1.1) in parallel.

The 5th ONF PoC's set-up was composed of 37 individual contributions of the aforementioned companies in total.

3 SDN Network Architecture and Configuration

3.1 Overview

The overall SDN architecture and configuration of the test set-up in the 5th ONF WTP PoC is illustrated in Figure 1 below:

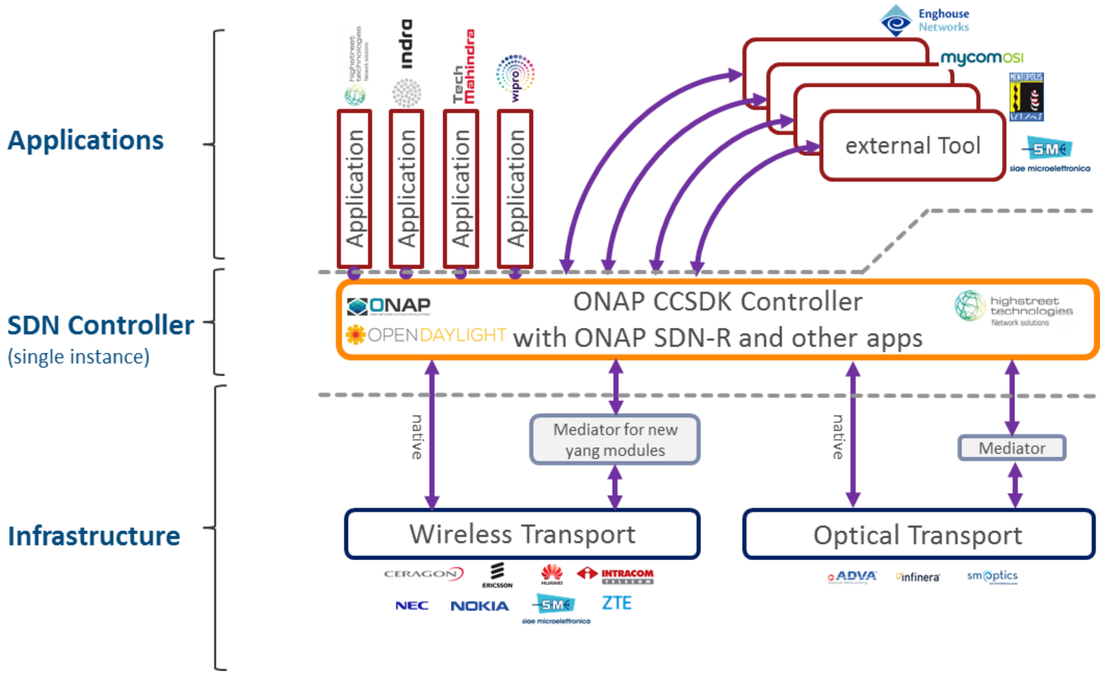


Figure 1: Overview of SDN architecture used in 5th ONF WTP PoC

The architecture identifies the main objects which were target of implementation and subsequent verification:

- Physical Network Functions (devices) of different technologies: wireless transport, Ethernet-PHY, optical transport supporting different Yang modules and Netconf capabilities

- SDN controller (ONAP CCSDK SDN-C[1] with OpenDaylight Oxygen SR3)
- Applications and external tools

Several device vendors implemented the Netconf servers inside the device itself. However, in order to avoid issues and dependencies with product development the new Yang modules were implemented as an external mediator in most cases.

The network deployment included a single SDN controller instance. This controller is based on the ONAP CCSDK SDN-C (OpenDaylight Oxygen SR3) and was enhanced with applications as OSGi bundles from the ONAP SDN-R project and other applications to demonstrate the targeted use cases. ONF Yang modules, which were supporting several different technologies also in different backward compatible releases, were simultaneously applied at the southbound as a Netconf interface and at the northbound as a Restconf interface. The SDN controller used the Candidate Data Store according to ONF TR-545 [4] for configuring devices, which are already supporting this optional feature of the Netconf standard IETF RFC6241 [8]. Otherwise, new configurations have automatically been written directly into the Running Data Store. Also the Netconf Call Home functionality has been supported by the controller.

All applications and external tools at the northbound of the SDN controller used the technology specific Restconf APIs or were provided by the SDN controller.

For development purposes and for preparation of the joined ONF and ONAP PoC, an ONAP CCSDK 3-node cluster was setup in DT-Cloud in Prague in addition to the Telefonica laboratory in Munich. It allowed the community to demonstrate that the ONF Yang models are also functional in an SDN controller cluster.

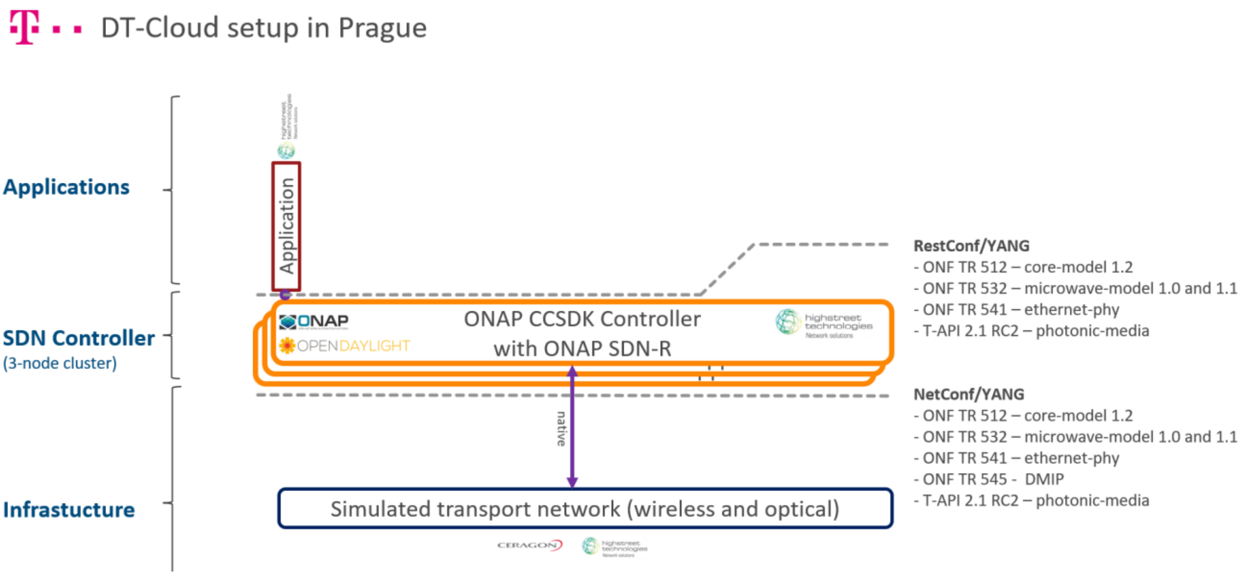


Figure 2: Clustered version of the SDN controller

This architecture used in DT-Cloud exists also in the ONAP Community Labs. In such labs the ONAP components are deployed as suggested by the ONAP community as Docker containers with Kubernetes on top of OpenStack.

[1] ONAP Common Controller SDK (CCSDK) project

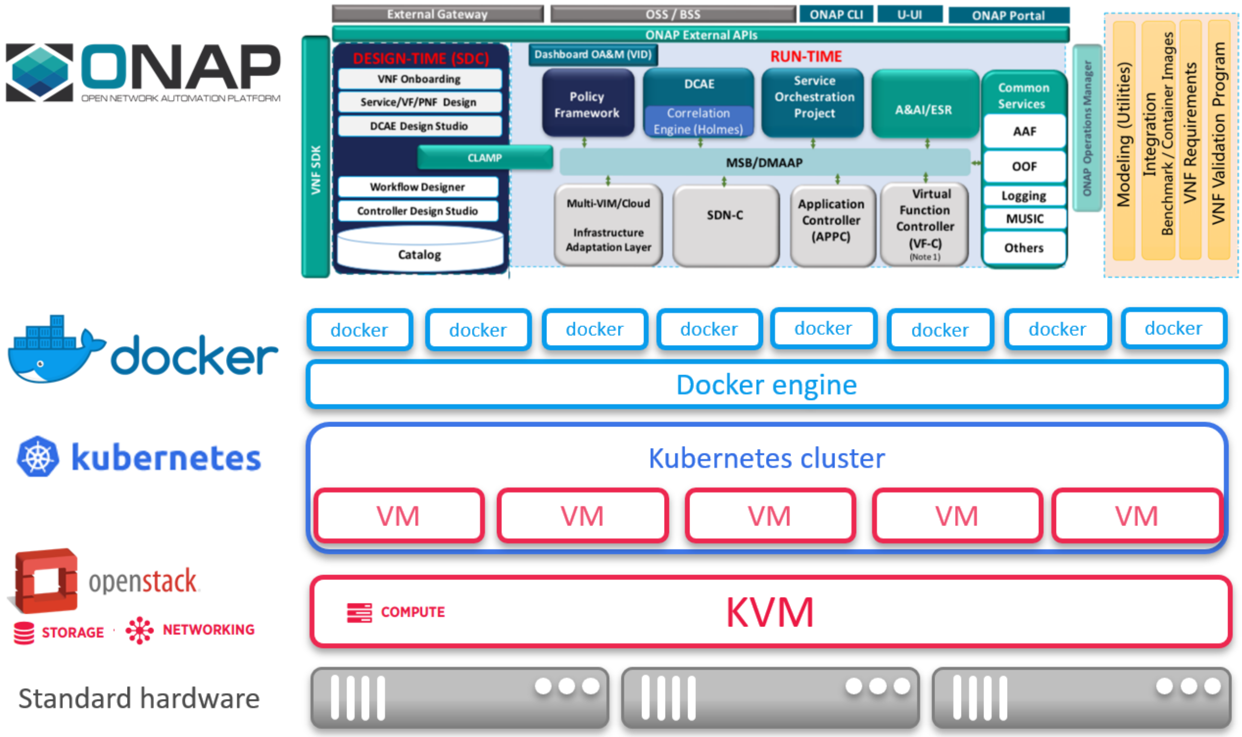


Figure 3: Software component deployment with OpenStack and Kubernetes

3.2 Test Network Setup

Telefonica provided one physical server (HPE DL560) in their laboratory in Munich-Freimann. Ubuntu 18.04 server has been used as host operating system and KVM as hypervisor with QEMU virtualization. Applications (northbound), controller, services (DHCP) and mediators (southbound) were run on this platform. Vendors were also free to use their own computers to run mediators and applications. Access to the environment was provided via VPN through Internet and locally via Wi-Fi.

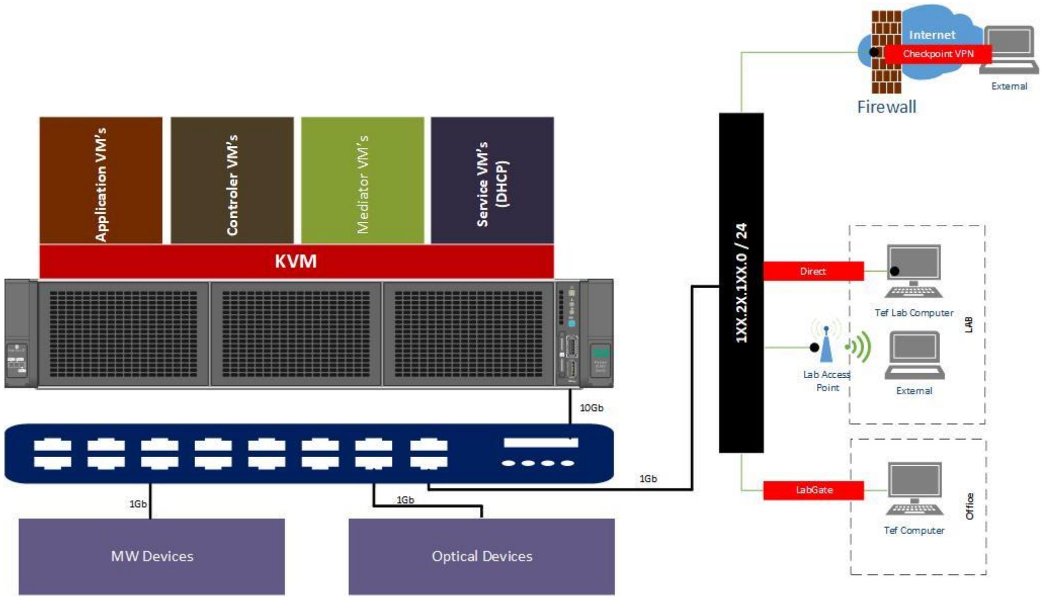


Figure 4: Lab Architecture and Topology

SDN control plane was implemented by directly connecting the management interfaces of the microwave and optical devices to a local Ethernet switch (star topology).

SDN data plane was implemented by connecting all the microwave devices in a linear daisy chain topology. Jperf application was used for generating test traffic on a data plane.

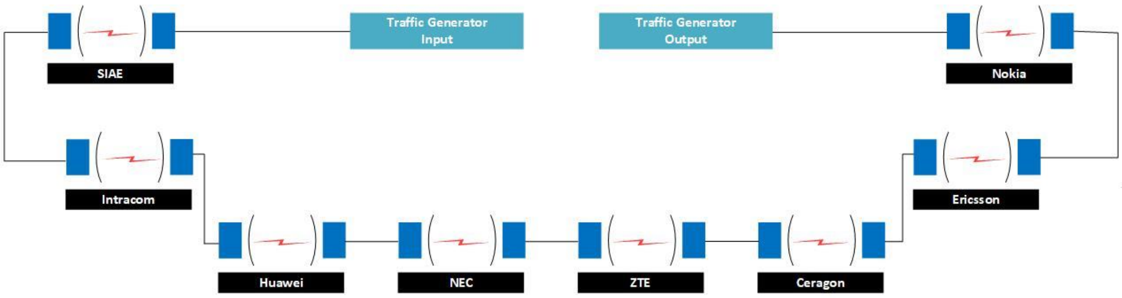


Figure 5: Microwave devices Data Plane connection

VIAVI MTS5800 handheld network tester were used for generating 1Gbit/s, 10Gbit/s and 100Gbit/s data traffic on an optical data plane as shown in Figure 6.

Figure 6: Optical devices Data Plane connection

4 Use Cases and Applications

4.1 Microwave

Microwave Information for Planners

Transport network planners shall see up-to-date configuration information about microwave links within their existing planning tools. This use case was an example for an existing software providing additional customer value by applying the vendor-agnostic microwave management interface. The Microwave Link

Information Use Case was focusing on retrieving and displaying Configuration, Status and Performance information about the microwave radio interface. It has been implemented by Enghouse and Mentopolis.

Preventive Maintenance

An SDN controller provides easy access to nearly real time information about the radio interface. Information of a granularity of minutes can be stored in a data store for later analysis. The Preventive Maintenance use case analyzed the stored information to prevent future network problems. Performance indicators of the radio interface have been analyzed for known patterns and correlated with external data like meteorological information. The Preventive Maintenance use case has been implemented by Indra Sistemas.

Microwave Network Performance

Operators need to have a detailed view on the performance of their microwave transport networks. Long-term monitoring and collection of performance values (e.g. availability indicators) creates a large amount of data to be analyzed and subsumed in handy overviews. Overviews shall allow drilling down into details about the most critical links' performance values. In the best case an outage can be prevented by immediate preventive action on these links.

An application provided by SIAE Microelettronica collected performance information from the links, correlating the data and presented it according to configurable KPIs.

Planning Data Reconciliation

Until planning using live network data will be implemented, the discrepancies between content of the planning data base and the actual live network is a major issue for operators. While automation of this data reconciliation is challenging, tool support is necessary.

SIAE Microelettronica provided an application that shows the initially planned configuration besides the currently active configuration and allows alignment operations, into both directions.

Smart Statistics Reporting

Every now and then, operators need very specific, but aggregated information about their network. Such need for information is often driven by operational or commercial questions. Some examples might be the exact vendor share or the number of operated outdoor units of a specific sub-band type.

An application provided by SIAE Microelettronica allows quickly configuring clear statistics (e.g. pie charts) about many equipment radio properties and other characteristics.

Configuration, Performance and Alarm Management

highstreet technologies provided a set of vendor-agnostic open source applications that covered the following activities

- automatically detecting new elements and displaying capabilities and current configuration and status of these elements
- continuously collecting and storing historical performance data, incl. comparing it with expected values and analyzing the devices' long-term behavior (e.g. unavailable seconds)
- receiving, displaying and storing notifications sent by the devices

Performance Assurance

For assuring and analyzing the experience on the currently active network, extracting time series of all kinds of performance data is required. MycomOSI provided an adaptor that retrieved microwave and Ethernet PHY performance values via the standardized interfaces from the SDN controller's north bound interface and imported them into their existing end to end performance assurance software.

4.2 Ethernet PHY

Ethernet Port Information for Planners

The Ethernet Port Information for Planners use case was focusing on retrieving and displaying Capability, Configuration and Status information about Ethernet ports, which are hosted by sub-structured hardware. It verified the capabilities of the new Ethernet PHY information model according to ONF TR-541 [3] together with the Equipment part of the Core Information Model TR-512 [1].

Enghouse equipped their existing network planning software with standard interfaces to implement this use case.

Ethernet Port Documentation Verification

Current transport planning processes often do not foresee any feedback loop from the live network to the planning database. Over the years, the content of the planning database diverges more and more from the actual live network. This leads to an increasing need for clarification of ambiguities during network deployment and device commissioning.

In this use case, an application provided by Enghouse compared the content of the planning database with uploaded information from the live network and marked discrepancies in the data base and on the graphical user interface of the planning software. The application supported the Ethernet PHY information model TR-541 together with the Equipment part of the Core Information Model TR-512.

4.3 Optical Transport

Basic Optical Transport Configuration

The Basic Optical Transport Configuration use case was focusing on a small set of configuration attributes at optical devices. It tested the capabilities of an optical model, which has been derived from attaching the new optical extension of the Transport API (TAPI v2.1) directly to the Core Information Model.

The application for this use case had been provided by Tech Mahindra.

4.4 Device Management Interface Profiling

Automated Commissioning

Current commissioning process requires skilled people on-site, causes significant administrative overhead and is error-prone. In future, devices, which get connected to the network, shall be automatically configured via Netconf protocol and compositions of technology specific interface definitions.

Such Automated Commissioning Process implies automatically establishing IP, TCP, SSH and Netconf connections between device and controller and associating the new hardware with the correct planning information while considering carrier grade security aspects. Currently, no standard is describing such process.

Phase 1 – DHCP

The DHCP use case represents the first phase of an Automated Commissioning Process and implies the device requesting for the necessary information for connecting to the IP network. This includes an own IP address for the device, but also the IP address of the SDN controller. Because there is no option code for identifying the IP address of an SDN controller in DHCP requests and responses defined in the DHCP standard, TR-545 is describing a method for using a generic option code for this purpose.

The DHCP use case tested, whether a well-known open source DHCP server (isc dhcpd) can exemplarily be used to implement this way of communicating the IP address of the SDN controller. Congiv handled the DHCP server configuration and SIAE Microelettronica supported the device side of the problem.

Phase 2 – Netconf Call Home

The Netconf Call Home use case represents the second phase of an Automated Commissioning Process and starts with the device announcing its presence to the SDN controller by establishing a TCP connection.

The Netconf Call Home use case tested, whether the Netconf client inside OpenDaylight and the Netconf server inside the device (based on openYuma platform can automatically establish a Netconf connection with each other.

Candidate Data Store

TR-545 contains several prescriptions for unifying the behavior of the southbound interface of the SDN controller. This includes appliance of functionalities, which are optional in the Netconf standard (IETF RFC 6241).

The Candidate Data Store use case analyzed, whether these prescriptions are supported by common Netconf servers and clients as e.g. in openYuma, ConfD and OpenDaylight.

Confirmed Commit

The Confirmed Commit is also an optional feature of the Netconf standard (IETF RFC6241 [8]). It defines the behavior of the Netconf server in cases, the connection to the Netconf client does not come up again after configuring some change, which cut this connection.

The Confirmed Commit Use Case tested, whether existing Netconf implementations are supporting this feature as described in TR-545.

5 Test Results

5.1 Microwave

Microwave Information for Planners

Device information, such as hardware, ports as well as internal connectivity, performance data and topology information was retrieved from the SDN controller and presented in Aktavara DNCM.

All devices, which were installed in the lab (Ceragon, Ericsson, Huawei, Intracom Telecom, NEC, Nokia, SIAE Microelettronica and ZTE), were visible.

The microwave link related status and measurement data was imported as well. As all devices were installed in a laboratory environment with no real air interfaces, the comparison between live network data and predicted link performance was unspectacular. Impairments had to be simulated. Required next step would be applying application and interface in a productive environment, for comparing predicted link performance with real live network data.

Preventive Maintenance

Communication and exchange of information according to TR-532 was possible with all device models. NEC and Nokia devices implemented TR-532 v1.0 directly at their native Netconf interfaces. Ceragon, Ericsson, Huawei, Intracom Telecom, SIAE Microelettronica and ZTE applied mediators to translate between TR-532 v1.1 and their proprietary management interfaces.
One out of eight vendors was identified to have a different interpretation of the concept of current performance values.
A single, specific status value (SINR), could be provided by only four out of eight device types, because of hardware limitations.
In general, it has been shown that interconnection using TR-532 is working and that the Preventive Maintenance use case can be implemented in a vendor-agnostic way. The presented PoC implementations are very close to commercial products already.

Microwave Network Performance

The Microwave Network Performance application, which had been provided by SIAE Microelettronica, successfully read all inventory, radio and Ethernet performance data of the devices provided by Ceragon, Ericsson, Huawei, Intracom Telecom, NEC, Nokia, SIAE Microelettronica and ZTE.

E.g. it could be verified that the bandwidth utilization values, which have been reported by the respective devices along the ring, were in accordance with the amount of traffic injected by the traffic generator (JDSU MTS5800).

The Microwave Network Performance application could also be used for suggesting adjustments of critical radio parameters to reduce impairment of the traffic injected in the chain.

Planning Data Reconciliation

Testing started with creating a dummy planning database by importing all the network data from the northbound interface of the OpenDaylight controller. A consistent configuration database describing all the devices could be generated by this. After establishing this reference, the application continuously compared all the current values of the TR-532 configuration parameters with the ones inside the dummy planning database and highlighted the deviations, which were cause by manual configuration on the devices.

It has been demonstrated that reconciliation of planning and live network data could be achieved with a single click, while the operator had just to decide about which side to align.

The Planning Data Reconciliation application, which had been provided by SIAE Microelettronica, successfully cooperated with devices provided by Ceragon, Ericsson, Huawei, Intracom Telecom, NEC, Nokia, SIAE Microelettronica and ZTE.

Smart Statistics Reporting

Smart Statistics Reporting application, which had been provided by SIAE Microelettronica, gave a quick and clear overview about the devices in the PoC network. It could draw pie charts for many equipment and radio interface related properties. The created reports could be exported into open file formats.

The application successfully retrieve the necessary data from equipment provided by Ceragon, Ericsson, Huawei, Intracom Telecom, NEC, Nokia, SIAE Microelettronica and ZTE.

Configuration, Performance, Alarm and Inventory Management

The values of attributes described in the corresponding technology specific interface definitions (TR-532, TR-541, TAPI) could be read from, respectively written to all microwave and optical devices in the PoC test installation.

Also, inventory data (TR-512) could be retrieved from all devices and displayed on the graphical user interface of OpenDaylight.

Performance data could be read from the all microwave devices and persistently stored in the database of the SDN Controller. Likewise, alarms coming from the microwave devices could be logged and listed on the graphical user interface.

Aforementioned applications have been operated on devices from Ceragon, Ericsson, Huawei, Intracom Telecom, NEC, Nokia, SIAE Microelettronica and ZTE.

Performance Assurance

The performance assurance and monitoring capabilities were provided by MYCOM OSI. The multi-vendor microwave infrastructure was visible in several dashboards' showing e.g. inventory information, health index KPIs or the worst performing elements. User were able to deepen the analysis by opening a second screen with information about e.g. errored seconds (ES), severely errored seconds (SES), unavailable seconds (UAS), and number of transmitted Bytes.

Purposely generated unavailable seconds on Ceragon and SIAE Microelettronica devices could be seen in these screens. Also the threshold crossing alarms, which were caused by interfering the connections, were raised in the application.

5.2 Ethernet PHY

Ethernet Port Information for Planners and Documentation Verification

Interface information down to the Ethernet PHY level was retrieved from the SDN controller and presented in Aktavara DNCM.

Ceragon and SIAE Microelettronica devices supported the Ethernet PHY model and could both be made visible in the Ethernet Port Information for Planners as well as the Documentation Verification view.

The comparison between live network data and planned port configuration could be successfully simulated.

5.3 Optical Transport

Basic Optical Transport Configuration

The following test cases had been successfully demonstrated with the Transponder Application from TechMahindra and devices from ADVA, Infinera and SM Optics:

- Switching off the laser from within the configuration window of the Transponder Application. Validating the result on the Traffic Generator by looking at the status of the flow turning red.

- Configuring the laser to be on again. Restarting the Traffic Generator to bring its status to green. After the refresh timer expired, the Transponder Application read the laser state and displayed it as on (small circle changed color from Grey to Green).
- Configuring an Rx minimum value, which is outside the allowed value range displayed in Transponder Application. Checking, whether this out of range value triggers an alarm, which is displayed in the problem list inside OpenDaylight.
- Retrieving and displaying performance data either automatically every 15 minutes or on-demand by clicking on a refresh button.
- Setting default values for Rx minimum and Rx maximum from within the Threshold Configuration window inside the Transponder Application. Verification of the configuration by element management system.

5.4 Device Management Interface Profiling

ONF TR-545 has been published in October 2018. No PoC participant, could implement any of the DMIP use cases on physical devices within the short time period before the PoC.

Automated Commissioning

SIAE Microelettronica supported the Automated Commissioning use case with their Mediator. Each time the Mediator Virtual Machine, which is hosting several tens or hundreds of Mediator instances, got connected to the IP network, it sent a DHCP request. After receiving its own IP address and the one of the SDN controller (option code 43 in accordance with TR-545), it started to open a TCP connection to the Netconf Call-Home TCP port at the Controller Virtual Machine.

It turned out that OpenDaylight is expecting the Netconf call-home on TCP port 6666 as a default, instead of port 4334, which would have been the one prescribed by IANA (Internet Assigned Numbers Authority). Because it has been found that the operator should be free to decide about the port to be used, it has been decided to amend TR-545 by a prescription about how the DHCP server shall communicate also the TCP port for the Netconf call-home.

Once the TCP connection was established the Netconf client inside OpenDaylight properly created a Netconf over SSH session to the Mediator and the Mediator showed up in the list of ‘unknown network elements’ inside the Connect application on top of the controller. As usual, for promoting the Mediator into the ‘required network elements' a manual action is needed.

For OpenDaylight to recognize the Mediator, the pairing of the Mediator’s name and its public key had to be pre-configured in a list of allowed devices. It had been found out that OpenDaylight uses the SSH host key (not to be confused with the user key) to identify the device and that this operation must be performed invoking the proper Restconf API at OpenDaylight NBI.

For those devices that do not appear in the list, any attempt to create an SSH session is aborted and therefore the connection to the Mediator is not possible.

Candidate Data Store

The Candidate Data Store use case has been supported by Ceragon, Ericsson and SIAE Microelettronica. In all three cases the Netconf server was implemented as a mediator. While implementations from Ceragon and SIAE Microelettronica were based on an open source platform, a licensed product had been applied by Ericsson.

During establishing the Netconf connection between SDN Controller (OpenDaylight) and mediator, the Netconf client inside OpenDaylight learns, whether the Netconf server inside the mediator supports the Candidate Data Store capability. Based on this information OpenDaylight addresses configuration requests automatically to the Candidate Data Store instead of the Running Data Store.

During the 4th PoC, it had been found that different mediator implementations reacted differently in case only some of the changes inside a single configuration request could be executed.

Now with the Candidate Data Store, all three implementations behaved exactly the same. Partly executed change requests got rolled-back until the original configuration was operational again and an error message was sent.

It could be shown that ONF TR-545 helps unifying the behavior of different interface implementations.

Confirmed Commit

The Confirmed Commit is an optional capability, which is extending the Candidate Data Store capability of the Netconf Protocol (RFC 6241).

It allows to increase operational stability by marking commit operations to be confirmed within a certain time period. If such confirmation is not received by the device until the configured timeout expires, the device changes all attributes back to their previous values. Applying this feature seems being particularly useful e.g. while making changes potentially affecting the SDN control plane to the remote site of a microwave link in a tree shaped network topology.

The Confirmed Commit has been implemented and demonstrated by Ceragon and SIAE Microelettronica on their OpenYuma based Mediator platforms.

The expected behavior and valuable contribution to operational stability (as soon as directly implemented on the device) could be demonstrated.

6 Conclusions

In this paper the main results of the 5th Wireless Transport PoC organized by ONF and held in Telefonica Germany premises in November 2018 have been described and commented.

The following scopes have been reached:

- Demonstration and validation of the latest version of the Microwave Model (according to TR-532 v1.1). The current model supports the required attributes and functions. The maturity of the model was proved by the adoption in wide range of advanced applications enabling among others configuration of devices, preventive maintenance, performance monitoring and analysis, planning and maintenance of microwave links.
- Demonstration of the implementation of Photonic model for vendor-agnostic management of optical transponders in partially disaggregated networks. The implementation is based on ONF Core information model (v 1.2) and TAPI technology specific extension (v 2.1). The supporting application enables configuration of the main parameters of optical transponders via graphical user interface.
- Demonstration of the implementation of Ethernet PHY model (according to TR-541 v0.3). The aforementioned microwave model was extended for supporting Ethernet over wire functionality.
- Demonstration of the first implementations of the Device Management Interface Profile (DMIP) (according to TR-545 v1.0). Even well-defined models are not sufficient for proper management of a multi-vendor network. Thus, the Device Management Interface Profile unifies the behavior on the southbound interface of the network domain controller to ensure unambiguous states and responses of different vendors' devices. As the initial step, the unified behavior was successfully demonstrated on a set of NETCONF capabilities such as Candidate Configuration and Confirm Commit (both IETF RFC6241). In addition, the first two phases of Automated (Zero touch) Commissioning were successfully demonstrated by implementing a unified communication with the DHCP server and the Netconf Call home function.

A record in participation with eight microwave vendors (Ceragon, Ericsson, Huawei, Intracom Telecom, NEC, NOKIA, SIAE Microelettronica, ZTE), three optical vendors (ADVA, Infinera, SM Optics) and eight software application providers (Enghouse Networks, highstreet technologies, Indra Sistemas, Mycom OSI, Mentopolis, SIAE Microelettronica, Tech Mahindra, Wipro) was reached in this PoC, supervised by three sponsoring operators (AT&T, Deutsche Telekom, Telefonica). In summary, 21 companies and more than 56 representatives were on site to perform tests and present the results. The wide participations is very promising and important for adoption of the open-source models and open interfaces across whole industry.

7 Contributions

7.1 ADVA – Optical Transponder, Optical Line System (OLS) and Interface Simulator

Context

ADVA Optical Networking developed, delivered and demonstrated SDN-controlled open disaggregated optical transport during this PoC. The optical use cases were based on an Open Line System (OLS), which served as disaggregated network infrastructure for open terminals also from other technology partners. These open terminals were used to deliver 1Gbit/s and 10Gbit/s services and they were connected for example to mobile backhaul switches.

The provided open terminals can be configured via Netconf interfaces, which use the ONF core model as a reference for describing and controlling the network element. These API were connected to the OpenDaylight SDN controller and its graphical user interface.

The operations and capabilities demonstrated in this PoC show the capability to control and monitor the laser operations for open optical terminals. The configuration of Open Line Systems using the ONF Transport API has been demonstrated in numerous research, commercial proof of concepts and interop activities, and the use of the same Yang model specifications in the Photonic Media conditional packages means that a single model can be used to define the configurations associated with the open terminals and the OLS.

The partially disaggregated (OLS + open terminal) control of optical systems provides a good trade-off for network operators, enabling them to easily integrate third-party transponders in their systems without a significant overhead in managing the optical line systems, and the use of common standards-based representation for both the open-terminal and the OLS further simplifies the software integration in these systems.

Value and flexibility of disaggregated optical network architectures for controlling optical devices in multi-vendor environments and seamless integration into open source SDN controllers has been demonstrated.

Implementation - Disaggregated Optical Transponder

The ONF core model is an extensible Yang model that allows describing a network element by a collection of Logical Termination Points (LTPs). LTPs are associated with a network layer or a layer protocol in the ONF core model terminology, and a hierarchical association of LTPs can be used to describe the capabilities and current operations of ports on a network element. Interconnections between LTPs are described using Forwarding Constructs (FCs) which are used to describe the switching behavior inside a network element. Finally, LTPs at specific layers can be associated with conditional packages that are used to describe the layer-specific capabilities. The logical association of these components is presented in Figure 7

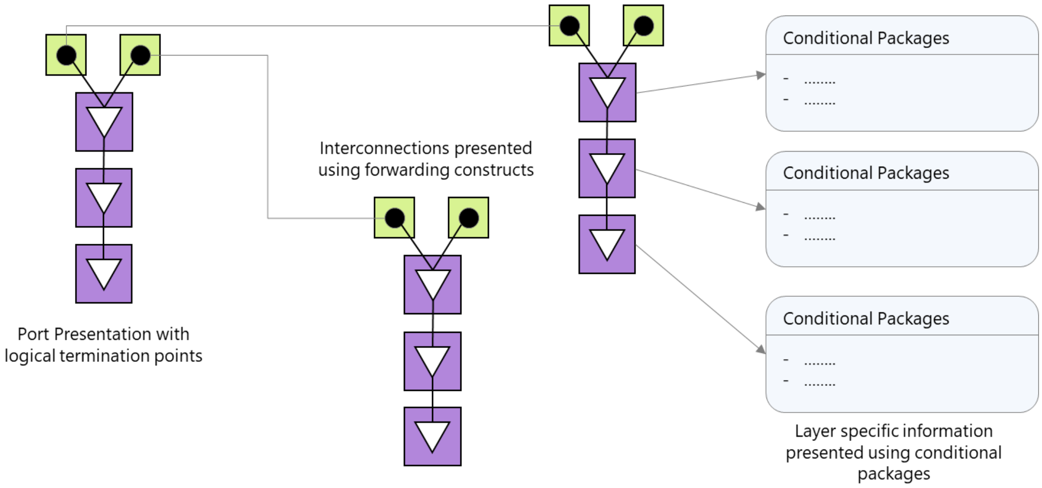


Figure 7: Logical termination points (LTPs) and Forwarding Constructs (FCs)

The use of conditional packages facilitates incremental development of standards at specific technology layers, while maintaining the base network element description. This is reducing the overhead for both hardware vendors and third-party application-developers that consume the ONF core model representation.

Already in past, extensions to the ONF core model have been successfully employed to demonstrate Ethernet, OTN Switching and Precision Time Protocol (PTP) configurations. During this PoC, it was extended to demonstrate Optical Media configuration and monitoring.

The ONF Core model also supports an equipment model that can be used to expose and configure the hardware inventory in a network element.

During the PoC, open terminals were discovered and managed over Netconf by an OpenDaylight (ODL) controller (as shown in Figure 8). A third-party application communicated with the SDN controller using the OpenDaylight northbound Restconf interface. The application was able to present the association of the physical inventory (lasers / SFPs) of the open terminals to the logical Optical Media termination in the ONF core model representation, and the control of the optical media layer using the Photonic Media conditional packages.

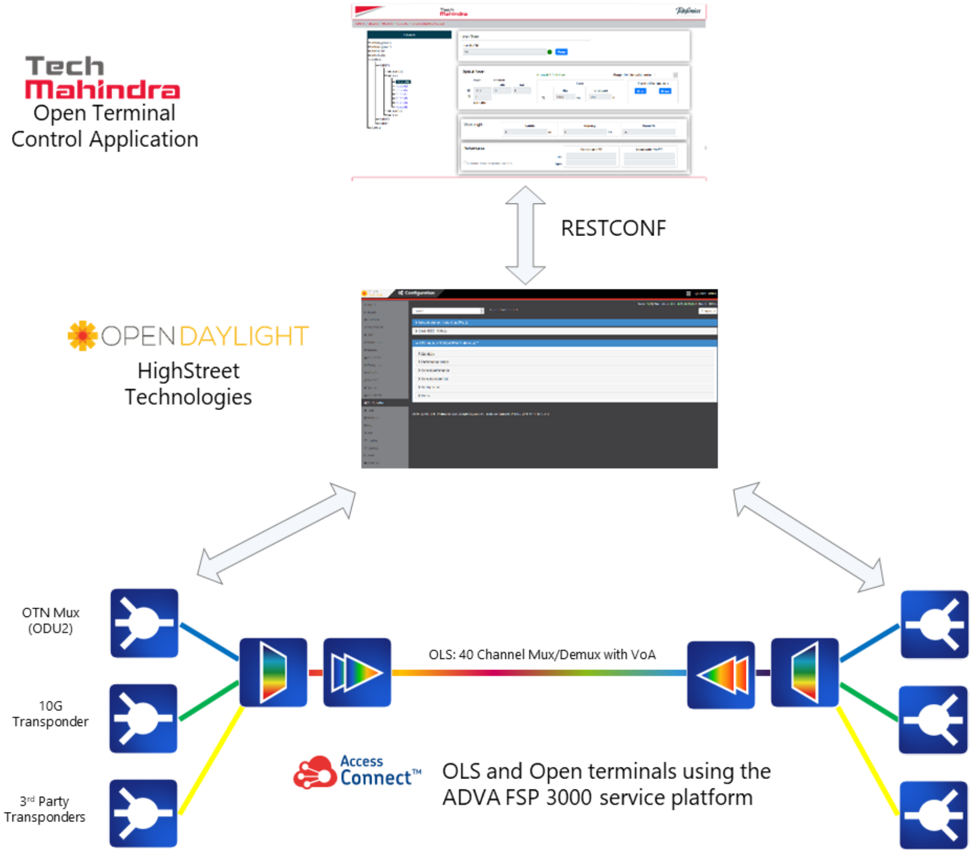


Figure 8: Architecture of the Optical use cases

The LTPs in the ONF core model can reference specific equipment instances in the same network element, making it feasible to reference, for example, an SFP with a photonic media LTP. The correlation with the equipment hierarchy provides a familiar handle for operators to configure the network element over the logical ONF core model based network element representation. The photonic media conditional package associated with the LTPs is based on the ONF T-API photonic media model, and provides configuration and monitoring capabilities for optical transmitters (including lasers) used in the open terminals. In the proof-of concept this model was primarily used to demonstrate monitoring and configuration capabilities.

The monitoring capabilities demonstrated include the accurate reporting of received (RX) and transmitted (TX) power at the laser. The values reported over the ONF core model by the open terminals was verified against the values as reported in device's element management system, and a Variable optical Attenuator (VoA) included in the OLS provided by ADVA was used to vary the attenuation, and confirm that the expected change in values was reported over the ONF core model implementations.

Control capabilities demonstrated in the PoC included the ability to configure the laser status in order to switch the laser on/off. The current laser status was also reported over the NBI, and test equipment connected to the open terminals were used to verify if the data transmission was in accordance with the reported laser status.

Receive power threshold configuration (high/low receive power thresholds) was supported over the photonic media conditional package to support soft-failure indication, and in case the received power was outside the specified threshold, associated problems were reported as alarms over the ONF core model.

Implementation – Default Value Mediator

Given the multiple integrations involved in the PoC activity, ADVA also delivered a so-called Default-Value-Mediator (DVM) to simulate Netconf devices supporting the Photonic Media application. The DVM was key to start end-to-end software integration and testing before the optical hardware was available in the PoC lab. The DVM could be controlled using a REST interface, and could instantiate any number of simulated (read-only) Netconf devices, with the ability to choose from three distinct simulated hardware configurations. For each configuration, the simulator could report potential interface configuration capabilities, current status information as well as a limited record of current and historical PM entries.

Resources

Yang Model and Default Value Mediator: https://github.com/openBackhaul/opticalTransponderInterface

ADVA Optical Networking web site: www.advaoptical.com

Contact partners:

- General: Stephan Neidlinger (sneidlinger@advaoptical.com)
- SDN API, ONF core model: Mohit Chamania (MChamania@advaoptical.com)
- Optical system: Jan Müsebeck (jmuesebeck@advaoptical.com)

7.2 Aktavara DNCM

Context

Enghouse Networks implemented its commercial product Aktavara DNCM (Dynamic Network Configuration Manager) to serve as an application within the Transport SDN Trial. Telefonica Germany is already using this platform for operational network planning across multiple vendors and technologies.

The Dynamic Network Configuration Manager compares the network planning information provided by x:akta with live network information from the SDN controller to identify configuration discrepancies, to analyze Ethernet performance data and to identify the network topology. The data is then also provided to the other applications in the project.

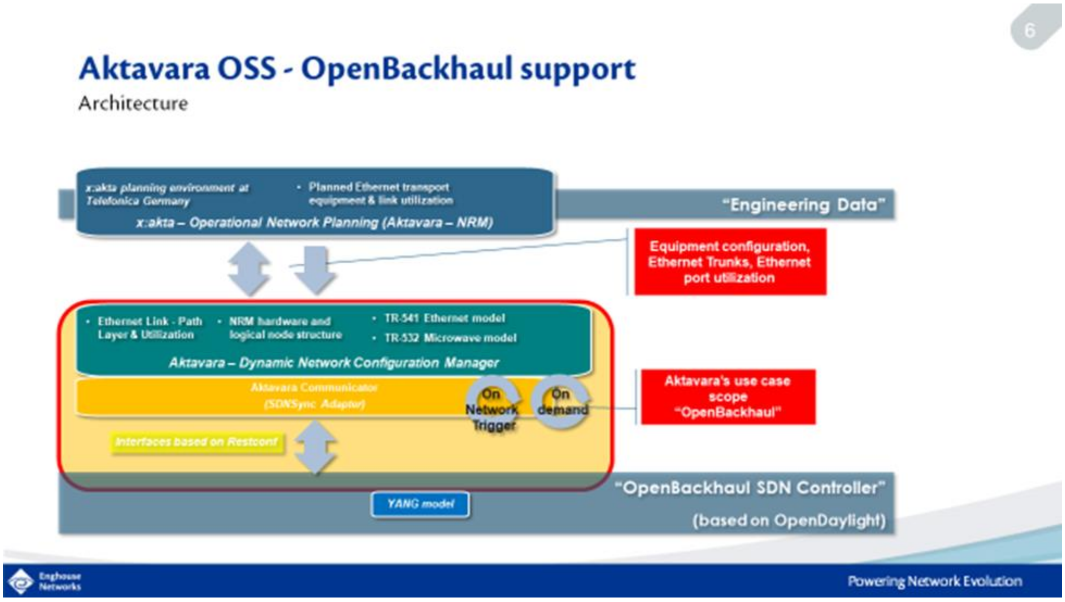


Figure 9: Solution Architecture

All microwave links announced by the controller are uploaded and processed for comparison. The microwave unit's configuration is automatically analyzed and displayed in a network diagram. The Network Diagram is a built-in interactive workspace in the Dynamic Network Configuration Manager.

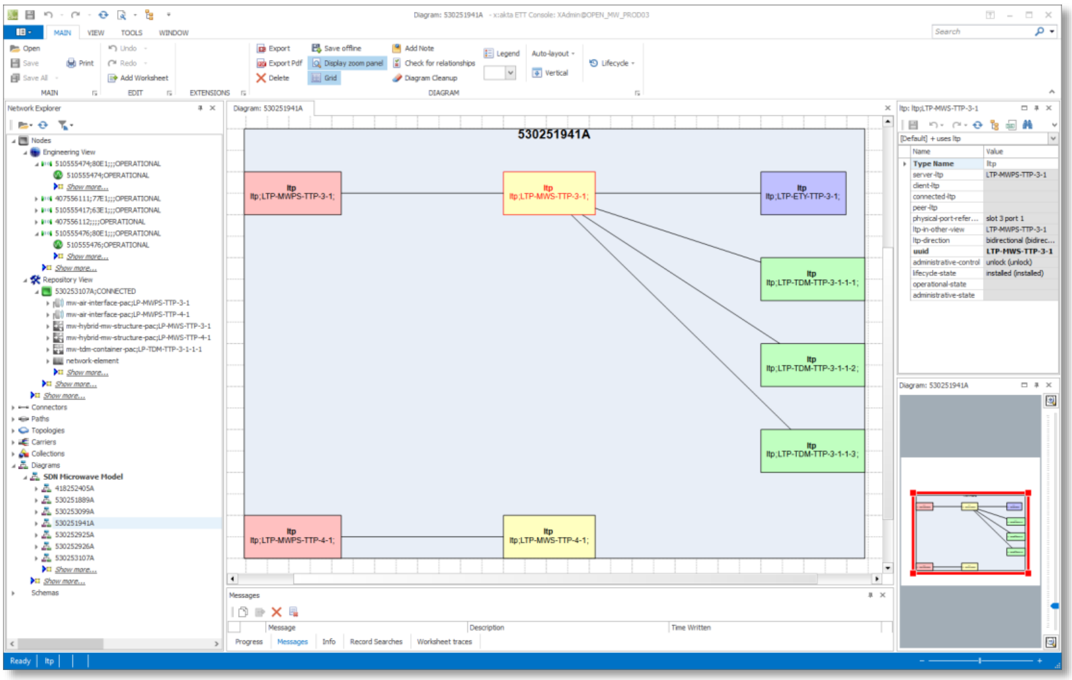


Figure 10: Auto-generated Display of Controller Unit Structure

The following functions have been covered within the 5th PoC:

- Network auto-discovery and automated operational status update for the planning layer. The update might be started on user demand or automatically after a change in the network. The Dynamic Network Configuration Manager reads the microwave configuration into its repository layer, identifies the planned microwave links and updates their provisioning status.
- Ethernet utilization reporting the most urgent upgrades. The live network Ethernet performance data utilization within a time period of 24h for the performance counters is analyzed, processed and compared to the planned Ethernet configuration. The function updates the microwave bearer information with the value in % and generates an interactive report with the most urgent updates network wide.

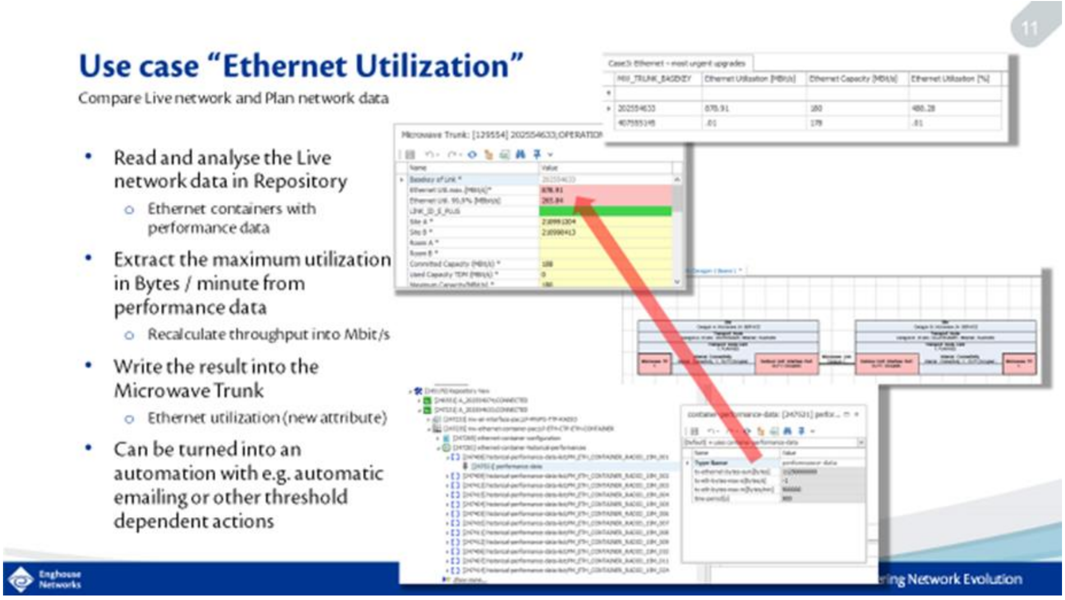


Figure 11: Ethernet Bandwidth Comparison Live vs. Plan

- The Ethernet PHY model is implemented to compare the live network configuration for the Ethernet ports with the planned configuration. Port parameters, e.g. wavelength, port status, remote port are retrieved and compared.

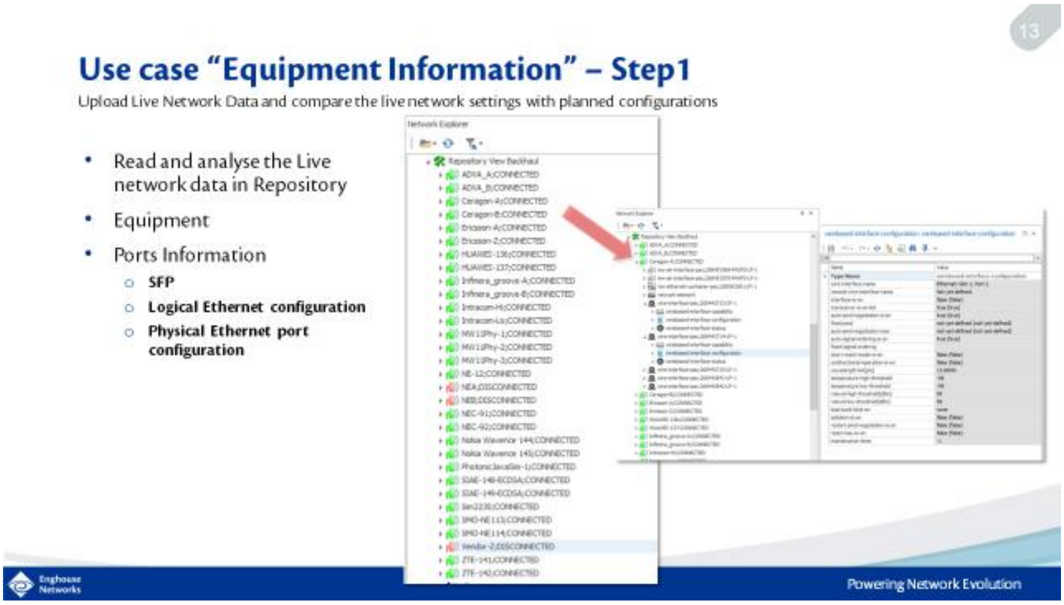


Figure 12: Equipment Information

- For the comparison process the Live Network View adapter presents the live network information directly linked with the planned network stage.

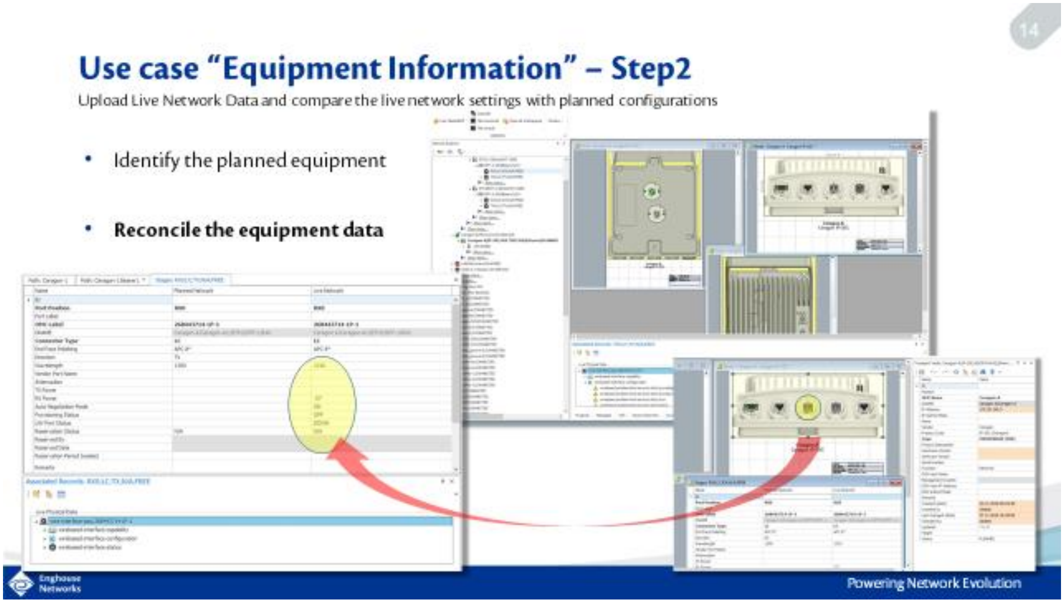


Figure 13: Ethernet PHY Information presented in the Live Network View

Implementation

The Dynamic Network Configuration Manager - Repository layer is automatically configured with the microwave data model according to TR-532 V1.1 as well as the Ethernet PHY model TR-541. A YANG Interpreter is used for that. The Interpreter creates the CI (Configuration Item) model by using the built-in design tools. The Aktavara Communicator establishes the connection to the SDN controller running the Restconf protocol. The YANG Interpreter is highly configurable, which means that changes (new releases) to the controller or mediators can be supported by DNCM within a single day.

The auto-discovery functionality in the Dynamic Network Configuration Manager identifies network changes in the controller instantly and updates the data into the Repository model. The data model covers:

- Equipment configuration
- Ethernet & TDM parameter settings
- Ethernet performance data
- Microwave link capacity reconciliation

Resources

- Enghouse Networks (www.enghouse.com) provides leading-edge COTS products to telecom service providers, communications network operators, utilities and the oil and gas industry. Its portfolios include Operations Support Systems (OSS), Business Support Systems (BSS), Mobile Value Added Services (VAS) solutions as well as process engineering and data conversion services.
- Aktavara AB is a wholly-owned Enghouse subsidiary based in Sweden and is responsible for some of the new key modules in the comprehensive Enghouse OSS suite, such as NRM, DNCM and Site Manager.

7.3 Ceragon – Microwave Device, Mediator and Interface Simulator

Microwave Device and Mediator

The Mediator, which Ceragon implemented for participating at the 5th PoC, supported the Microwave Information Model (TR-532 v1.1) and the Ethernet PHY Information Model (TR-541 v0.3).

With the help of this Mediator, the FibeAir IP20 microwave devices that were brought to the PoC site were able to be connected to the SDN controller and to be used by the SDN microwave applications implemented for the PoC.

The Mediator is designed to run on a single Linux virtual machine and serves multiple microwave network elements, while being lightweight and scalable. It is based on a combination of C and Python. The C implementation extends the openYuma open source framework that provides a Netconf server

More information can be found here: http://blog.ceragon.com/blog/powering-your-5g-network-addressing-challenges

Ethernet PHY UML, Class Diagram, GenDoc Export, YANG files and Wireless Transport Emulator can also be found on https://github.com/openBackhaul/wireInterface.

Default Value Mediator

The Default Value Mediator was developed for enabling the testing of SDN applications in preparation for the 5th PoC. It is a software tool that provides a Netconf interface exposing the Microwave Information Model (TR-532 v1.1) and the Ethernet PHY Information Model (TR-541 v0.3).

Its purpose is eliminating the need of SDN application developers to own physical network elements in order to test their code or to wait for the hardware vendors to finish their interface implementations.

The Default Value Mediator implemented for the 5th PoC is based on the Wireless Transport Emulator (WTE) and was called WTE PoC5. The WTE has already been used in the 4th PoC, for the exact same purpose: exposing the information models used in the PoC, such that SDN application developers could test their code prior to the PoC.

The WTE Poc5 extends the previous version by implementing the latest release of the microwave model (TR-532) and the not yet published Ethernet PHY model (TR-541). The implementation is still based on openYuma open source framework that provides a Netconf server on a Python core, which is in charge of the whole emulation environment.

For every simulated network element, a Docker container is created, having the Netconf server running inside. Each such instance can then be mounted in the OpenDaylight SDN controller and default values are returned for the attributes inside the information models.

In contrast with the previous version, the simulated devices were capable of being configured to have the Netconf Candidate Data Store available. With this new capability, the Candidate Data Store and the Confirmed Commit Use Cases could be implemented according to DMIP TR-545 and demonstrated with the help of the simulator.

WTE PoC5 is part of the MeLaCon project. Details about installation and usage guidelines can be found in GitHub: https://github.com/Melacon/WTE_PoC5

Microwave UML, Class Diagram, GenDoc Export, YANG files and Wireless Transport Emulator can also be found on https://github.com/openBackhaul/airInterface.

7.4 Ericsson – Microwave Device and Mediator

Within the 5th PoC, Ericsson successfully demonstrated the implementation of the Microwave Information Model (TR-532 v1.1) and the Candidate Data Store use case based on the Device Management Interface Profile (DMIP; TR-545 v1.0).

The Microwave Information Model TR-532 v1.1 extends its predecessor TR-532 v1.0 with additional attributes identified within previous POCs and supplier implementations.

During former PoCs, different interface implementations reacted differently on just partly executable configuration changes. In order to improve this situation, the DMIP TR-545 is prescribing the interface implementations to follow the concept of Candidate Data Stores according to IETF RFC6241. Within this concept the collection of intended configuration changes is send to the microwave node or to the associated Mediator, in case the device itself doesn't natively support the Netconf protocol. The receiver of the selection of intended changes will just store these and will wait for receiving also a Commit command before processing a validation and the configuration of all stored changes in a single step. In case not all the intended changes could be successfully executed, the device or Mediator reverts back all the already configured changes from the same batch and sends an error message back to the controller. Only if all changes could successfully be configured, an ok message will be answered to the controller. With this mechanism the device will always be in a well-defined status, which is of particular importance in cases the transport of the management interface itself is affected by the changes, too.

7.5 highstreet technologies – ONAP SDN-R Applications

highstreet technologies delivered and operated the open source SDN controller in the 5th PoC. The provided SDN controller bases on the ONAP Common-Controller-SDK SDN-C project and comes along with a couple of ONAP SDN-R project applications and functions. At Telefonica Lab, this SDN controller run in a single Ubuntu 18.04 virtual machine.

A clustered version of the SDN-R was installed at DT-Cloud and another 3-node clusters in accordance to the ONAP SDN-C deployment requirements and rules of ONAP version Casablanca were installed in the North-American ONAP Community lab at Winlab Rutgers and the European ONAP Community lab at Open SDN & NFV Lab (OSNL) in Berlin.

All 10 SDN controller nodes (3 x 3-node cluster + 1x single node) are OpenDaylight SDN controller version Oxygen SR3. The software extensions are publicly available on the following GitHubs:

- https://github.com/OpenNetworkingFoundation/CENTENNIAL/tree/master/code-Oxygen-SR3
- https://github.com/onap-oof-pci-poc/
- https://gerrit.onap.org/r/#/admin/projects/?filter=ccsdk%2F
- https://gerrit.onap.org/r/#/admin/projects/?filter=sdnc

During the PoC, it has been demonstrated that several network function and network technologies, which had been provided by different hardware and software providers, could be controlled by the same SDN controller. This also demonstrated that maintaining separate SDN controller instances just for distinguishing vendors and technologies is mainly for organizational reasons, but no technological necessity.

In addition, also different releases of the microwave model (TR-532 v1.0 and v1.1) have been supported by the same SDN controller. The following list shows all models and releases used in the PoC setup:

- Core Information Model TR-512 v1.2
- Microwave Information Model TR-532 v1.0
- Microwave Information Model TR-532 v1.1 (incl. Ethernet PHY Information Model TR-541 v0.3)
- Photonic Media Model using TAPI Object Classes from TAPI 2.1 (RC2)

Compared to previous PoCs the ONF Centennial applications were enhanced with the following feature functionality:

Connection

The Connection application has been amended by displaying and supervising the connection state to Netconf servers supporting the PhotonicMedia Model.

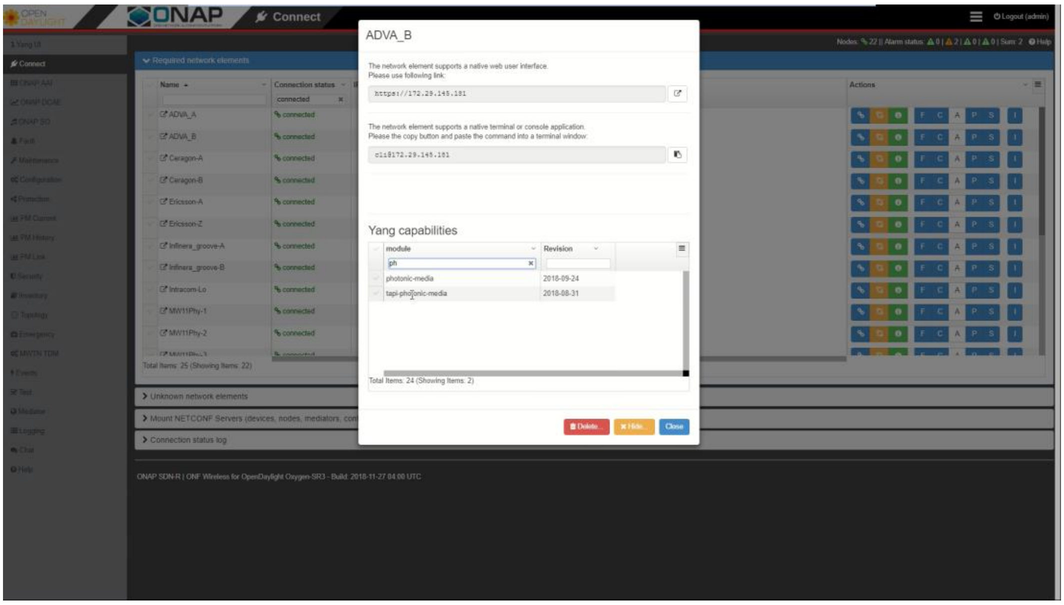


Figure 14: Support of the PhotonicMedia Model

MaintenanceMode

The MaintenanceMode application supports the suppression of alarm notifications at the northbound of the SDN controller towards [ONAP DCAE](). The corresponding user interface offers the suppression of alarms within a user defined interval, starting somewhere in the future, within the next hour or the next eight hours.

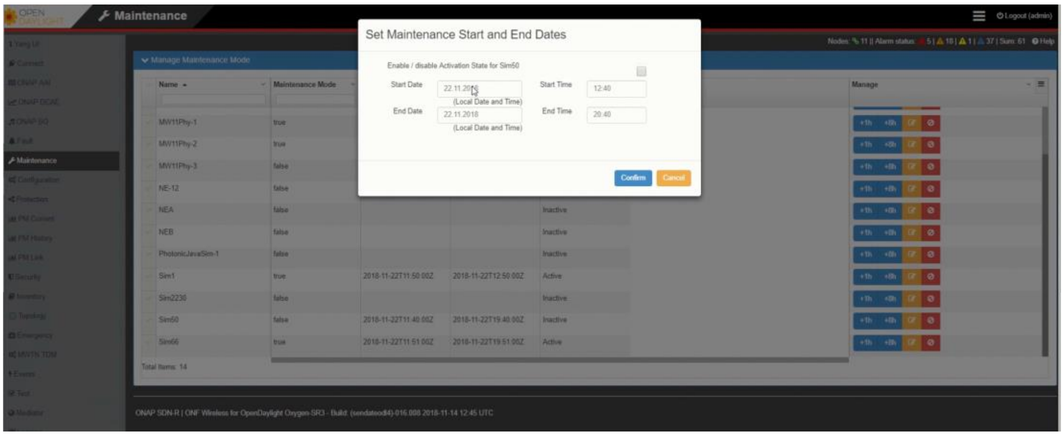


Figure 15: UI of the Maintenance Mode app

Inventory

The Inventory application supports now microwave, millimeter-wave, distributed antenna system and optical devices. The Netconf server implementing the equipment model as described in Core Model v1.2. is key for this application.

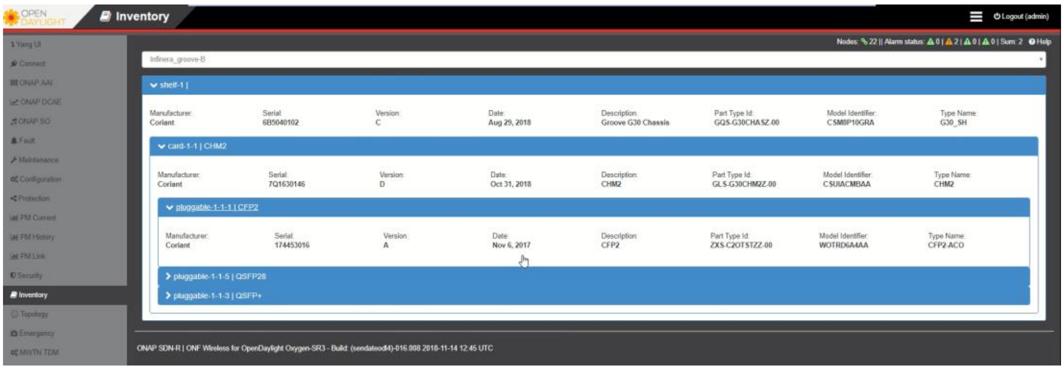


Figure 16: Inventory support Photonic Media Devices.

Configuration

The Configuration application supports all relevant models and their object classes.

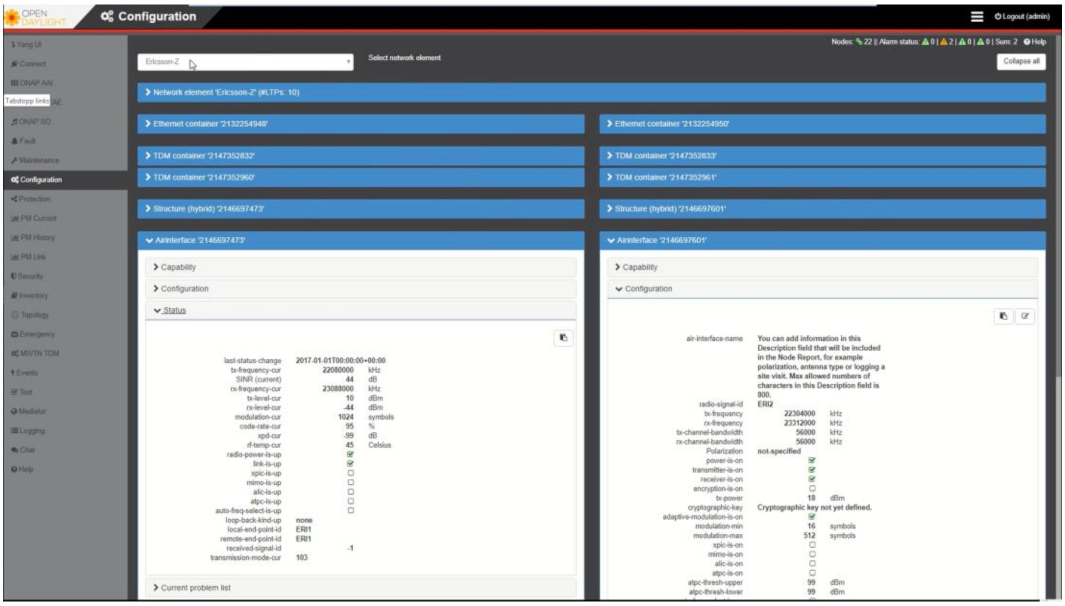


Figure 17: Microwave 1.1 Model in Configuration

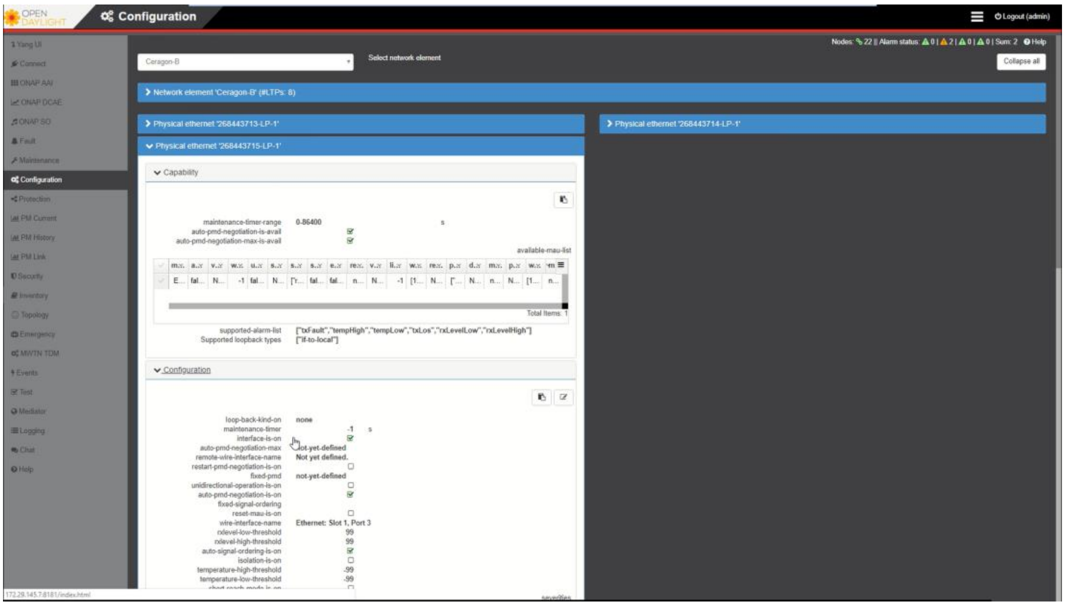


Figure 18: Ethernet PHY Model in Configuration

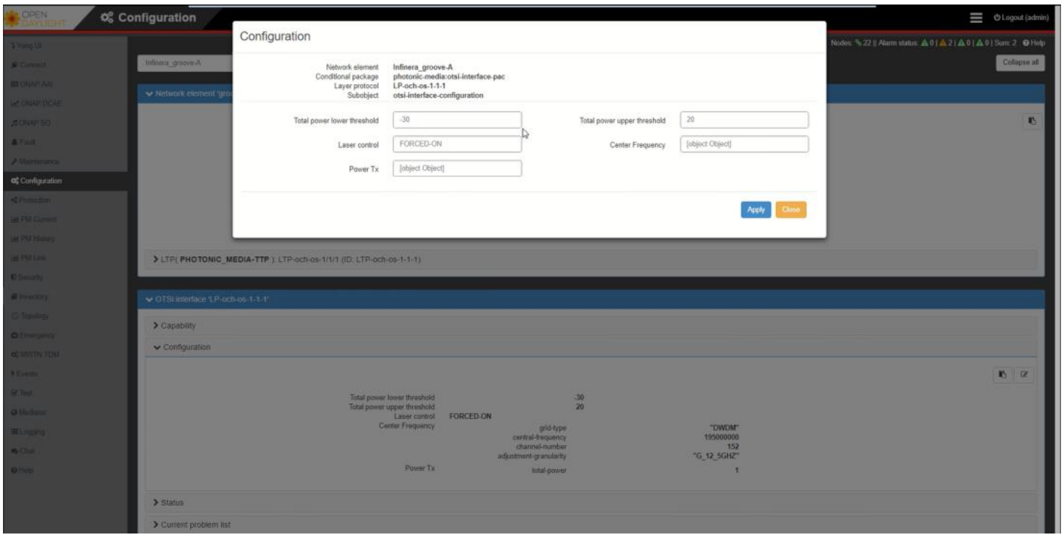


Figure 19: Photonic Model in Configuration

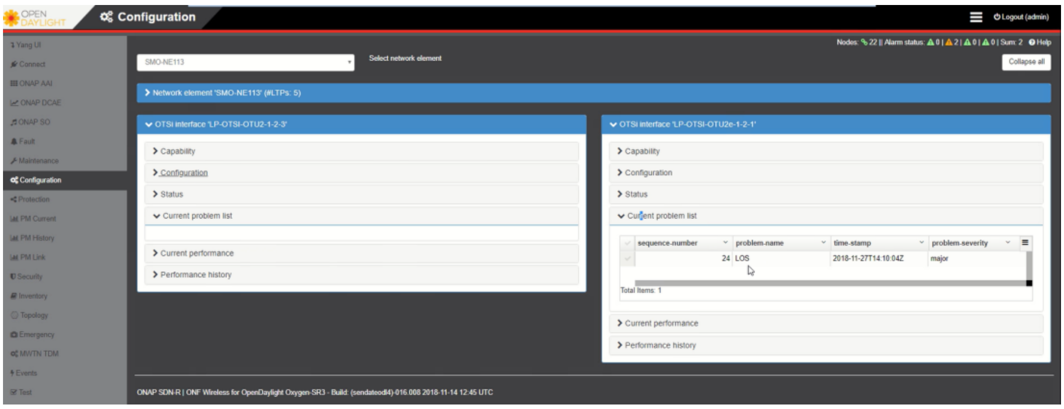


Figure 20: Current Problems of Photonic Media in Configuration

Fault

The Fault application within the DeviceManager supervises the connection between the SDN controller (cluster) and the expected Netconf servers. In case a connection is lost, the DeviceManager sends an event to ONAP DCAE.

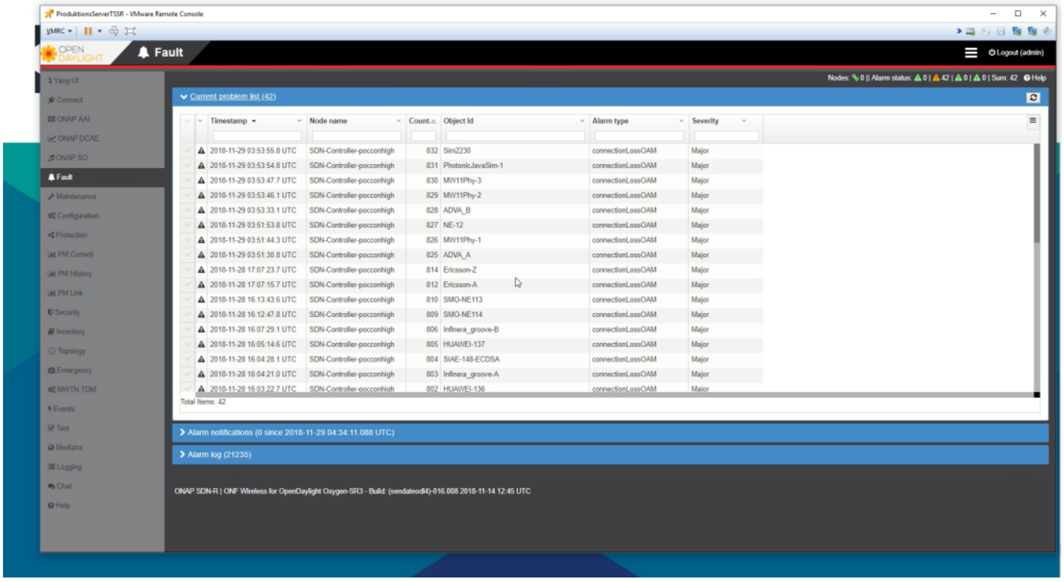


Figure 21: Netconf Server connection supervision an alarming

7.6 Huawei – Mediator for RTN905

The Huawei SDN Mediator is used for adapting own RTN microwave equipment to third party SDN controllers. It provides a Netconf interface with ONF standard information models (Core TR-512 and Microwave TR-532) towards the SDN controller and a legacy management interface towards the RTN equipment. The Mediator performs the translation between the two interfaces.

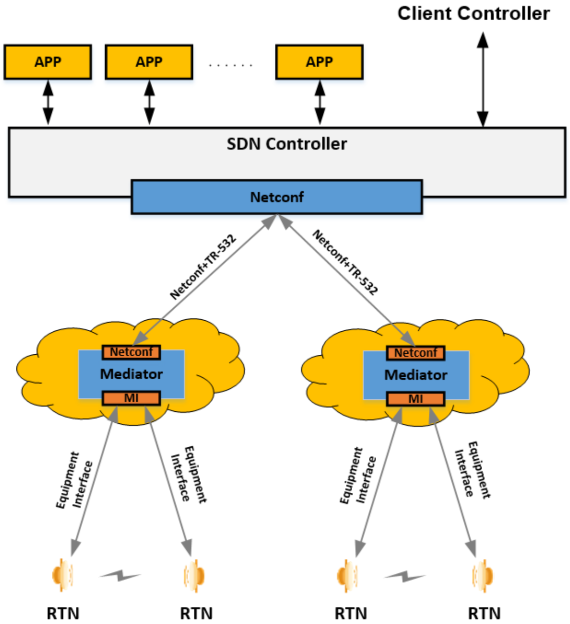


Figure 22: Huawei SDN Architecture Overview

During the 5th ONF PoC, the Mediator worked on an Ubuntu operation system, which was deployed on the given server. The applications could query and modify the RTN parameters through the third party SDN controller and the mediator.

To save IP addresses and virtual machine resource, a single Mediator virtual machine contains multiple Mediator instances. A TCP/IP connection had to be configured between a Mediator instance and a network element. All the Mediator instances shared a single IP address, and were distinguished by TCP/IP port.

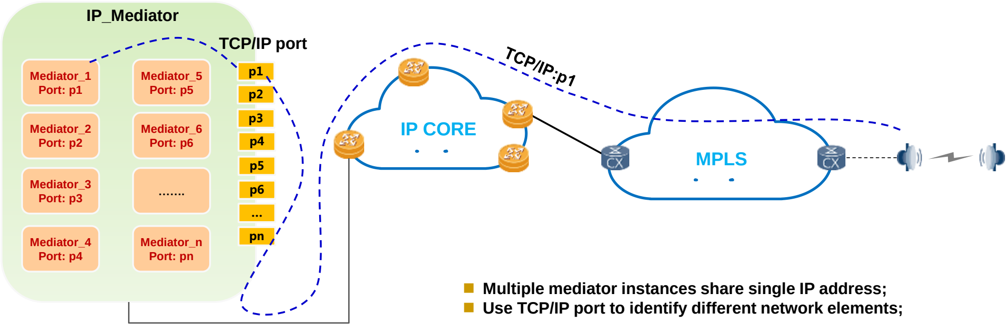


Figure 23: Huawei SDN Control Plane Connection

As a result, a single Mediator virtual machine can manage multiple network elements. There is a Mediator Instance Manager for initiation and maintenance of the Mediator instances.

Each Mediator instance includes three components:

- The Netconf component is used for communication with the SDN controller. It processes standard Netconf protocol and its interface definitions follow the ONF Microwave Information Model defined by TR-512 and TR-532.
- The Qx interface component is used for communication with RTN equipment.
- The translation component translates between Netconf and Qx protocol and performs the mapping between ONF standard and legacy proprietary data model.

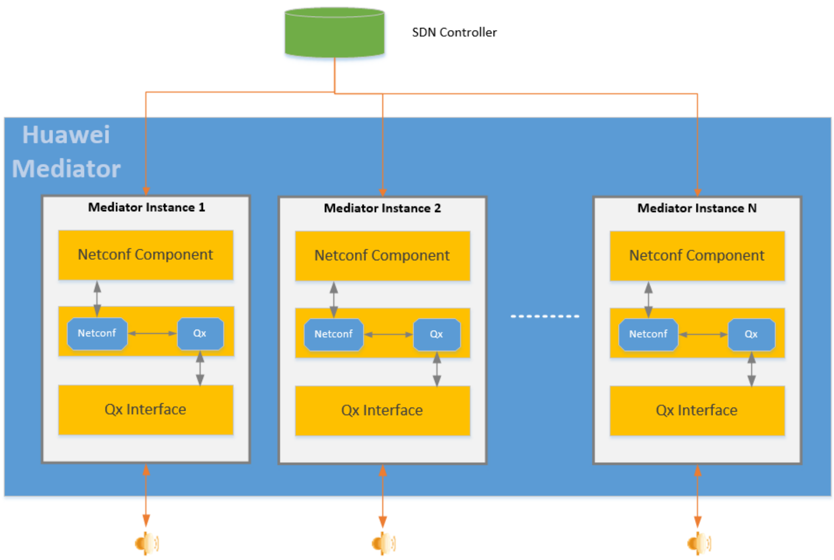


Figure 24: Huawei Mediator Virtual Machine Overview

In principle, each Mediator instance is used specifically for one RTN device. In this PoC, one RTN905/RTN950 link is managed by 2 Mediator instances running in a single virtual machine.

The Mediator used for managing RTN905/RTN950 in this PoC has been working with the software component named as OptiX RTN905 V100R007C10SPC500 and OptiX RTN950 V100R007C10SPC500.

7.7 Infinera- Optical Transponder

Context

Given the prevalence of standard and proprietary Yang models in the industry, optical disaggregated transponder from the family Groove G30 was designed so that supporting additional models is a simple task, without the need to change the actual software.

The device has its own native (or internal) model, so whenever another model needs to be supported, mapping rules need to be defined, translating the external model into the internal model, and vice-versa.

In the case of this PoC, the additional external model to be supported was the ONF Core Information Model, as well as the photonic-media (based on TAPI). These models would need to be partially supported by the device, via mapping rules.

Implementation

The actual implementation was composed of two parts:

- Model transformation engine
- Model mapping rules themselves

The transformation engine is software that is part of the device base functionality, so no extra implementation was needed for this particular PoC.

The mapping rules were written as Yang files, using proprietary extensions, which create rules from XPath based expressions. These expressions are then handled in runtime, and any Netconf commands using the ONF models are automatically converted into internal/proprietary model representation. The entire application software is agnostic from the external model that was used to trigger a request.

An OpenDaylight controller was used between the device and an Optical Transponder Application; this version had native knowledge of ONF models and allows transparent access to the device via its northbound Restconf API (endpoints generated via mounted device Yang models).

The Yang models used by the application would auto-discover the device hierarchy and interfaces according with the example given in the Figure 25.

Discover mechanism would be:

- Get top-level-equipment list;
- Use that to get equipment instances;
- Follow the hierarchy of the equipment instances, namely the occupying-fru that would point to other equipment instances;
- Repeat until the entire hierarchy was discovered;
- For equipment entries that would have a connector, an ltp instance would exist, where the physical-port-reference would reference the equipment itself;
- The ltp would contain a lp with name PHOTONIC_MEDIA, which would be augmented by an otsi-interface-pac with the same name;
- The otsi-interface-pac would then contain multiple specialized fields, both configuration and state, which the application could use to control the traffic.

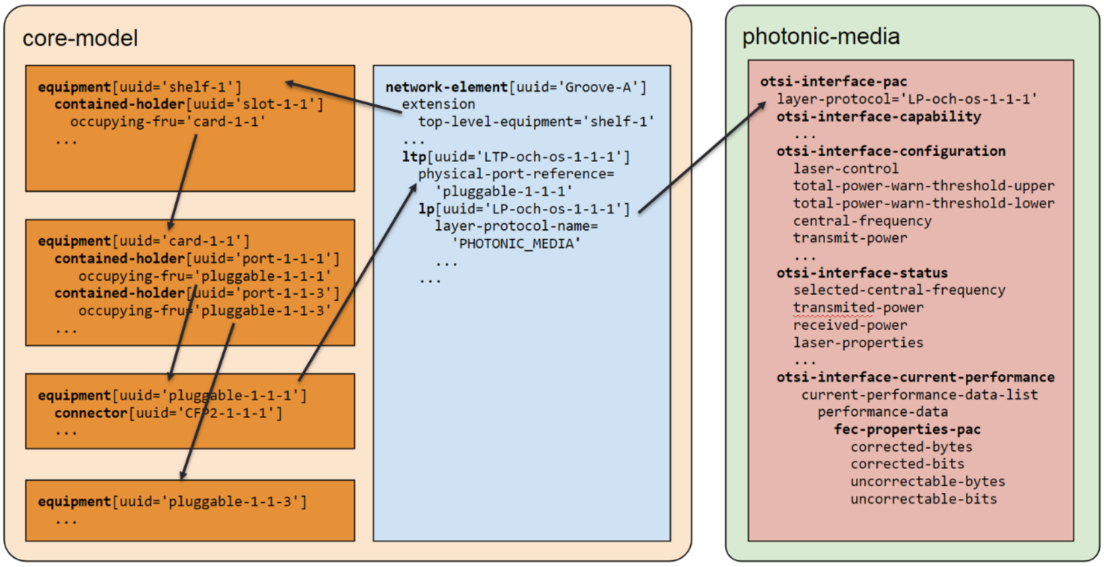


Figure 25

References

OpenDaylight release (OpenDaylight Oxygen)

https://github.com/OpenNetworkingFoundation/CENTENNIAL/tree/master/code-Oxygen-SR3

Core Information model (ONF TR-512)

- Specification https://www.opennetworking.org/images/stories/downloads/sdn-resources/technical-reports/ONF-CIM_Core_Model_base_document_1.1.pdf
- Yang Model https://github.com/OpenNetworkingFoundation/CENTENNIAL/blob/master/models/Yang/core-model.Yang

TAPI

- Homepage https://wiki.opennetworking.org/display/OTCC/TAPI
- Yang Model https://github.com/OpenNetworkingFoundation/TAPI

Photonic-media.Yang

- Yang Model https://github.com/openBackhaul/opticalTransponderInterface

7.8 Indra Sistemas – Predictive Maintenance Tool

Context

Microwave links continuously report air interface information. This data is usually discarded and only some basic statistics are stored for basic monitoring. The SDN controller can provide easy access to almost real time KPIs of the air interface, as modelled in ONF TR-532. It is possible to collect these KPIs with a granularity of minutes and store them in a data lake for further analysis.

The deepMW application from Indra/Minsait can use this information from the data lake for predictive maintenance. Based on patterns found in the microwave link's KPIs, it is possible to infer future network problems and correlate this information with external data like meteorological information to enrich the original data.

Implementation

Instead of using the SDN controller's northbound interface, it is possible to directly retrieve the live network data from the OSGi container OpenDaylight runs on. Subscribing to notifications to be aware of topology changes and periodically fetching performance data from within the OSGi container minimizes performance hit on both the controller and the deepMW application. The deepMW application takes advantage of this possibility and includes a module that runs inside the OpenDaylight Karaf container, processing and filtering the needed information for the deepMW application engine.

The deepMW data collector has been implemented as a set of OSGi bundles in Java language. It follows the OpenDaylight developing guide lines. All the bundles and their dependencies are packaged in a kar file (KAraf aRchive) that can be easily installed on the Karaf container that runs the OpenDaylight controller by using console commands.

The following bundles are part of the deepMW collector module:

- API. Defines the supported RPC operations of the module
- Impl. Implementation of the module functionality
- Config. Needed configuration files
- Module configuration and Main module configuration
- Database configuration. Triggers the creation of a data source service by pax-jdbc-config

The deepMW data collector requires the following services from the OSGi container:

- dataBroker. Used to subscribe to notifications on the OpenDaylight MD-SAL broker
- mountService. For the creation of mount points and query configuration, performance and alarm information from the devices
- dataSource. Created by pax-jdbc-config by detection of the database configuration file deployed by the deepMW database configuration bundle

When the deepMW data collector is initialized:

- It subscribes itself with the dataBroker to receive notifications about Netconf topology changes
- It queries the currently operational Netconf topology and writes a suitable representation to the deepMW database

When the deepMW data collector is running:

- It reacts on changes in the Netconf topology and writes them to the database. It discards those devices not providing the expected microwave-model capability
- It periodically fetches performance and alarm information of the nodes in its topology representation and stores it to the database

This information will be completed with additional external sources that will be used to improve prediction performance. One of these data sources will be meteorological data, this information will be gathered from official weather information providers and saved for posterior analysis.

Once there is enough data in the database it is processed with different data mining tools like Spark and some R and Python based libraries. Recognition models are created with these tools to predict network impairment situations before they would be detected by typical alarm mechanisms.

This approach can be further extended to include all the information required from the TR-532 models and dump it to a more generic data lake. This could include not only performance information from the microwave links (and also other link technologies: optical, Ethernet,…), but also configuration data that can be correlated to allow the recognitions models to grasp a better view of the potential problems.

This information can be even consumed by different applications for business information models and therefore save performance on the controller side since the data has to be extracted only once.

As summary, we have used OpenDaylight as a fundamental part of a very modular solution. Performance data from different hardware vendors is extracted seamlessly and stored in a database to be consumed by algorithms written in Python and R. The result of the algorithms and the historical performance data can be accessed through a REST interface, providing the means to attach the system to different user interfaces like the one that is included with the solution.

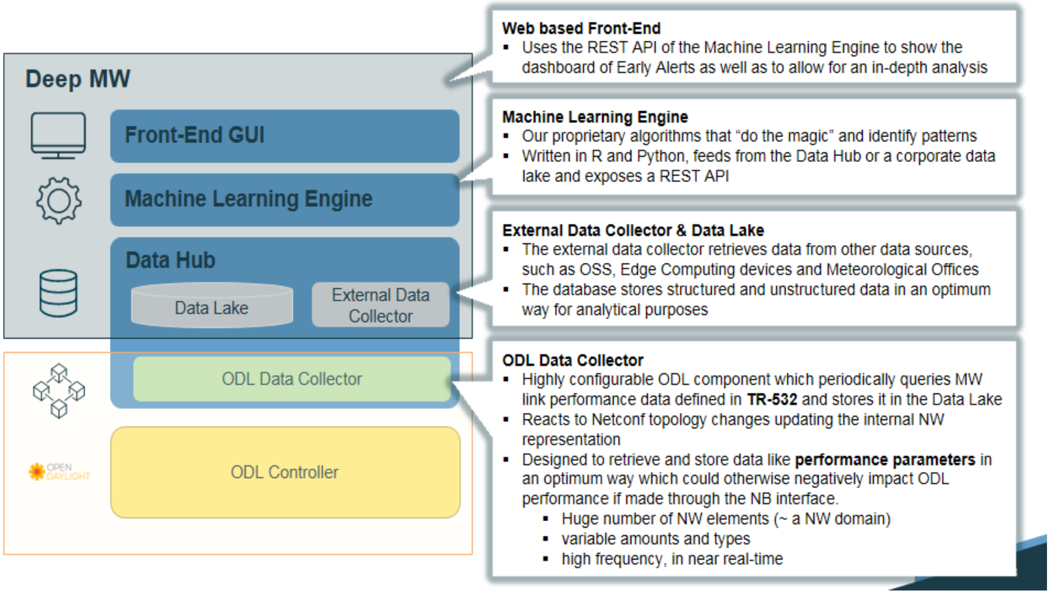


Figure 26: DeepMW Architecture

References

https://www.minsait.com/en/industries/telecommunications

https://wiki.OpenDaylight.org/view/OpenDaylight_Controller:MD-SAL

7.9 Intracom Telecom – Microwave Device with native Support

For the multi-domain and multi-vendor network topology setup of the 5th ONF PoC, Intracom Telecom contributed with an OmniBAS™ OSDR microwave link that had natively implemented the ONF TR-532 v1.1 microwave model.

The OmniBAS™ OSDR provided configuration, status and performance attributes directly to the SDN controller via the management port as indicated by the Microwave Link Information, the Preventive Maintenance and other Microwave Use Cases.

For additional information about the OmniBAS™ OSDR equipment, please refer to:

http://www.intracom-telecom.com/en/products/wireless_network_systems/4G5G_backhaul_fronthaul/omnibas_osdr.htm

7.10 Mentopolis – Automated Microwave Acceptance Application

Actual setup of a microwave link has to comply with link-specific parameters prescribed in the frequency assignment by the regulatory authority. To ensure compliance, the configured parameter values have to be compared with the frequency assignment and operation of the link approved or rejected afterwards. This microwave link acceptance process is performed by persons, who are different from the microwave link setup team (4-eyes principle). At some operators, the acceptance process is even outsourced to external auditing companies. Today's microwave link acceptance process is expensive and time consuming and can be simplified by using data retrieved from the network by SDN.

Mentopolis provided an application to the 5th PoC that automated the microwave link acceptance by comparing planning parameters plus regulatory input parameters with the data about the network gathered at the northbound of the SDN controller.

A standalone version of the LinkVis/APT application was used during the PoC. A limitation of the planning data base to the links composing the PoC network was the only difference to the software version, which is currently in commercial use at Telefonica Germany.

Instead of just prefilling the Acceptance-Report with values from the planning data base for later manual verification with information from the Local Craft Terminal, now actual configuration data is retrieved via northbound interface of the SDN controller and written into the Acceptance-Report; verification with data from the Local Craft Terminal is not necessary any more. (During the PoC, functionality was limited on the AirInterface Pac of the Microwave Information Model TR-532.) On user request (button pushed), the data is collected and processed. Once the processing is finished, the user may view the results and/or generate the report.

For the communication with the northbound interface via Restconf calls, a standalone library has been developed to use it in several products. The library method is first accessing the network topology to gather all available network elements from the live environment. After that, equipment and microwave information about every relevant network element is queried. Finally, the data is provided in a structure equal to the standard Yang model.

The processing and comparison of the planning and live data is done by the LinkVis/APT application.

7.11 Mycom OSI – Experience Assurance and Analytics Application

Context

On the operator side, there is need for a robust platform to monitor performance, configuration, status and topology of multi-vendor transport networks. Intuitive KPIs, dashboards and online reporting plus analysis is required for facilitating preventive and pro-active measurements on the network.

In order to meet this need, time series performance data needs to be extracted from the infrastructure. By integrating with OpenDaylight, and the open information models developed within the ONF framework, this can be achieved in a vendor-agnostic, means efficient way.

Implementation

For the 5th PoC, Mycom OSI was providing the Experience Assurance and Analytics (EAA) PrOptima™ application for end-to-end multi-vendor performance and configuration management including:

- Deployment of an API for extraction of time series performance data from the microwave infrastructure with an integration with OpenDaylight SDN
- Deployment of microwave model (mapping the provided YANG model) through a NIMS Adaptor for the import of performance data into EAA PrOptima™
- Provision of intuitive KPIs, dashboards for on-line reporting and analytics
- Models Supported: Microware Radio TR-532 V1.1, Ethernet PHY TR-541

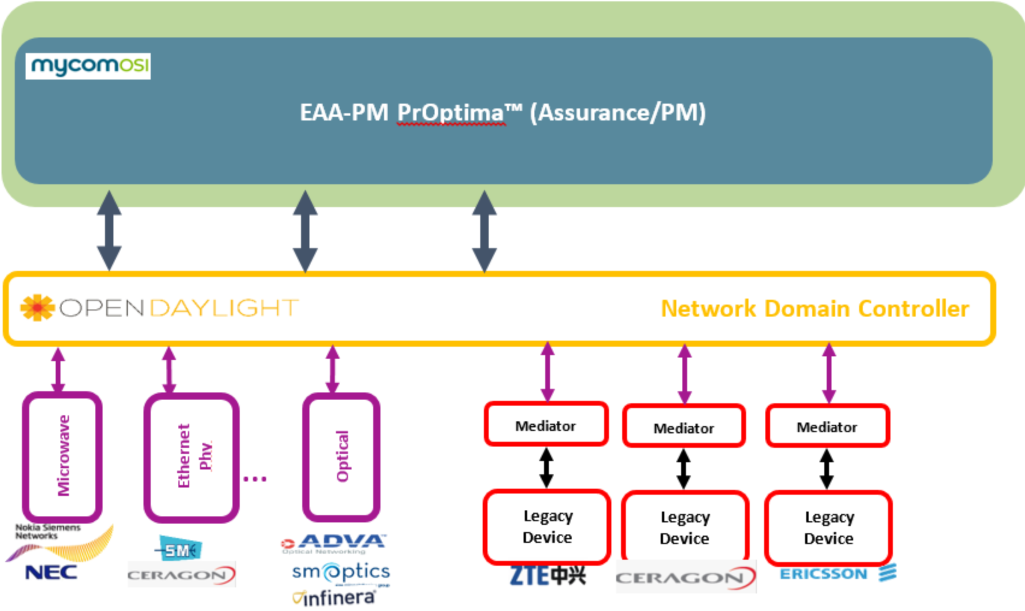


Figure 27: Experience Assurance and Analytics Application

A number of specific on-line reporting dashboards have been built and populated with data from a short time window during the 5th ONF PoC. This includes relevant KPIs and dashboards with the aim of enabling preventive and proactive performance management.

Dynamic baselining of TCAs are reported, including KPIs such as Link availability, ES, SES, UAS.

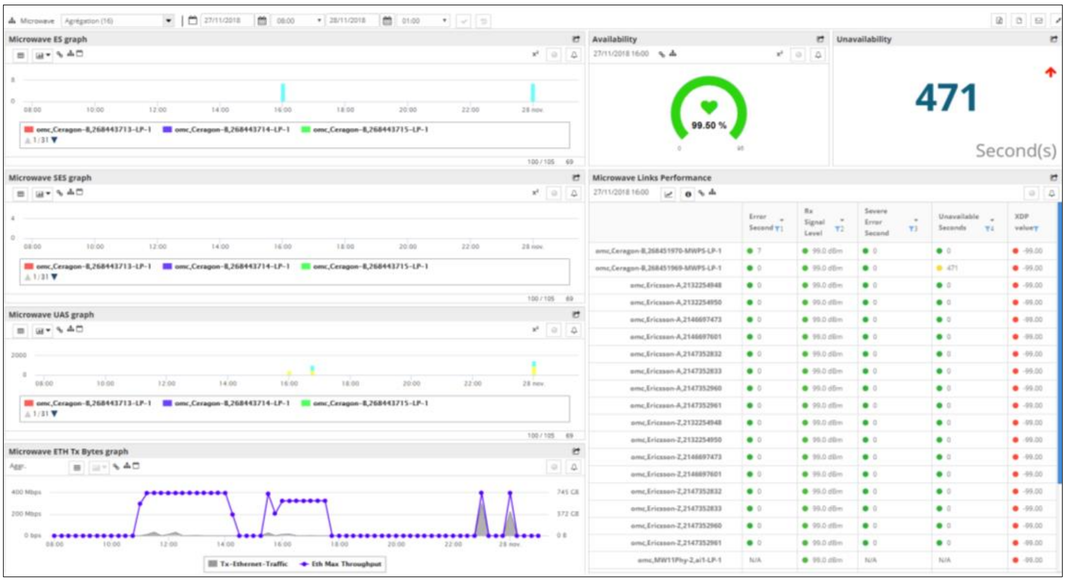
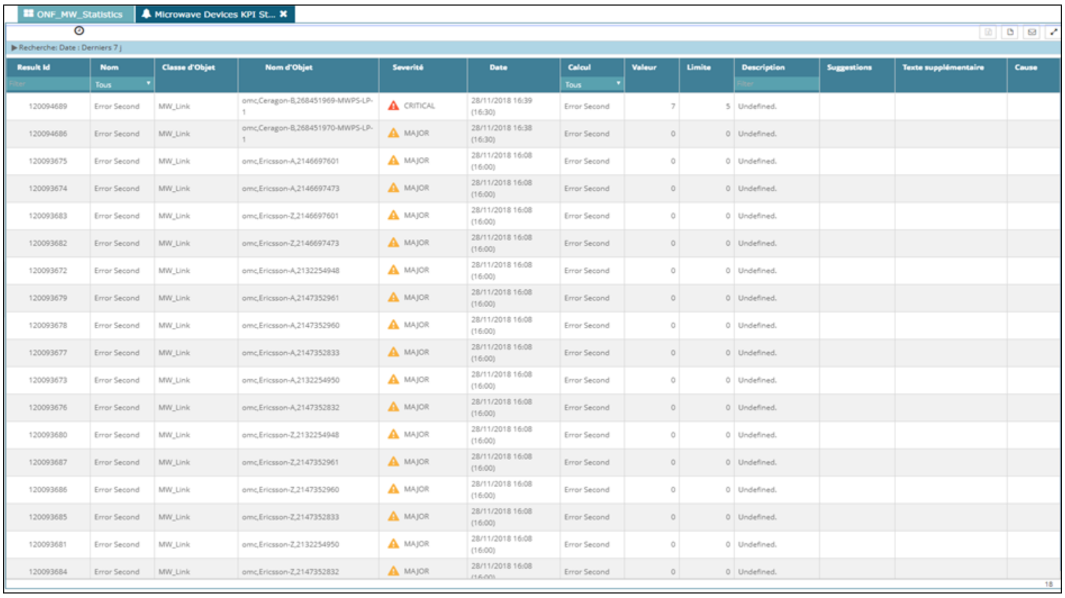


Figure 28: Resource Performance KPIs and Node/Link Inventory


ONF_MW_Statistics
Microwave Devices KPI St...
▶ Recherche: Date : Derniers 7 j

Result Id	Nom	Classe d'Objet	Nom d'Objet	Severité	Date	Calcul	Valeur	Limite	Description	Suggestions	Texte supplémentaire	Cause
Filter	Tous					Tous			Filter			
120094689	Error Second	MW_Link	omc,Ceragon-B,268451969-MWPS-LP-1	CRITICAL	28/11/2018 16:39 (16:30)	Error Second	7	5	Undefined.			
120094686	Error Second	MW_Link	omc,Ceragon-B,268451970-MWPS-LP-1	MAJOR	28/11/2018 16:38 (16:30)	Error Second	0	0	Undefined.			
120093675	Error Second	MW_Link	omc,Ericsson-A,2146697601	MAJOR	28/11/2018 16:08 (16:00)	Error Second	0	0	Undefined.			
120093674	Error Second	MW_Link	omc,Ericsson-A,2146697473	MAJOR	28/11/2018 16:08 (16:00)	Error Second	0	0	Undefined.			
120093683	Error Second	MW_Link	omc,Ericsson-Z,2146697601	MAJOR	28/11/2018 16:08 (16:00)	Error Second	0	0	Undefined.			
120093682	Error Second	MW_Link	omc,Ericsson-Z,2146697473	MAJOR	28/11/2018 16:08 (16:00)	Error Second	0	0	Undefined.			
120093672	Error Second	MW_Link	omc,Ericsson-A,2132254948	MAJOR	28/11/2018 16:08 (16:00)	Error Second	0	0	Undefined.			
120093679	Error Second	MW_Link	omc,Ericsson-A,2147352961	MAJOR	28/11/2018 16:08 (16:00)	Error Second	0	0	Undefined.			
120093678	Error Second	MW_Link	omc,Ericsson-A,2147352960	MAJOR	28/11/2018 16:08 (16:00)	Error Second	0	0	Undefined.			
120093677	Error Second	MW_Link	omc,Ericsson-A,2147352833	MAJOR	28/11/2018 16:08 (16:00)	Error Second	0	0	Undefined.			
120093673	Error Second	MW_Link	omc,Ericsson-A,2132254950	MAJOR	28/11/2018 16:08 (16:00)	Error Second	0	0	Undefined.			
120093676	Error Second	MW_Link	omc,Ericsson-A,2147352832	MAJOR	28/11/2018 16:08 (16:00)	Error Second	0	0	Undefined.			
120093680	Error Second	MW_Link	omc,Ericsson-Z,2132254948	MAJOR	28/11/2018 16:08 (16:00)	Error Second	0	0	Undefined.			
120093687	Error Second	MW_Link	omc,Ericsson-Z,2147352961	MAJOR	28/11/2018 16:08 (16:00)	Error Second	0	0	Undefined.			
120093686	Error Second	MW_Link	omc,Ericsson-Z,2147352960	MAJOR	28/11/2018 16:08 (16:00)	Error Second	0	0	Undefined.			
120093685	Error Second	MW_Link	omc,Ericsson-Z,2147352833	MAJOR	28/11/2018 16:08 (16:00)	Error Second	0	0	Undefined.			
120093681	Error Second	MW_Link	omc,Ericsson-Z,2132254950	MAJOR	28/11/2018 16:08 (16:00)	Error Second	0	0	Undefined.			
120093684	Error Second	MW_Link	omc,Ericsson-Z,2147352832	MAJOR	28/11/2018 16:08	Error Second	0	0	Undefined.			

18

Figure 29: Threshold Crossing Alarms

The results demonstrate seamless end-to-end multi-vendor integration through an off-the-shelf adaptor, which is supporting the microwave and Ethernet models specified by ONF, which will lead to closed loop orchestration by SDN in the future.

Resources

For more information, please visit: www.mycom-osi.com.

7.12 NEC – Microwave Device with native Support

For this 5th ONF POC, NEC provided its latest iPASOLINK split mount VR4 microwave communications system, which is part of their Next Generation Converged Radio solution for LTE & beyond. It is equipped with a Netconf interface that is natively supporting ONF TR 532v1.0 and does not require the use of an external mediator.

During the PoC, the microwave information for planner, the preventive maintenance and some use cases of the 4th ONF PoC (now with native support instead of mediator) have been successfully supported. Connectivity to the ONAP platform and compatibility with the OSS/Applications attached on top of the ONAP orchestrator have been demonstrated. Backwards compatibility of microwave applications in the OpenDaylight controller has also been supported and shown.

For additional information about the NEC iPASOLINK product and solution please visit https://www.nec.com/en/global/prod/nw/pasolink/index.html

7.13 Nokia – Microwave Device with native Support

In the 5th ONF PoC, Nokia provided a link of the Wavence microwave platform with a Netconf agent running natively with the ONF TR-532 v1.0 models. The Wavence microwave node was able to connect to the OpenDaylight controller and to provide Alarms and Performance data for the microwave link.

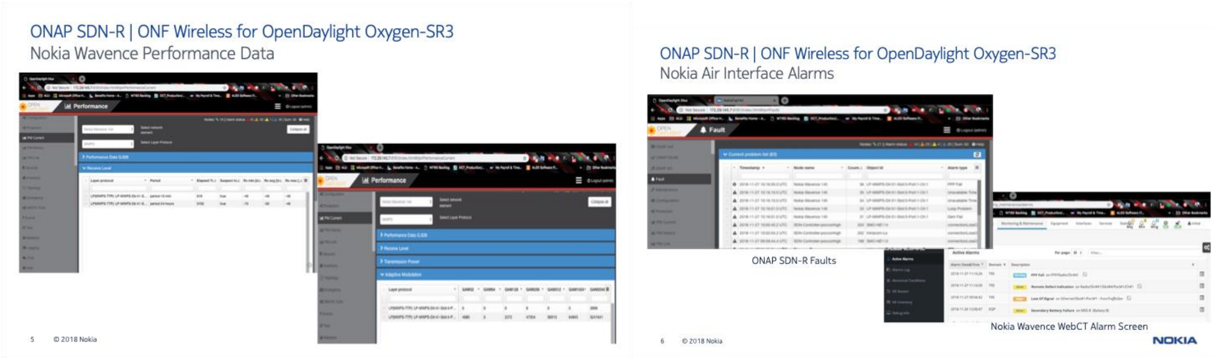


Figure 30: Wavence devices in OpenDaylight controller

Since the Netconf agent was running natively on the Wavence platform, no mediator virtual machines were needed to support translations. This is allowing operators to generate OPEX savings from deploying SDN without setting up additional virtual machines for mediators. Nokia has introduced native Netconf support with the Wavence 18.0 platform and will continue to enhance the Netconf support with the ONF TR-532 v1.1 models.

7.14 SIAE – Microwave Network Performance Application

The Microwave Network Performance Application named iVeritas is a preventive maintenance software providing analytics of Key Performance Indicators (KPIs) for microwave radio and Ethernet.

The major issue for vendors, due to the large amount of data to be processed, is to have a detailed overview about their network. iVeritas offers immediate feedback on how the network is performing and allows the operator to drill down to the most critical links impacting the custom defined KPI. This facilitates immediate preventive action, respectively addressing critical issues before they become outages.

iVeritas allows analysis using KPIs based on performance measurement raw data:

- Radio link performance according to ITU-T G.826 (ES, SES, UAS)
- Ethernet traffic utilization
- Ethernet traffic utilization based on busy hours dynamic criteria
- Radio link received power level
- Radio signal to noise ratio

Based on ONF TR-532 Microwave Information Model and the northbound interface of the SDN controller, iVeritas collects and stores traffic information from a multi-vendor microwave network.

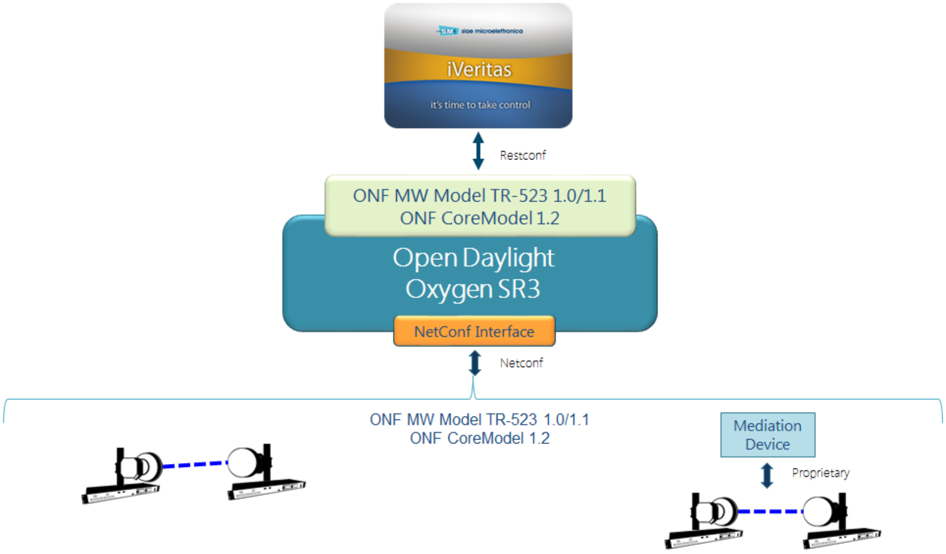


Figure 31: iVeritas Network

iVeritas keeps a configurable amount (limited only by the capacity in terms of gigabytes of disk space of the machine where the application is hosted) of historical data. The data collection is applicable to all network topologies.

In the 5th PoC implementation, the application reads historical performance values about the Ethernet Container and the AirInterface in 15 minutes intervals and provides a configurable and user customizable dashboard, graphical views and data tables.

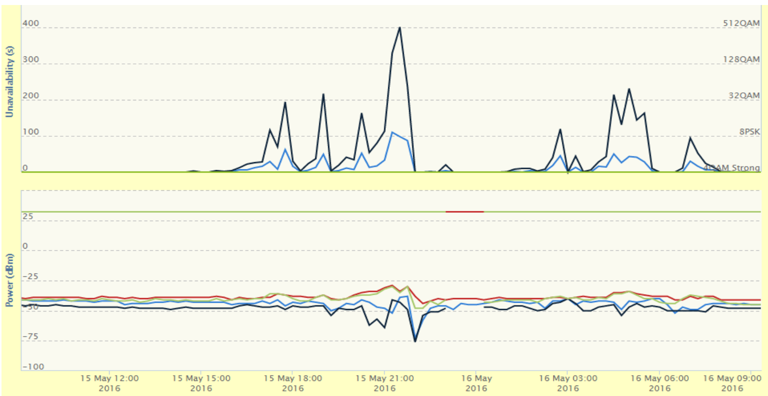


Figure 32: Time Series of Performance Values

The implementation is divided into server parts and client parts. They supervise the connectivity to the SDN controller and periodically request for performance monitoring data. Using the client plugin the application is able to retrieve data directly from the northbound exposed by the SDN controller performing Restconf calls. All the collected performance and remote monitoring values are processed and sent to the server plugin running on an Apache Tomcat web service that receives the data and stores all the data in a local data base. iVeritas itself is also running in an Apache Tomcat Web Server and reads the data previously stored in the local data base.

7.15 SIAE – Planning Data Reconciliation Application

A major issue for vendors, is to constantly have a detailed overview about their own network and to keep differences between the planned network configuration and the real network configuration under control. This includes shaving tools supporting the alignment of the two configurations.

Planning Data Reconciliation Application shows for each network device its actual configuration compared to what has initially been planned by the business and it allows to apply alignment operations, in both directions. It guides users to take actions for obtaining the desired alignment just with a couple of clicks.

This application works on top of the SDN controller

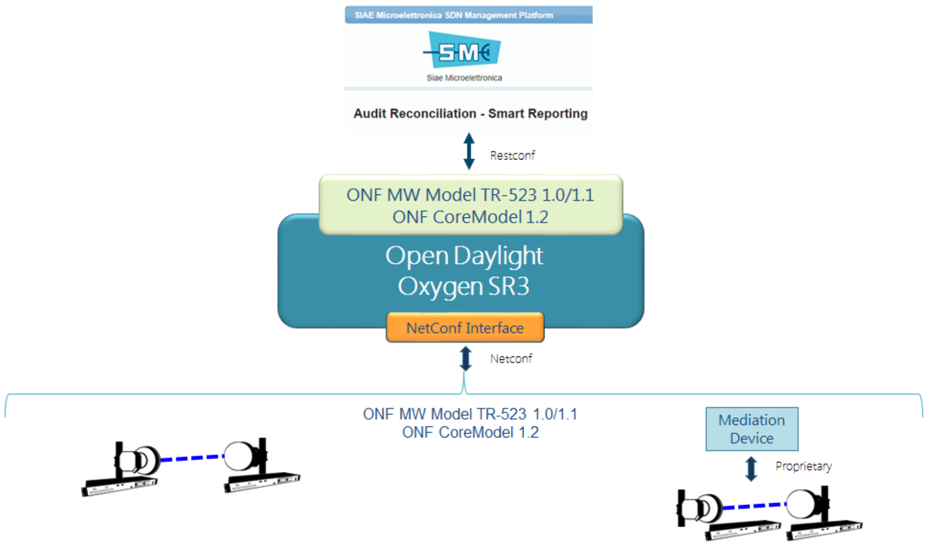


Figure 33:

Audit functionality is performed comparing values of parameter defined as “Comparable” read from:

- Network (via Restconf from OpenDaylight using TR-532 microwave model)
- Planning DB

Planning DB is an internal DB that can be:

- Created as a one shot snapshot from the network
- Importing data from a custom planning

Reconciliation is available for mismatched parameters in two different modes:

- Modifying network element setting via Restconf on OpenDaylight
- Updating local planning DB

Planning Data Reconciliation SDN application provides a configurable and customized (by user) dashboard where user can straight away visualize radio port properties which determine the configuration differences between network and planning, in addition to execute alignment operations.

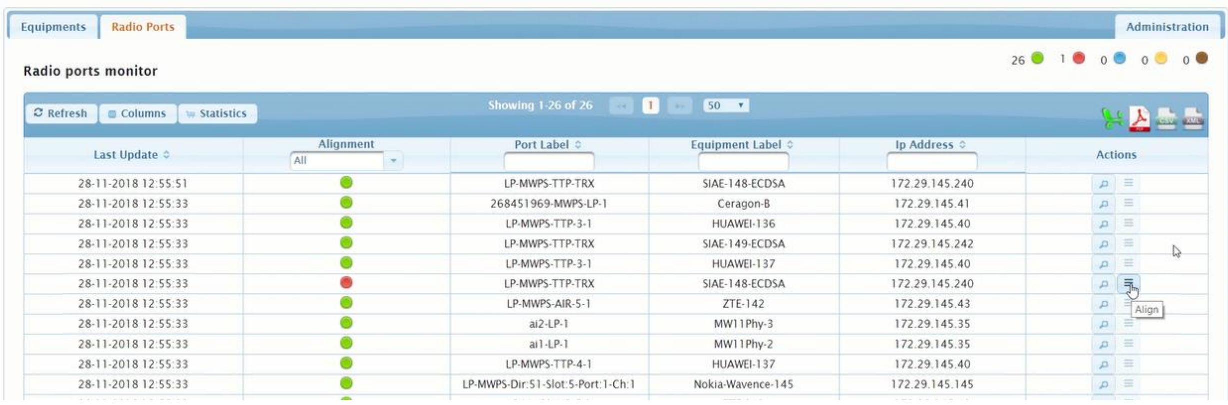


Figure 34: Radio Port details

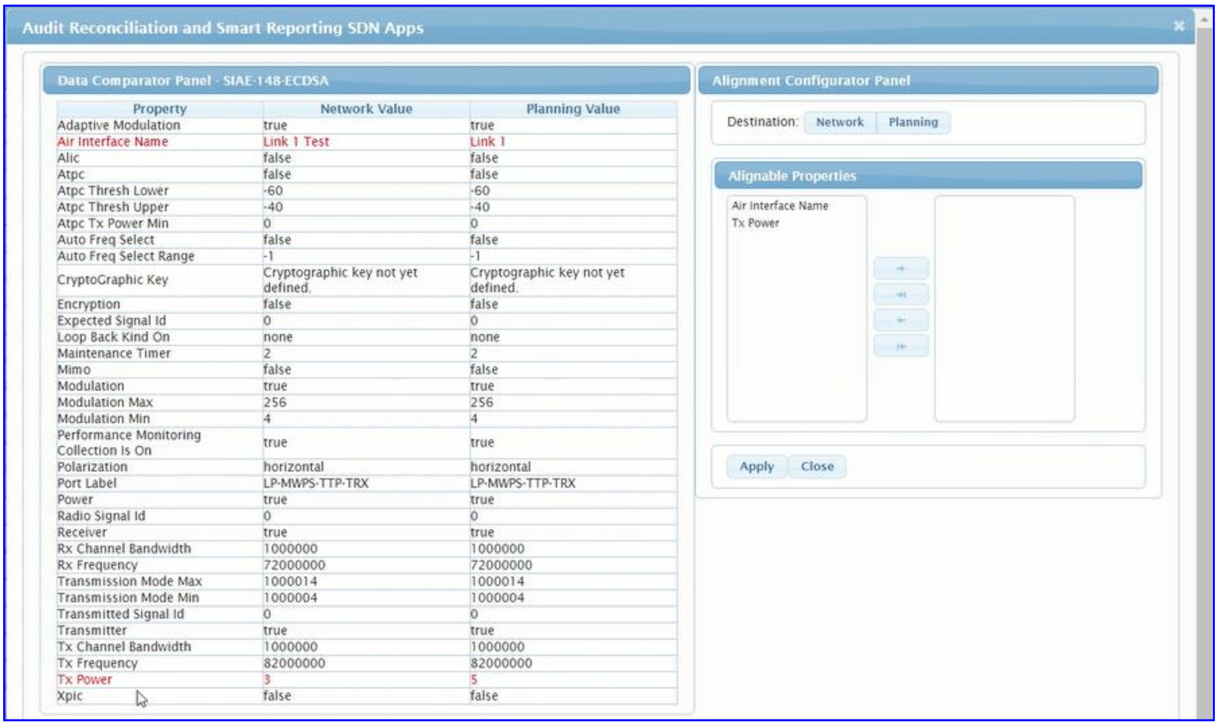


Figure 35: Alignment window

The implementation of the business logic is in Java language. It supervise the connectivity to the SDN controller and periodically requests current parameter status directly from the northbound exposed by the controller performing Restconf calls.

All the values collected are elaborated by the web service and stored in a local database.

7.16 SIAE – Smart Reporting Application

Smart Statistics Reporting is an application for network administrators becoming able to immediately give a quick and clear overview about the devices in a given network. Smart Statics Reporting allows to draw pie charts for many equipment and radio interface related properties. It gives a really clear graphical tool to visualize the values of each important equipment property.

This application works on top of the SDN controller. It applies the Restconf interface together with ONF TR-512 Core Information Model (for inventory information) and ONF TR-532 Microwave Information Model (for information about the radio interface)

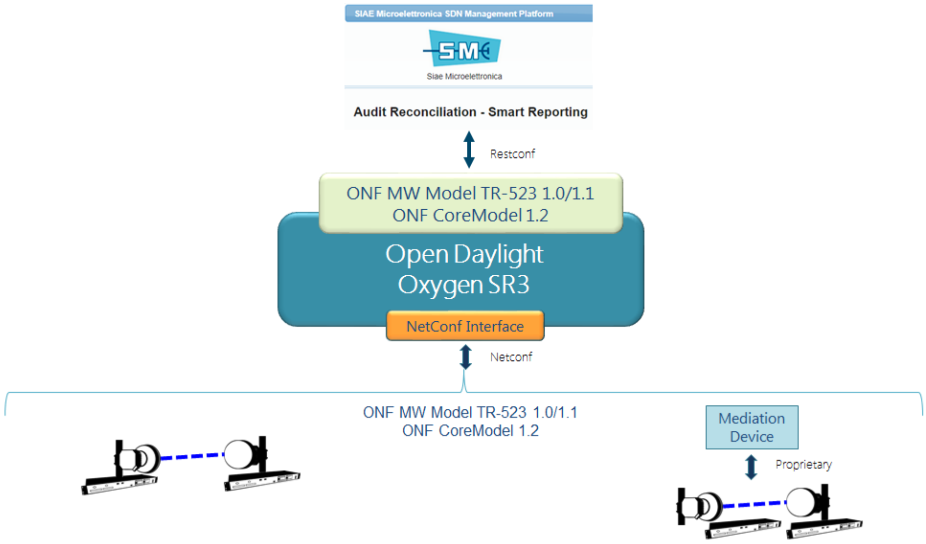
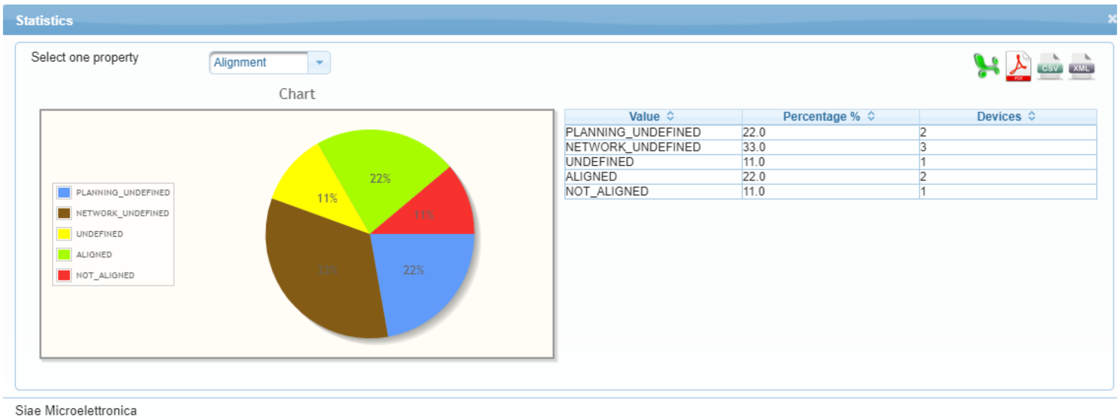
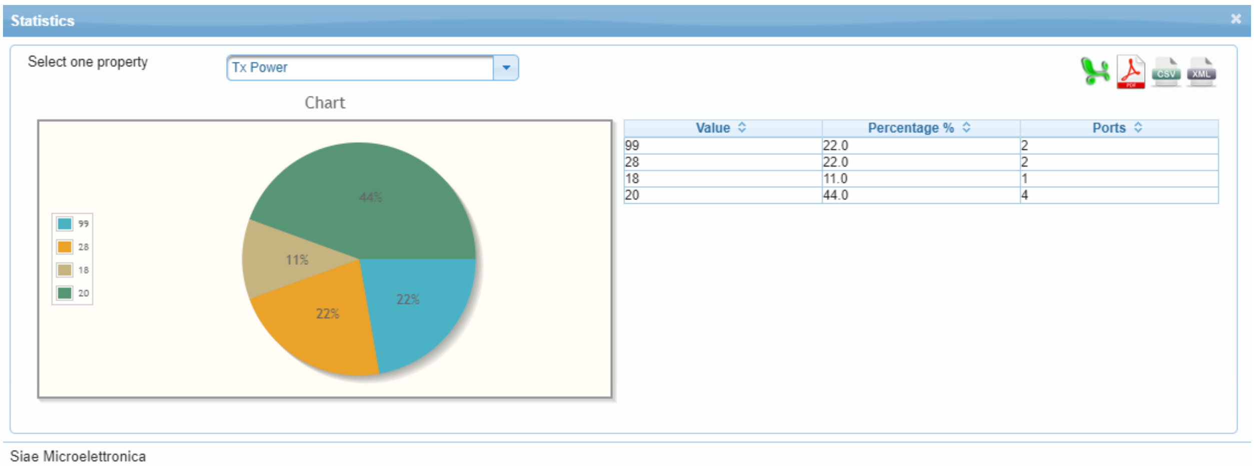


Figure 36:

The application provides a listed representation and a configurable pie chart based on selected parameter. A detailed view is available for radio ports and it is possible to configure how to manage each single parameter including defining, whether it can be read, compared and aligned. The application allows to create different types of pie charts based on the property that the operator wants to analyze.



The created reports can be exported in the following file formats: excel, pdf, csv and xml



The business logic is implemented in Java language. It supervise the connectivity to the SDN controller and periodically requests current parameter status directly from the northbound exposed by the controller performing Restconf calls.

All the values collected are elaborated by the web service and stored in a local Database.

7.17 SIAE – Microwave Device and Mediator

A Mediator translates Netconf queries, which it receives at its northbound interface, into a proprietary management interface (usually SNMP) at its southbound. Applying such Mediator has several advantages:

- It allows managing legacy devices that will never be updated with a native Netconf interface
- It allows to test new releases of Yang models while the native Netconf agent still implements the official model
- It allows to flexibly change the model; e.g. at events like PoCs new structures can be evaluated and errors can be fixed in a very short time period compared to rebuilding and installing a complete new firmware version on a system

SIAE's Mediator for the 5th PoC supported the following ONF definitions:

- TR-512 (v1.2) Core Information Model
- TR-532 (v1.1) Microwave Information Model
- TR-541 (v0.3) Ethernet PHY Information Model
- TR-545 (v1.0) Device Management Interface Profile

Besides the microwave, this Mediator supported the following DMIP and Ethernet PHY use cases:

- Automated Commissioning Phase 1 – DHCP
- Automated Commissioning Phase 2 – Netconf Call Home
- Candidate Data Store
- Confirmed Commit
- Ethernet Port Information for Planners
- Ethernet Port Documentation Verification

The Mediator is based on the openYuma Netconf server and implemented as a Virtual Network Function (VNF). It can be instantiated on the same host machine as often as required, because it takes advantage from the concept of containers. A Docker platform provides the ability to package and run an application in a loosely isolated environment called a container. The isolation and security allow you to run many containers simultaneously on a given host. Containers are lightweight, because they don't need the extra load of a hypervisor, but run directly within the machine kernel of the host.

An Instance Manager is responsible for creating and deleting Mediator instances and their connectivity. As shown in Figure 37, the Instance Manager reads kind and number of the necessary Mediators from a data base. It creates the containers that always comprise a single Mediator instance and connects their Ethernet interface (eth0) with a new virtual Ethernet interface (veth) at a virtual switch. The virtual switch itself gets automatically created at the time the first Mediator gets instantiated. By adding the Ethernet interface of the host machine to the virtual switch, all Mediator instances get connected to the outside network.

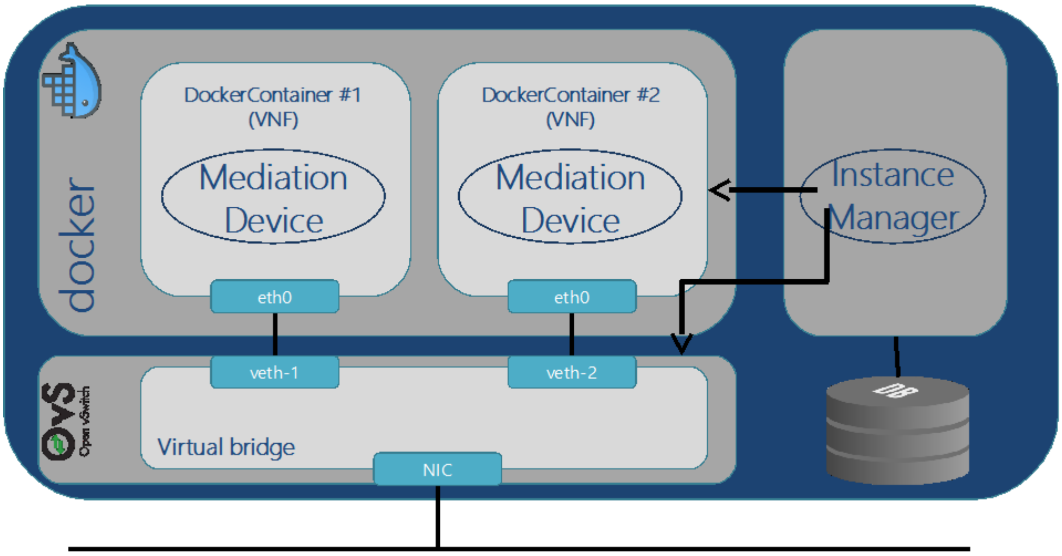


Figure 37: Mediators as flexibly instantiated Containers

At startup of the Mediator Virtual Machine (incl. Instance Manager etc.), it sends a DHCP request through its Ethernet interface. The DHCP server reachable through the outside network leases an IP address to the Mediation Virtual Machine. The same DHCP offer contains also the IP address of the SDN controller. For proper interworking, the DMIP TR-545 defines that the controller's IP address has to be marked with option code 43, which originally has been defined for transporting vendor-specific information.

The SDN controller IP address is used for opening a TCP connection to a specific Netconf Call-Home TCP port at the Controller Virtual Machine. OpenDaylight use the port 6666 as a default value. That's different from port 4334, which should be standard. Once the TCP connection got opened, the Netconf client inside OpenDaylight creates a Netconf over SSH session to the Netconf server of the Mediator.

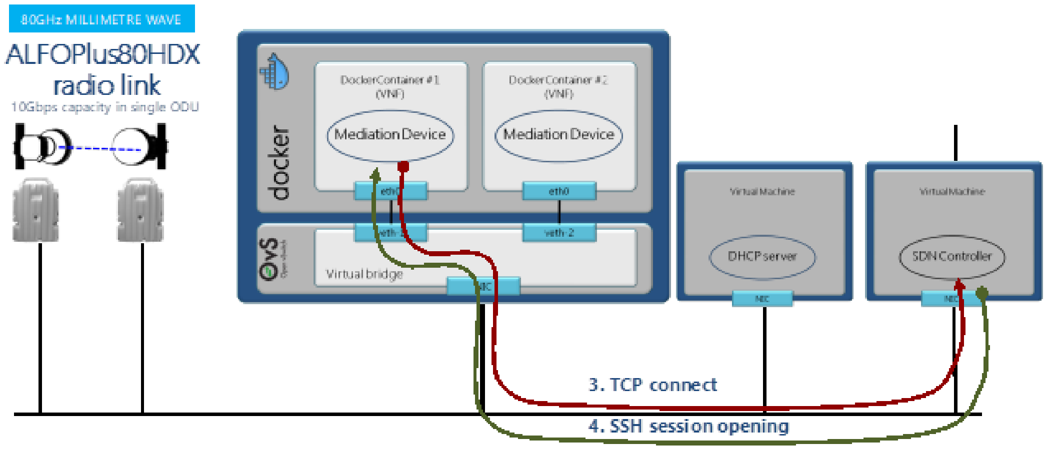


Figure 38: Automated Commissioning of a Mediator Instance

From this time on, the Netconf client can apply the management connection.

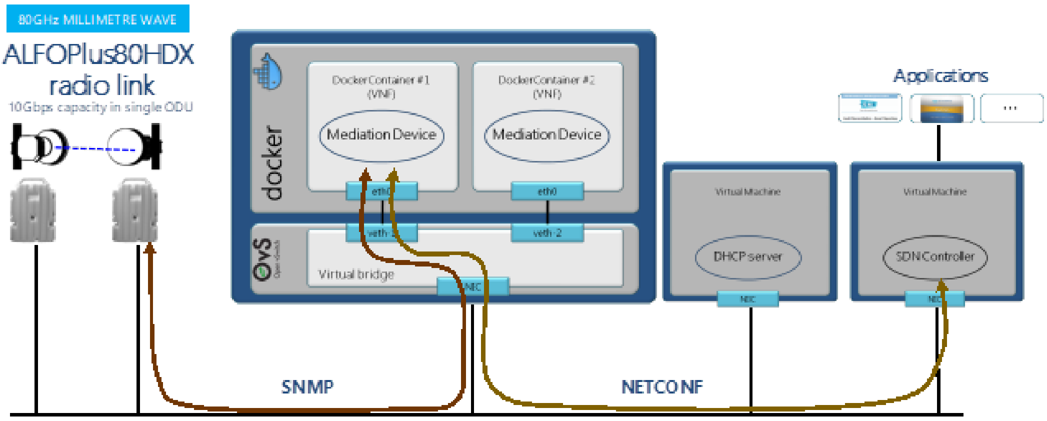


Figure 39: Mediator supported Management Connection

7.18 SM Optics - Optical Transponder

During the 5th Proof of Concept (PoC), SM Optics has participated with a Disaggregated Optical Transponder, which was mapping 1Gbit/s and 10Gbit/s services over OTN (OTU2/OTU2e) on a third party WDM system. Scope was implementing a common Yang model covering photonic transport for showing the possibility to control a multi-vendor optical network with a single SDN controller and common applications.

The following information models have been covered:

- ONF Core Model TR-512 v.1.2
- ONF Core Model Extensions for Photonic Model (part of TAPI 2.1)
- ONF Device Management Interface Profile TR-545

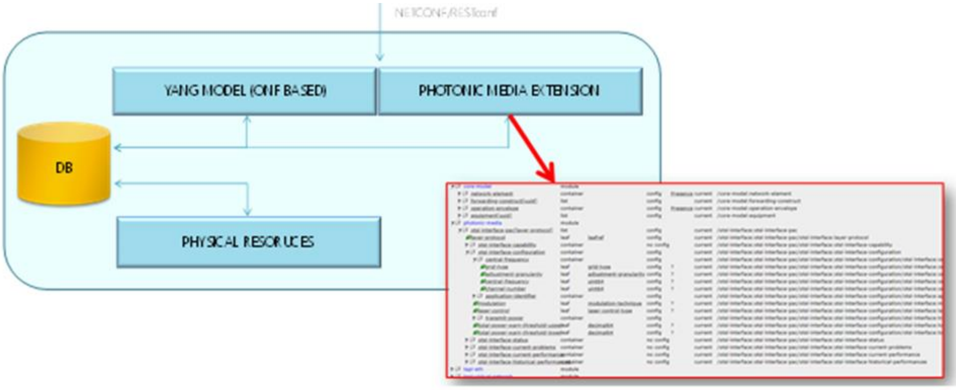
The following activities have been implemented:

- Supervision of optical products from a common OpenDaylight instance
- Read Capabilities and Status for the photonic parameters
- Configure some parameters (either in OpenDaylight or via optical application)
- Optical Performance monitoring reporting
- Reporting of alarms in case of failures

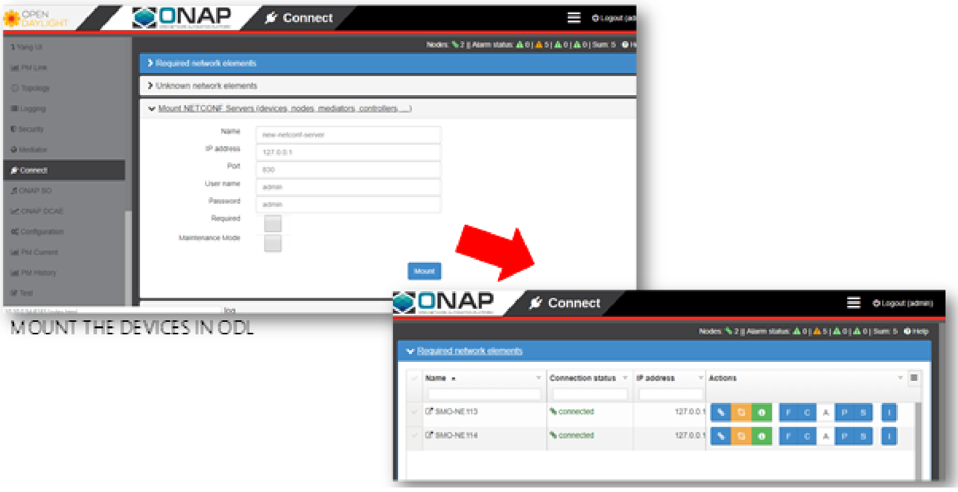
The following hardware components have been applied:

- LM1 (Lightmode 1 Rack Unit) OTN 10G
- Configured with:
- 2 x OTU2/OTU2e lines (colored DWDM)
- 1 x 10Gbit/s client
- 2 x 1Gbit/s clients

Implementation has been based on native support of Yang model by the network element and with Netconf server running directly in the network element. For being able to manage the complete node, the PoC Photonic Yang models have been added to the models already existing on the device.

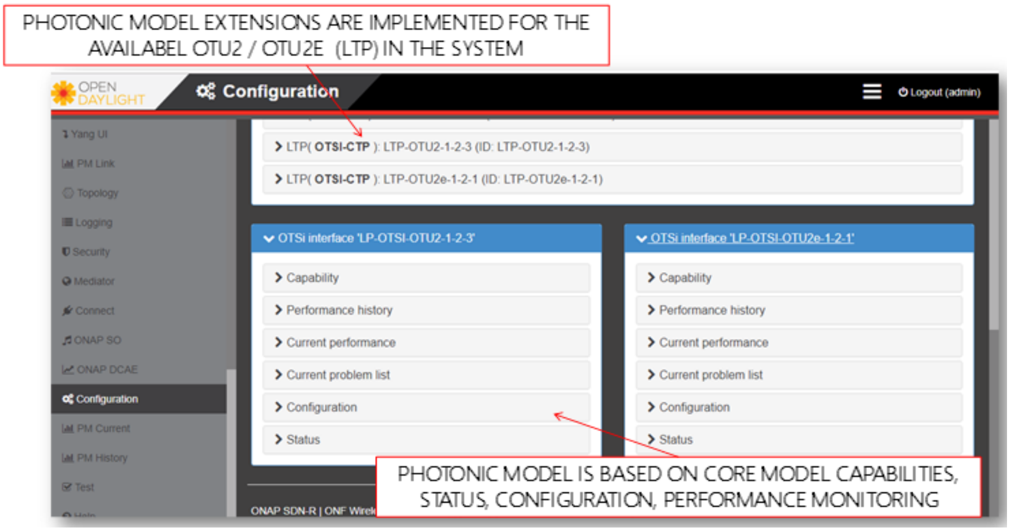


Mounting the network element in OpenDaylight allowed establishing a TCP connection between Netconf client and Netconf server (over SSH):



From that on, the Netconf client could manage and configure the network element based on the common Photonic Model.

The parameters, which are defined in the photonic extension of TAPI 2.1, allowed to access the capability, the configuration, the status, the performance monitoring and the problem list.



Parameters were then made available at the northbound Restconf interface of the OpenDaylight controller and applications were retrieving the data from there.

7.19 TechMahindra – Transponder Application

Integrational of microwave, router and optical equipment is required for implementing an end to end SDN, e.g. in the mobile backhaul network. A combination of Yang models, which are consolidating all these device types and OSI network layers, is required to get a holistic view on the network and to allow e.g. network analysis or loop back automation looking at the network as a whole.

The ONF Core IM (TR-512) is very much qualified to be the base of such a combination of Yang models. It has just to be amended by technology specific information models like TR-532 for microwave or TR-541 for Ethernet PHY. For covering optical equipment, the newly developed optical extension to the TAPI (Transport API) has been directly attached to the Core IM.

Within the 5th PoC, first steps of testing this implementation of a vendor-agnostic management interface for optical transport equipment have been taken. This included verification of proper establishment of the connection between device and controller. A Transponder Application Portal has been created for presenting the values of a couple of Configuration attributes and for modifying a couple of these attributes.

TechMahindra contributed such Transponder Application Portal to the 5th PoC and successfully demonstrated the following use cases:

1. Presenting the optical-media parameters
   - Laser State
   - Optical power values
   - Wavelength information
   - Performance data
2. Configuration window to modify below parameters values
   - Laser State
   - Threshold Configuration values for Rx min and Rx max.
3. Switch off the laser from configuration window and validated on Traffic Generator flow impacted by creating fault counts and turning its status to red

The Transponder Application Portal is implemented in Angular 6 UI Framework and NodeJS.

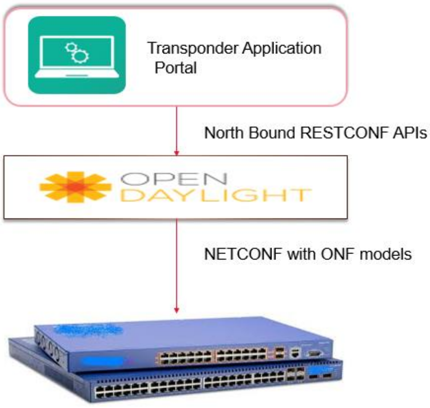


Figure 40: High Level Architecture of the Transponder Application Portal

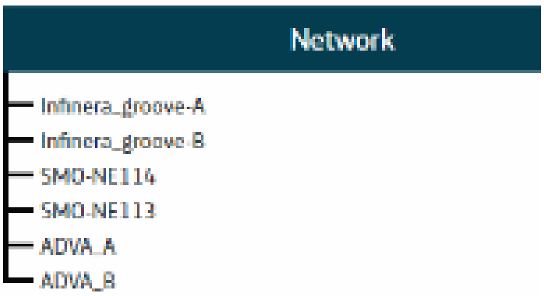
It interacts with the northbound Restconf interface of the SDN controller (OpenDaylight) to get all the parameter values, which are presented in its graphical user interface. Also PUT requests are made on this interface for modifying configuration parameters at the optical device.

NOTE: The Restconf requests below are just examples. Actual URLs always depend on the specific case.

The network element's details are fetched via GET request: http://<ODL-URL>:8181/restconf/operational/network-topology:network-topology/topology/topology-netconf/. Below the information is provided with the response

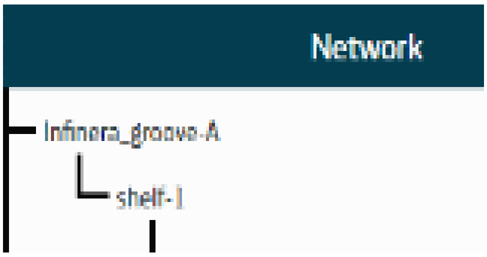
```
<capability>
(urn:onf:params:xml:ns:yang:otsi-interface?revision=2018-09-24)photonic-media
<connection-status xmlns="urn:opendaylight:netconf-node-topology">connected</connection-status>
```

The node id's which are matching with above info are getting displayed as NE as below:



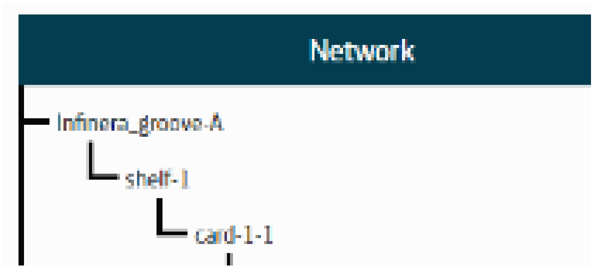
GET requests like http://<ODL-IP>:8181/restconf/operational/network-topology:network-topology/topology/topology-netconf/node/Infinera-groove-A/yang-ext:mount/core-model:network-element, resulted in the following inventory information:

```
<extension>
<value-name>top-level-equipment</value-name>
<value>SHELF-1</value>
```

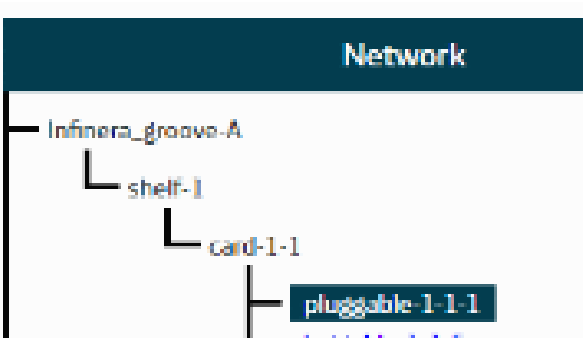



The following GET request http:<ODL-IP>//:8181/restconf/operational/network-topology:network-topology/topology/topology-netconf/node/Infinera-groove-A /Yang-ext:mount/core-model:equipment/SHELF-1 resulted in further details about this board:

```
<contained-holder>
<uuid>Card-1-1</uuid>
<administrative-state>unlocked</administrative-state>
<spatial-properties-of-type>
<length/>
<width/>
<height/>
</spatial-properties-of-type>
<occupying-fru>Card-1-1</occupying-fru>
[..]
```

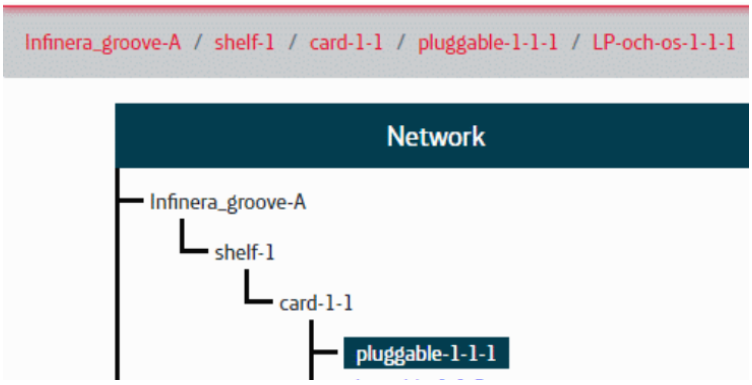


To get the port details, GET request: http://<ODL-IP>:8181/restconf/operational/network-topology:network-topology/topology/topology-netconf/node/Infinera-groove-A/Yang-ext:mount/core-model:equipment/Pluggable-1-1-1 has been sent. In this case, there is no “occupying-fru” -> meaning end-of-algorithm > the Pluggable-1-1-1 is a “leaf”.



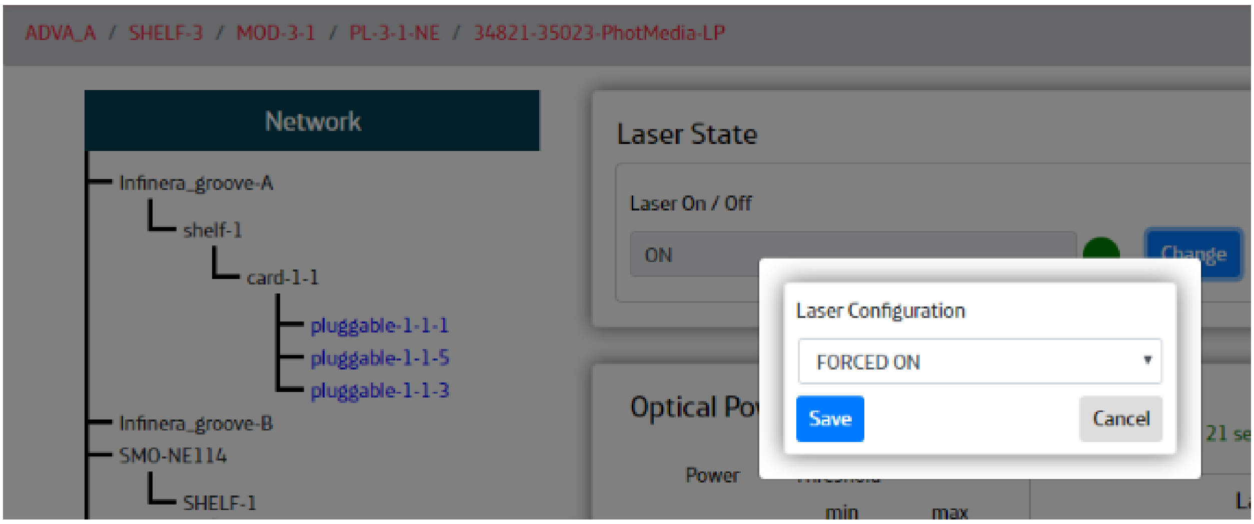
Now search in the Network-element LTP list for such “leaf” (pluggable-1-1-1) http://<ODL-IP>:8181/restconf/operational/network-topology:network-topology/topology/topology-netconf/node/Infinera-groove-A/Yang-ext:mount/core-model:network-element

<ltp>
<uuid>82003-85922-och</uuid>
<administrative-state>locked</administrative-state>
<ltp-direction>bidirectional</ltp-direction>
<physical-port-reference> **Pluggable-1-1-1**</physical-port-reference>
<lifecycle-state>planned</lifecycle-state>
<client-ltp>82003-85922-otu2</client-ltp>
<label>
<value-name>type</value-name>
<value>Photonic Media TTP</value>
</label>
<lp>
<uuid>**LP-och-os-1-1-1**</uuid>
<administrative-state>locked</administr



For reading configuration and status data, below GET requests are invoked http://<ODL-IP>:8181/restconf/operational/network-topology:network-topology/topology/topology-netconf/node/Infinera-groove-A/Yang-ext:mount/photonic-media:otsi-interface-pac/LP-och-os-1-1-1/otsi-interface-configuration.

For writing/Put data to optical devices, below PUT request is used.



Resources

For more information contact:

- Vinay Prakash - VP0036927@TechMahindra.com
- Prasanna – PK00106937@techmahindra.com

7.20 Wipro – Interface Validator

Context

The Interface Validator supports device vendors in testing their interface implementations, but also allows network operators to automate acceptance testing. It is a protocol validation software that verifies the Netconf protocol operations as triggered by third-party applications, which are run on top of the SDN controller.

The Interface Validator that has been provided by Wipro to the 5th PoC is in principle agnostic to the technology, respectively information model under test. The respective test cases are formulated in XML and can also be scheduled and executed as a batch.

Implementation

libnetconf is yet another open source implementation by CESNET that provides both the server and client side implementations for Netconf according to IETF RFC6241. It enables the developers to add Netconf functionality to any heterogeneous network. The library provides the necessary APIs required to manipulate the data store, to carry out operations on the device based on the content of the data store and also to handle custom RPC operations.

CESNET has made available example implementations of Netconf servers, client command line interfaces and a client graphical user interfaces; all of them leverage libnetconf library under the hood to perform Netconf operations.

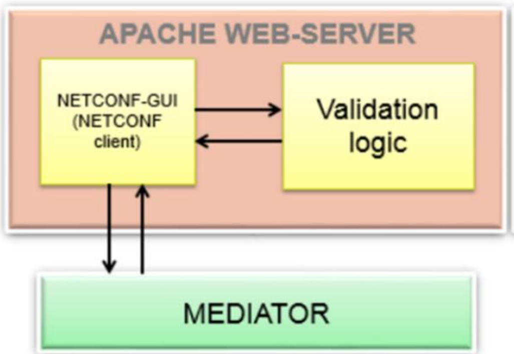


Figure 41: Architecture of the Netopeer GUI

The Interface Validator is an enhancement made by Wipro to the existing Netconf client graphical user interface called Netopeer GUI. The architecture of the Netopeer GUI is as shown in Figure 41. In the context of the PoC, the graphical user interface connects to the Mediator as a Netconf client although in general it can connect to any device with Netconf server implementation.

The graphical user interface is designed and developed based on the Symfony PHP framework. It communicates with an apache module with Netconf client capabilities. The tool has been enhanced with a new set of webpages and an additional test case validation logic. The Netconf client command line interface implementation from CESNET allows connecting with a single Netconf server and thus the test case execution option is allowed only on the device with an active Netconf session. The test case execution capability, which is available in the command line interface, has been integrated with the Netopeer GUI. This enables seamless session establishment with several Netconf capable devices and scheduling the test case execution on a device of choice. The Netopeer GUI maintains persistent connections with the

Netconf servers and provides a user interface to carry out several operations pertaining to test case execution. The following operations are supported:

- Upload a test case
- Delete a test case
- Start execution of test cases
- Stop execution of test cases
- Download all the logs pertaining to all the test cases or download logs per test case
- Set the delay time for the test case execution to start and so on

Resources

For further information about Netopeer please visit https://github.com/CESNET/netopeer.

7.21 ZTE – Microwave Device and Mediator

ZTE participated with their ZXMW NR8120A microwave device at the 5th PoC. A Mediator solution has been applied to expose the latest TR-532 v1.1 microwave management model over Netconf protocol towards the OpenDaylight controller and the proprietary interface towards the device.

The Mediator could be deployed as a VM software (e.g. based on Docker container). During the 5th PoC, it has been operated on a laptop for simplification.

The connections between the components of the SDN control plane are depicted in Figure 42.

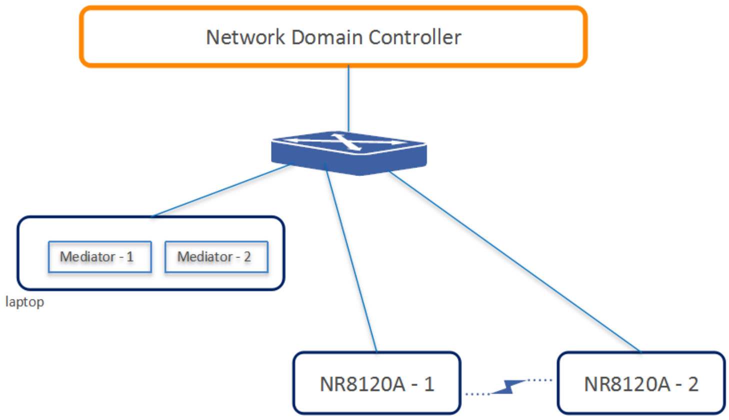


Figure 42: SDN control plane with Mediator support

8 References

[1]	ONF TR-512 Core Information Model www.opennetworking.org/software-defined-standards/models-apis/
[2]	ONF TR-532 Microwave Information Model www.opennetworking.org/software-defined-standards/models-apis/
[3]	ONF TR-541 Ethernet PHY Information Model github.com/openBackhaul/wireInterface
[4]	ONF TR-545 Device Management Interface Profile www.opennetworking.org/software-defined-standards/models-apis/
[5]	ONF Transport API (TAPI) wiki.opennetworking.org/display/OTCC/TAPI
[6]	IEEE 802.3 Ethernet physical layer standards.ieee.org/standard/802_3-2018.html

[7]	CENTENNIAL GitHub www.github.com/OpenNetworkingFoundation/CENTENNIAL
[8]	IETF RFC6241 NETCONF tools.ietf.org/html/rfc6241

9 Terminology

Abbreviation	Meaning
API	Application Programming Interface
BSS	Business Support Systems
CI	Configuration Item
COTS	Commercial off the shelf
DB	Data Base
DHCP	Dynamic Host Configuration Protocol
DMIP	Device Management Interface Profile
DVM	Default Value Mediator
ES	Errored Seconds
GUI	Graphical User Interface
IANA	Internet Assigned Numbers Authority
IETF	Internet Engineering Task Force
IM	Information Model
IP	Internet Protocol
KPI	Key Performance Indicator
KVM	Kernel-based Virtual Machine
LTE	Long Term Evolution
LTP	Layer Termination Point
MD-SAL	Model Driven Service Abstraction Layer
NBI	Northbound Interface
NE	Network Element
Netconf	Network Configuration Protocol
NFV	Network Function Virtualisation
ODL	OpenDaylight
OLS	Optical Line System
ONAP	Open Network Automation Platform
ONF	Open Networking Foundation
OSGi	Open Service Gateway Initiative
OSI	Open Systems Interconnection Model
OSS	Operations Support Systems
OTN	Optical Transport Network
OTCC	Open Transport Configuration & Control
QEMU	Quick Emulator
PoC	Proof of Concept
PM	Performance Measurement
PTP	Precision Time Protocol
RFC	Request for Comments
RPC	Remote Procedure Call
Rx	Receiver
SES	Severely Errored Seconds
SBI	Southbound Interface
SDN	Software Defined Networking
SNMP	Simple Network Management Protocol
SSH	Secure Shell

Abbreviation	Meaning
SINR	Signal to Interference plus Noise Ratio
TCP	Transmission Control Protocol
TAPI	Transport Application Programming Interface
TR	Technical Recommendation
Tx	Transmitter
UAS	Unavailable Seconds
UI	User Interface
VAS	Value Added Services
VoA	Variable optical Attenuator
VPN	Virtual Private Network
VNF	Virtual Network Function
WTP	Wireless Transport Project
WTE	Wireless Transport Emulator
XML	Extensible Markup Language