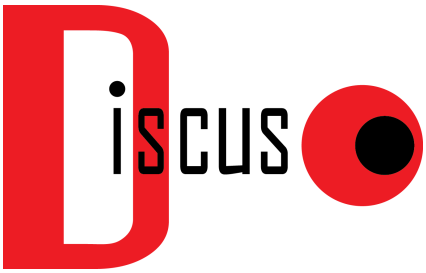


D6.5

Final report on the specification of the metro/core node architecture

Dissemination Level: PU

- **Dissemination level:**

  PU = Public,
  RE = Restricted to a group specified by the consortium (including the Commission Services),
  PP = Restricted to other programme participants (including the Commission Services),
  CO = Confidential, only for members of the consortium (including the Commission Services)

Abstract:

This document reports the final outcome resulting from WP6, including Task T6.1, "metro/core node architecture design" as well as description of the implementation work carried out in Tasks T6.2 "scalable switch fabric" and T6.3 "openflow-based control plan design and implementation" (as APPENDIX). A set of network services that are identified in the previous deliverable D6.1 have been updated during the course of the project, covering the applications to be carried out through both long-reach passive optical network (LR-PON) and optical flat core in DISCUS architecture. The final structure design of the overall DISCUS metro/core node architecture is presented, which contains the functions for different layers supporting the specified DISCUS network services. The functions included in DISCUS final metro/core node architecture are: Layer 1 optical switching and optical transport, Layer 1/2 optical line terminal towards access segment dealing with time and wavelength division multiplexing, Layer 2 multiprotocol label switching (MPLS) Access and Core switches as well as the corresponding control plane design. The deliverable presents the performance evaluation of the final DISCUS metro/core node architecture in terms of different aspects, namely resiliency, optical power budget, energy consumption, and cost. The results have shown DISCUS metro/core node architecture is a promising solution with respect to the considered aspects.

COPYRIGHT

The DISCUS Consortium consists of:

Participant Number	Participant organization name	Participant org. short name	Country
Coordinator			
1	Trinity College Dublin	TCD	Ireland
Other Beneficiaries			
2	Alcatel-Lucent Deutschland AG	ALUD	Germany
3	Coriant R&D GMBH	COR	Germany
4	Telefonica Investigacion Y Desarrollo SA	TID	Spain
5	Telecom Italia S.p.A	TI	Italy
6	Aston Universtity	ASTON	United Kingdom
7	Interuniversitair Micro-Electronica Centrum VZW	IMEC	Belgium
8	III V Lab GIE	III-V	France
9	University College Cork, National University of Ireland, Cork	Tyndall & UCC	Ireland
10	Polatis Ltd	POLATIS	United Kingdom
11	atesio GMBH	ATESIO	Germany
12	Kungliga Tekniska Hoegskolan	KTH	Sweden

Authors:

Name	Affiliation
Jiajia Chen	KTH
Marija Furdek	KTH
Lena Wosinska	KTH
Di Giglio Andrea	TI
Marco Schiano	TI
Laura Serra	TI
Julio Montalvo Garcia	TID
Thomas Pfeiffer	ALUD
Klaus Pulverer	COR
Adam Hughes	POLATIS
Nick Parsons	POLATIS
Marco Ruffini	TCD
David Payne	TCD
Giuseppe Talli	TYNDALL

Internal reviewers:

Name	Affiliation
Rohde, Harald	COR
Klaus Pulverer	COR
Juan Pedro Fernandez-Palacios Gimenez	TID

Due date: 31st July, 2015

Table of Contents

1 Introduction

The overall DISCUS architecture builds on the concept of Long-Reach Passive Optical Network (LR-PON) in the access, and a flat backbone partitioned into optical transparent islands, aiming to deliver ubiquitous high speed broadband access to all users in a an economic and sustainable way. For simplification of the telecommunication network architecture, DISCUS network nodes located at the edge of core segment directly connect LR-PON access segments, which are also referred to as metro/core nodes. These nodes are the only place to provide packet processing interfaces between access and core segments. In this way, access capability is enabled to the core edge and hence the metro network is eliminated, which is one of the major features of the DISCUS architecture. Meanwhile, it should be noted that in contrast to the network nodes in today's deployments, DISCUS metro/core nodes should be able to efficiently handle both user and core oriented applications, which introduces a great challenge on scalable and flexible node architecture design.

We have reported the preliminary metro/core node architecture as well as supported services in the previous deliverable D6.1. During the course of the project, the node architecture and the corresponding services have been updated. Therefore, the goal of this deliverable is to reports on final structure and design of the overall metro/core node architecture, including supported services, and interface to metro/core node control plane. Meanwhile, with consideration of different architectural aspects, we have also carried out performance evaluation in the node level with respect to resiliency, optical power budget, cost and energy consumption, which have demonstrated that the DISCUS metro/core node is a promising solution. The network performance by employing DISCUS metro/core node will be reported in the deliverables for the other work packages, such as WP7 deliverable D7.7 "Consolidated/integrated optical core design" and WP2 deliverable D2.9 "Final report on DISCUS architecture".

1.1 A brief overview of DISCUS Metro/Core Node

In this deliverable, we refer to the nodes located at the edge of core network in the DISCUS architecture as DISCUS metro/core nodes, since they have a similar architectural position as what are often called metro/core nodes in today's networks. In the overall DISCUS architecture, DISCUS metro/core nodes are the only place providing a packet processing interface between two different network segments (i.e., between long reach access and flat core transport network segments). It implies that DISCUS metro/core nodes should handle the traffic from/to access side (facing the LR-PON), and core side (facing the optical circuit switched based flat backbone) as well as interconnection between access and core segments.

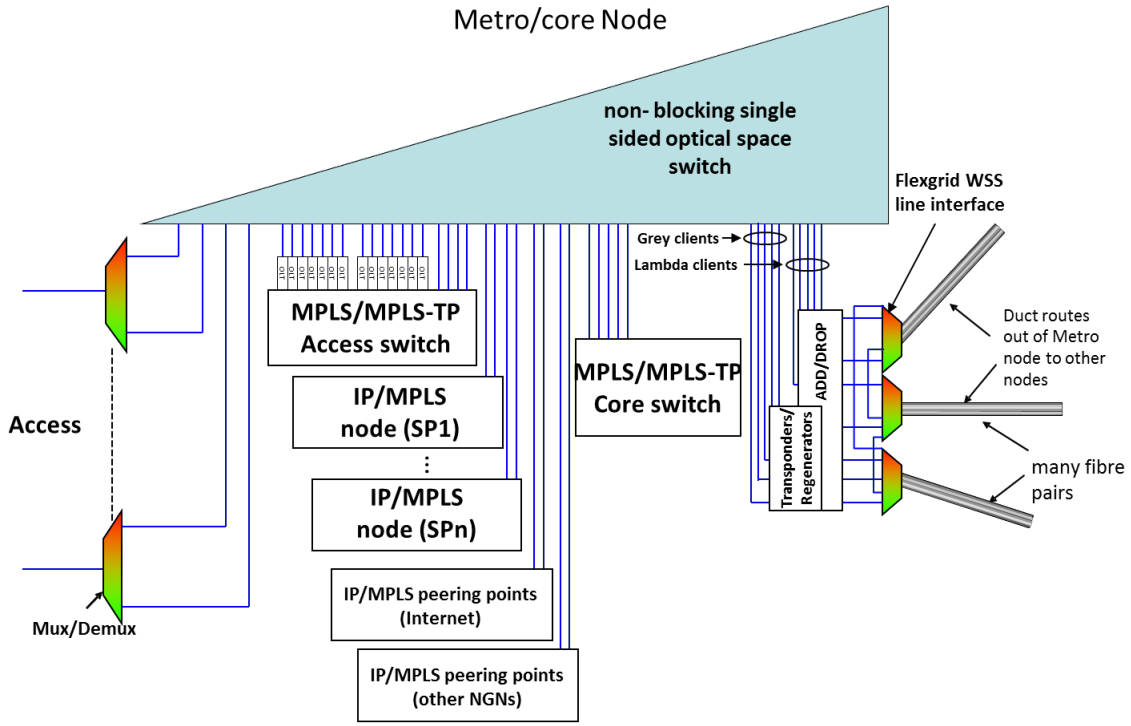


Figure 1-1: Overall DISCUS reference functional architecture, as reported in D6.1

A preliminary DISCUS metro/core node design has been presented in the first WP6 deliverable D6.1 (see Figure 1-1). The principle of the node structure is to have a transparent optical layer in the form of an optical switch with fibre links towards both access and core segments and electronic layers (e.g. Layer 2/Layer 3 switches) that can flexibly connect to. The involved optical switch does not necessarily distinguish ports for access or core segment. It also enables direct connection of optical paths between access and core networks. In this way, installation and operation can be simplified, i.e. freely connecting the optical switch to the interfaces of access/core segments.

In this document, based on the preliminary design of DISCUS metro/core node reported in D6.1, we present the updates of node architecture in different layers that have been carried out during the course of DISCUS architecture.

1.2 Outline of the deliverable

The remainder of this document is structured as follows. In Chapter 2 we specify the final list of network services that DISCUS architecture should support based on the preliminary one reported in D6.1. They are divided into two main categories: end user-oriented and core-oriented. It should be noted that in this deliverable we mainly focus on the updates that are performed during the course of the project and would not like to repeat everything that has been included in the previous deliverable D6.1. In Chapter 3 we present the overall DISCUS metro/core node design based on the initial work depicted in D6.1. It will be found the main structure of the overall DISCUS metro/core node is kept as reported in D6.1. The major efforts that we have put are to refine the design of each layer considered in the node architecture. The final DISCUS metro/core node includes the following four main components: 1) Layer 1 (L1) optical switching and optical transport, 2) Layer 1/2 (L1/L2) optical line terminal (OLT) towards access segment dealing with time and wavelength

division multiplexing, 3) Layer 2 (L2) multiprotocol label switching (MPLS) based Access switch and Core switch, and 4) the corresponding control plane. Chapter 4 provides the study in terms of the different aspects that include resiliency, optical power budget, energy consumption, and cost. For resiliency, we proposed to apply the concept of architecture on demand to enhance DISCUS metro/core node reliability in two ways, namely supporting fibre switching and enabling self-healing. For optical power budget, we update the evaluation considering multi-stage Clos optical switch architecture that is presented in this deliverable. Finally, conclusions are drawn in Chapter 5. Moreover, two appendixes are also attached, which are the summary of the implementation work for Task T6.2 "Scalable switch fabric" and Task T6.3 "OpenFlow-based control plane design and implementation".

2 Updated DISCUS Supported Network Services

The purpose of this chapter is to update the set of network services described in D6.1 that are delivered by the DISCUS metro/core nodes.

This scope can be properly achieved starting from a simplification of the Metro/Core (MC) node architecture developed in D6.1 that can be done according to the worldwide evolution on control plane architectures based on Software Defined Networking (SDN) solutions, as tackled in D6.3. In general, the SDN approach allows to separate the control plane from the data plane, at every layer of the OSI protocol stack. In particular, for the DISCUS Network Provider (NP) we can speak of Transport SDN (T-SDN), because it is applied at Layer1/2 transport levels, both on the access side (LR-PON and MPLS/MPLS-TP Access switch) and on the core side (MPLS/MPLS-TP Core switch and Photonic layer). For the Service Provider (SP) network we generically speak of SDN, because it is applied at Layer 2/3 levels on an IP/MPLS network composed by routers and servers. Therefore, the present evolution in SDN and T-SDN concepts do substantially remove any difference between the MPLS and the MPLS-TP approaches, because the control plane will be in both domains (i.e. the SP and the NP domains) separated by the data plane and the Label Switched Path (LSP) is provisioned in a connection-oriented manner, according to the packet transport (MultiProtocol Label Switching-Transport Profile MPLS-TP) approach, i.e. without the mediation of IP distributed protocols. Furthermore, as described in D6.3, the NP T-SDN is interfaced with the SP SDN, in order to create the end-to-end LSP and Pseudo-Wire (PW). For all these reasons, we can generically speak of MPLS LSP. This general concept can be applied to the DISCUS metro/core node architecture shown in Figure 2-1, **where the reference Layer 2 functional blocks are generically called, from now on, as MPLS Access switch and MPLS Core switch (highlighted in yellow)**. It is a major update introduced in this deliverable from the functional point view of the metro/core node architecture. The other updates of the node architecture are in details of each function, which are described in Chapter 3 of this deliverable.

In D6.1, the services supported by the DISCUS network have been divided into two categories:

- end user-oriented network services;
- core-oriented network services.

The former are the services provided to the final customers (either residential, business or mobile) through the LR-PON DISCUS access network, while the latter are the transport services delivered to the service providers through the DISCUS core network.

In this deliverable, **updates are envisaged especially for end user-oriented services**, in order to get aligned with customers' services evolution, including services already analysed in WP2 deliverable D2.4. Some updates are also necessary in order to better specify the services that need high bandwidth, available over a lambda of the Time and Wavelength Division Multiplexing (TWDM) and/or Dense Wavelength Division Multiplexing (DWDM) LR-PON.

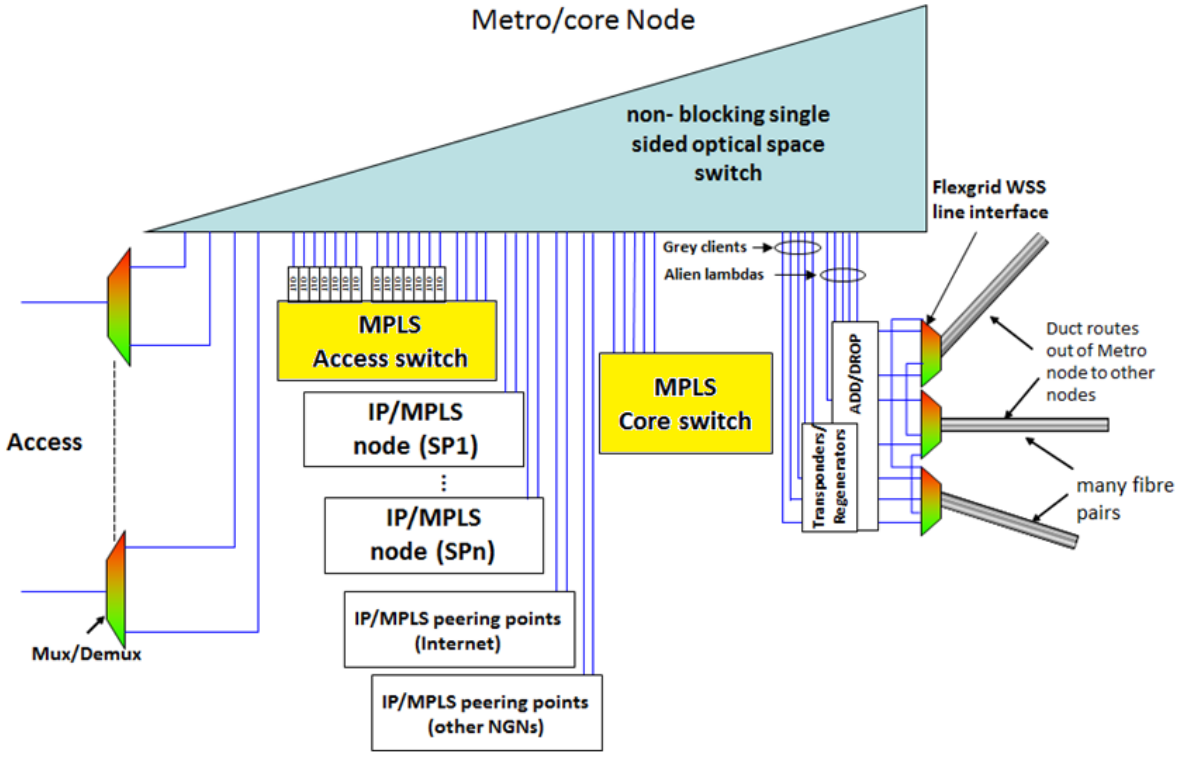


Figure 2-1: Update on Metro/Core node reference functional architecture

The services construction always refers to some open access models already described in D6.1, that are necessary to equally support service transport and delivery in a multi-Service Providers environment, offering to the final customers the freedom to choose between different end-user services offered by different SPs.

2.1 End user-oriented network services

Among different open access models described in D6.1, we focus on "Bitstream" open access and "Wavelength" open access. The bitstream open access model is based on service delivery at Layer 2 Ethernet/MPLS level, and it is supposed to be applied to all the services that do not need a huge amount of bandwidth per single customer. For example, most of the residential and small business customers as well as some large business customers and the 2G/3G mobile antennas, need a maximum bandwidth of 100-150 Mbit/s. The wavelength open access means that a dedicated wavelength, managed by the Service Provider (SP) itself, is used to serve customers needing 1 or 10 Gbit/s bandwidth, e.g., very large business, cloud computing customers and/or the 4G mobile e-node B.

As deeply described in D6.1, the bitstream approach means that the end user-oriented services are divided into three categories: E-LINE (p2p), E-LAN (mp2mp) and E-TREE (p2mp) services, each one with a proper mechanism of VLAN creation and mapping over MPLS. Each customer category (residential, business, cloud, mobile) needs different end user-oriented services, provided with these three service modes. All these cases are summarized in Figure 2-2 as a comprehensive representation of the end user services deployment through the LR-PON and the MPLS Access switch on a metro/core node, for a generic service provider.

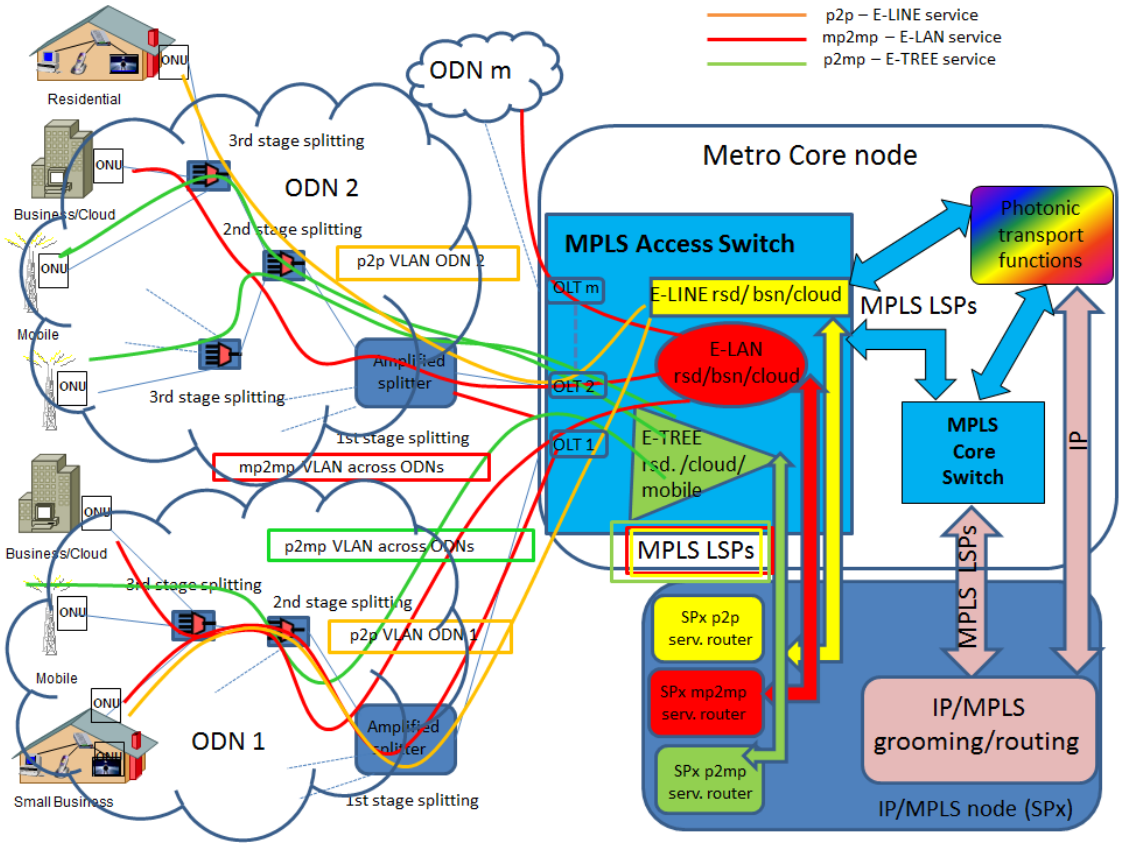


Figure 2-2: E-LINE, E-LAN and E-TREE services for residential, business, cloud and mobile customers through LR-PON and MPLS Access switch on a Metro/Core node for a generic Service Provider (ODN: Optical Distribution Network, ONU: Optical Network Unit, OLT: Optical Line Terminal)

The updates on end user services are listed in the following, according to the E-LINE, E-LAN and E-TREE service modes depicted above and in line with the description done in D6.1 and D2.4. Some of these new services, with their bandwidth needs, derive from the H2020 European Digital Agenda requirements.

The **E-LINE (p2p)** services, for each customer profile, can be classified in the following way:

- for *residential* customers**:**

  - Voice over IP (*VoIP*);

  - Internet services, i.e.:

    - *surfing*, *e-mail* exchange, *internet banking*, *small e-commerce* (1-2 Mbit/s bandwidth);

    - the *e-life* service named *e-health*, that consists in patient health records exchange among doctors and public institutions, digital pharmaceutical requests, hospital remote assistance and diagnostic at the patient premises (this last service requires 10 Mbit/s BW downstream);

  - Multi Media Services (MMS):

- *Video on Demand* (*VoD*), with today standard and high definition video(SD/HD, bandwidth from 2 to 5 Mbit/s) and future full HD (FHD, 10 Mbit/s) and ultra HD (UHD, 50 Mbit/s) video;
- *Video Communication* (*VC*), at low definition-LD (0,5 Mbit/s), SD (1 Mbit/s) and HD (2 Mbit/s);
- *on line gaming* (5 Mbit/s, both upstream and downstream);

- Pure Data Services (PDS): *peer-to-peer file sharing* (4 Mbit/s downstream);

- for *small/large business* and *Public Administration* (*PA*) customers:

- VoIP, VC and Internet , this last one also in terms of the *Internet of Things* (*IoT*), i.e. smart cities info, automotive (connected-car), vending machines and e-commerce services able to increase the user experience, for example "virtual goods shopping" (3D printing, virtual mannequin, 5 Mbit/s downstream bandwidth) or "virtual car showroom" (10 Mbit/s downstream).

The **E-LAN (mp2mp)** services, for each customer profile, can be classified in the following way:

- for *residential* customers:

- Internet e-life cloud service like the *e-learning* mediated by a Learning Management System platform, based on virtual classrooms/labs, interactive whiteboards, collaboration between teacher and students and among students (1-2 Mbit/s upstream and 5 Mbit/s downstream);

- for *small/large business* and *PA* customers:

- L2/L3 VPN, including *data-back-up* (5 Mbit/s both directions) and *multi-tenant HD VC* (10 Mbit/s both directions);

- for *cloud* services:

- all the services that need the mediation of Data Centres, as described in D6.1, including services for *e-Government* like *Digital PA*, *e-Health*, *e-School.*

The **E-TREE (p2mp)** services, for each customer profile, can be classified in the following way:

- for *residential* customers:

- *IPTV* service, as described in D6.1;

- for *large business* and *PA* customers:

- Internet of Things (IoT) like *Smart Cities*, *Safe Cities* (5 Mbit/s), *e-Tourism, Smart Metering* (for electrical networks, gas networks, etc., 1 Mbit/s bandwidth);

- for *mobile* services:

- as described in D6.1 for 2G and 3G.

Wavelength open access refers to the situation in which an SP wants to manage an optical wavelength in the LR-PON in a dedicated way at a physical level. In this sense, the wavelength open access will be offered by a LR-PON infrastructure provider as a wholesale service to another service provider.

In the bitstream open access, the owner of the wavelength and the owner of the end-user may be different, and the owner of the wavelength determines the service definition and service provisioning, because the wholesale service is offered using its equipment. Bitstream open access also allows to efficiently share the bandwidth of an optical channel between end-users of several service providers.

On the opposite, in the wavelength open access scenario, the same services previously described will be offered to end-users, but in this case the wavelength will be managed by the same service provider that owns the wavelength, and this service provider can define with total freedom the end-user services (or another wholesale services) and can configure and provision services in an independent way, because it uses its own network equipment.

A typical situation using wavelength open access can be a mobile network operator (MNO) using a dedicated wavelength or set of wavelengths to provide backhaul to its mobile stations using a TWDM-PON system over the LR-PON. Using this dedicated wavelength, the MNO can freely (and with total privacy) manage the Quality of Service (QoS) and the access speed of its mobile customers. If the MNO is vertically integrated with the LR-PON network/infrastructure operator of a Fixed Network Operator, fixed and mobile services may share the same wavelength resources obtaining some optical resources efficiency [1]. Nevertheless, the independent management of wavelengths for fixed and mobile services has operational advantages, allowing flexible and independent service configuration and provisioning. Due to the high bandwidth and specific latency requirements of LTE and LTE-Advanced radio access networks, these radio technologies fit very well the Wavelength open access approach.

Another relevant possibility for wavelength open access is to reserve specific wavelengths in the LR-PON to provide Point to Point services for one or various small/large business companies or for MNOs using centralized baseband processing (cloud RAN approach).

In the middle between bitstream and wavelength open access, a temporary approach named Virtual Unbundled Loop Access (VULA) is being considered as a regulatory approach in Europe, aiming to allow Other Licenced Operators the total flexibility for service definition to their end-users, even using the network equipment from another provider.

In DISCUS LR-PONs, Wavelength open access can be inherently achieved by employing wavelength division multiplexing.

The LR-PON infrastructure provider must control the wavelength usage allowed to each service/network provider, and must guarantee the proper operation of multi-wavelength systems in the same LR-PON. Equipment synchronization or coordination between service/network providers may be required to achieve this.

2.2 Core-oriented network services

As described in D6.1, these transport services involve the MPLS core switch and the photonic layer (as represented in Figure 2-1 and Figure 2-2). D6.1 has already given an exhaustive description of these network services, including client ports restrictions and line ports required, with the proper DWDM modulation formats necessary to support the different reaches. A final decision concerning MPLS Core switch application, in terms for example of client ports capacity and hub functionalities, will be given in WP7. In any case, Table 1 shows a reviewed list of network services. For MPLS Core switch, the resilience schemes reserved for packet transport layer are limited to “unprotected” and “restored” because we

assume that this layer provides its own resilience functions independently from the photonic layer.

Table 1 Network services

CLIENT LAYER	NETWORK SERVICE DESCRIPTION	RESILIENCE	INTERFACES
IP/MPLS SERVICE NODES AND PEERING POINTS	100 GE circuit connection	• Unprotected • Restored • 1:1 protected	On client side: • 100 GE • Lambda client On line side: • 32 Gbaud DP-16QAM single carrier (dual 100 GE client) • 32 Gbaud DP-QPSK single carrier • 32 Gbaud DP-BPSK dual carrier
	400 GE circuit connection	• Unprotected • Restored • 1:1 protected • Combined protection and restoration	On client side: • 400 GE • Lambda client On line side: • 32 Gbaud DP-16QAM dual carrier • 32 Gbaud DP-QPSK quad carrier
MPLS TRANSPORT (CORE) SWITCH	100 GE circuit connection	• Unprotected • Restored	On client side: • 100 GE • Lambda client On line side: • 32 Gbaud DP-16QAM single carrier (dual 100 GE client) • 32 Gbaud DP-QPSK single carrier • 32 Gbaud DP-BPSK dual carrier
	400 GE circuit connection	• Unprotected • Restored	On client side: • 400 GE • Lambda client On line side: • 32 Gbaud DP-16QAM dual carrier • 32 Gbaud DP-QPSK quad carrier
MPLS ACCESS SWITCH	100 GE circuit connection	• Unprotected • Restored • 1:1 protected • Combined protection and restoration	On client side: • 100 GE • Lambda client On line side: • 32 Gbaud DP-16QAM single carrier (dual 100 GE client) • 32 Gbaud DP-QPSK single carrier • 32 Gbaud DP-BPSK dual carrier
	400 GE circuit connection	• Unprotected • Restored • 1:1 protected • Combined protection and restoration	On client side: • 400 GE • Lambda client On line side: • 32 Gbaud DP-16QAM dual carrier • 32 Gbaud DP-QPSK quad carrier
ENTERPRISE CUSTOMERS (TRANSPARENT LAMBDA BASED	40 GE circuit connection	Unprotected on access side, on core side: • Unprotected	On client side 40 GE On line side depending on LR-PON constraints

SERVICES ACROSS ACCESS AND CORE)		• Restored	
	100 GE circuit connection	Unprotected on access side, on core side: • Unprotected • Restored	On client side 100 GE On line side depending on LR-PON constraints

3 Final DISCUS Metro/Core Node Design

This chapter includes a description of the final overall DISCUS metro/core node structure and design. Figure 3-1 shows an abstract view of functions as well as the associated interfaces to control plane that should be included in final DISCUS metro/core node architecture. It can be seen that the high-level view (i.e., the main elements of DISCUS metro/core node) are kept almost the same compared to the preliminary one reported in D6.1, in order to well accommodate the DISCUS supported network services. The minor difference is introduced to access and core MPLS switch, where the reference Layer 2 functional blocks are generically called (which is explained in the beginning of Chapter 2 to be in line with the supported services). During the course of the project, our major efforts have been put to refine the detailed design of each element. The recent updates are presented in this chapter.

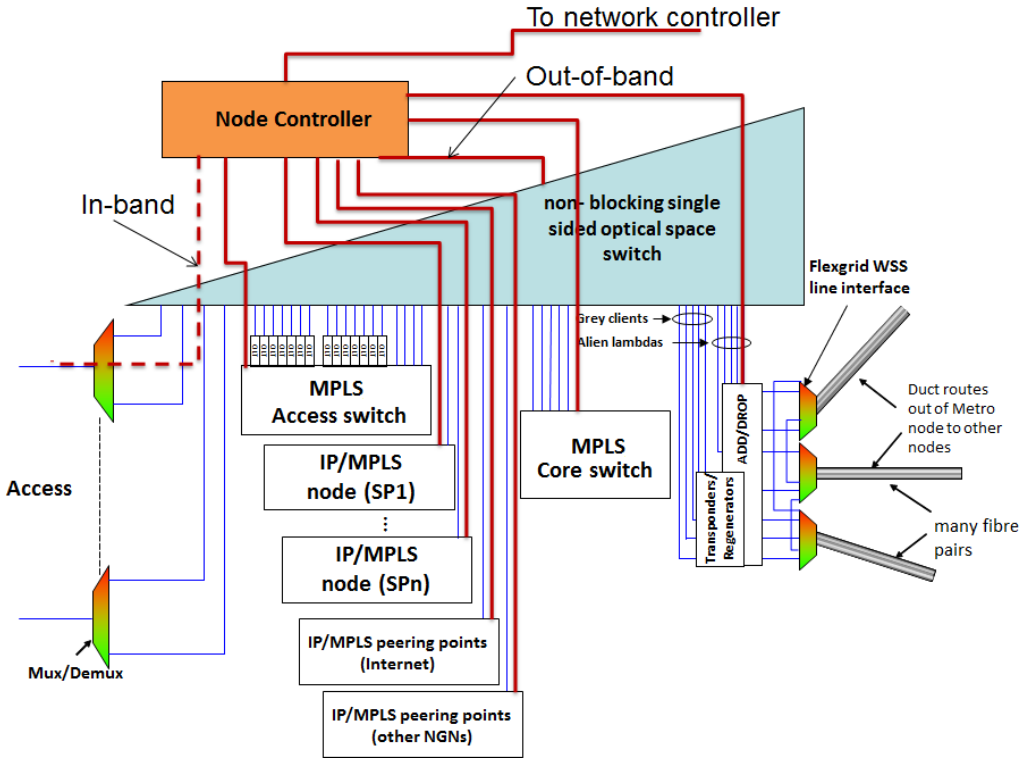


Figure 3-1: An abstract view of final DISCUS metro/core node architecture as well as the required interface for control plane

3.1 Layer 1: Optical switching and transport technologies

In this sub-chapter, Layer 1 (L1) function is concentrated. The recent updates on the considered technologies for optical switching and optical transport are presented. Similar as D6.1, it should be noticed that the L1 function at the Optical Line Terminal (OLT) dedicated to LR-PON is included in Sub-Chapter 3.2.

3.1.1 Optical Switching

The optical fabric of the metro/core node has been described in detail in D6.2 and D6.3. The purpose of this section is to summarise those discussions and to describe the work done since after D6.3.

3.1.1.1 Optical switch architecture

Using Polatis' [3] technology to provide a 192 port single sided optical crossbar switch, a single sided 12288 port switch fabric can be built in a folded Clos architecture [4].

Figure 3-2 shows the folded Clos architecture with g single sided edge switches each with g ports and m single sided centre switches, each with g ports. We partition the edge switches with m internal connections and n external connections, so $n+m < g$, and then Clos' theorem asserts that $m = 2n-1$. We define the total number of ports $N = ng$.

To have strictly non-blocking, we need to kee $3n - 1 < g$. So the maximum value for n is $(g+1)/3$, which gives us $N = g(g+1)/3$ and $m = 2(g+1)/3 - 1$ which constrains g to keep m a whole number. Using $g = 192$ and $p+1 = 192$, give us $2N = 12288$ ports.

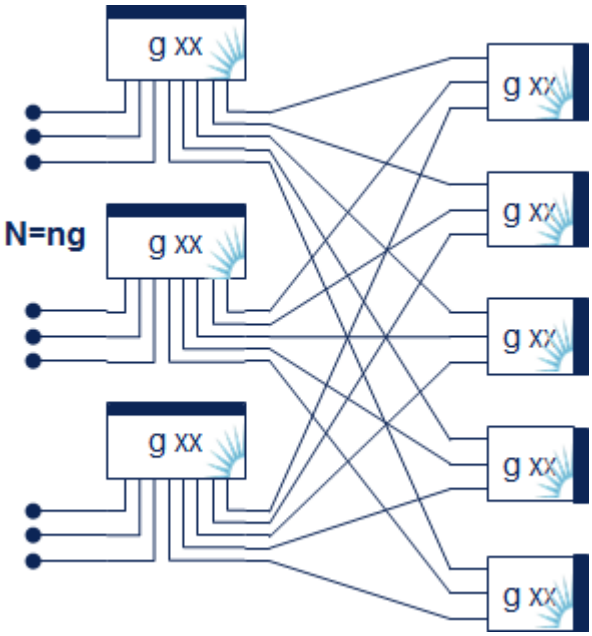


Figure 3-2: Strictly non-blocking single sided Clos switch composed by single sided edge and center switch matrix

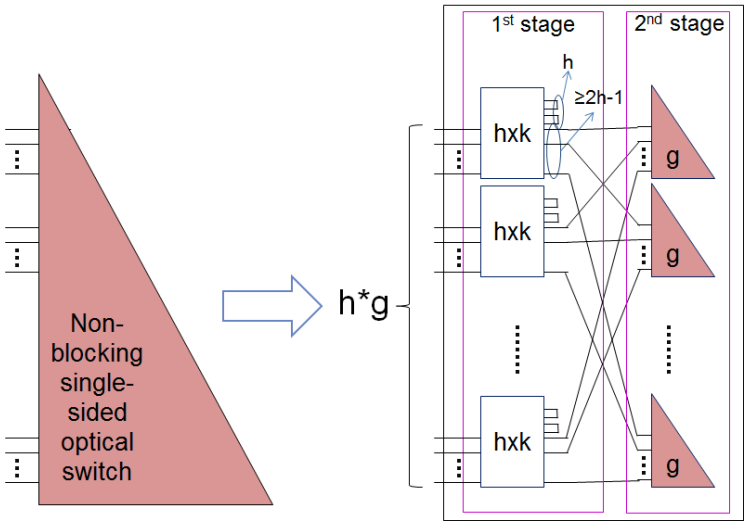


Figure 3-3: Strictly non-blocking single sided Clos switch composed by double sided edge switch matrix and single sided center switch matrix

Figure 3-3 shows an alternative solution of the folded Clos architecture with *g* double sided edge switches each with *hxk* ports and *n* single sided centre switches, each with *g* ports. We partition the edge switches with *h* internal connections (each pair is connected, see the figure above) and *n* external connections (to connect centre switch). Clos' theorem asserts that $n \geq 2h-1$ in order to have strictly non-blocking. We define the total number of ports $N = h*g$. Considering the total number of ports of each switch matrix is up to 192, $N=12288$.

Hitless scalability was also addressed in D6.2. There are two standard approaches to scale a strictly non-blocking Clos switch, "edge fill" and "centre fill". With edge fill, the centre stage is completed initially, and the incomplete edge gains switches to scale. With centre fill, the edge stage is completed initially, and the incomplete centre gains switches to scale. With interchangeable centre and edge switches, it is clear that starting with the smaller centre stage filled and the larger edge stage incomplete gives the lower initial cost, although the final cost remains constant.

3.1.1.2 Control of the Optical switch

The OpenFlow protocol was proposed in D6.2 as an open SDN standard that would deliver a consistent external interface, while allowing the control architecture to change over the lifetime of this project. In D6.3, three functions were identified for the optical switch:

1. Connect input to output port
2. Read power at given port
3. Trigger alarm for power below threshold

The OpenFlow interface available for the optical switch is built against the OpenFlow v1.0, using circuit switching extensions (v0.3). The circuit switch extensions enable Function 1. Power for a given port is not a standard feature, and needs to be implemented as Vendor extensions to the port_status message. The alarm state for a port can be signaled using the port state field (OFPSS_LINK_DOWN). With the future implementation of OpenFlow 1.4, all three of these functions are available as standard features.

In D6.3 a multi-module router was proposed. This software component computes paths through the Clos architecture, between two external connections. In D6.3 two possible locations for the router were mentioned in the context of SDN:

1. the router is part of the Polatis switch and the switch presents a single SDN agent on a "shelf controller" with only the external connections,
2. the switch presents one SDN agent for each module in the switch, presenting internal and external connections, and the router is an application in the SDN runtime used by the SDN controller to compute the paths. The SDN controller then configures the path using the multiple SDN agents.

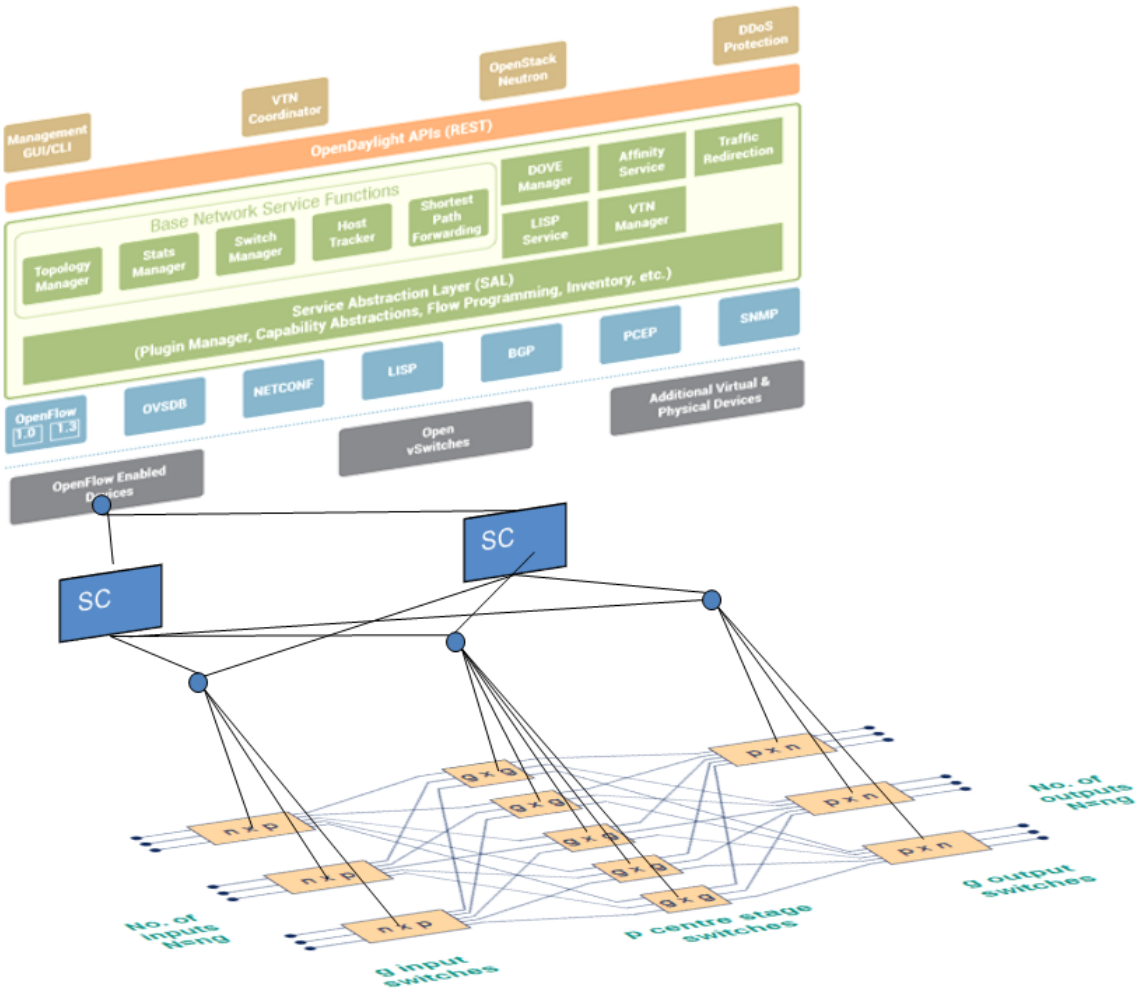


Figure 3-4: Shelf controller architecture for a multi-module switch fabric

In the context of this project Polatis have implemented the first option, with the routing algorithm part of the "shelf controller" software (see **Figure 3-4**), because it makes the Clos switch fabric (the component switches plus shelf controller) become more easily controlled SDN resource.

The multi-module router described in D6.3 has been implemented and tested as a component of the switch fabric, which has been reported in Milestone 19.

A rudimentary 3-stage optical switch is realized from seven available Polatis Optical Cross Connect (OXC) modules to prove the embedded routing algorithm and shown to perform as predicted. The synthesised strictly non-blocking Clos architecture and routing algorithm work as designed, allowing non-blocking connections between endpoints. Measurements of insertion loss and switching speed are in line with expectations (4dB of insertion loss and 20ms of switching time). The routing algorithm has been run successfully in simulation mode for 100x100 ports. Further work is in progress to test the algorithm against fabric sizes up to 12,000 single- and double-sided switch ports, and to test the control of larger numbers of components.

A 4x4 switch was built using seven available component switches, with sizes ranging from 24x24 to 96x96. For demonstration purposes, we modified configuration files to create two input layer switches of 2x3, three center stage switches of 2x2 and two output stage switches of 3x2. The positions of the active fibers within the switch were chosen to make the physical

connection process and the software configuration (pathmap) as simple as possible. The switch functioned as a fully-connected three stage 4x4 switch. Insertion loss data and switching speed data was gathered for a representative set of cases.

The switching speed results showed connection times similar to those for a single module switch (measured 9 ms to 20 ms as shown Figure 3-14 below). In one or two cases the switching trace showed a clear discontinuity, indicating that two of the component switches are completing the transitions at slightly different times. It is possible that in some of the measurements, only one or two of the switching components are actually switching because the routing algorithm preferentially uses the paths it finds first.

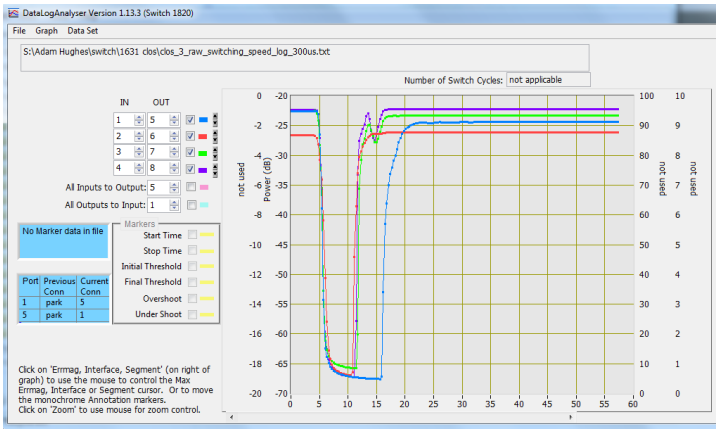


Figure 3-5: Light levels when switching all four paths

3.1.2 Photonic switching and add/drop functionalities supporting core network services

The metro/core node transport technologies and architectures have already been identified and discussed in D7.1 (photonic switching functionalities) and in D6.1 (optical line interfaces) for supporting the core network services. In this sub-chapter we revise those architectural choices based on the latest information on reference networks size and dimensioning data. The proposed metro/core node photonic switching and add/drop functionalities dedicated to core network are based on Wavelength Selective Switches (WSSs) and Multicast Switches (MSs) as shown in Figure 3-6.

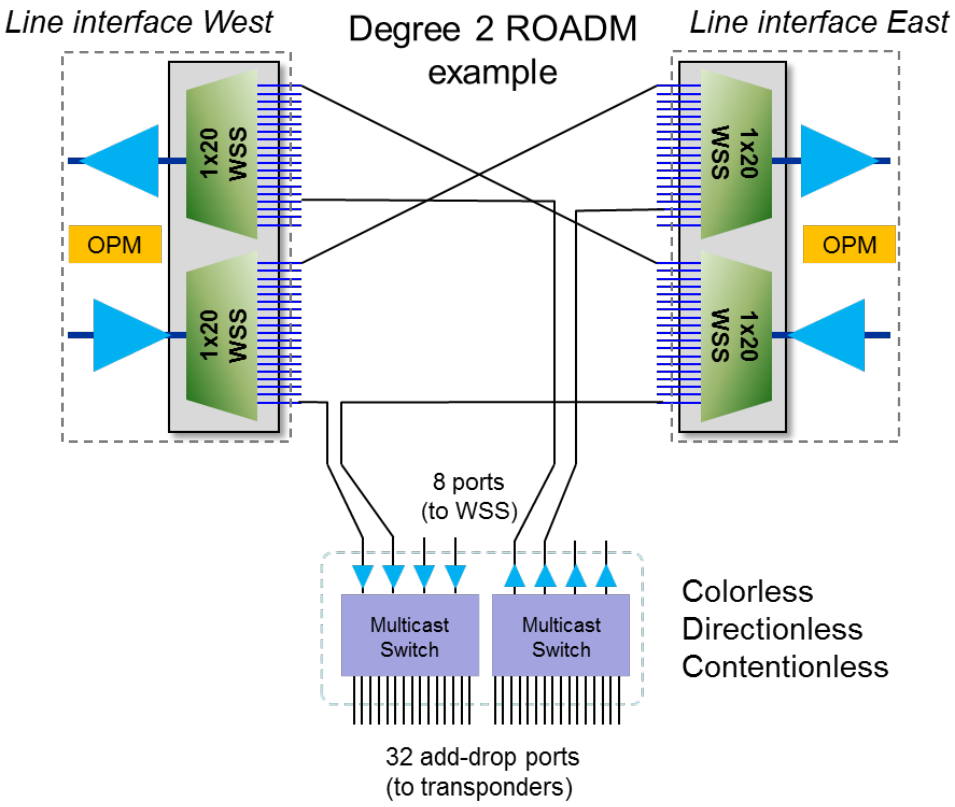


Figure 3-6: Example of Degree 2 ROADM of the photonic switching and add drop functions implemented by Wavelength Selective Switches and Multicast Switches. Only one add-drop block with 32 ports is shown in the picture, but additional blocks can be added if more add-drop ports are needed

This simple architecture has been selected among the ones proposed in D7.1 for the following reasons:

- It provides Colorless Directionless Contentionless (CDC) add/drop (A/D) thus enabling optical restoration without any wavelength blocking constraint in the end nodes;
- It provides flex-grid spectrum management functions improving the efficiency of spectrum utilization;
- The number of WSS and MS ports is compatible with the tentative dimensioning of the largest metro/core nodes of one of the most challenging DISCUS reference networks.

The last point deserves a further comment. A UK reference core network with 73 nodes is currently under study and some preliminary dimensioning data are available. In such a network the larger node has degree 7 and the number of add-drop ports required in some nodes is very large. Therefore we have decided to use 8x32 MS which recently have become commercially available.

From preliminary dimensioning results it also appears that in the most challenging traffic scenario, an optical bandwidth broader than the C band is required (see deliverable D7.7). Therefore we plan using WSS components with 100 nm bandwidth that can be built by three conventional WSS each covering 1/3 of the bandwidth [5].

The cost model of these components is taken from IDEALIST project 90 [6] and is shown in Table 2.

The cost model and power consumption of these components are estimated scaling the cost and power consumption of the corresponding C band components as shown in Table 2.

Table 2 Cost and power consumption model of the photonic components

Item	Cost (ICU) (*)	Power consumption (W)	Number of elements in MC nodes
Dual WB WSS 1x20/20x1 (**)	1.5	60	1 for each line interface
Dual Multicast Switch 8x32 (**)	1.3	45	1 for each group of 32 transponders (***)
WB Raman Amplifier (**)	0.8	100	2 for each line interface

(*) the Idealist Cost Unit (ICU) is the reference cost unit of the IDEALIST [6] cost model and corresponds to the cost of a 100 G coherent transponder

(**) this component is not included in IDEALIST cost model and its cost has been estimated by TI

(***) additional components are required if the number of add-drop ports is larger than 32

The cost and power consumption model of the boards actually used in core network equipment that incorporate the described components is shown in Table 3.

Table 3 Cost and power consumption model of the photonic equipment boards

Item	Cost (ICU)	Power consumption (W)	Notes
Line interfaces	3.1	260	1 dual WB WSS and 2 WB Raman amp.
Line amplifier	1.6	200	2 WB Raman amp.

3.1.3 Stacked OXC architecture

Another central result coming from early dimensioning studies is that, even using wideband components, the number of ports of 1x20 WSS is not sufficient to accommodate all add-drops and lines in many of the network nodes. Therefore in those nodes we will adopt the stacked OXC architecture presented in [7]. This architecture is shown in **Figure 3-7**.

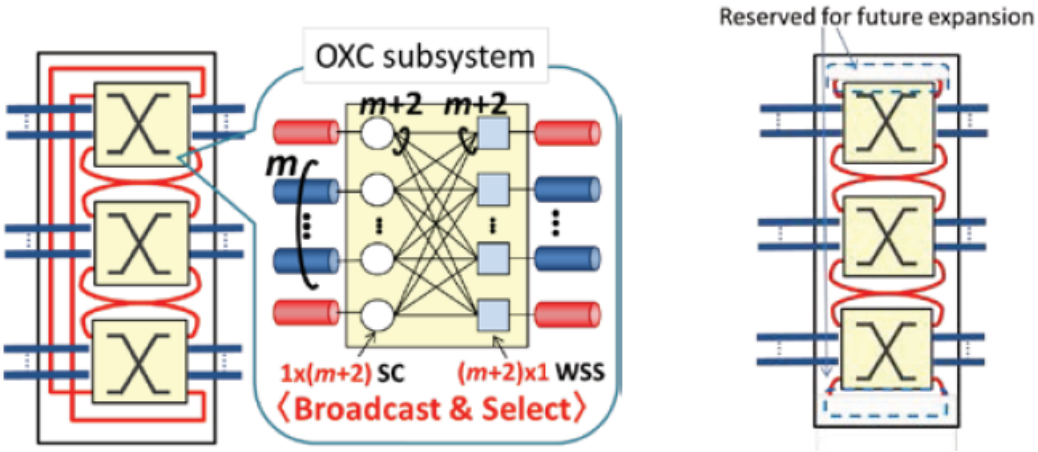


Figure 3-7 OXC stacked architecture [7]

3.1.4 Bandwidth Variable Transponders

In deliverable D6.1 we have proposed three kinds of Line Interfaces (LIs): a single carrier LI, a quadruple carrier partially configurable LI and a quadruple carrier fully configurable LI. These kinds of LIs are commonly called Bandwidth Variable Transponder (BVT) or Sliceable-BVT (S-BVT). The terminology will be used in the following.

While this choice remains valid, it is worthwhile reducing the kinds of S-BVT to two just to ease the operation and maintenance activities and reduce spare parts. Table 4 shows the characteristics of the two selected S-BVT:

Table 4 Characteristics of S-BVTs for the DISCUS core network and the related cost and power models

1. Single-carrier BVT

No. of optical carriers	No. of 37.5 GHz slots	Configurations (services)	Client signals	Modulation format	Reach (km)	Spectral Efficiency (bit/s/Hz)	Cost (ICU)(*)	Power consumption (W)
1	1	Single 40 G	1x40GE	DP-BPSK	2430	1.067	0.95	140
		Single 100 GE	1x100GE	DP-QPSK	1170	2.667	1	150
		Dual 100 GE	2x100GE	DP-16QAM	500	5.333	1.1	160

(*)the Idealist Cost Unit (ICU) is the reference cost unit of the IDEALIST cost model and corresponds to the cost of a 100 G coherent transponder

2. Quadruple-carrier S-BVT

No. of optical carriers	No. of 37.5 GHz slots	Configurations (services)	Client signals	Modulation format	Reach (km)	Spectral Efficiency (bit/s/Hz)	Cost (ICU) (**)	Power consumption (W)
4	4	Single 400 GE	1x400GE	DP-QPSK	1170	2,667	3	440
		Twin 400 GE	2x400GE	DP-16QAM	500	5,333	3.2	450
		Twin 100 GE	2x100GE	DP-BPSK	2430	1,333	3	440

(**) estimated by Telecom Italia

The single carrier BVT provides 40, 100 and 2x100 GE services just changing the modulation format. The reach varies accordingly, while the cost remains almost the same, minor changes being due to the cost of client grey interfaces.

The quadruple carrier S-BVT provides 100, 2x100 and 2x400 GE services configuring the modulation format and carriers grouping. This S-BVT is sliceable in the sense that the four carriers can be routed independently on the network and therefore they can support multiple traffic demands with different destination nodes. The cost has been estimated based on the conservative assumption that photonic integration technologies will allow a 25% cost reduction compared with conventional discrete components technology.

Depending on the bandwidth of a given traffic demand and on the length of its associated light-path, the most appropriate BVT and S-BVT will be selected based first on the reach and second on the cost per Gbit/s specific of each configuration.

3.2 Layer 1/2: Optical Line Terminal

In deliverable D6.1, OLT architecture aspects for Wavelength Division Multiplexing (WDM) based PONs have been described for both TWDM and DWDM flavors, which exhibit a lot of commonalities. These descriptions remain valid and are not repeated here.

WDM-PON endpoints need to be managed from the OLT, which has been standardized recently as part of the ITU-T G.989 suite of standards. The transport of management information, however, differs between approaches.

For TWDM, the transport of management content is being accommodated by management channels that had been defined already for previous system generations, such as Gigabit-enabled PON (GPON) and XGPON1.

For DWDM, the situation is different, because the payload is not necessarily terminated by the PON, but passed through the metro/core node transparently. This is intended, e.g., for mobile fronthaul transport, in order to avoid any added latency. In such cases, an Auxiliary Management and Control Channel (AMCC) is defined to convey wavelength assignment, allocation information and OAM data, and to enable automated ONU tuning and monitoring. The AMCC is added to each individual wavelength in both, downstream and upstream direction.

Two different physical-layer implementations of the AMCC exist, determining the way the AMCC content is transported over the physical channel: *transparent* and *transcoding*. The difference of the two methods is the way the AMCC content is transported over the physical channel, while the content of the AMCC remains independent from the transmission method.

For the case that the DWDM system has to transparently transport a payload bitstream, without terminating any part of its frame structure, the AMCC has to be added to the payload at the same wavelength with only minor interference of the AMCC and the payload data. This case is called *transparent AMCC*. The framed AMCC data is added to the payload with low-modulation-index baseband over-modulation (i.e., a low-frequency amplitude modulation which is superimposed to the (amplitude) modulation of the payload bitstream). The principle is illustrated in Figure 3-8 below.

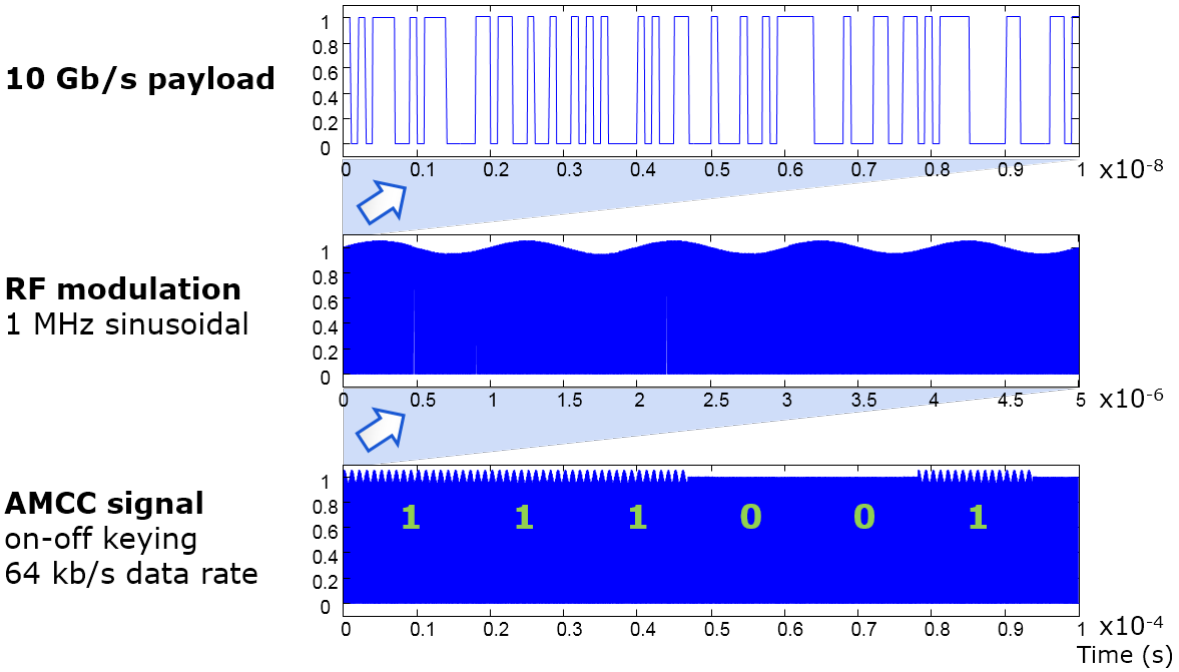


Figure 3-8: Low frequency modulation for AMCC

Two implementations of the transparent AMCC are currently under discussion. These are based on baseband over modulation and RF (radio frequency) pilot tones, respectively, and contrasted in Figure 3-9.

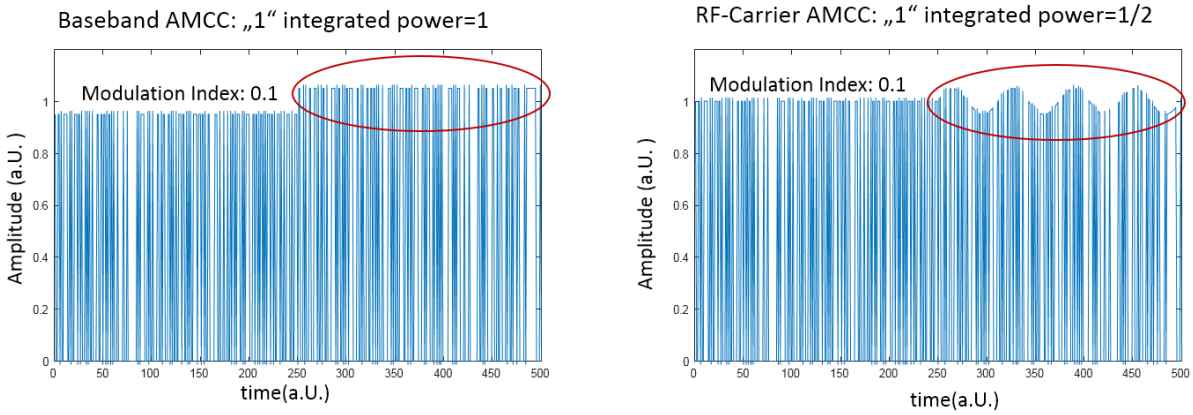


Figure 3-9: Baseband AMCC vs. RF-carrier AMCC

With baseband overmodulation, the framed AMCC data is amplitude-modulated on top of the payload bitstream as a baseband signal. It can be detected with a simple threshold detector after low-pass filtering of the receive signal. Currently, low modulation index of ~10% is discussed to guarantee penalties on the payload data signals of <0.5 dB. Further, overmodulated AMCC bit rates of 150 kb/s are discussed and have been demonstrated.

The second implementation of the transparent AMCC is based on the use of a low-frequency RF pilot tone. At the transmitter side, the client signal is mixed with the management and control channel in a low-frequency domain so as to offer transparent AMCC with minor interference between the AMCC and the payload data. The modulation depth of the management and control channel will be very shallow: it will be optimized not to harm the quality of the client signal. At the receiver side, the management and control signal is demultiplexed after the detection of the optical signal.

In case that the DWDM system is allowed to modify the framing, a *transcoding* approach can be used to make room for the management information within the framing structure. The approach is based on reducing the coding overhead while retaining the original bit rate, e.g. by moving from 8b/10b to 32b/34b. The space gained by this transcoding is made available for transporting the AMCC data (see Annex K of G.989.3).

The two AMCC implementations are compared in Figure 3-10:

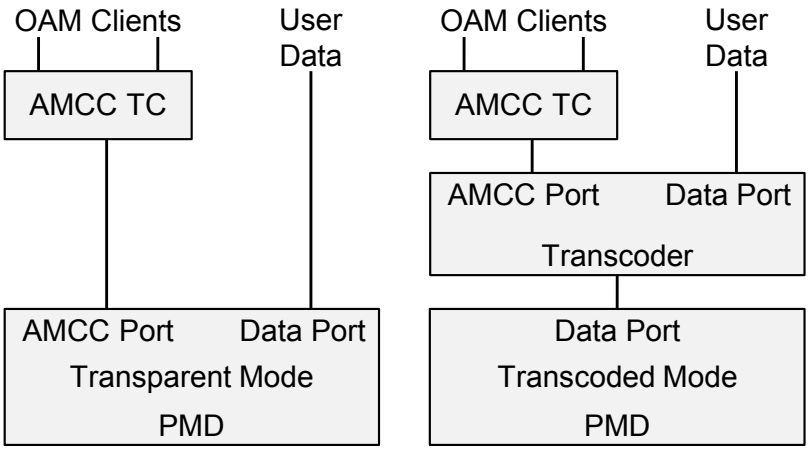


Figure 3-10: Relationship of the AMCC TC layer with other system functions for the transparent (left) and transcoded (right) options (from ITU-T G.989.3)

3.3 Layer 2: MPLS switches

This chapter describes the final design of the MPLS switches inside the DISCUS metro/core node architecture, with the functionalities required in order to support the network service models described in Chapter 2.

As already described in D6.1, for a network provider the main requirements are for connection-oriented transport capabilities that can be achieved with the MPLS-TP (Packet Transport) framework. Its main characteristic is to have the control plane separated by the data plane. On the contrary, the MPLS framework (developed for L2/L3 switches and routers composing the service provider network) foresees an IP distributed control plane that does not allow to create a Label Switched Path (LSP) with connection-oriented characteristics. As discussed in D6.1, this approach represents also the main evolution foreseen for next generation PONs. For these reasons, in D6.1 we have considered hybrid equipment called MPLS/MPLS-TP access switch and MPLS/MPLS-TP core switch, because they had to support at the same time the MPLS-TP core-oriented network services inside the NP DISCUS network and the MPLS end-user oriented services between the OLT and the SP node.

As explained in Chapter 2 of the present deliverable, the evolution of the SDN control plane concept allows now to generically re-name these equipment as MPLS access switch and MPLS core switch, because there will be no more substantial difference between the MPLS and MPLS-TP approaches, in the sense that the DISCUS T-SDN control plane orchestrates, together with the service provider SDN control plane, the creation of an end-to-end LSP in a connection-oriented manner, according to the packet transport solution, both in the NP and SP domain.

It should be noted that MPLS grooming is preferred here rather than OTN grooming because of native packet nature of data traffic, collected by LR-PONs. Ethernet framing is used since it is cheaper. Standards over 10Gbit/s implement Forward Error Correction (FEC) over Ethernet frame. In the case using appropriate modulation format, FEC usage might be mitigated. MPLS gives LSP that in certain cases traffic can reach huge bit-rate. Then S-BVTs are used to carry this traffic over multiple carriers.

Regarding the access and core switches design, first of all we must consider the MPLS protocol stack needed to support the end user services all over the DISCUS network (i.e. from the SP node border to the ODNs of the LR-PON), as already described in D6.1 and D6.3 deliverables. To summarize, the MPLS LSP is created inside the SP network and on the access switch connected to the Optical Distribution Networks (ODNs), thus identifying the SP and the access switch connected to the SP customers. Furthermore the VLAN, that in the specific case matches one-by-one with the MPLS Pseudo Wire (PW), identifies the service for each SP and for each OLT port. Therefore, the access switch manages VLANs, PWs and LSPs, while the core switch only manages the LSPs.

The detailed functional blocks design of the MPLS access switch is represented in **Figure 3-11**. The picture is the integration into the Access switch of the OLT and NT cards functionalities, as described in D6.1 for what concerns their evolution towards MPLS that properly matches the access switch design done in the same deliverable. VLANs from/to ODN are mapped/de-mapped into MPLS PWs on the OLT card, that has switching capabilities both at Ethernet level (MAC addresses and VLANs) and at MPLS PW level (for traffic switching between ODNs belonging to the same OLT card). All the OLT cards inserted into the access switch are connected by an electrical backplane to the Network Terminal (NT) card, where MPLS PW switching is available (for traffic switching between ODNs belonging to different OLT cards) as well as MPLS LSP switching (after LSP encapsulation/de-capsulation) to/from the SP service node or to/from the MPLS core switch or directly to/from the photonic layer.

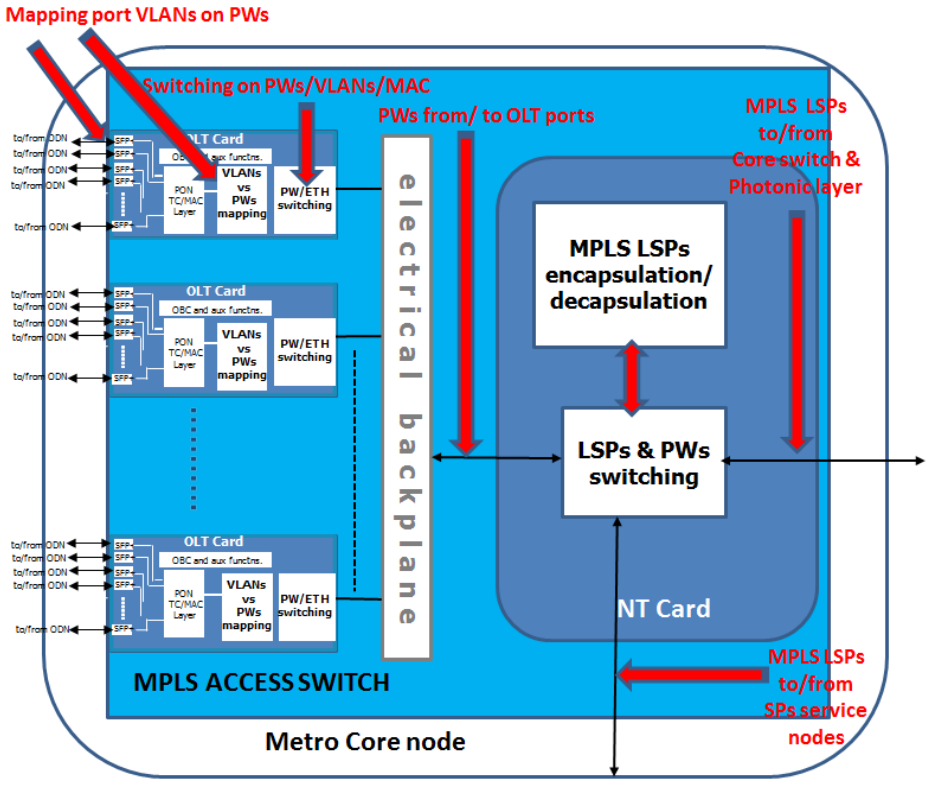


Figure 3-11: MPLS access switch detailed design in terms of functional blocks on OLT and NT cards

Considering the end user services implementation inside the access switch, Figure 3-12 illustrates the case in which the SP service edge routers (i.e. Broadband Remote Access Server BRAS, Provider Edge PE, etc.) are directly connected to the access switch, i.e. the case in which the SP customers are co-located with these edge routers. We can see that the E-LINE, E-LAN and E-TREE services (as described in D6.1) are linked to the corresponding PWs and the VSI (Virtual Service Instance) is used to perform PW switching. In particular, a hierarchical PW switching is available either on the OLT card (if switching involves customers attached to different ODNs connected the same OLT card) or on the NT card (if switching involves customers attached to ODNs connected to different OLT cards).

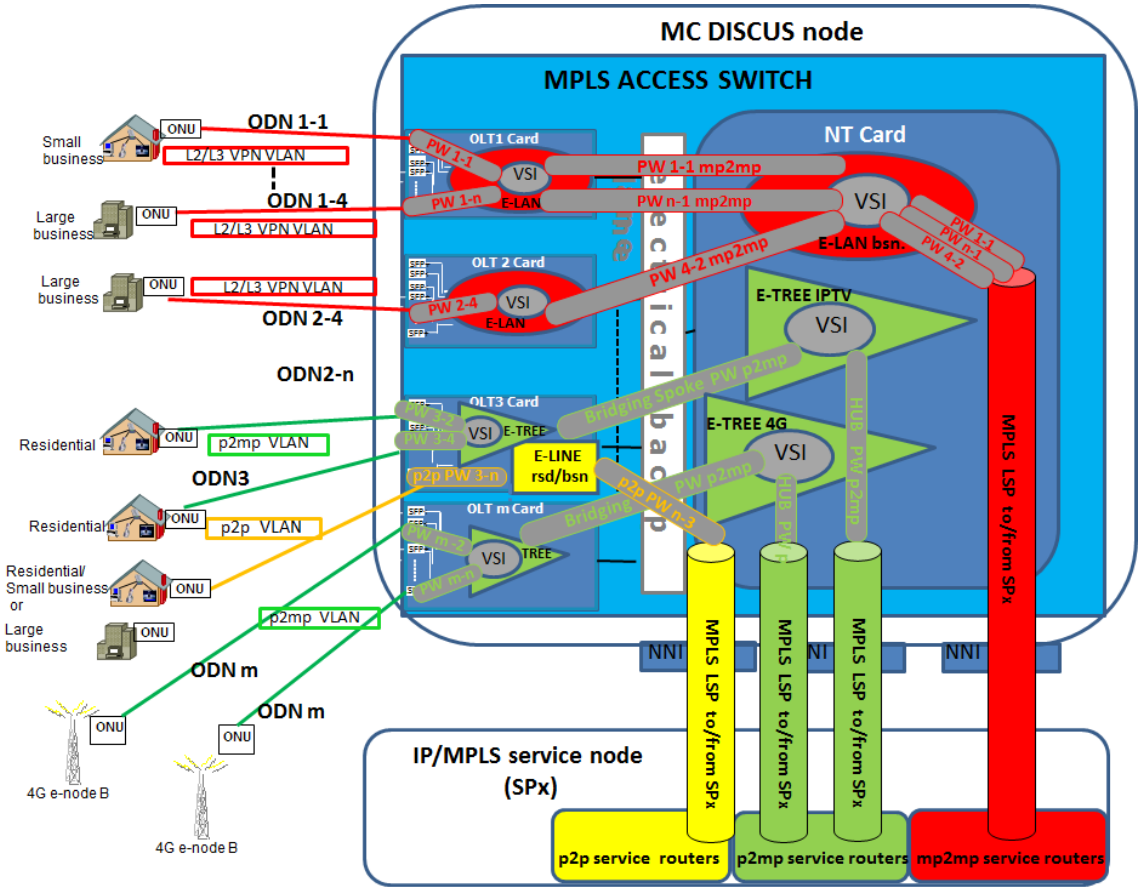


Figure 3-12: End user oriented network service support on access switch: 1) Ethernet VLANs encapsulation over PWs; 2) PWs switching for E_LAN and E-TREE services mediated by VSI; 3) transport over MPLS LSPs to SP edge service routers

Concerning the MPLS core switch design, we must consider that the core switch is used for DISCUS core oriented network services that need packet aggregation before transmission over the photonic layer. In particular, the core switch is useful in two cases: 1) for traffic aggregation and transmission of SP customers not co-located with their SP edge routers and 2) for aggregation and transport support of the IP/MPLS grooming and routing functions present in the SP node.

Figure 3-13 illustrates the design of the core switch that manages the two core-oriented network services described above.

For the first network service, the core switch is used to aggregate the low bandwidth traffic coming from the access switch. The MPLS LSPs of the access switch are connected to the 10 Gbit/s or 40 Gbit/s client ports on the core switch. Here these LSPs are switched and groomed into higher bandwidth (100 Gbit/s or 400 Gbit/s) line ports, for packet core transmission over the photonic layer. Moreover, the figure shows that the remote traffic delivery to the SP edge router is performed by the core switch itself, in order to save client interfaces on the access switch and in order to balance the work load between access and core switches.

For the second network service, i.e. for transport support of the SP grooming and routing functions, we can say again that the Core switch is useful in order to apply grooming of lower bandwidth traffic (10/40Gbit/s) into 100/400 Gbit/s packet core traffic.

Otherwise, as shown in Figure 3-13, for high bandwidth LSPs (100/400 Gbit/s) from the Access switch or from the SP grooming and routing function, traffic can be directly managed by the Photonic layer.

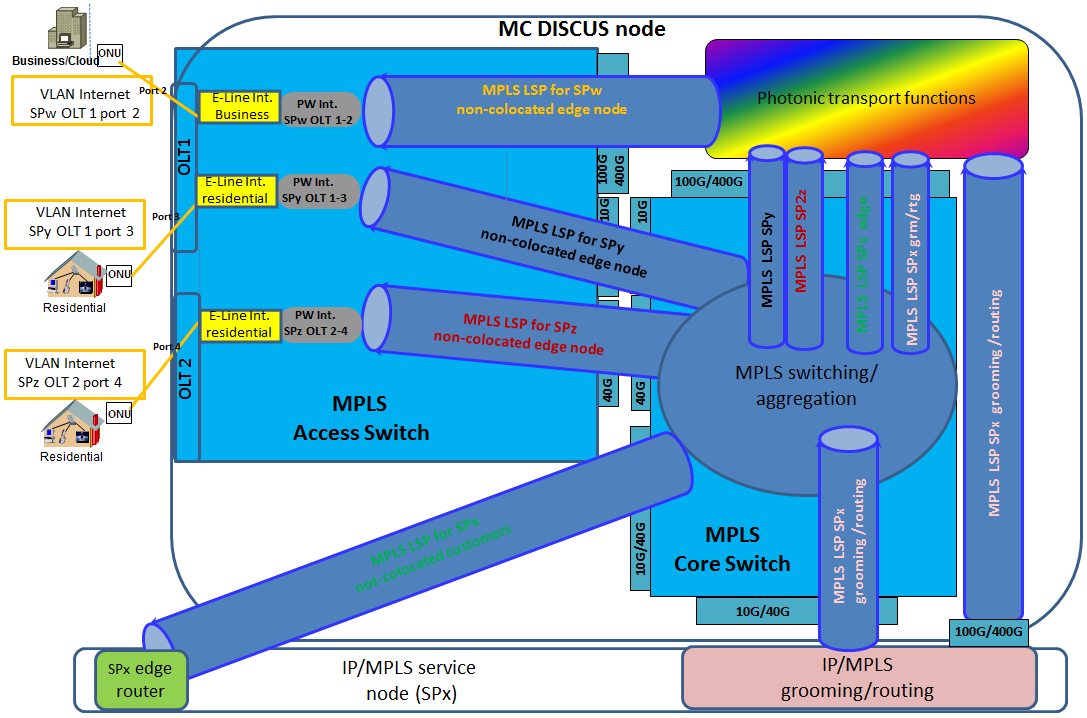


Figure 3-13: Core-oriented network services implementation on access switch and core switch

Operation, Administration and Maintenance (OAM) over the DISCUS network is of paramount importance in order to allow fast protection mechanisms. In recent years, IETF standardization on MPLS, and its cooperation with ITU-T on MPLS-TP requirements, led to OAM tools both in the MPLS and in the MPLS-TP domains which work at data plane level and support proper protection mechanisms for both of them.

In the context of the SDN approach developed in the DISCUS project, the network provisioning should include some OAM configuration on PWs and LSPs but, in order to have fast failure detection, the SDN control plane should be supported by OAM data plane

protocols. In particular, considering the DISCUS service model, the OAM & protection implementations could be developed as described in the following.

Regarding *OAM monitoring*, the following functionalities must be taken into account.

- *End-to-end OAM on PW* from the OLT access side to the SP service node: this OAM can be properly managed with VCCV (Virtual Circuit Connectivity Verification), according to MPLS IETF standard *RFC 5085*. Since it is an inter-domain PW OAM, coordination between NP and SP domains is needed, supported by the DISCUS and the SP SDN control planes;
- *End-to-end OAM on LSP* and *OAM on sub-LSP in DISCUS domain:* the final evolution towards a generic MPLS LSP can lead to a common OAM (developed starting from the 2 possible standardized implementations described below) for the end-to-end LSP, with a Sub-Path Maintenance Elements (SPMEs) Tandem Connection Monitoring (TCM) for LSP OAM in the DISCUS domain. The two OAM mechanisms could be the following ones:
  - considering that the SDN approach leads to explicitly routed MPLS LSPs, the more obvious OAM choice would be an OAM inspired by MPLS OAM ITU-T standard *Y.1711* and MPLS-TP ITU-T standard *G8113.1/2* (based on Continuity Check-CC, Connectivity Verification-CV, Fast Failure Detection – FFD, etc.); in this case SDN control planes in DISCUS and SP network should manage the different OAM labels (Alert Label 14 for MPLS and G-ACH Label 13 for MPLS-TP) presently established by *RFC 5586* & *RFC 6423*, *RFC 5654* and *RFC 5860*;
  - another OAM approach could be based on MPLS-BFD (Bidirectional Forwarding Detection) signaling, that has been developed for MPLS and MPLS-TP (see *RFC 4379* &*RFC 5884* as well as *RFC6428*). Also in this case the DISCUS and SP SDN control planes should coordinate network updates into their databases, after failure detection at data plane level, in order to properly support protection.

Regarding *Protection*, D6.3 has foreseen PW and LSP redundancy in order to support network protection on a back-up OLT and a back-up MC node. PW and LSP redundancy are useful for end-to-end protection, from the ODN side to the SP active or stand-by router, as schematically represented in Figure 3-14. This means that an inter-domain coordination between DISCUS and SP networks must be implemented at SDN level. For protection from failures occurring inside the DISCUS domain, a sub-LSP OAM signaling must be exploited, as shown in Figure 3-15, where we can see that the active PW is encapsulated both on the active LSP (green path) and on the stand-by sub-LSP (blue path). In this way, a failure occurring inside the NP domain does not involve the SP domain, i.e. it doesn't involve the back-up PW and the back-up end-to-end LSP, still maintaining available the connection to the SP active service edge router.

In details, the required protection functionalities are described in the following.

- *PW redundancy* means that a back-up PW is used, for a certain SP service VLAN, from a back-up OLT through a back-up MC node to a back-up service edge router in the SP domain. The MPLS PW OAM can properly support it, with the IETF standard R*FC 6718*, also augmented with BFD [*RFC 5880*] for faster time convergence [*RFC5885*]. As usual, a coordination between NP & SP SDNs is needed.
- *LSP protection with back-up LSP* can exploit one of the two possible OAM solutions described above, in order to have a common protection behavior on the generic end-to-end

MPLS LSP. In the first case, the IETF *Y.1720* and ITU-T *G.8131* standards are available. In the second case, a 1:1 LSP protection mechanism based on MPLS-BFD OAM (like the one used for MPLS-TP) can be considered, with a coordination through SDN with the SP that, at present, can use the Loop Free Alternate Fast Re-Route (LFA-FRR) protection mechanism (*RFC 5286*). Sub-LSP protection inside the DISCUS network is available with the support of the OAM sub-path connection monitoring (SPME TCM).

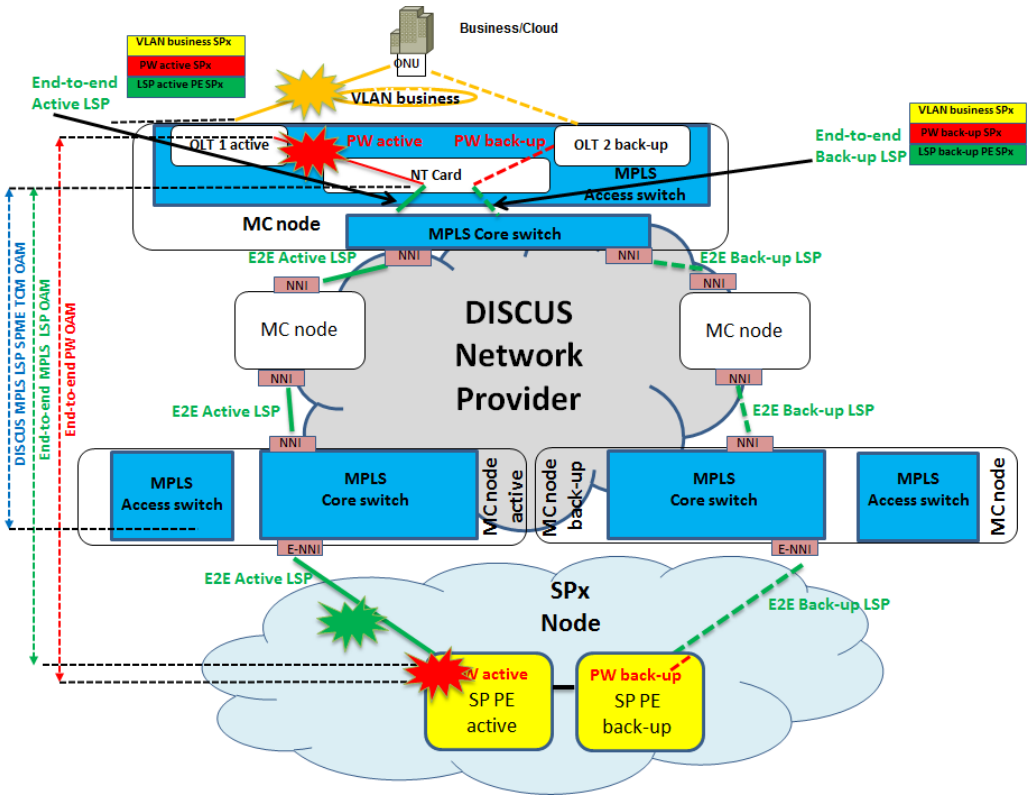


Figure 3-14: End-to-end protection between ODNs and SP edge routers performed with PW redundancy on back-up LSP

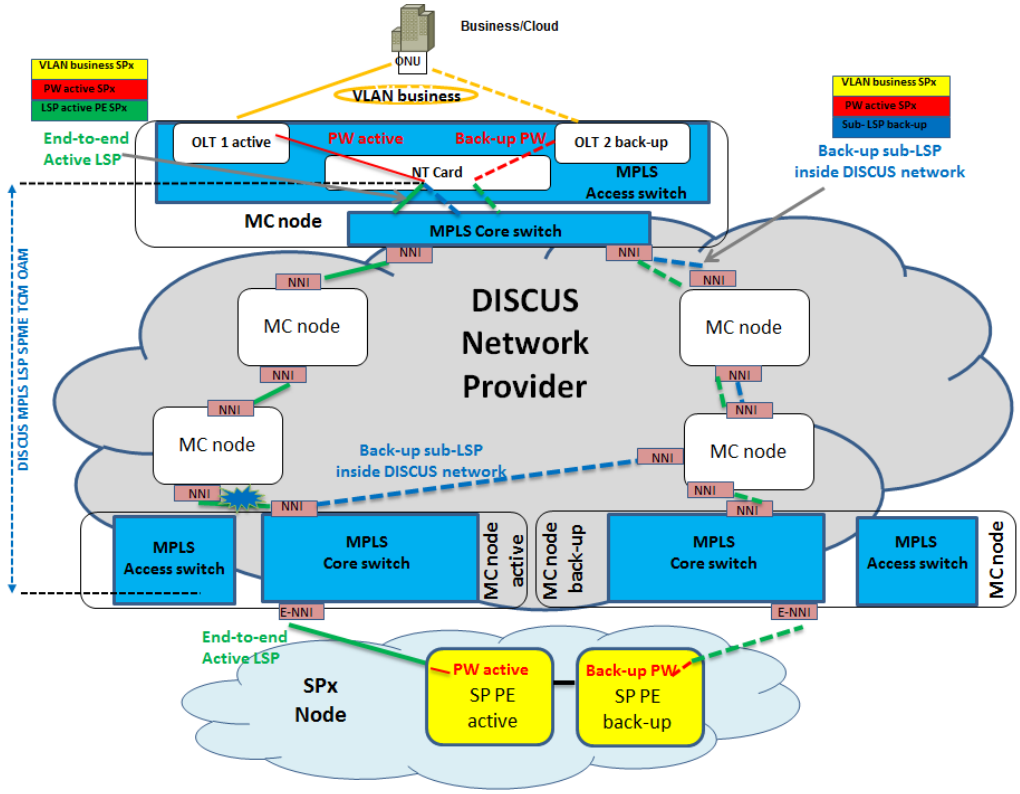


Figure 3-15: Sub-path (sub-LSP) protection inside the DISCUS network

3.4 Control plane design

This section describes the functional architecture and diagram for the control plane, awhile the implementation details are reported in Appendix 1. As previously mentioned (i.e., D6.3) the control plane was designed to provide an OpenFlow implementation of the DISCUS access side of the MC node that could be demonstrated on a testbed.

The interaction between control plane and DISCUS network elements is illustrated in **Figure 3-16**. The main entities considered are:

- A web portal with which the user interacts, which is owned by a Service Provider (SP)[1].
- A network orchestrator, which collects requests from the SPs and forwards them to the node controllers, coordinating their interaction. As discussed in D7.7, the orchestrator also embeds the core network controller in the DISCUS architecture.
- A node controller, receiving requests from the orchestrator and interacting with the physical devices through OpenFlow-based device-control plane interfaces.

There are two main types of scenarios considered for the demo: one is on service on demand and the other is fast PON protection on a dual-homed architecture. The first type is divided into two scenarios, as we differentiate between a simpler single-channel scenario and a multi-

[1] The portal could also be owned by a broker that supports multiple service providers, but in this implementation SP ownership is assumed for simplicity

wavelength scenario, where the ONU can tune to a different wavelength dynamically if required by the service.

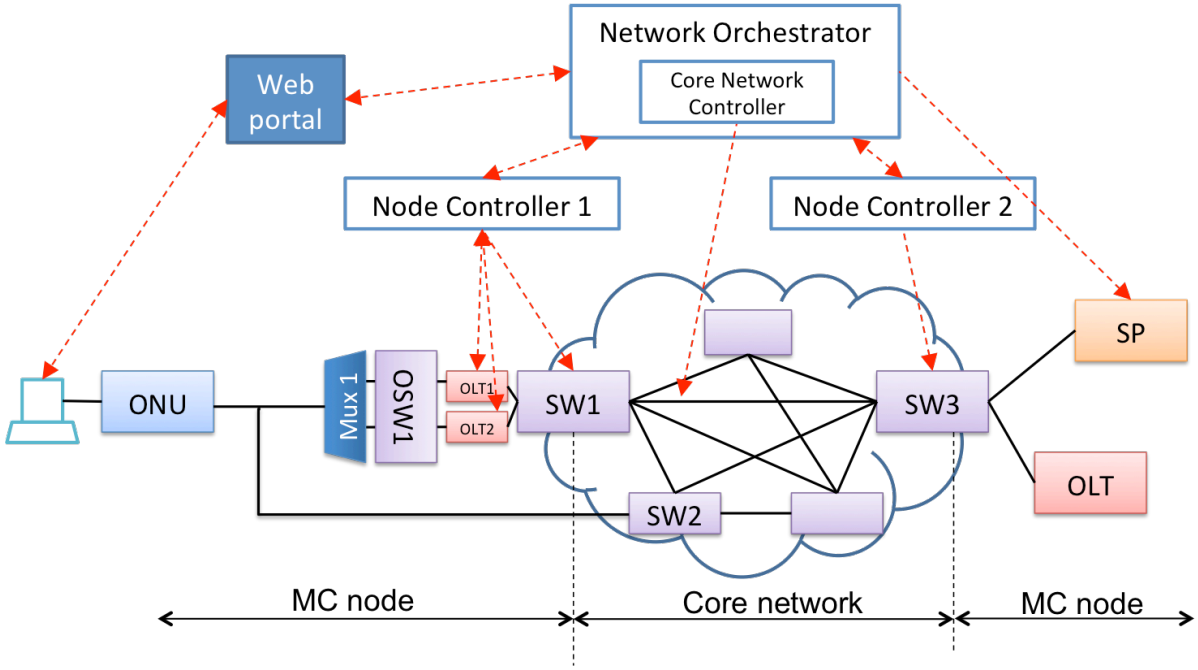


Figure 3-16 Interaction between control plane and DISCUS network elements

There is also a distinction in the service implementation between residential and business service. The differentiation is in the association between Pseudo Wires (PWs) and services at the data level. It should be noted that at the access switch QoS is handled at PW grabularity. A PW service request is associated with establised Committed Information Rate (CIR) and Peak Information Rate (PIR) parameters. The first is used to appropriately reserve capacity towards the output port of the access switch, while the second is used to police traffic at the input.

In the residential case the same PW serves all users in a PON, who receive the same service type from the same provider. Since quality of service is handled at the PW granularity at the access switch, this means that it is up to the SP to police individual users and assure they receive the service requested and do not exceed their peak rates. The SP can dynamically increase the capacity associated to any given PW, as user demand for a service increases, but the MC node has no view on how Quality of Service (QoS) is operated within the PW. We also refer to this case as that of aggregated PW service.

In the business case, a PW is associated to the ONU for a given service, meaning that the QoS can be assured for the individual user by the access switch. This incurs a higher cost, due to the scalability issues of handling QoS on too many PW separately. We also refer to this case as that of individual PW service.

It should be noticed that in our implementation we have decided to consider differentiation at the service level, rather than the user level, meaning that any given user could have a mix of residential type and business type services.

For the first scenario, we assume a user request of video on demand, which the SP needs to deliver as an assured service. Our implementation handles this as an individual PW service in order to demonstrate the dynamic setup of PW, however in a commercial environment we could consider most VoD services handled by aggregated PW services, while only top-

quality VoD services (e.g., 4K video at high-fidelity compression rates) are handled through individual PW.

The flow diagram is synthetically represented in **Figure 3-17**.

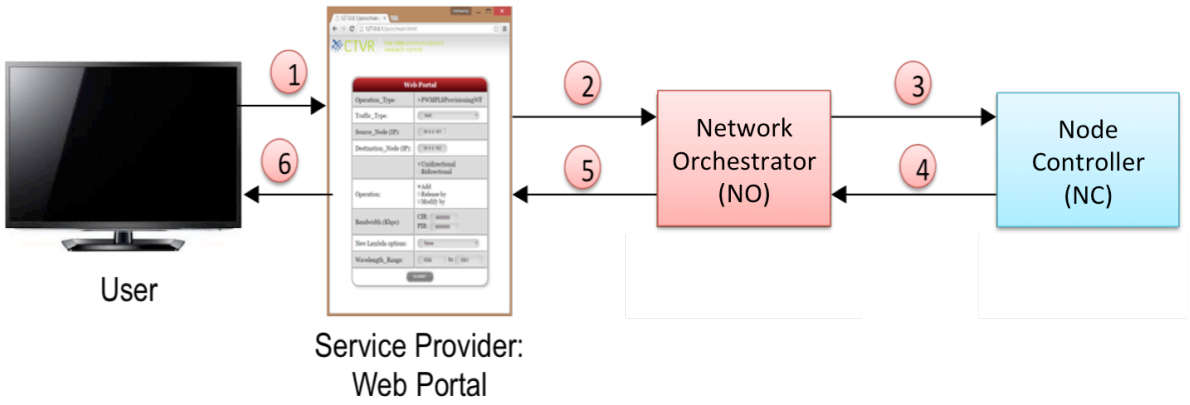


Figure 3-17 Flow diagram for service request to the control plane.

For the first scenario the user requests the service through a web portal from the SP (1). Once selected, the SP computes the required parameters to support the service and, using the Application-controller plane interface (A-CPI) RESTful API (defined in D6.3), sends a request, through a JSON interface, to the orchestrator (2). Such request could include the increase of capacity on an existing PW or the creation of a new one. The orchestrator identifies the MC node involved in the request, translates the A-CPI request into Intermediate-controller plane interface (I-CPI) APIs requests and forwards them to the node controllers (3), (the orchestrator will also handle the interaction with its embedded core network controller where required). Finally the node controllers communicate with the physical devices through the OpenFlow-based Device-controller plane interface (D-CPI). It should be noticed that this implementation is fully OpenFlow compliant: the access switch is an OpenFlow-enabled Ethernet switch supporting OF v1.4; the Polatis optical switch has an embedded OpenFlow agent supporting OF v1.0 with optical extension v0.3; the OLT connects to the control plane with an OpenFlow agent that implements functionalities[2] of OF v1.4.

There is then an Acknowledgment phase to close the loop, where the NCs involved in the request communicate back to the orchestrator that the service was successfully setup (4), and the orchestrator communicating back to the SP that it can now start the VoD service streaming from one of its servers (5). Finally, the user is notified through the web portal (6).

The implementation of the second scenario is similar to the first, with the difference that the ONU is tunable and thus it can move to a different PON when required. The service selection is also similar (although for this scenario we target a more general Bandwidth on Demand application). Besides the ONU communicating its wavelength tuning range, it can also specify rues for moving to a different wavelength channel.

We have defined four options for requesting a dedicated wavelength channel:

- Dedicated: the service request should be granted only if a new dedicated wavelength is available. This for example could be used for some business applications (e.g., X-hauling), that for security reasons want to operate over a non-shared wavelength.

[2] Only the functionalities necessary for the demonstration were implemented in the agent.

- Preferential: the service should be granted preferentially on a new dedicated wavelength; however if this is not possible the user is willing to accept the service on a shared wavelength.
- Necessitated: the user does not request a new (shared or dedicated) wavelength channel, but the ONU can support the operation if required for capacity management purpose.
- None: the ONU does not support tuning to a different channel. If there is not enough capacity in the current PON, the service request is rejected[3].

Finally, the third scenario implements dual-home feeder fiber PON protection. The details for the implementation of this scenario have been reported in section 2.3 of deliverable D4.13 "Resilience in heterogeneous long reach access networks".

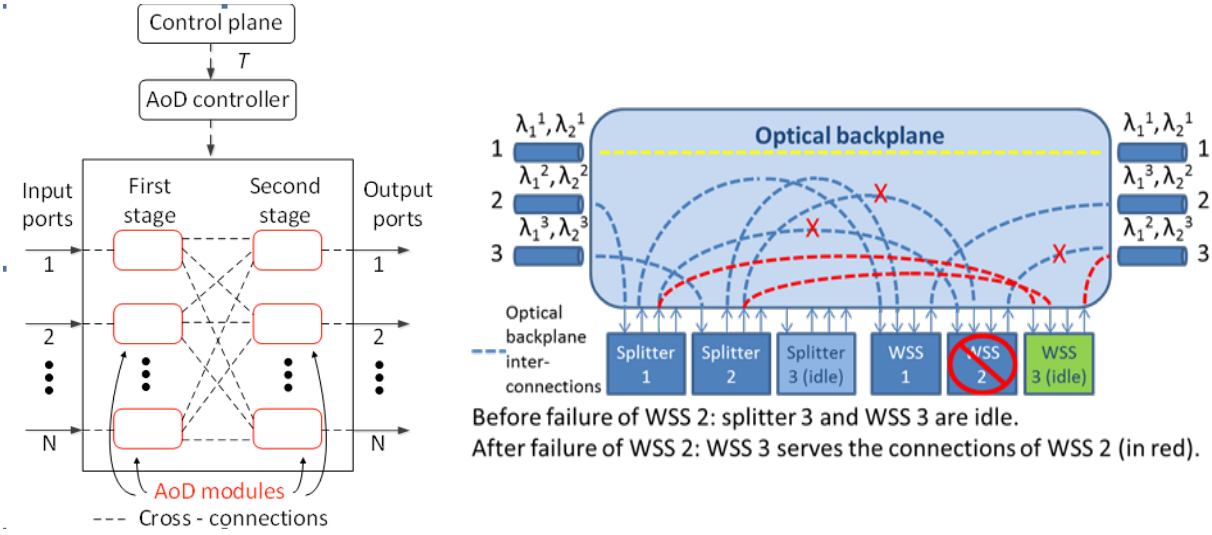
[3] In principle the operator could choose to move other tunable ONUs to a different channel in order to free capacity for the ONU that cannot tune, however we have not implemented this option for our demonstration.

4 Performance Assessment

This chapter provides the study in terms of the different aspects that include resiliency, optical power budget, energy consumption, and cost as well as the associated performance assessment. Regarding resiliency, we proposed to employ the concept of Architecture on Demand (AoD) to enhance DISCUS metro/core node reliability performance in two ways: 1) supporting fibre switching and 2) enabling self-healing. For optical power budget, we update the evaluation consider multi-stage Clos optical switch architecture that is presented in this deliverable.

4.1 Resiliency

Due to the high data rates which must be supported by the DISCUS network, the nodes need to provide a high level of resiliency. Recently, synthetic and reconfigurable optical switching nodes implemented by architecture on demand (AoD) have been proposed [8] in order to address the limitations of existing optical nodes by providing flexible processing and switching of optical signals through programmable architecture. Unlike hard-wired node architectures, optical components inside an AoD Reconfigurable Optical Add/Drop Mulitplexer (ROADM), e.g., splitters, wavelength-selective switches (WSSs), amplifiers, are not fixed within the architecture but can be interconnected in a completely arbitrary manner via an optical backplane (e.g., Polatis piezoelectric optical switch), as illustrated in Figure 4-1 (a). AoD nodes have been shown to have superior performance in terms of cost-effectiveness [9], energy-efficiency [10], scalability [11], and resiliency [12].



a) b)

Figure 4-1: (a) A schematic view of an AoD node and (b) fiber switching (denoted in yellow) and self-healing in an AoD node.

The concept of AoD can be easily applied to the DISCUS node to improve its flexibility and reliability via two key principles: *(i)* supporting fiber switching and *(ii)* enabling self-healing of node component failures. Fiber switching refers to the feature where all signals from one input fiber port of the node can be forwarded directly to an output fiber port without traversing any components in between. Such component bypass can reduce the number of active node components which has a beneficial impact to the power consumption, while it also reduces the associated risk of connections being affected by component failures.

Bypassed components can be put to idle, low-power mode, and can be reused as redundancy in case an active component of the same type fails, thus enabling recovery of affected connections at the node level, without rerouting them through the network. The principles of fiber switching and self-healing are illustrated in Figure 4-1 (b). Note that in this study we are focusing on the core side of the DISCUS metro/core node. The similar idea can be also applied to the access side of the DISCUS node.

4.1.1 Synthesis algorithm

In order to build the appropriate node architecture, a fast and efficient synthesis and selection algorithm is required, which takes into account the availability of building modules and the traffic demands. Given a number of building modules available for the AoD composition and a finite port-count optical backplane, it is important to synthesize the node in a way that uses a minimal number of components, thus lowering the complexity and cost.

The node synthesis algorithm takes a valid traffic matrix *T* (such as the one shown in Figure 4-2 a)) as input and iteratively searches it in three consequtive steps to identify the necessary modules and their cross-connections in the optical backplane. In Step 1, the algorithm checks for fiber switching, which is possible if any row of matrix *T* contains all same elements. If this condition is satisfied, a new fiber-switched connection is created between the pair of identified nodes and added to the set of node connections to be established. Prior to this, a check is performed on whether any other signal from any other inputs needs to be switched to the desired output port, which requires adding an optical coupler at that output.

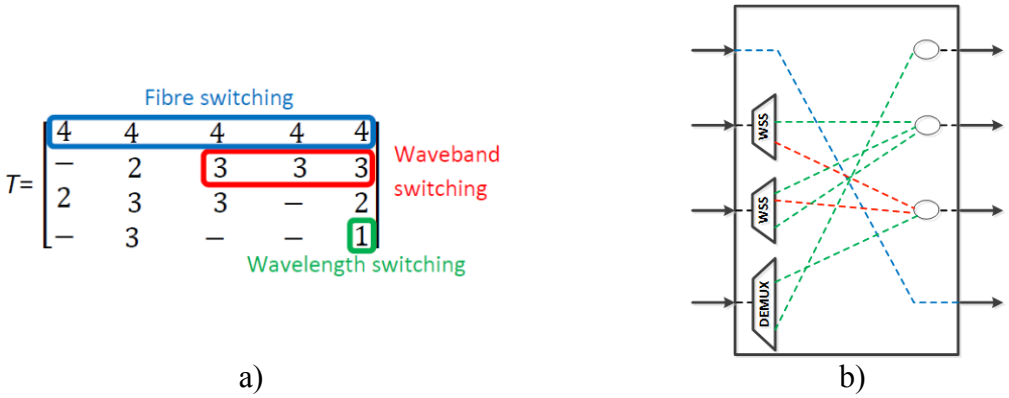


Figure 4-2: (a) An example of a valid traffic matrix T and (b) The node architecture generated by the synthesis algorithm based on matrix T.

In Step 2, the algorithm searches for signals that can be grouped together and switched as a waveband. To do so, it checks each row of matrix *T* (excluding the fibre switching requests identified in first step) for groups of signals on neighboring wavelengths directed to from the same input to the same output port. When a wavebanded signal is found, a check is performed to identify whether a WSS is already attached to the current input port, and if yes, a connectoin is created using the first free port of the WSS. Otherwise, a WSS is added to the input. As in Step 1, a coupler is placed at the output to support signals directed to the same output port if needed.

In Step 3, additional connections needed to support the remaining wavelength-switched signals are set up by first checking if a WSS or a demux have already been placed at the input port. If there are no optical devices, a demux is placed instead of WSS in the fixed-grid scenario to reduce cost and power consumption. As in Steps 1 and 2, a coupler is placed at the output port if necessary to support wavelength-switched signals from other input ports.

The resulting node architecture generated by applying the 3-step synthesis algorithm to matrix *T* is shown in Figure 4-2 b).

4.1.2 Connection availability model

In general, the availability A(t) of a component is defined as the probability that the component works correctly at the time of observation t. Steady-state availability (hereinafter availability) of a component can be calculated from its Mean Time Between Failures (MTBF) and Mean Time To Repair (MTTR). The MTBF and MTTR values, as well as the power consumption and cost of components used in the reliability performance evaluation are shown in Table 5.

Table 5 The MTTF, MTTR, power consumption and cost values of basic AoD and hard-wired node components.

Component	**Mean Time Between Failures (MTBF) [h]**	**Mean Time To Repair (MTTR) [h]**	**Power consumption [W]**	**Cost [a.u.]**
Fixed-grid WSS 1:9	500.000	2	20	73
(DE)MUX	4.000.000	8	-	24
Single-sided switch	13.000.000*	2	40	2000
Optical power splitter 1:8	9.000.000	8	-	1,8
OXC	182.000	2	60	1467

*We assume a single traversal of a switching module.

The availability model of each connection traversing an AoD switch depends on the type of switching and on the existence of redundant components within the node. When there are no redundant components, availability A_n of a connection is the product of the availabilities A_i of components *i* serially traversed by the connection, as illustrated in Figure 4-3. In the figure, C denotes coupler, while OB denotes one traversal of the optical backplane (i.e., single-sided switch in DISCUS metro/core node). If we consider a 192x192 switch matrix, this traversal encompasses one pair of switch ports.

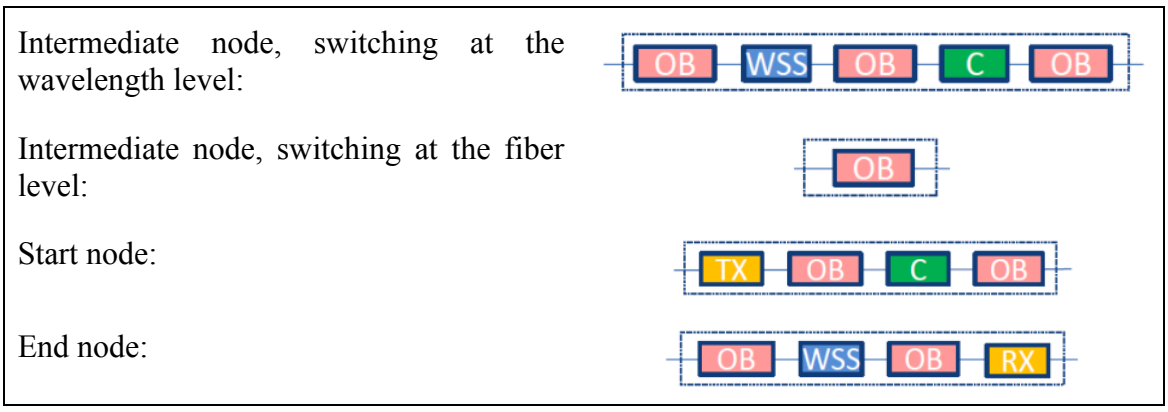


Figure 4-3: Connection availability model for a node without redundant components.

The presence of redundant components in the node changes the serial availability model into an *r-out-of-n* structure where the connection is available if at least *r* our possible *n* components of a certain type are available (here, *r*=1), as shown in Figure 4-4.

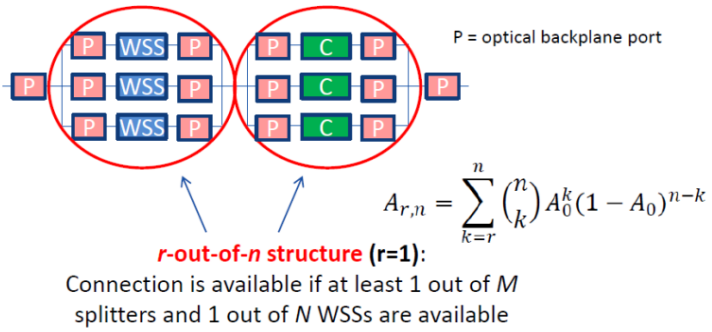


Figure 4-4: Connection availability model for a node with redundant components

Due to the absence of fixed, hard-wired interconnections between components, a redundant component can be shared by all connections in the node. A schematic view of the connection availability model in an AoD node with one redundant WSS shared among three connections is shown in Figure 4-5.

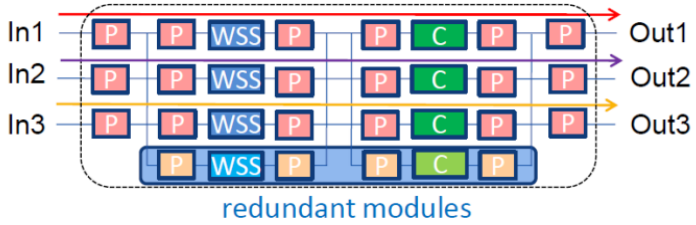


Figure 4-5: Redundant modules shared among all connections within the node

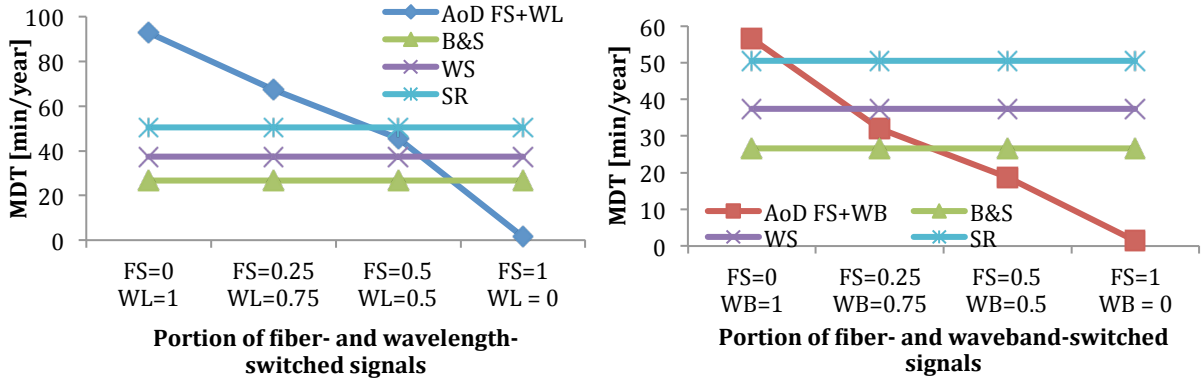
4.1.3 Reliability performance evaluation

To evaluate availability of the considered node architectures we modeled component failures and reparations by means of Monte Carlo simulation, using the data from Table 5Table 5 The MTTF, MTTR, power consumption and cost values of basic AoD and hard-wired node components.. The simulated events cause transitions of connections between fault-free (*UP*) and faulty (*DOWN*) states, so the overall availability can be expressed as $T_{UP}/(T_{UP}+T_{DOWN})$. We assume 88 wavelengths per fiber and analyse the performance of a 4-degree (connecting 4 fiber ports towards core network) and a 6-degree (connecting 6 fiber ports towards core network) node for varying portions of signals switched at the fiber, waveband and wavelength level. A detailed description of the simulation settings can be found in [13].

To evaluate the benefits of introducing the AoD implementation principle to the core-side of the DISCUS metro/core node, we compare its performance against three different conventional, hard-wired node architectures. The first benchmarking architecture is the Wavelength Selective (WS) architecture, implemented by passive demuxes at the input ports and muxes at the output ports, interconnected by an optical cross-connect. All DWDM channels from the N input fibres are demultiplexed into $N \times K$ channels, where K is the total number of wavelengths per fibre. The second benchmarking architecture uses a set of optical splitters at the input ports and a set of WSSs at the output ports in a Broadcast and Select (B&S) configuration. Despite the fact that this architecture is simple and widely accepted, the loss introduced by optical splitters (proportional to the splitting degree) limits its scalability. The third type of architecture, referred to as Spectrum Routing (SR), is realized by placing WSSs at the input and at the output ports. The basic advantage of this architecture in comparison to the B&S architecture is that the lightpath loss does not depend on the nodal

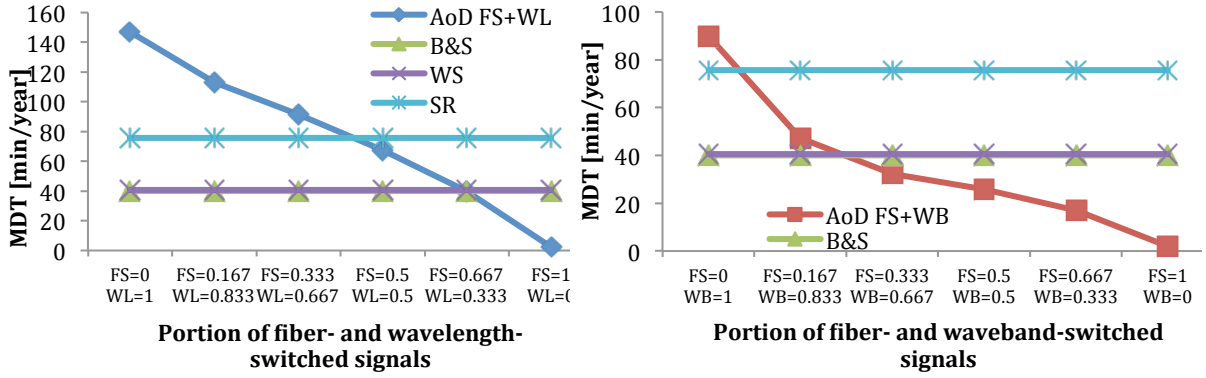
degree. However, it requires additional WSSs at the input fibres, which makes it more expensive to realize.

Figure 4-6 and Figure 4-7 show the Mean Down Time (MDT) of a 4-degree and 6-degree node, respectively, defined as the average number of minutes per year in which at least one signal is in *DOWN* state due to node component failures. The results are recorded for a combination of switching at the fiber and wavelength level (denoted as FS+WL) and for the combination of fiber- and waveband-switching (denoted as FS+WB).



a) b)

Figure 4-6: The mean down time for a 4-degree node and varying switching type portions

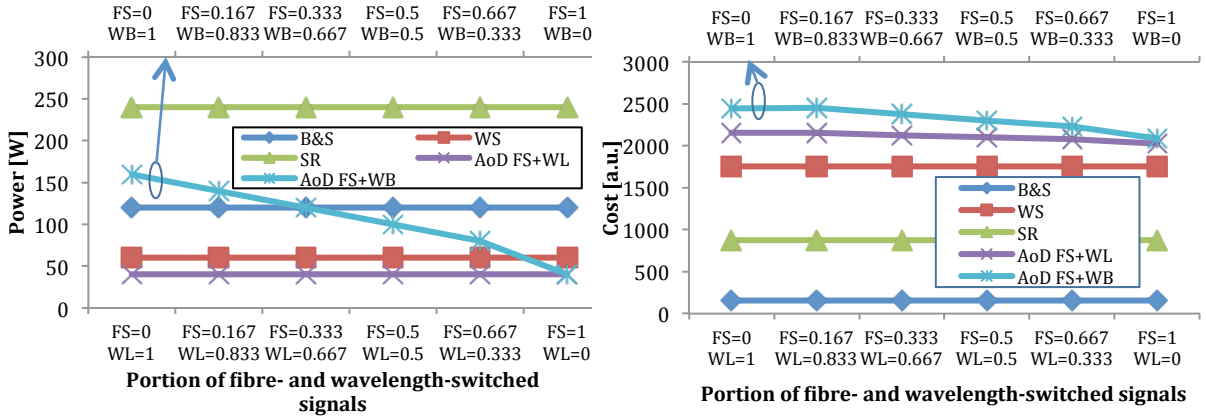


a) b)

Figure 4-7: The mean down time for a 6-degree node and varying switching type portions

As can be seen from the figures, the performance of hard-wired nodes does not change with respect to different switching types since these architectures do not support flexibility in component usage. On the contrary, the MDT for AoD node decreases as the portion of fiber-switched connections increases. This stems from the fact that a greater number of components become idle and can be reused as redundancy for failure recovery. Furthermore, a higher portion of signal wavebanding is also beneficial in decreasing the MDT. This is due to the fact that wavebanded signals occupy a lower number of optical backplane ports, which reduces the correlated risk of connection failures. Note that the slightly higher MDT of AoD compared to hard-wired nodes when FS=0 arises due to contribution of the optical backplane to the overall failure probability. However, this balances out as soon as redundant components are released via increased FS portion with a tendency of further MDT decrease.

The power consumption and costs associated to the different switch architectures are shown in Figure 4-8 for the 6-degree node (the results for the 4-degree node are analogous). As can be seen in Figure 4-8 a), the AoD node with fiber- and wavelength-switching has the lowest power consumption which is equal to the power consumed by the optical backplane. Since passive devices are used for wavelength-switching, the power consumption of this solution does not change with the portion of fiber-switched signals. When waveband switching is taken into account, the added active WSSs increase the power consumption of the AoD node implementation. As the portion of fiber-switched connections increases, the power consumption of AoD drops. Note that the worst-case performance of AoD is well below the power consumption of the spectrum routing architecture which deploys WSSs at all input and output ports.



a) b)

Figure 4-8: (a) The power consumption and (b) the cost of different switch implementations

The cost of different solutions is depicted in Figure 4-8 b), showing that AoD requires slightly higher initial costs than the conventional architecture types. Note that the shown AoD costs include one redundant component of each type for redundancy purposes. Combined with their lower power consumption and lower MDT values, the initial extra costs of AoD could be covered by the savings in Operational Expenditures (OPEX) and revenue losses over a period of 5-7 years [12].

The analysed scenarios primarily focused on fixed grid and unicast traffic, but can easily be applied to flex-grid scenarios as well. Since flex-grid WSS has a lower failure rate than fixed-grid WSS, the MDT of flex-grid case would be lower than the fixed-grid one, while their relation to the hard-wired implementation would likely follow the same general trend. Note also that the passive demuxes would need to be replaced by flex-grid WSSs, which might slightly increase the MDT, power consumption and cost in scenarios with dominating portion of wavelength-switched signals. An additional functionality required by the node is to support manycast. In this case, the multicast fiber-switched signals would need to use one extra splitter at the input port and one additional traversal of the optical backplane, which would slightly increase their failure probability. However, given the very low failure rate of these components, these modifications are not expected to have a significant impact on the trends of the results.

4.2 Optical power budget

The LR-PON power and Optical Signal Noise Ratio (OSNR) budget presented in D2.1, D2.3 and D5.4 have taken into account the preliminary configuration of the metro/core node including the expected losses of the various components (wavelength mux/demux, optical switch, etc.). The configuration of the metro/core node on the access side can be seen in **Figure 4-9**, which is also showing the LR-PON in the lollipop configuration. It should be noted that the metro/core node configuration on the access side does not have a significant change compared to earlier deliverables and it is also the same for all variants of the DISCUS LR-PON. Optical amplifiers are placed at the ingress/egress of the node followed by circulator and a DWDM component (or alternatively two DWDM one each for down- and up-stream). The demultiplexed inputs/outputs of the DWDM are then connected to the ports of the optical switch. The OLTs down- and up-stream ports are also connected to the optical switch, which enables the proper connection between DWDM and OLTs.

In this section we present an update from the previously presented analysis, where a more accurate estimate of the optical switch insertion loss is introduced. Due to the large size of the port cross connect required by the metro/core node design (around 12k) a Clos optical switch architecture (see Sub-chapter 3.1.1) is necessary. The loss of a multi-stage optical switch should simply be several times that of a single stage, plus internal interconnects. Assuming pluggable single stage modules with reasonably well engineered connectors introducing a 1dB max insertion loss, and a max loss of 2dB per switch stage, the max loss of the overall (passing switch matrix 3 times) is 7dB. It should be noted that although 7dB max loss for a 12,000 port switch is quite impressive, better insertion losses could be obtained taking the 3 sigma path loss. However, for power and OSNR budget calculation the max loss should be taken into account.

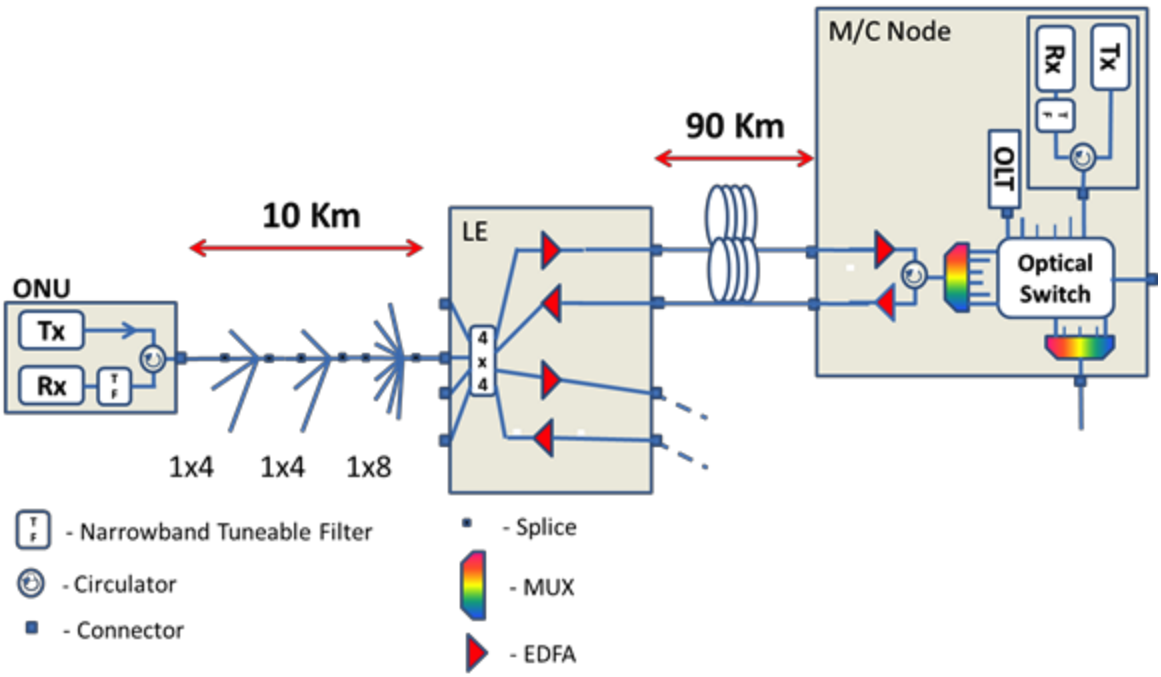


Figure 4-9: DISCUS LR-PON architecture for single LE (lollipop).

Due to the presence of optical amplifiers (Erbium Doped Fibre Amplifier EDFAs) at the ingress/egress of the node, small deviations in the component losses can be accommodated with no impact on the overall performance of the link, which is mostly limited by the access

and metro portions of the LR-PON. For example the higher loss introduced by the 3-stage switch, which is 7dB in the worst case compared to 2 dB for a single stage switch, has a very small impact on the ONSR of the upstream signal. The OSNR calculations are reported in **Table 6** and **Table 7** for a single stage switch and a multi-stage (passing switch matrix three times) switch respectively, for the lollipop configuration with 512 split, 10km ODN and 90km metro link. The OSNR is practically unchanged and the only drawback of the higher loss in the switch is the need for a higher gain in the EDFA at the input of the metro/core node. Power and OSNR budget calculations for all the other variants of the DISCUS LR-PON return the same conclusion that the OSNR is practically unaffected by an increased loss in the optical switch.

However, introducing other high loss components between the EDFAs and the OLTs might become unpractical because of the high gain required from the EDFAs and hence another stage of optical amplification might be required, which would increase the component count.

Table 6 DISCUS LR-PON OSNR and power budget for single LE (lollipop) configuration for 512 split 10km ODN and 90km metro link, considering a single stage switch in the metro/core node (2dB loss).

	Components	Gain/Loss(Min) [dB]	Gain/Loss(Max) [dB]	P_Sig(Nom) [dBm]	OSNR(Nom) [dB]	P_Sig(Max) [dBm]	OSNR(Max) [dB]	P_Sig(Min)[dBm]	OSNR (Min) [dB]
1	Transmitter	0	0	5	40	5	40	5	40
2	Circulator	-0.2000	-0.6000	4.6000	40.0000	4.8000	40	4.4000	40
3	Connector	0	-0.5000	4.3500	40.0000	4.8000	40	3.9000	40
4	Splice	0	-0.3000	4.2000	40	4.8000	40	3.6000	40
5	1:4 Splitter	-5.4000	-7.1000	-2.0500	40	-0.6000	40	-3.5000	40
6	Splice	0	-0.3000	-2.2000	40	-0.6000	40	-3.8000	40
7	Splice	0	-0.3000	-2.3500	40.0000	-0.6000	40	-4.1000	40.0000
8	1:4 Splitter	-5.4000	-7.1000	-8.6000	40.0000	-6.0000	40	-11.2000	40
9	Splice	0	-0.3000	-8.7500	40	-6.0000	40	-11.5000	40.0000
10	Splice	0	-0.3000	-8.9000	40.0000	-6.0000	40	-11.8000	40.0000
11	1:8 Splitter	-7.9500	-10.5000	-18.1250	40	-13.9500	40	-22.3000	40.0000
12	Splice	0	-0.3000	-18.2750	40.0000	-13.9500	40	-22.6000	40.0000
13	Fibre (0/10 Km)	0	-3	-19.7750	40	-13.9500	40	-25.6000	40.0000
14	Connector	0	-0.5000	-20.0250	40	-13.9500	40	-26.1000	40.0000
15	1:4 Splitter	-5.4000	-7.1000	-26.2750	40.0000	-19.3500	40	-33.2000	40.0000
16	EDFA1	32.2000	32.2000	5.9250	25.7491	12.8500	32.2892	-1	19.2090
17	Connector	-0.5000	-0.5000	5.4250	25.7491	12.3500	32.2892	-1.5000	19.2090
18	Fibre (90 Km)	-27	-27	-21.5750	25.7491	-14.6500	32.2892	-28.5000	19.2090
19	Connector	-0.5000	-0.5000	-22.0750	25.7491	-15.1500	32.2892	-29	19.2090
20	EDFArx	21.1000	21.1000	-0.9750	24.4607	5.9500	31.0995	-7.9000	17.8220
21	Circulator	-0.6000	-0.6000	-1.5750	24.4607	5.3500	31.0995	-8.5000	17.8220
22	Mux/DeMUX	-6	-6	-7.5750	24.4607	-0.6500	31.0995	-14.5000	17.8220
23	Optical Switch	-2	-2	-9.5750	24.4607	-2.6500	31.0995	-16.5000	17.8220
24	Connector	-0.5000	-0.5000	-10.0750	24.4607	-3.1500	31.0995	-17	17.8220

Table 7 DISCUS LR-PON OSNR and power budget for single LE (lollipop) configuration for 512 split 10km ODN and 90km metro link, considering a multi–stage switch (passing switch matrix three times) in the metro/core node (7dB loss).

	Components	Gain/Loss(Min) [dB]	Gain/Loss(Max) [dB]	P_Sig(Nom) [dBm]	OSNR(Nom) [dB]	P_Sig(Max) [dBm]	OSNR(Max) [dB]	P_Sig(Min)[dBm]	OSNR (Min) [dB]
1	Transmitter	0	0	5	40	5	40	5	40
2	Circulator	-0.2000	-0.6000	4.6000	40.0000	4.8000	40	4.4000	40
3	Connector	0	-0.5000	4.3500	40.0000	4.8000	40	3.9000	40
4	Splice	0	-0.3000	4.2000	40	4.8000	40	3.6000	40
5	1:4 Splitter	-5.4000	-7.1000	-2.0500	40	-0.6000	40	-3.5000	40
6	Splice	0	-0.3000	-2.2000	40	-0.6000	40	-3.8000	40
7	Splice	0	-0.3000	-2.3500	40.0000	-0.6000	40	-4.1000	40.0000
8	1:4 Splitter	-5.4000	-7.1000	-8.6000	40.0000	-6.0000	40	-11.2000	40
9	Splice	0	-0.3000	-8.7500	40	-6.0000	40	-11.5000	40.0000
10	Splice	0	-0.3000	-8.9000	40.0000	-6.0000	40	-11.8000	40.0000
11	1:8 Splitter	-7.9500	-10.5000	-18.1250	40	-13.9500	40	-22.3000	40.0000
12	Splice	0	-0.3000	-18.2750	40.0000	-13.9500	40	-22.6000	40.0000
13	Fibre (0/10 Km)	0	-3	-19.7750	40	-13.9500	40	-25.6000	40.0000
14	Connector	0	-0.5000	-20.0250	40	-13.9500	40	-26.1000	40.0000
15	1:4 Splitter	-5.4000	-7.1000	-26.2750	40.0000	-19.3500	40	-33.2000	40.0000
16	EDFA1	32.2000	32.2000	5.9250	25.7491	12.8500	32.2892	-1	19.2090
17	Connector	-0.5000	-0.5000	5.4250	25.7491	12.3500	32.2892	-1.5000	19.2090
18	Fibre (90 Km)	-27	-27	-21.5750	25.7491	-14.6500	32.2892	-28.5000	19.2090
19	Connector	-0.5000	-0.5000	-22.0750	25.7491	-15.1500	32.2892	-29	19.2090
20	EDFArx	26.1000	26.1000	4.0250	24.4591	10.9500	31.0979	-2.9000	17.8202
21	Circulator	-0.6000	-0.6000	3.4250	24.4591	10.3500	31.0979	-3.5000	17.8202
22	Mux/DeMUX	-6	-6	-2.5750	24.4591	4.3500	31.0979	-9.5000	17.8202
23	Optical Switch	-7	-7	-9.5750	24.4591	-2.6500	31.0979	-16.5000	17.8202
24	Connector	-0.5000	-0.5000	-10.0750	24.4591	-3.1500	31.0979	-17	17.8202

4.3 Cost and power consumption (Dave will update)

In the original proposal document we proposed that the DISCUS architecture should significantly reduce power consumption compared with business as usual networks. A comparison graph which compared a business as usual (BAU) power consumption curve with a projected power consumption curve for the DISCUS architecture was shown in the proposal and is reproduced in Figure 4-10.

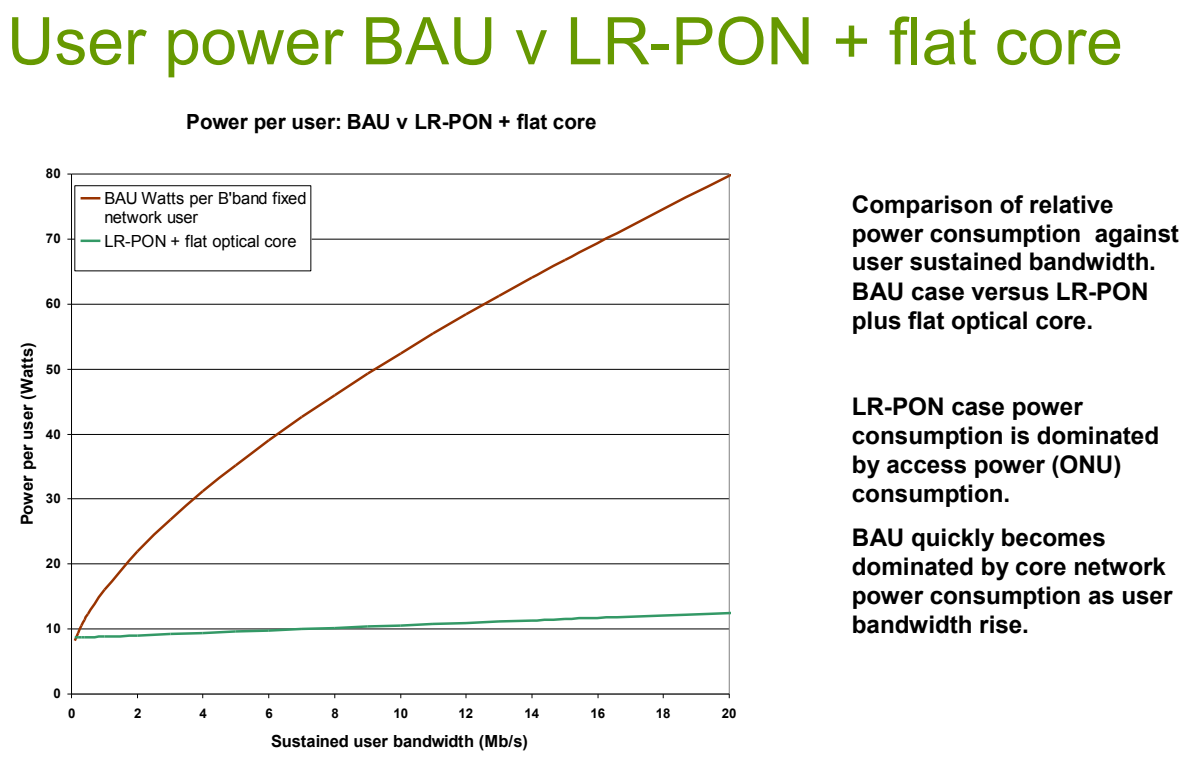


Figure 4-10 Original estimate of power consumption benefits of the DISCUS architecture compared with BAU.

Except for the Customer ONU much of the power consumption arises in the metro-core node for the DISCUS network and the power consumption of these nodes is indicative of the

consumption for the whole network. This section provides an updated power consumption calculation for the metro-core node.

4.3.1 Power consumption calculations

The details of the power consumption modelling methodology is described in D2.8 and is not reproduced here, however an outline of the modelling process will be given. The power consumption modelling has proved to be more difficult than originally envisaged; this is because of the wide variation in reported power consumption of equipment and components in the published literature. A good publication that tries to address this problem is a paper by Heddeghem [14] which recognised the problem and tried to give a set of compromise figures based on some published figures from equipment suppliers. However the authors needed to make projections for power consumption of future equipment and the number of data points for these projections was very small (often only two points being available) thus still leading to anomalous results across the various items of equipment. A general problem with the power consumption analysed in the literature is a lack of a more fundamental basis for power consumption reduction and the link to the technology families used in equipment. The major impact of power consumption reduction comes from die size or feature size of integrated circuits. The model described in D2.8 uses the evolution of integrated circuit technology to provide the fundamental basis for projecting future technology power consumption. In this section the results of that model applied to the technology used within the metro core node is given and an updated curve for the power consumption benefits of the DISCUS metro-core node is given.

4.3.2 Metro-core node power consumption model results

The main equipment items requiring electrical power in the metro-core node are the OLT shelves and racks, the layer 2 switch and the layer 3 router, and switch and transponders that interface to the core network fibres. There is also the optical switch although this is low power compared to the above items of equipment and only amounts to ~6% of the total metro-core node power.

The power dissipation of the metro-core node equipment components as a function of user sustained bandwidth in the busy hour is shown in Figure 4-11 and Figure 4-12. These figures show the same data but one with a linear scale for the power consumption axis and one with a logarithmic scale. The linear scale shows that after about 20 Mb/s of user bandwidth the power dissipation for all component tends to a linear relationship with user bandwidth. At lower user bandwidths, which is the likely situation for the next decade at least, the relation is non-linear. This is particularly the case for the OLT and transponders and to a lesser extent the optical switch. This is because regardless of user bandwidth there is a minimum OLT count just to pass all the customers that connect to the metro-core node; similarly there is a minimum number of transponders to enable the flat optical core to have a full logical mesh of light path interconnects to all other metro-core nodes.

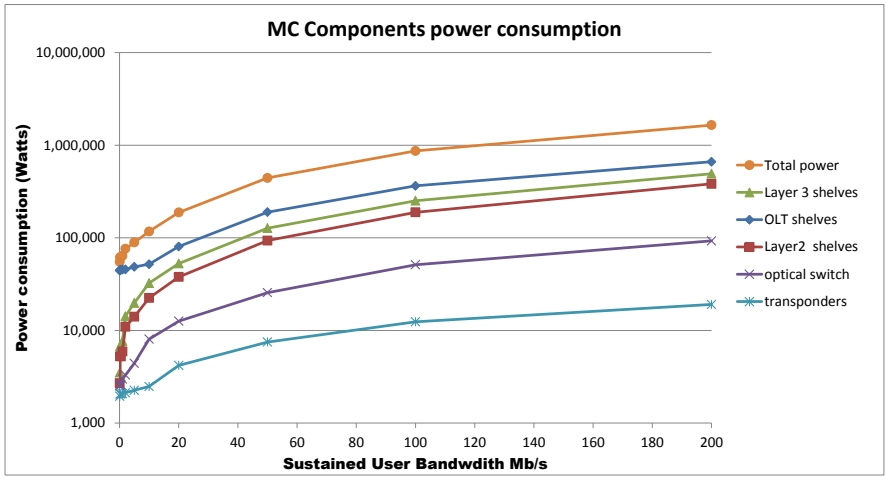


Figure 4-11 Metro-core node - power dissipation of equipment components (log scale)

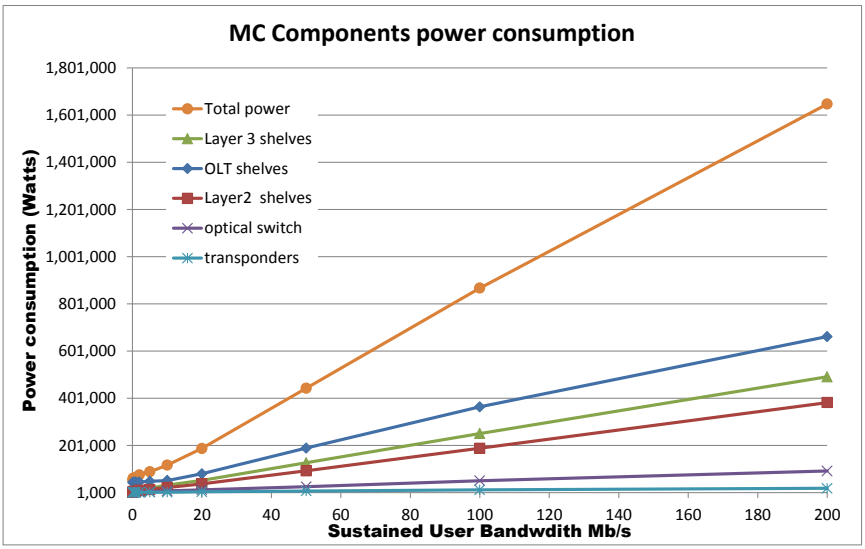


Figure 4-12 Metro-core node - power dissipation of equipment components (linear scale)

These results are based on a LR-PON with 512-way split with 80% utilisation (i.e. 20% spare capacity for customer growth with 100% customers passed at day one build). The capacity utilisation was set to 75% after which the LR-PON capacity is increased by adding an additional LR-PON wavelength through an additional OLT at the metro-core node site. The results are calculated for 100% take up, which would be the highest power dissipation. In the model the take-up is a variable that can range from 1% to 100%. The metro-core node size is set to 380,000 primary sites, which is the average metro-core node size for the UK network with a flat core design of 73 nodes. Much of the flat core modelling and design work in the DISCUS project has used the 73 metro-core node network design and this number was also adopted here for compatibility purposes. In addition, the model can be set to any size of metro-core node between 50,000 and 1,500,00 primary sites connected.

The most power hungry components are the OLT shelves, followed by the layer-3 routers, as expected, and then the layer-2 switch. The optical switch and transponders are relatively low power. This is because the optical switch components are inherently low power devices ~50mW per switching element coupled with the large beam steering arrays which lead to very low power switch configurations. The low power of the transponders simply reflects the relatively small number of these components compared to the number of OLTs and switch ports on the access side of the network.

The results shown in Figure 4-11 and Figure 4-12 are for a configuration with OLT shelves that connect via grey ports to the layer-2 access switch. The power consumption and cost of these components could be further reduced by embedding the OLTs into the access switch. That is to build line cards that plug into the slot cards of the layer-2 switch that consist of OLT ports rather than grey point to point ports. This would save two sets of grey interface ports: one set on the OLT shelves and the corresponding set on the layer-2 switch. For the example metro-core node used to produce the results there would be a further modest power saving of about 5% achieved by embedding the OLTs into the layer-2 switch. This is a significant saving at high bandwidths when power consumption could be 1.6 MegaWatts per metro core node or ~123 MegaWatts for the all the metro-core nodes in the UK network.

As mentioned above, the model can work with a wide range of metro-core nodes sizes and that enables exploration of power requirements as average node size and the total number of nodes in the network vary. A plot of total network power against number of metro-core nodes in the flat core network is shown in Figure 4-13.

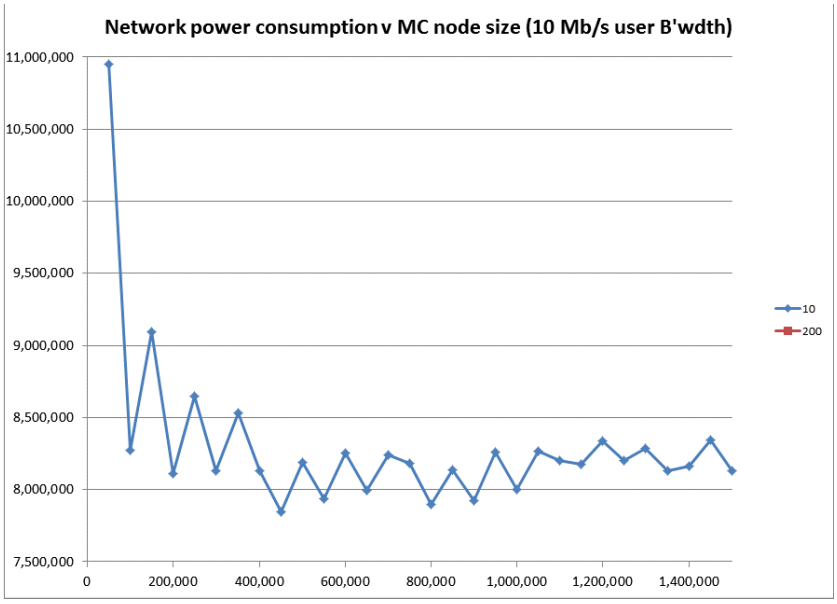


Figure 4-13 Network power consumption v metro-core node size

The graph shows a minimum consumption around a metro-core node of size 450,000 sites which corresponds to a network for the UK of about 60 MC nodes. If very small nodes of only 50,000 sites are used then the power consumption rises rapidly. However much of the power consumption benefit is obtained once the average site size reaches the 100,000 sites, although there is a downwards trend up until the 450,000 sites size, after which power consumption rises slightly. The curve in Figure 4-13 has been plotted with a course granularity of metro-core node size increasing in steps of 50,000 sites and produces the saw tooth shape to the curve the discontinuities are produced by the transitions to different volumes of the various port capacities that are calculated within the model as at each data point the port population is recalculated, rather than adding ports incrementally.

The power consumption model we have created is significantly more detailed and sophisticated compared with the initial estimate shown in Figure 4-10, however the results show significant improvement relative to our early estimates. Figure 4-14 shows the comparative results of the power consumption per user on a 450,000 site MC node with the original estimate. One important reason for the improved figure is the updated energy efficiency of today's and projected technology. The new result were obtained considering

technology used for energy efficiency projects for the year 2016, whereas the original estimate used some early published data that were about an order of magnitude poorer in energy efficiency.

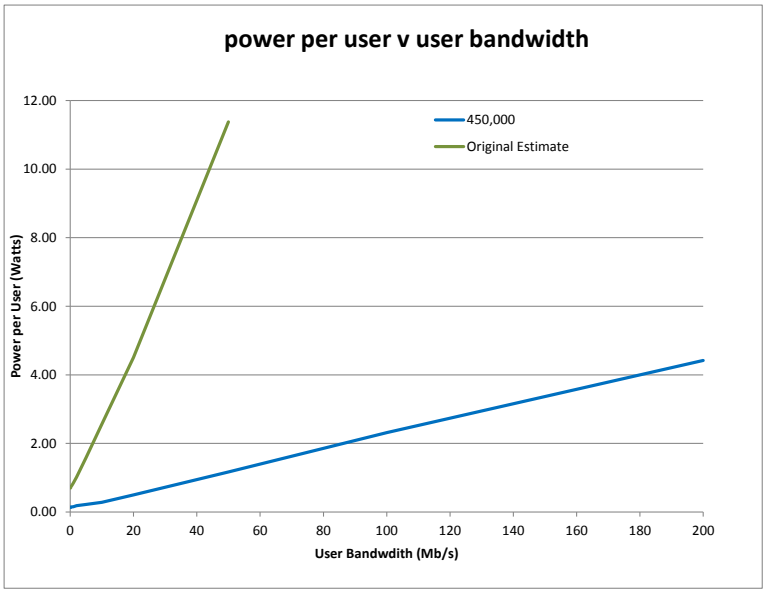


Figure 4-14 Power per user v user bandwidth for a 450,000 site MC node

When these curves are re-plotted on the original Figure 4-10 over the restricted user bandwidth range up to 20Mb/s then the comparative chart in Figure 4-15 is obtained and the improvement clearly shown.

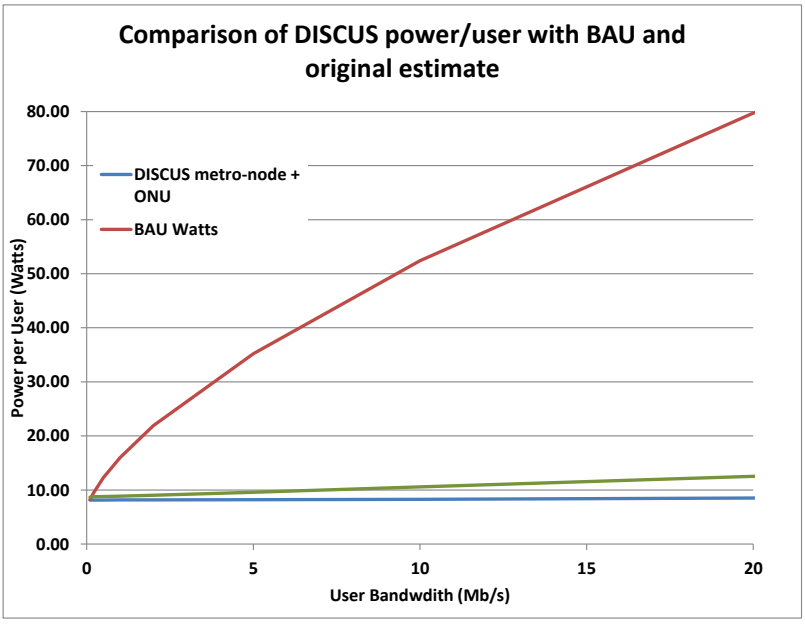


Figure 4-15 Updated curve for power per user plotted on original proposal chart

4.3.3 Power consumption conclusions

The power consumption model is a detailed model constituting an integral part of the cost and cash flow model reported in D2.8. The model dimensions all the component volumes for the equipment layers within the metro core node and then computes the power consumption for any size and configuration of the metro-core node. The model is capable of analyzing a wide range of power consumption figure and trends, some of which have been shown in this

section. The overall conclusion however is that the major power consumption benefits produced for the initial DISCUS architecture proposal have actually underestimated the power savings brought about by a large scale deployment of the DISCUS architecture, which are indeed much greater than originally anticipated.

5 Conclusions

In the deliverable, we have reported the final outcome resulting from Task T6.1, "metro/core node architecture design". First, a set of network services have been updated, which DISCUS metro/core node need to support. They are divided into two major categories, namely end user-oriented and core-oriented, which highly impact on the metro/core node modules handling the traffic from/to LR-PON and optical flat core segments in DISCUS architecture, respectively. Compared with the previous deliverable D6.1, service list updates are envisaged especially for end user-oriented services, getting aligned with customers' services evolution. Some updates are also necessary in order to better specify the services that need high bandwidth, available over a lambda (wavelength service towards the end user) of the LR-PON in the DISCUS architecture.

Then the overall DISCUS metro/core node design is refined based on the preliminary work depicted in D6.1 by containing the functions for different layers supporting the specified DISCUS network services. The main functions are maintained as the same as the design reported in D6.1. The major efforts have been put in the design of each function in different layers, including optical switching and transport technology towards optical flat core, OLT towards LR-PON as well as Layer 2 MPLS access and core switch. In the final design of DISCUS metro/core node, we concentrate on single sided switch architecture, which brings high flexibility for operation and installation process. Two alternatives to realize large-size single sided optical switch has been reported, where the total number of switching ports can be up to 12288 when using 192 optical switch matrix.

It should be noted that the optical space switch used in DISCUS metro/core node offers a transparent optical layer that the fibre links towards both access and core segments, and on the other hand also impact on performance in terms of different aspects. Performance assessment carried out in this deliverable concentrates on node level and analyze the impact of introducing the large-size of optical space switch (as optical switch backplane) in the final DISCUS metro/core node design. Regarding resiliency, by using architecture on demand concept we have found that the reliability performance can be improved in two ways: *(i)* supporting fiber switching and *(ii)* enabling self-healing of node component failures. The results have shown that compared to the other possible designs, the architecture with optical switch backplane can reduce mean down time and power consumption, particularly for the case with more traffic applicable for fiber switching. It should be noted that this resiliency enhancement is at the expense of the number of switching ports, implying higher cost. For optical power budget, we updated the evaluation to consider multi-stage optical switch architecture that can support more than 10k switching ports. In the worst case, 7dB is introduced by passing optical switch backplane in the DISCUS metro/core node, but the optical signal noise ratio is kept almost the same.

It should be noted that this document is a final report of Task T6.1 on the specifications of the DISCUS metro/core node architecture. All the performance analyses included in this deliverable are carried out in node level. Apart from the work done within WP6, the studies on DISCUS metro/core node design have been also interacted with the investigations of service requirement, cost and energy models (in WP2), LR-PON components (developed in WP4 and WP5), and optical flat backbone network (in WP7). The associated performance assessments can refer to the deliverables within the related work packages.

6 Appendix 1 Control Plane Implementation

This appendix describes the implementation of the OpenFlow-based control plane for the Metro-Core node. Although the focus of the implementation is in the Metro-Core node controller and specifically in the control of the access switch, optical switch and OLT/ONU, we have also implemented a basic network orchestrator in order to test the interfaces to the node controller and to allow the testing of the three scenarios described in D6.3.

6.1 Implementation on Network Orchestrator

The function of the network orchestrator is described in deliverable D6.3, and has the task of orchestrating connectivity between the MC node controllers, i.e. collecting an abstract view of the network, receiving requests from applications (e.g., from SPs or network management agents) and sending requests to MC node controllers and core network controllers to instantiate end-to-end connectivity.

Our implementation of the orchestrator is greatly simplified, providing two basic functions, which are required for the demonstration:

- Communication with the Service Provider in order to receive service requests and acknowledge the service creation.
- Communication with the MC node controllers in order to send connectivity request and receive acknowledgment of the connectivity created.

The NO is implemented in python and the communication towards the SP and the controllers is implemented through socket programming (the NO listens on port 8888). The messages are formatted using JavaScript Object Notation (JSON). JSON is a syntax for storing and exchanging data. It is written in a lightweight data-interchange format, which is less verbose than XML. JSON also describes data structures which includes arrays, whereas XML does not. Therefore, JSON is used to communicate between the web portal and NO, and between NO and NC. The messages are then transferred using the CURL library. CURL is a tool to transfer data from/to a server using one of supported protocols. In the demonstration, it is used to send JSON request and response messages with the html protocol.

This is an example on how the request by a user who has requested a VoD service is processed by the NO. The user request is captured by the SP portal, which forwards a capacity request to the NO, by formulating a JSON message, as shown in Figure 6-1. “Trafic_Type” specifies VoD which is video on demand. “Operation_Type” specifies that the request is of PW (PseudoWire) type. “Source_Node” specifies the ID of the SP server, while “Destination_Node” specifies the ID of the ONU. The “Direction_Type” indicates that the bandwidth reservation is only unidirectional. The “operation” specifies that the request is to add a new PW, while the “New_Lambda” parameters set as “none” specifies that the service can only be accepted at the ONU on the current wavelength. The wavelength range is not used in this example, as a new wavelength cannot be accommodated by the ONU. Finally the assured bandwidth is specified by a parameter “CIR”, which is 100 Mbps and a “PIR” of 1Gb/s.

{'Traffic_Type': 'VoD', 'Operation_Type': 'PWMPLSProvisioningWF', 'Source_Node': '15.0.0.1', 'Destination_Node': '10.0.1.1', 'Direction_Type': 'Unidirectional', 'Operation': 'add', 'New_Lambda': 'None', 'Wavelength_Range_from': '1556', 'Wavelength_Range_to': '1561', 'Bandwidth': {'CIR': '100000', 'PIR': '1000000'}}

Figure 6-1 Example of request from web portal in JSON format

A similar message is forwarded by the NO to the node controller (Figure 6-2). It should be noted here that since this is a request on a PW, the destination is that of the OLT, which will be terminating the PW.

{'Traffic_Type': 'VoD', 'Operation_Type': 'PWMPLSProvisioningWF', 'Source_Node': '15.0.0.1', 'Destination_Node': '10.0.1.1', 'ONU_IP': '10.0.0.1', 'Direction_Type': 'Unidirectional', 'Operation': 'add', 'New_Lambda': 'None', 'Wavelength_Range_from': '1556', 'Wavelength_Range_to': '1561', 'Bandwidth': {'CIR': '100000', 'PIR': '1000000'}}

Figure 6-2 Example of request from NO to NC in JSON format

The commands are then submitted to the NC using the CURL tool. This is an example for adding a service:

pycurl.URL = http://<NC IP address>:8080/stats/flowentry/add

pycurl.POST = 1 (1 calls the POST method, 0 calls the GET method)

pycurl.POSTFIELDS = JSON output message

pycurl.perform() (this sends the message out to the NC).

6.2 Implementation on Network Controller

6.2.1 Overall architecture of network controller

The network controller (NC) is in charge of translating incoming requests from the NO into instructions for the physical devices, such as access switch, optical switch and OLT. Figure 6-3 shows the controller architecture. The NC consists of JSON (REST/API), Application module, Database, and RYU controller.

The JSON (REST/API) module is an interface that translates a JSON request, which is received from the NO via I-CPI on port 8080, into an array list which is sent to the application module.

The application module, which is implemented using python, processes the incoming request, considering the state information present in the database. This module implements functionalities such as path calculation, path recovery, wavelength selection, bandwidth assignment, PW assignment and LSP assignment. Based on the request and the network state, it determines whether the request can be satisfied or should be declined. If the request is declined, the application module returns a decline message back to the NO, which is parsed by the JSON interface module. Otherwise, the application module triggers the appropriate OpenFlow commands using the RYU OpenFlow controller.

The database module is implemented in MySQL and is one of the core elements as it stores detailed connectivity state information such as paths, devices ports, wavelengths, bandwidth, flow entries and local topology.

The OpenFlow controller is version 3.22 of RYU, which supports OpenFlow 1.4 and is developed in python. The role of the controller is to translate the requests from the applications into OpenFlow protocol syntax.

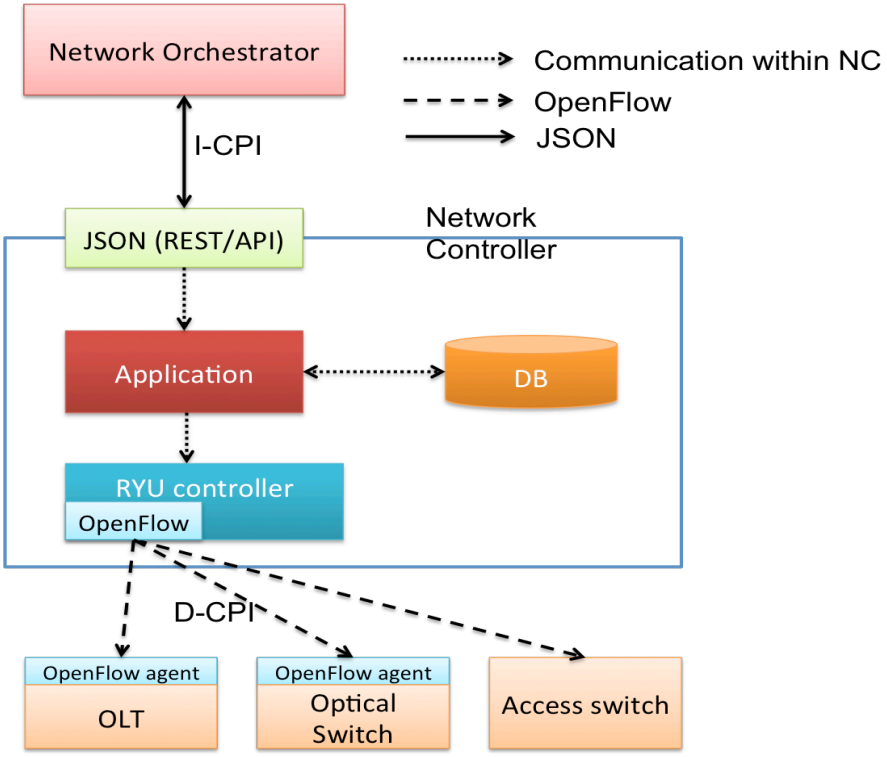


Figure 6-3 Network controller architecture

6.2.2 Application Module

6.2.2.1 Interaction with NO through I-CPI

Once it receives a message from the NO through the JSON module, the controller processes the request using information available in the database.. If the request cannot be satisfied, for example because there is not enough available capacity to satisfy the request, the controller replies to the NO with a decline message including the reason for the decline and the maximum capacity available. This message, in JSON format, is shown in Figure 6-4, for example in response to the message show in Figure 6-1

If otherwise the request can be satisfied, the NC sends an acknowledgment message back to the NO via I-CPI. An example is reported in Figure 6-5, where the NC reports an error code and suggests that the maximum available capacity is 50Mb/s.

```
{'ID_Operation': 1001, 'Operation_Type': 'PWMPLSProvisioningWF', 'Source_Node': '15.0.0.1', 'Destination_Node': '10.0.1.1', 'Error_Code': 'Bandwidth full', 'Suggested_BW': '50000'}
```

Figure 6-4 Example of decline message from NC to NO

{'ID_Operation': 1001, 'Operation_Type': 'PWMPLSProvisioningWF', 'Source_Node': '15.0.0.1', 'Destination_Node': '10.0.1.1', 'Result': 'PATH_CONFIGURED', 'BW': 100000, 'Error_Code': 'NO_ERROR'}

Figure 6-5: Example of acknowledge message from NC to NO

6.2.2.2 Interaction with RYU controller

The application module interacts with the RYU controller in order to forward OpenFlow-based messages to the network devices it controls. The RYU controller generates a set of parameter in JSON format if the request is accepted, while the OpenFlow module converts such parameters into OpenFlow commands.

6.2.2.2.1 Control of the access switch

The network controller interacts with the access switch in order to add, modify or delete flow in the switching tables. This includes the management of quality of service, which is implemented through the Peak Information Rate (PIR) and Committed Information Rate (CIR) mechanisms. In this implementation, we employ two flow tables in each switch, as shown in Figure 6-6. When packets from a given flow arrive at the switch, they are first checked against flow table_id #0 for PIR. If the rate is above PIR the packets are dropped. Packets that are not dropped are then passed through table_id #1, whose task is to remark as low priority (i.e., as prec_level=0) packets that exceed the PIR value. The switch then forwards the packets to the output port defined by the switch flow table. It should be noted that two meters and two flows entries are needed for each flow in each switch.

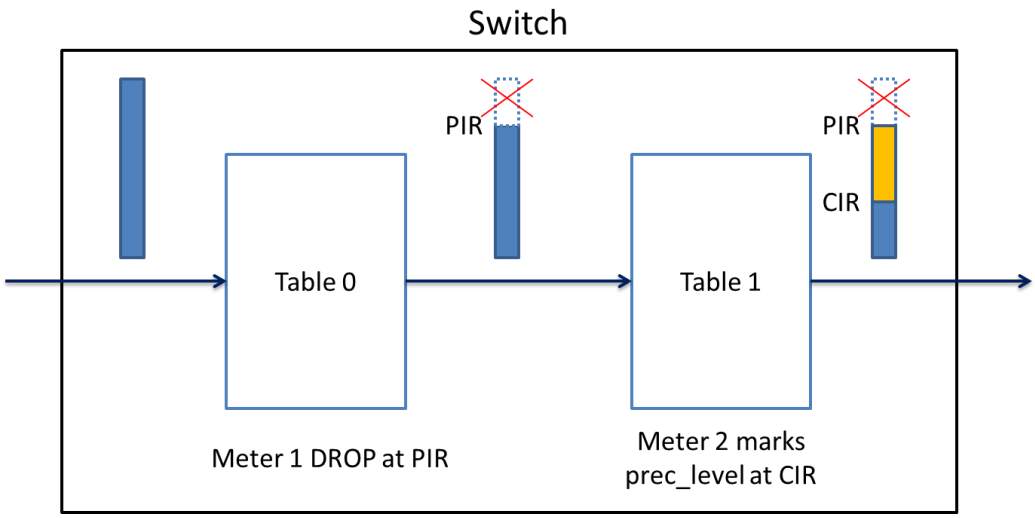


Figure 6-6: Using meters and flow tables to implement QoS

The commands to set up meters through JSON are as follows. Figure 6-7 and Figure 6-8 show the JSON message associated to the OpenFlow "OFPMC_ADD" command to add meters, respectively, for the PIR and CIR classes.

{'dpid': <*switch id*>, 'meter_id': <*meter number for PIR*>, 'flags': 'KBPS', 'bands': [{'type': 'DROP', 'rate': <*PIR*>}]}

Figure 6-7 JSON message to add meter for PIR

{'dpid': <*switch id*>, 'meter_id': <*meter number for CIR*>, 'flags': 'KBPS', 'bands': [{'type': 'DSCP_REMARK', 'rate': <*CIR*>, 'prec_level': 0}]}

Figure 6-8 JSON message to add meter for CIR

Figure 6-9 and Figure 6-10 show the JSON message associated to the OpenFlow "OFPFC_ADD" command to add flows, respectively, to the PIR and CIR meter entries. In the figure, the 'dl_type' of 34887 is used to define the Ethertype for MPLS.

{'dpid': <*switch id*>, 'table_id': 0, 'actions': [{'table_id': 1, 'type': 'GOTO_TABLE'}, {'meter_id': <*meter number for PIR*>, 'type': 'METER'}], 'match': {'in_port': <*input port number*>, 'dl_type': 34887, 'mpls_label': <*MPLS_LABEL*>}}

Figure 6-9 JSON message to add flow for PIR

{'dpid': <*switch id*>, 'table_id': 1, 'actions': [{'port': <*out port number*>, 'type': 'OUTPUT'}, {'meter_id': <*meter number for CIR*>, 'type': 'METER'}], 'match': {'in_port': < *input port number* >, 'dl_type': 34887, 'mpls_label': <*MPLS_LABEL*>}}

Figure 6-10 JSON message to add flow for CIR

Other operations on flows are those used to delete meters, through the command "OFPMC_DELETE", with parameters shown in Figure 6-11. Deletion of flow entries are instead operated through the "OFPFC_DELETE_STRICT" command, with parameters specified in Figure 6-12.

{'dpid': <*switch id*>, 'meter_id': <*meter id*>}

Figure 6-11 Parameter in JSON format to delete meter band

{'dpid': <*switch id*>, 'table_id': <*table_id*>, 'match': {'dl_type': 34887, 'mpls_label': <*pw*>, 'in_port': <*input port number*>}}

Figure 6-12 Parameter in JSON format to delete flow entry

Another functionality implemented by the NC, used for the protection scenario, is the protection mechanism that is triggered by the alarm condition from the OLT. Once the failure occurs, the OLT that connects to the failed link detects the failure through a loss of signal condition. The alarm message from the OLT to the NC, in JSON format, is shown in Figure 6-13, and is forwarded using CURL message

pycurl.URL = http://<NC IP address>:8080/stats/flowentry/chg_to_bk.

pycurl.POST = 1

pycurl.POSTFIELDS = JSON message

```
{'s_link': '102', 'd_link': '14560991494939486229', 'Operation': 'chg_to_bk'}
```

Figure 6-13 Parameter in JSON format from OLT to inform failed link to NC

In the JSON message, the failed link is identified through the source and destination entities for that link. The long number for the 'd_link' value is typical when the devices is a switch.

6.2.2.2.2 Control of the OLT

The OLT and ONU do not present native Openflow interfaces, but instead are controlled over a high-speed serial UART running at 406kbps. The Microblaze on the host FGPA boards presents an interface for directly programming and interrogating PON control registers, which are then accessible over the high-speed UART interface. Run-time control of the PON is executed through the interfaces on the OLT which in turn relay control instructions to the remote ONU using PLOAM messages. The run-time functionality includes configuration of the laser frequencies of the OLT and ONU tunable lasers, the configuration of Alloc_id at the ONU for appropriate XGEM packets, and the re-homing of ONU from one OLT to another.

An Openflow agent wrapper around both OLT1 and OLT2 was developed so as to present an Openflow v1.4 compatible interface to the Metro-Access Controller. Openflow v1.4 facilitates the control of optical parameters of Openflow compatible switches and devices through the OFPPortModPropOptical method. These parameters include the transmission centre frequency or wavelength, a frequency offset from the centre frequency and the transmission power level (dB) and are a subset of those which we are looking to control within the PON. Because we need runtime control of additional non-standard parameters, we enhanced both the v1.4 protocol and the agent to allow configuration of the XGEM, Alloc_ID and PseudoWire tags associated to a given flow through the Metro-Access Controller

Logically, using the paradigm of Software Defined Networking, the Metro Access controller communicates with the OLTs through the OpenFlow agent translating OpenFlow commands into proprietary control messages sent through the UART interface, such as the set up of wavelengths and the set up of protection paths.

6.2.2.2.2.1 PON (OLT and ONU) interface

The communication with the OLT and ONU is accomplished through a single Class FPGA, through which a number of methods are defined. Instantiation of the class opens a serial interface port to the particular device. The sendcmd() method is the base method to issue a string to the interface and receive back a response. sendcmd() is used by almost all other FGPA methods to issue commands and gather responses. The raw FPGA interface presents a menu structure, which is invoked by the FPGA class. The following table outlines the range of based methods which may issued on the PON devices.

Method	Explanation
$device->reset()	Resets Device
$device->enable_mpls()	Enables MPLS tagging interpretation on the device. This command is issued at the head-end OLT.

$device->disable_mpls()	Disables MPLS tagging interpretation on the device. This command is issued at the head-end OLT.
$device->set_ds_laser($fpga_channel)	Sets downstream laser to channel designated by $fpga_channel (Skylane mapping). This command is issued at the head-end OLT.
$device->set_us_laser($onu_id, $fpga_channel)	Sets downstream laser to channel designated by $fpga_channel (Skylane mapping) on the ONU $onu_id. This command is issued at the head-end OLT.
$device->set_alloc_id($onu_id, $alloc_id)	Sets the alloc id / XGEM port on the ONU $onu_id
$device->create_flow($mac, $xgem, $mpls_tag, $cam)	Creates a flow denoted by $mac mac address, XGEM port $xgem, MPLS tag $mpls_tag on cam $cam. This command is issued at the head-end OLT.
$device->delete_flow($cam)	Deletes flow in cam $cam_id
$device->dwa_set()	Completes the set up of the DWA mapping on the PON. This command is issued at the head-end OLT.
$device->dwa_reset()	Resets the DWA mapping on the PON. This command is issued at the head-end OLT.
$device->getstatus()	Returns the status of the device. This may be on any deivce
$device->readreg()	Returns the register value. This may be on any device

6.2.2.2.2.2 OpenFlow agent

As stated, the Openflow standard V1.4 introduced support for querying and managing the characteristics of Optical ports. This includes the ability to configure and monitor transmit and receive frequency of a laser, as well as its power. The Optical ports may be Ethernet based optical ports (i.e. Ethernet through SFP+ interfaces) or optical ports on circuit switches. This allows an Openflow controller to configure the optical ports through the Optical port mod property (ofp_port_mod_prop_optical), monitor the optical ports through Optical port stats property (ofp_port_stats_prop_optical) and to describe the optical ports through the Optical port description property (ofp_port_desc_prop_optical).

6.2.2.2.2.2.1 Data structures

To support the mapping between the real device, OF_Agent has a number of data structures defined on a per-port basis to handle the attributes required by the Openflow Controller.

6.2.2.2.2.2.2 Port Capabilities

The **ofp_port_desc_prop_optical** data structure stores attributes for

- Minimum TX Frequency/Wavelength
- Maximum TX Frequency/Wavelength
- TX Grid Spacing Frequency/Wavelength

- Minimum RX Frequency/Wavelength
- Maximum RX Frequency/Wavelength
- RX Grid Spacing Frequency/Wavelength
- Minimum TX power
- Maximum TX power
- Features

Features is a bit mask, with the bits representing the capabilities of the particular port. The right-hand most bit (OFPOPF_RX_TUNE) denotes if the Receiver is tunable or not (OFPOPF_TX_TUNE), the second bit if the transmitter is tunable or not (OFPOPF_TX_PWR), the third bit if the Power is configurable and lastly, the fourth bit denotes whether to use frequency or wavelength (OFPOPF_USE_FREQ). For the OF_Agent, all four bits are set to the on position. The minimum, maximum, and grid spacing are specified for both transmit and receive optical ports as either a frequency in MHz or wavelength (lambda) as nm * 100. For ports that are not tunable, the minimum and maximum values need to be identical and so specify the fixed value. The tx_pwr_min and tx_pwr_max are dBm * 10.

6.2.2.2.2.2.3 Port Configuration

Messages that configure the optical ports are implemented using the **ofp_port_mod_prop_optical** data structure in the OF_Agent. The attributes include

- OFPOPF Configure – The configure field describes optical features to change for this port. Frequency is specified in MHz, wavelength (lambda) as nm * 100. Also OFPOPF_USE_FREQ bit match the advertised port feature in the **ofp_port_desc_prop_optical** data structure.
- Centre Frequency
- Signed frequency offset
- Size of the grid for this port
- TX power setting

6.2.2.2.3 OF_Agent functionality

The OpenFlow Agent (OF_Agent) is composed of 2 main sets of Classes. The first set of classes details the messages which are exchanged between OF_Agent and the upstream Openflow Controller. OF_Agent acts as a client, and the Openflow Controller acts as the server in the relationship. It is the function of OF_Agent to initiate the relationship, by contacting the controller. It is the function of the OF_Agent to sufficiently mediate between the non-native Openflow device and the controller, and behave in a manner that will give the impression that the device is a real Openflow Switch.

6.2.2.2.3.1 Behaviour

The overall behaviour of the Openflow Agent is described by the OF_Agent class. It has two main routines or methods, wait_on_controller() and get_optical_message(). On instantiation, the OF_Agent makes a network connection to the standard TCP listening port for Openflow (6633).

The wait_on_controller() routine is a loop that waits for Openflow messages. Once the upstream Openflow controller accepts the TCP connection, it issues an Openflow Hello message to which OF_Agent responds with Hello. Thereafter, wait_on_controller() handles responses to upstream controller requests for message types OFPT_ECHO_REQUEST, OFPT_BARRIER_REQUEST, OFPT_FEATURES_REQUEST, OFPT_PORT_MOD, OFPT_MULTIPART_REQUEST and OFPT_FLOW_MOD. Wait_on_controller() uses the OFPT_PORT_MOD message with subtype OFPT_PORT_MOD_OPTICAL as a trigger to generate a message (get_optical_message()) to the downstream Optical device, for which the OF_Agent is acting as agent. get_optical_message() is a stub, which is over-ridden when OF_agent is instantiated.

6.2.2.2.3.2 Messages

All Openflow messages follow a common format, however, the format has grown from a simple one format in the 0x01 wire standards to the more complex formats of wire standards 0x04 and later. The increase in complexity of the messages reflects the complexity of the type of functions and controls which more recent type of Openflow switches are required to exercise. Switches are required to be stateful, and have wider varieties of port attributes. An example of complex messaging introduced is the Barrier Message, which must be interpreted by OF_Agent. When the controller wants to ensure message dependencies have been met or wants to receive notifications for completed operations by OF_Agent, it uses an OFPT_BARRIER_REQUEST message. This message has no body. Upon receipt, the OF_AGent must finish processing all previously-received messages, including sending corresponding reply or error messages, before executing any messages beyond the Barrier Request. When such processing is complete, the OF_Agent must send an OFPT_BARRIER_REPLY message with the transaction id of the original request.

The base class of OF_Agent message is ofPkt. All messages derive their attributes for transaction ID and payload from the ofPkt Class. OfPkt also details the versions of Openflow which the OF_Agent can sustain. It is possible for the OF_Agent and the Controller to negotiate to the most optimal version of Openflow to be used in the relationship.

All messages have a common header with fields for the size of the message as well as detailing the message type. Depending on the message type, the payload of the message will vary in length as well as format. Depending on the switch characteristics the size of messages can vary also. For example, there is a common format of data structure and command structure to manipulate a single port, however, different switches have different numbers of ports so the size of the total structures is sufficiently expandable.

6.2.2.2.3.3 Upstream Message types

Messages can be characterised as either synchronous or asynchronous. Asynchronous messages can be sent by either the OF_Agent or the Controller, and elicit a response from the other party. Example of asynchronous messages are Hello and EchoRequest. Hello elicits Hello in return, and is used to initiate the relationship between the OF_Agent and the Controller, however in practice, this usually is initiated by the Openflow controller. EchoRequest elicits EchoReply in response and is typically used as a health check or keep-alive message.

Synchronous messages generally are initiated by the controller, once the relationship has been created. Synchronous messages are used to elicit information from the OF_Agent by the Controller. Examples of Synchronous messages are OFPT_FEATURES_REQUEST, OFPT_PORT_MOD and OFPT_FLOW_MOD. OFPT_FEATURES_REQUEST is the request for the characteristics of the switch as well as the capabilities of all ports.

6.2.2.2.3.4 Downstream Message types

The main downstream message type is get_optical_message() which is generated when OF_Agent receives a OFPT_PORT_MOD_OPTICAL message from the controller. get_optical_message() takes the following parameters :

- target port number (port_no).
- switch address (addr) or data path Identifier (dpid),
- wavelength or frequency to be tuned (freq_lmda),
- offset frequency or lambda (fl_offset),
- grid span (grid_span). The grid_span is the amount of bandwidth consumed by this port, which is mainly used for Flex Grid.
- transmission power at which laser is transmitted or generatered (tx_pwr), measured as dBm * 10

The tuned frequency is the sum of freq_lmda and fl_offset. For some tuning options, including Flex Grid, the tuned frequency is based upon a centre frequency and an offset. The centre frequency is also in passive filters.

6.2.2.2.3.5 Control of the OF_Agent

To control the OF_Agent, the standard Openflow V1.4 syntax is used. Firstly, the Feature Mask should be set to include capabilities for OFPPC_PORT_DOWN, OFPPC_NO_RECV, OFPPC_NO_FWD and OFPPC_NO_PACKET_IN. Secondly the OFPPortModPropOptical properity is built up based on freq_lmda, fl_offset, grid_span, tx_pwr. Lastly, the OFPPortMod method is called on the OF_Agent datapath ID (or the MAC address) with the OFPPortModPropOptical properites and Feature Mask just described. The OFPPC_PORT_DOWN bit indicates that the port has been administratively brought down and should not be used by OpenFlow. The OFPPC_NO_RECV bit indicates that packets received on that port should be ignored. The OFPPC_NO_FWD bit indicates that OpenFlow should not send packets to that port. The OFPPFL_NO_PACKET_IN bit indicates that packets on that port that generate a table miss should never trigger a packet-in message to the controller.

6.2.3 Database module

The database module stores all information from the MC node on routing, wavelengths, capacity, MPLS labels, and detail of flows and meters being used. There are two kinds of database implemented, one for infrastructure configuration and another for network state information.

1.1.1.1 Infrastructure configurations DB

This database stores information that is statically defined by a management or network control system and consists of the following tables: PATHS, WAVELENGTHS, SW_PORTS and HOST_IP.

The **PATHS** table defines the route between sources and destinations. While in a more complex implementation this table should be dynamically allocated by routing protocols running information gathered by discovery protocols, in this case they are allocated through a simple shortest path algorithm.

Table 8 Example of information in the PATHS table

id	src_IP	dst_IP	pri_path
37	15.0.0.1	10.0.1.1	1,SW1,SW2, 2
40	15.0.0.1	10.0.1.2	1,SW1,SW2, OSW1, 3

src_IP indicates an the source IP address of the path, for example the address of a SP video server; *dst_IP* indicates the IP address of the destination path, for example the address of an OLT that terminates a PW paths; *pri_path* defines a path between *src_IP* and *dst_IP* (this uses internal addressing, which are mapped in the HOST-IP table): packet switches are indicated as SW, while optical switches like OSW in this table.

The **WAVELENGTH** table associates wavelength ITU-T IDs with the wavelength ID recognised by the SFP+ tunable component.

Table 9 Example of information in the WAVELENGTH table

wavelength_id	ITU_ID	Wavelength (nm)	SFP_ID
1	21a	1560.61	21
2	21b	1560.2	22

wavelength_id defines an ID number for the wavelength; *ITU_ID* is the standard wavelength ID defined by the ITU, *Wavelength* defines the value of the wavelength expressed in nm; *SFP_ID* is the wavelength identifier used for controlling the tunability of the SFP+ transceiver.

The **SW_PORTS** table defines the port connectivity of each switch. In our case this is statically allocated, while in more complex implementations this information could be retrieved through discovery protocols.

Table 10 Example of information in table "SW_PORTS"

SW_ID	ports_connection
SW2	0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,OSW1,0,0,0,0,0,0,SW1, 1, 2
SW1	0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,OSW1,0,0,0,0,0,0, 1,0,SW2,0,0,0,0
OSW1	0,0,0,0,0,0,0,0,0,0,SW1,0,0,0,0,SW2,0,0,0,0,3,0,0,0

SW_ID shows the id number of a switch; *ports_connection* defines the port number of the switch and SW_ID that connects to the switch. In the example, SW2 is connected to host 3 at port 45, to OSW1 at port 39, to SW1 at port 46, and to host 2 at port 47; SW1 is connected to OSW1 at port 35, to host 1 at port 42 and to SW2 at port 44; OSW1 connects to SW1 at port 11, to SW2 at port 16 and to device 3 at port 21.

The **HOST_IP** table defines the IP address for each host.

Table 11 Example of information in the HOST_IP table

host_id	ip	PRI_OLT_IP	BK_OLT_IP
1	15.0.0.1		
2	10.0.1.1		
3	10.0.1.2		
4	10.0.0.3	10.0.1.1	10.0.1.2

host_id indicates the ID number of the host (it could be an SP server, ONU, or OLT); *ip* shows the IP address of the host; *PRI_OLT_IP* shows an IP address of a primary OLT (if the device is an ONU); *BK_OLT_IP* shows an IP address of a backup OLT (if the device is an ONU).

1.1.1.2 Network state information DB

This database contains the information used to identify the dynamic state of the network, including the flow table, the meter table, and the capacity table. This information is added, removed or modified dynamically following network operations. It consists of the following tables: MPLS_LABEL,

The **MPLS_LABEL** table defines LSP and PW numbers for each flow.

Table 12 Example of information in the MPLS_LABEL table

src_IP	dst_IP	lsp	pw	traffic_type
15.0.0.1	10.0.1.1	5001	2001	Internet

15.0.0.1	10.0.1.1	5002	2002	VoD

src_IP indicates the IP address of the source of the MPLS PW path; *dst_IP* indicates the IP address of the destination of the MPLS PW path; *lsp* defines the higher level LSP that containes the PW; *pw* defines the label for the path, *traffic_type* indicates the traffic type carried by the PW.

The **SERVICES** table stores information on active services in the MC node.

Table 13 Example of information in the SERVICES table

ID_Operation	SW_ID	Operation_Type	meter1	Meter2	flow_id	wavelength_no	pw	dst_host_id
1001	SW1	PWMPLSProvisioningWF	101	102	37	1	2001	4

ID_Operation is a reference ID number associated to an incoming request and is automatically generated after the NC receives the request; *Operation_Type* indicates the type of service; *meter1* and *meter2* are meter numbers being used by the following *flow_id*, controlling, respectively, PIR and CIR; *flow_id* refers to the *id* in table FLOWS; *wavelength_no* refers to *walength_id* in table WAVELENGTH; *pw* refers to the *pw* field in table MPLS_LABEL; *dst_host_id* refers to the destination ONU ID.

The **METERS** table stores the configuration of the OpenFlow meters. Meters are unidirectional, thus if Qos is required on both direction of a connection, it is configured into two separate entries.

Table 14 Example of information in the METERS table

ID_Operation	SW_ID	stream	meter_id	band_type	rate	arguments
1001	SW1	15.0.0.1 to 10.0.1.1	101	DROP	6000000	
1001	SW1	15.0.0.1 to 10.0.1.1	102	DSCP_REMARK	4000000	prec_level=0

ID_Operation is the reference ID number for the request associated to the meter; *SW_ID* is the ID number of the switch where the meter is configured; *stream* indicates the direction for the meter (i.e., form source IP to destination IP address); *meter_id* is an identifier for the meter and is automatically generated after a request; *band_type* defines the action on the packet, (i.e. DROP for PIR or DSCP_REMARK for CIR); *rate* is the capacity limit for the meter considered; *arguments* indicates an optional argument for the meter (i.e., following the OpenFlow syntax).

The FLOWS table stores the flows in use on the OpenFlow switches, indicating both matching condition and meters.

Table 15 Example of information in the FLOWS table

id	SW_ID	ID_Operation	stream	match_condition	meter1	meter2
37	SW1	1001	15.0.0.1 to 10.0.1.1	{'dl_type': 34887, 'mpls_label': 5001, 'in_port': 42}	101	102
38	OSW1	--	--	{‘in-port’: 21, ‘out_port’24}	--	--

id is a reference number for the flow; *SW_ID* is the ID number of the switch where the flow is installed; *ID_Operation* is the reference ID number of the request; *stream* indicates the direction of the flow (from source IP to destination IP address); *match_condition* refers to match field used in the OpenFlow table for the switch; *meter1* and *meter2* are meter numbers attached to the flow, controlling, respectively, PIR and CIR capacity.

The **BW** table stores the available capacity for each wavelength on each link in the network (i.e., both for PON channels and for capacity between switches or between switches and other hosts). It is calculated and updated by the NC upon every new request.

Table 16 Example of information in the BW table

s_link	d_link	MAX	wavelength_ID	available_capacity	ID_Operation
1	SW1	10000000	1	4000000	1001,1002
SW1	SW2	10000000	1	5000000	1001,1002
SW1	SW2	10000000	2	2000000	1001,1002
SW2	2	10000000	1	4000000	1001,1002
3	4	10000000	1	6000000	1003
3	4	10000000	2	5000000	1004

s_link is device_id at the beginning of the link; *d_link* is device_id at the end of the link; *MAX* is the total capacity of the link in bps; *wavelength_ID* is the identifier for the wavelength used; available_capacity shows remaining CIR bandwidth in bps.

It should be noted that multiple wavelengths are simply added on a link by adding further rows.

The BW table is used to assess the current available capacity on a end-to-end connection. For example, in the case of a VoD request with defined CIR and PIR parameters, the NC will first determine the end-to-end path from the PATH table. It then checks the available capacity on each link making up the end-to-end path in the BE table. The availability of bandwidth of a wavelength on a path is the minimum available bandwidth of the set of links on the path. For example, if the path between a source and destination is 1→SW1 →SW2 →2, the availability of bandwidth on wavelength_ID 1 between the two intermediate switches is 5 Gbps, while on wavelength_ID 2 is 2 Gbps. Considering that the available capacity from 102 and to 102 are 4 Gb/s, the maximum end-to-end link capacity that can be instantiated will be 2Gb/s on wavelength_ID 2 or 4Gb/s if using wavelength_ID 2.

6.3 Example scenario

This subsection reports an example showing the control plane interactions for a request for a Bandwidth-on-Demand (BoD) service. It is assumed that while an ONU is receiving a service with a given reserved capacity (which we refer to as service-A) by an OLT (which we refer to as OLT-1), it also requests capacity for an additional service, for example Bandwidth-on-demand (which we refer to as service-B). The scenario is setup so that the wavelength channel used by the ONU does not have enough capacity to support the requested BoD service. The controller needs thus to activate a second OLT (which we refer to as OLT-2), in order to deliver service B. Since the ONU only has one (tunable) transceiver, both services A and B need to be delivered over the same wavelength, by OLT-2. This implies redirecting

service A to OLT-2, which requires communication with the SP to move service A into a different PW which can be routed to OLT-2. The aim of the experiment is to show the ability of OpenFlow and SDN to handle such end-to-end service provisioning and to measure the time required to provision service-B and the impairment generated for rerouting service-A.

A logical view of the testbed components and operations is shown in Figure 6-14, showing the different steps required to provision the service[4].

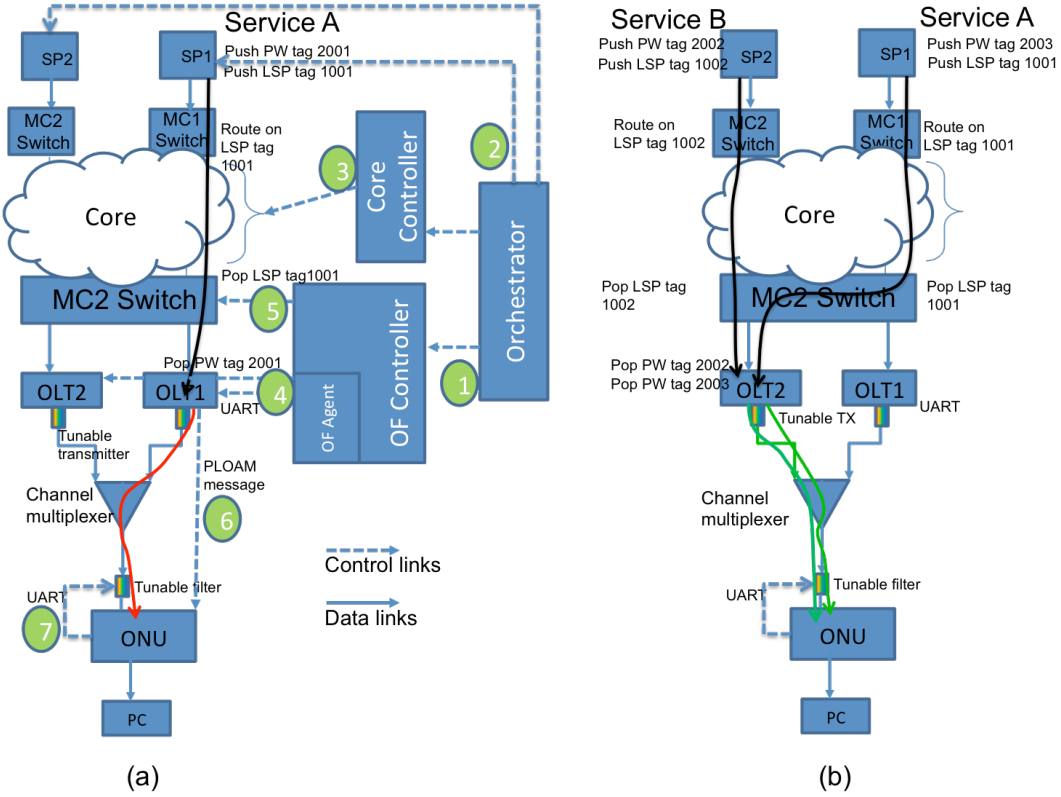


Figure 6-14 Operation of SDN control plane to provision service B while operating service-A; (b) network state after provisioning of service-B and re-direction of service A to OLT-2

Following the user request for a new BoD service-B, the Orchestrator forwards a JSON message to the OF controller (step 1) indicating the capacity requested and the range of wavelengths the ONU can support (Figure 6-15): this is a request for a PW with 8Gb/s assured capacity, unidirectional, with a 'necessitated' parameter for a new wavelength between 1556 and 1561nm. The necessitated parameter (introduced in section 3.4 of this document), indicates that a new wavelength channel should only be used if no capacity is available in the current channel.

[4] Although we described the process in sequential steps, some of the actions described are actually carried in parallel in order to speed up the provisioning process.

{'Traffic_Type': 'BoD', 'Operation_Type': 'PWMPLSProvisioningWF', 'Source_Node': '15.0.0.1', 'Destination_Node': '10.0.1.1', 'ONU_IP': '10.0.0.1', 'Direction_Type': 'Unidirectional', 'Operation': 'add', 'New_Lambda': 'Necessitated', 'Wavelength_Range_from': '1556', 'Wavelength_Range_to': '1561', 'Bandwidth': {'CIR': '8000000', 'PIR': '10000000'}, }

Figure 6-15 Request to increase bandwidth by adding dedicated traffic type

Once the controller has verified the capacity availability it acknowledges the orchestrator, which communicates with SP2 to activate the new service-B and with SP1 to reroute service-A into a different PW (Step 2). The orchestrator then informs the core controller to add a new LSP path for routing service-B to the correct MC node (step 3) (Figure 6-16): the first two lines in the meter table indicate the current CIR and PIR requirements for this LSP; in the flow table, the first entry indicates to apply the PIR meter and then go to the second table, while the second entry applies the CIR meter and sends the packets to the output port.

Meter entries

{'dpid': 7461348157366609941, 'meter_id': 101, 'flags': 'KBPS', 'bands': [{'type': 'DROP', 'rate': 10000000}]}

{'dpid': 7461348157366609941, 'meter_id': 102, 'flags': 'KBPS', 'bands': [{'type': 'DSCP_REMARK', 'rate': 8000000, 'prec_level': 0}]}

Flow entries

{'dpid': 7461348157366609941, 'table_id': 0, 'actions': [{'table_id': 1, 'type': 'GOTO_TABLE'}, {'meter_id': 101, 'type': 'METER'}], 'match': {'in_port': 42, 'dl_type': 34887, 'mpls_label': 1002}}

{'dpid': 7461348157366609941, 'table_id': 1, 'actions': [{'port': 44, 'type': 'OUTPUT'}, {'meter_id': 102, 'type': 'METER'}], 'match': {'in_port': 42, 'dl_type': 34887, 'mpls_label': 1002}}

Figure 6-16 Flow and meter entries for switch in MC-2

In the meantime the access controller communicates with OLT-2 through an OpenFlow agent that translates OF commands into proprietary control messages sent through the UART interface, indicating to activate its service on the desired wavelength (step 4). The controller then installs the new entry for the PW, including meter tables for QoS and capacity reservation in the switch flow table (step 5) (Figure 6-17): the first two entries determine the PIR and CIR for service-B, while entries three and four determine PIR and CIR for the rerouted service-A (using a new PW 2003); the first three entries of the flow table are associated to service-B: the first entry pops the LSP label and forwards the packet for processing to table 1, where the PIR policy is applied and the packet forwarded to table 2, which applies the CIR policy and sends the packet out; the last three entries carry out the same operations but on the flow associated to service-A.

In the meantime the ONU is informed, by a PLOAM message sent by OLT-1, to tune the filter to the new wavelength (step 6) which triggers a message from the ONU to the tunable filter (this is an external link in our testbed setup) via a UART interface (step 7).

Meter entries

{'dpid': 14560991494939486229, 'meter_id': 101, 'flags': 'KBPS', 'bands': [{'type': 'DROP', 'rate': 10000000}]}

{'dpid': 14560991494939486229, 'meter_id': 102, 'flags': 'KBPS', 'bands': [{'type': 'DSCP_REMARK', 'rate': 8000000, 'prec_level': 0}]}

{'dpid': 14560991494939486229, 'meter_id': 103, 'flags': 'KBPS', 'bands': [{'type': 'DROP', 'rate': 100000}]}

{'dpid': 14560991494939486229, 'meter_id': 104, 'flags': 'KBPS', 'bands': [{'type': 'DSCP_REMARK', 'rate': 50000, 'prec_level': 0}]}

Flow entries

{'dpid': 14560991494939486229, 'table_id': 0, 'actions': [{'ethertype': 34887, 'type': 'POP_MPLS'}, {'table_id': 1, 'type': 'GOTO_TABLE'}], 'match': {'in_port': 46, 'dl_type': 34887, 'mpls_label': 1002}}

{'dpid': 14560991494939486229, 'table_id': 1, 'actions': [{'meter_id': 101, 'type': 'METER'}, {'table_id': 2, 'type': 'GOTO_TABLE'}], 'match': {'in_port': 46, 'dl_type': 34887, 'mpls_label': 2002}}

{'dpid': 14560991494939486229, 'table_id': 2, 'actions': [{'port': 47, 'type': 'OUTPUT'}, {'meter_id': 102, 'type': 'METER'}], 'match': {'in_port': 46, 'dl_type': 34887, 'mpls_label': 2002}}

{'dpid': 14560991494939486229, 'table_id': 0, 'actions': [{'ethertype': 34887, 'type': 'POP_MPLS'}, {'table_id': 1, 'type': 'GOTO_TABLE'}], 'match': {'in_port': 26, 'dl_type': 34887, 'mpls_label': 1001}}

{'dpid': 14560991494939486229, 'table_id': 1, 'actions': [{'meter_id': 103, 'type': 'METER'}, {'table_id': 2, 'type': 'GOTO_TABLE'], 'match': {'in_port': 26, 'dl_type': 34887, 'mpls_label': 2003}}

{'dpid': 14560991494939486229, 'table_id': 2, 'actions': [{'port': 47, 'type': 'OUTPUT'}, {'meter_id': 104, 'type': 'METER'}], 'match': {'in_port': 26, 'dl_type': 34887, 'mpls_label': 2003}}

Figure 6-17 Flow and meter entries for MC-2 switch

Finally, once the operations are completed at the lower interface, the NC returns an acknowledgment to the orchestrator (Figure 6-18), reporting the ID_Operation, which was provided riginally by the NC, the IP address of the OLT (Destination_Node), wavelength id, and bandwidth (BW).

{'ID_Operation': 1002, 'Operation_Type': 'PWMPLSProvisioningWF', 'Source_Node': '15.0.0.1', 'Destination_Node': '10.0.1.1', 'Result': 'L1_PATH_CONFIGURED', 'Wavelength_id': 2, 'BW': '8000000', 'Error_Code': 'NO_ERROR'}

Figure 6-18 Acknowledgement

7 Appendix 2 Optical Switch Implementation

This appendix provides detailed test results on the prototype single-sided optical switches created for the DISCUS project under task 6.2. Polatis objectives in this task have been to:

- Design low-loss 3-stage switches scalable to over 3000x3000 ports
- Design and develop technology for 192x192 switching matrix to support 3-stage switches with port count above 12000x12000
- Design and develop a sparsely populated 3-stage switch to be used in the WP8 testbed
- Contribution to the design of the OpenFlow control plane and interfaces

As summarised in section 3 above and described in detail in earlier deliverables, a key feature of the DISCUS metro-core node architecture requires large-scale software-defined fibre layer connectivity with of order 10,000 endpoints to provision and protect network resources on demand. Architecture modelling presented in D6.2 showed that a single-sided (any fibre to any fibre) optical layer switching fabric has more attractive scaling properties than a conventional double-sided Clos network, where ingress and egress ports are defined in separate groups. Because of the industry-leading optical loss of the Polatis DirectLight optical switch platform (typically 1dB), it becomes feasible to design multi-stage optical switch fabrics that can scale to meet this requirement, but with an impact on median link loss budgets of only 4-5dB including interconnects. Consequently, the focus in task 6.2 has been to

a) extend the pre-existing 192x192 optical cross-connect platform to design and implement single-sided optical switch modules with embedded OpenFlow agents;
b) explore elements required to further improve the density and energy efficiency of the DirectLight technology; and
c) study control and interconnect methods for large-scale multi-element optical cross-connects.

During the project, three prototype single sided optical switch modules have been designed, to provide non-blocking connectivity between 48, 96 and 192 fibre ports, respectively. These switches are being made available for the final test bed in WP8.

In addition, two further optical switches have been provided to project partners for systems experiments: a 24x24 (serial 1709, part code N-VST-24x24-LU1-MMHNS-300) to Trinity College Dublin and a 32x32 (serial 0896, part code I-VST-32x32-FA1-GSENS-200) to Tyndall National Institute.

In the following sections, the design of single-sided optical switch modules is outlined and the resulting optical performance is detailed. Control of the optical switch elements using an embedded OpenFlow agent in a software defined network is described, along with the initial integration with a multi-module SDN application running on top of an OpenDaylight controller.

7.1 Single-sided Optical Switch Design

Polatis DirectLight technology is a versatile, ultra-low loss optical switching platform that can be configured to create fully transparent strictly non-blocking fibre layer cross-connects with arbitrary MxN matrix sizes ranging from 4x4 to 192x192 fibre ports.

As described more fully in D6.2, DirectLight is a 3-dimensional beam-steering technology for optical matrix switches [15],[16], combining patented piezoelectric actuation with integrated position sensors to make connections between 2D arrays of collimated fibres directly in free space. Significantly, switching occurs completely independently of the power level, colour or direction of light on the path, enabling pre-provisioning of dark fibre paths and avoiding concatenation of switching delays across mesh or multi-stage switch networks.

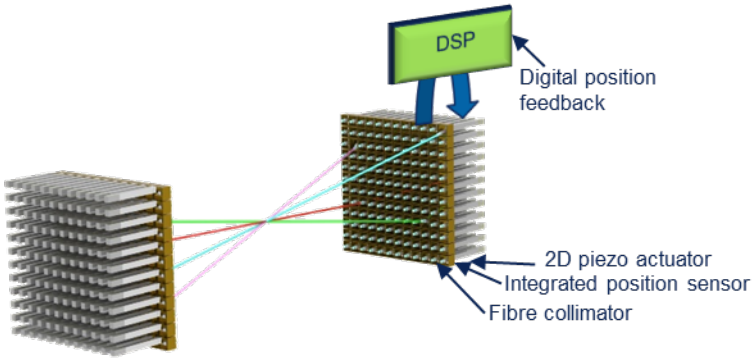


Figure 7-1 DirectLight beam-steering optical switch

The principle of operation is shown in Figure 1, where opposing 2-D arrays of fibre-pigtailed collimators are individually steered by piezoelectric actuators via a low-stress flexure pivot. Voltages applied to the actuators independently control collimator orientation in two angular dimensions. The pointing angles of the actuators are monitored by high accuracy absolute position sensors.

Conventional DirectLight double-sided optical matrix switches contain separate ingress and egress fibre arrays, which are built up in rows using modular slices containing 12 fibre ports. This allows configuration of a wide range of switch matrix sizes to address multiple requirements while maintaining a consistent cost per port.

During factory alignment, the optical switch elements are trained to find the optimum target positions for every path between ingress and egress ports. Target values are stored in memory and are used by a digital control loop to drive and hold the actuators in the correct position for each connection, regardless of the light level on the fibre path. Variable attenuation can be introduced on a connected path if required by controlled misalignment of one or more axes from the optimum target position.

A single-sided optical switch core where any fibre can connect to any other fibre can be realized by adding a plane optical quality mirror or interleaving element in the beam path [17], so that ingress and egress arrays are degenerate. The following sections describe the design of 48-, 96- and 192-fibre single sides switches developed in the DISCUS project using arrays of the same modular 12-port slice assembly.

7.1.1 48-fibre single-sided optical switch

A 48-fibre single sided optical switch can be realised by using a plane mirror to reflect beams from an array of four 12-port slices back onto itself, as shown in figure 2. In this way, a fibre can act as both an ingress and egress port and connect to any other fibre in the array.

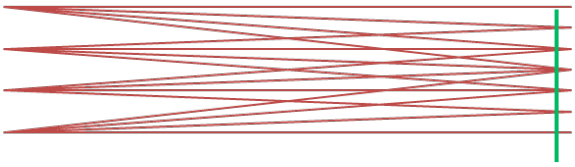


Figure 7-2 Vertical beam paths between four 12-port slices in a single-sided 48-fibre optical switch module

The folded core arrangement has been used to realise a prototype 48-fibre module with a compact integrated controller that incorporates a highly efficient piezoelectric driver ASIC (ARGOS) developed by Polatis and IMEC on a separate project. The low profile module fits comfortably within the popular 1RU height for rack-mount equipment.

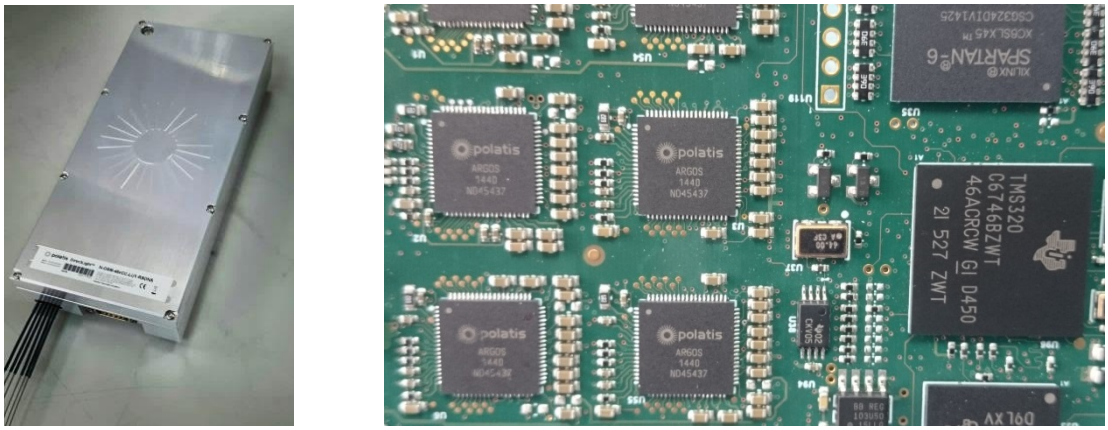


Figure 7-3 a) A 48-fibre single-sided optical switch module using a single board ASIC-based controller; b) Detail of the controller card showing the IMEC-developed Polatis DirectLight driver ASICs

7.1.2 96-fibre single-sided optical switch

A 96-fibre single sided optical switch can be realized by adding an interleaving element in the beam path between two 48-fibre arrays, as illustrated in Figure 4. Connections to the opposite array are shown as blue paths, whereas reflective connections to ports on the same array are shown in red. Care is taken to ensure that the reflective stripes are interleaved exactly between the intersections of the transmissive paths at the midpoint. This core is essentially a special case of the 1RU 48x48 switch, but with degenerate input and output arrays.

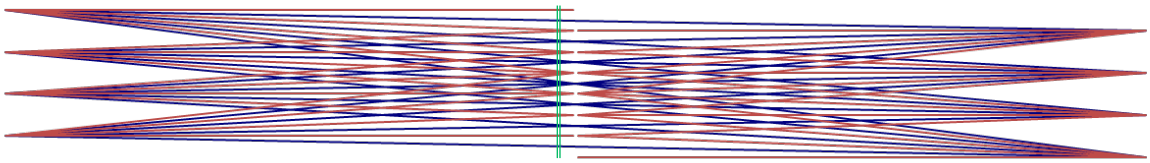


Figure 7-4 Vertical beam paths between two arrays of four 12-port slices in a single-sided 96-fibre optical switch module

7.1.3 192-fibre single-sided optical switch

A 192-fibre single sided optical switch can be realized by adding an interleaving element at the midpoint in the beam path between two 96-fibre arrays, where each array consists of eight 12-port slices as illustrated in Figure 5. A prototype 192-fibre single-sided switch was built and integrated into a 3RU chassis as shown in figure 6. The unit was first demonstrated publically at OFC 2014 and formally launched as a product at ECOC 2014.

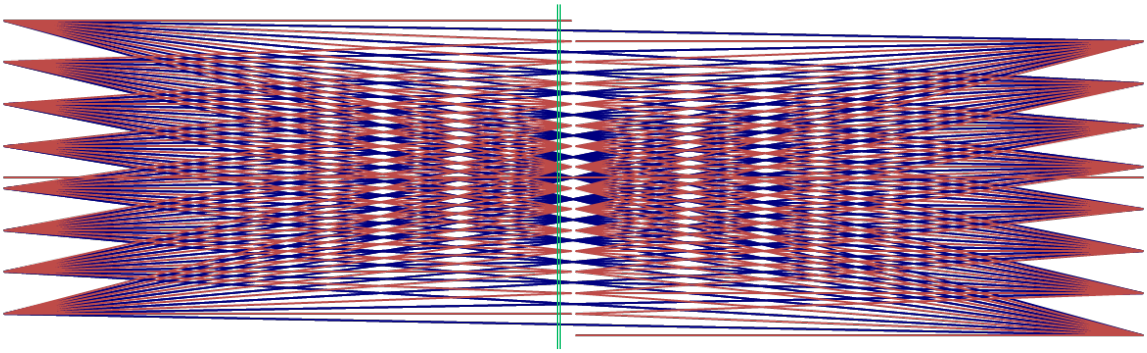


Figure 7-5 Vertical beam paths between two arrays of eight 12-port slices in a single-sided 192-fibre optical switch module

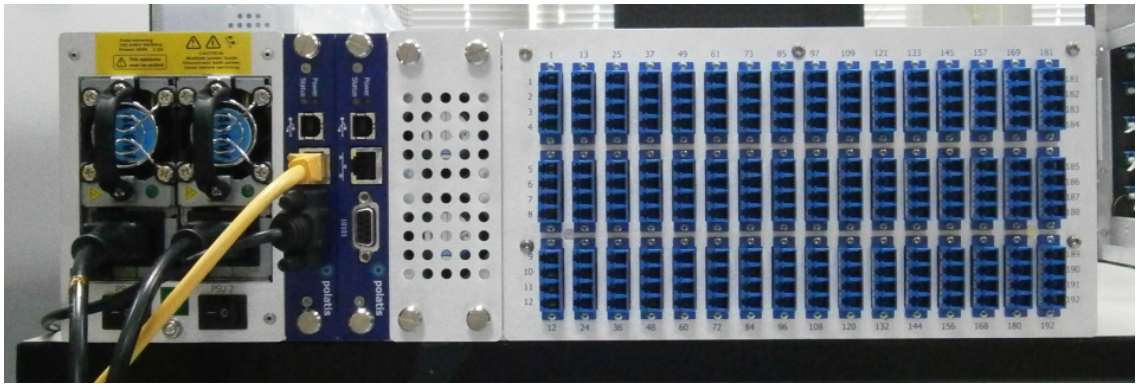


Figure 7-6 Rear panel of prototype 192-fibre single sided optical switch

7.2 Single-sided Optical Switch Test Results

This section summarises the test results from the three prototype single sided optical switch module variants designed on the DISCUS project, which provide non-blocking connectivity between 48, 96 and 192 fibre ports, respectively. A test station was commissioned to provide full bidirectional characterisation of single sided optical switches up to 192 fibre ports. Associated changes were also made to the relevant software test tools. Figure 7 shows the test station configuration with a fully-connected 96-fibre 1RU optical switch under test (arrowed) together with a close up of the test tool GUI.

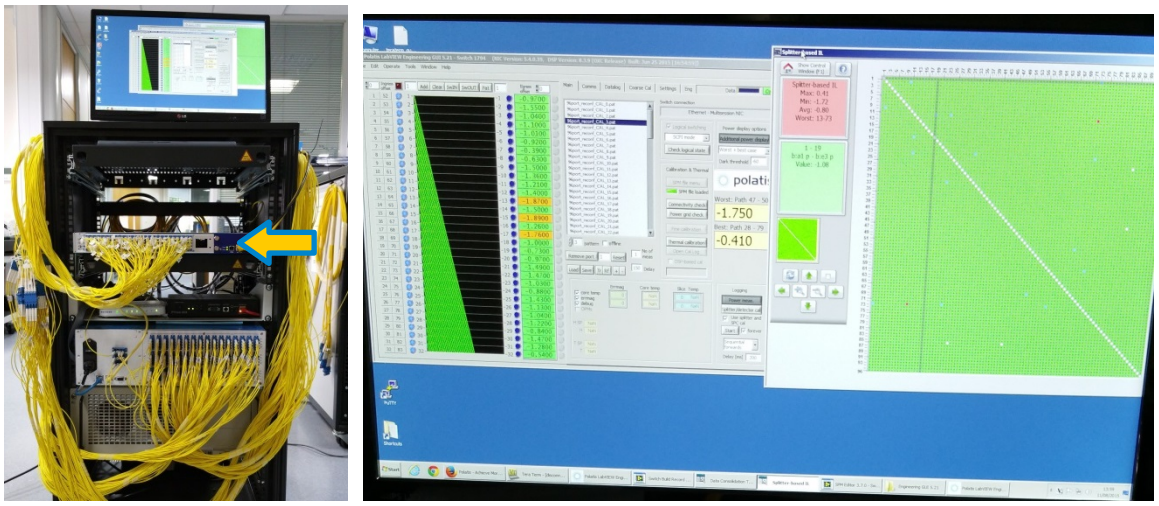


Figure 7-7 a) Bidirectional 192-fibre test station for single-sided optical switches with a 1RU 96-fibre OXC connected for characterisation; b) Close-up of test interface showing windows for loss measurement and connectivity grid traffic light analysis

The 3 switches built for the DISCUS project are listed in Table 1 below. Switch 1793 (48-fibre module) is still progressing through manufacture so the results presented below are preliminary from initial integration tests. Even so, the optical performance for all 3 switches is remarkably uniform and for the 192-fibre optical switch the median optical performance figures of 1.0dB loss, 0.04dB repeatability and 68dB return loss across all path combinations represents the state of the art for photonic cross-connects of this size.

Table 17 Prototype single sided optical switches for the DISCUS project

Unit serial	Part code	Description
1500	N-OST-192xCC-LU1-DMHNR	Series 6000 Network grade Single mode 9/125um Optical Matrix Switch configured in a 3U high Rack-mount Tray with 192 Reconfigurable ports, LC/UPC connectors, Dual Multisession Ethernet & serial comms, Multiprotocol control interface, Hot swap twin 100/240VAC power, Normal environment, Rear panel connections
1794	N-OST-96xCC-HU1-DMHNS	Series 6000 Network grade Single mode 9/125um Optical Matrix Switch configured in a 1U high Rack-mount Tray with 96 Reconfigurable ports, LC-HD/UPC connectors, Dual Multisession Ethernet & serial comms, Multiprotocol control interface, Hot swap twin 100/240VAC power, Normal environment, Standard configuration
1793	N-OSM-48xCC-LU1-RSDNS	Series 6000 Network grade Single mode 9/125um Optical Matrix Switch configured in a Module with 48 Reconfigurable ports, LC/UPC connectors, Serial comms, SCPI control interface, DC +12V power, Normal environment, Standard configuration

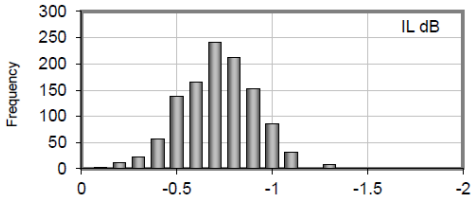
7.2.1 48-fibre OSM test results

polatis

OXC test results summary

Wavelength	1550 nm
Temperature	20 °C

Device	OST-48xCC-LU1-MMHNS
Serial	1793
Date	5-Aug-2015
Operator	PK



Insertion Loss

Units	dB
Max	-1.51
Median	-0.77
Min	-0.19

OXC test results record

Device	OST-48xCC-LU1-MMHNS	Date	5-Aug-2015	Wavelength	1550 nm
Serial	1793	Operator	GV	Temperature	20 °C

Insertion loss dB

Port	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48
1		-0.9	-1.0	-0.9	-1.1	-1.1	-1.1	-1.0	-1.0	-0.7	-1.4	-0.8	-0.9	-0.9	-1.1	-1.1	-0.6	-1.0	-1.1	-0.8	-1.0	-0.9	-1.1	-0.8	-1.1	-1.0	-0.8	-0.8	-0.8	-1.0	-0.8	-0.9	-1.0	-0.8	-0.9	-1.0	-0.8	-1.2	-1.0	-0.8	-1.2	-0.8	-1.0	-1.0	-1.1	-1.1	-1.0	-0.8
2			-0.6	-0.6	-0.7	-0.8	-0.8	-0.9	-0.6	-0.4	-1.0	-0.7	-0.6	-0.5	-0.8	-0.8	-0.4	-0.7	-0.6	-0.6	-0.7	-0.7	-0.7	-0.7	-0.5	-0.7	-0.6	-0.5	-0.5	-0.5	-0.9	-0.5	-0.7	-1.0	-0.6	-0.6	-0.7	-0.6	-0.7	-0.7	-0.5	-0.8	-0.5	-0.6	-0.6	-0.8	-0.7	-0.7	-0.6
3				-0.5	-0.8	-0.8	-0.9	-0.9	-0.7	-0.5	-1.0	-0.6	-0.7	-0.6	-0.8	-0.8	-0.4	-0.7	-0.8	-0.6	-0.8	-0.7	-0.8	-0.6	-0.8	-0.7	-0.6	-0.6	-0.5	-0.5	-0.9	-0.6	-0.6	-0.7	-0.9	-0.6	-0.6	-0.7	-0.6	-0.8	-0.8	-0.6	-0.9	-0.5	-0.7	-0.7	-0.8	-0.8	-0.7	-0.6
4					-1.0	-1.0	-1.1	-1.1	-0.9	-0.7	-1.3	-0.8	-0.9	-0.8	-1.0	-1.0	-0.6	-0.9	-1.0	-0.8	-1.0	-0.9	-1.0	-0.8	-1.0	-0.9	-0.8	-0.8	-0.7	-0.7	-0.7	-1.0	-0.7	-0.8	-1.1	-0.8	-0.8	-0.8	-0.9	-0.8	-1.1	-0.8	-0.7	-0.9	-0.9	-1.0	-0.8	-1.0	-0.8
5						-0.9	-1.0	-1.0	-0.8	-0.6	-1.1	-0.7	-0.8	-0.7	-0.9	-1.0	-0.5	-0.8	-0.9	-0.7	-0.9	-0.8	-0.9	-0.7	-0.9	-0.8	-0.7	-0.6	-0.6	-0.6	-0.9	-0.7	-0.8	-1.0	-0.7	-0.8	-0.8	-0.7	-1.0	-0.9	-0.7	-1.0	-0.7	-0.8	-0.8	-1.0	-0.9	-0.9	-0.7	
6							-0.9	-0.9	-0.7	-0.6	-1.2	-0.7	-0.7	-0.7	-0.9	-0.8	-0.5	-0.8	-0.9	-0.6	-0.8	-0.8	-0.9	-0.7	-0.9	-0.9	-0.6	-0.6	-0.6	-0.8	-0.6	-0.8	-0.8	-1.0	-0.6	-0.7	-0.9	-0.8	-0.6	-1.0	-0.6	-0.8	-0.8	-0.9	-0.9	-0.8	-0.6		
7								-1.0	-0.9	-0.8	-1.3	-0.8	-0.9	-0.9	-1.1	-1.0	-0.6	-0.9	-1.0	-0.7	-0.9	-0.9	-1.0	-0.8	-1.1	-0.9	-0.8	-0.8	-0.8	-1.0	-0.7	-0.9	-0.8	-0.8	-1.0	-0.7	-0.8	-0.8	-0.8	-1.2	-1.0	-0.8	-1.1	-0.8	-1.0	-1.0	-1.0	-0.8		
8									-0.8	-0.7	-1.4	-0.6	-0.7	-0.9	-0.9	-0.8	-0.5	-0.7	-0.8	-0.6	-0.8	-0.7	-1.0	-0.7	-0.9	-0.8	-0.7	-0.6	-0.6	-0.6	-0.5	-0.8	-0.7	-0.8	-0.8	-0.8	-0.5	-1.1	-0.9	-0.6	-1.0	-0.7	-0.9	-0.9	-0.9	-0.9	-0.7		
9										-0.5	-1.0	-0.7	-0.6	-0.8	-0.8	-0.6	-0.7	-0.7	-0.8	-0.6	-0.8	-0.7	-0.6	-0.5	-0.6	-0.8	-0.6	-0.9	-0.5	-0.6	-0.9	-0.8	-0.6	-0.9	-0.8	-0.6	-1.0	-0.6	-0.8	-0.8	-0.8	-0.9	-0.5	-0.7	-0.7	-0.8	-0.8	-0.6		
10											-1.1	-0.7	-0.6	-0.8	-0.9	-0.6	-0.8	-0.7	-0.9	-0.6	-0.8	-0.7	-0.6	-0.6	-0.6	-0.9	-0.6	-0.7	-0.7	-0.7	-0.6	-0.9	-0.8	-0.6	-0.9	-0.5	-0.8	-0.8	-0.8	-0.9	-0.8	-0.8	-0.6	-0.8	-0.8	-0.8	-0.6		
11												-1.0	-0.9	-0.8	-1.1	-1.2	-0.7	-1.0	-1.1	-0.8	-0.9	-1.0	-1.0	-0.9	-1.1	-0.9	-0.9	-0.8	-0.8	-1.2	-0.9	-0.9	-1.3	-0.8	-0.8	-0.9	-0.9	-1.1	-1.0	-0.9	-1.1	-0.7	-1.0	-0.9	-1.1	-1.0	-1.0	-0.8		
12													-0.8	-0.9	-1.1	-0.9	-0.6	-0.9	-1.0	-0.7	-0.9	-0.9	-1.1	-1.0	-0.9	-0.8	-0.8	-0.7	-0.9	-0.7	-0.9	-0.9	-0.9	-0.9	-0.7	-1.2	-1.0	-0.7	-1.2	-0.8	-1.0	-1.0	-1.1	-1.1	-1.0	-0.9			
13														-0.4	-0.7	-0.6	-0.2	-0.4	-0.6	-0.3	-0.5	-0.4	-0.6	-0.3	-0.6	-0.5	-0.4	-0.3	-0.3	-0.5	-0.3	-0.4	-0.6	-0.4	-0.5	-0.5	-0.4	-0.7	-0.6	-0.4	-0.7	-0.3	-0.6	-0.5	-0.6	-0.4			
14															-0.9	-1.0	-0.5	-0.8	-1.0	-0.7	-0.8	-0.9	-0.9	-0.7	-0.9	-0.8	-0.7	-0.7	-0.7	-1.0	-0.7	-0.9	-1.2	-0.7	-0.7	-0.8	-0.8	-0.9	-0.7	-1.0	-0.6	-0.8	-0.8	-1.0	-0.9	-0.8	-0.7			
15																-1.1	-0.7	-1.0	-1.1	-0.8	-1.0	-1.0	-1.0	-0.8	-1.1	-0.8	-0.8	-0.8	-0.8	-1.1	-0.8	-0.9	-1.1	-0.8	-0.9	-0.9	-0.8	-1.0	-0.9	-0.7	-1.0	-0.8	-1.0	-1.0	-0.9	-0.8				
16																	-0.4	-0.6	-0.8	-0.5	-0.7	-0.6	-0.8	-0.6	-0.8	-0.7	-0.5	-0.5	-0.6	-0.6	-0.7	-0.7	-0.4	-0.9	-0.7	-0.5	-0.9	-0.5	-0.8	-0.8	-0.7	-0.8	-0.6	-0.8	-0.7	-0.8					
17																		-0.4	-0.6	-0.2	-0.4	-0.4	-0.5	-0.2	-0.5	-0.4	-0.3	-0.3	-0.2	-0.5	-0.2	-0.4	-0.5	-0.3	-0.3	-0.4	-0.3	-0.6	-0.5	-0.2	-0.6	-0.2	-0.4	-0.4	-0.5	-0.5	-0.5	-0.3			
18																			-0.7	-0.9	-0.8	-1.0	-0.8	-0.8	-0.7	-0.9	-0.7	-0.5	-0.5	-0.7	-0.5	-0.9	-0.8	-0.7	-0.7	-0.9	-0.7	-1.1	-1.0	-0.7	-1.1	-0.7	-0.9	-0.9	-1.0	-1.1	-0.8				
19																				-0.6	-0.7	-0.8	-0.6	-0.8	-0.7	-0.6	-0.5	-0.5	-0.5	-0.6	-0.4	-0.6	-0.6	-0.6	-0.7	-0.7	-0.9	-0.9	-0.8	-0.8	-0.7	-0.9	-0.9	-0.9	-1.0	-1.1	-1.0	-0.8				
20																					-0.8	-0.7	-0.8	-0.6	-0.8	-0.7	-0.5	-0.5	-0.5	-0.7	-0.5	-0.7	-0.6	-0.6	-0.6	-0.6	-0.8	-0.7	-0.6	-0.7	-0.5	-0.7	-0.7	-0.7	-0.8	-0.7	-0.6				
21																						-0.8	-0.9	-0.6	-0.8	-0.7	-0.6	-0.6	-0.6	-0.8	-0.6	-0.8	-0.7	-0.7	-0.7	-0.7	-0.9	-0.8	-0.7	-0.8	-0.6	-0.8	-0.8	-0.8	-0.9	-0.8	-0.7				
22																							-1.0	-0.8	-1.0	-0.8	-0.7	-0.7	-0.7	-0.8	-0.7	-0.8	-0.7	-0.8	-0.7	-0.8	-0.6	-1.1	-0.9	-0.7	-1.1	-0.7	-0.9	-0.9	-0.9	-1.0	-0.9	-0.8				
23																								-0.7	-0.9	-0.8	-0.6	-0.6	-0.6	-0.9	-0.6	-0.7	-1.0	-0.7	-0.7	-0.8	-0.7	-0.9	-0.8	-0.7	-0.9	-0.6	-0.8	-0.7	-0.9	-0.8	-0.8	-0.6				
24																									-0.8	-0.7	-0.6	-0.5	-0.5	-0.8	-0.6	-0.7	-0.9	-0.6	-0.7	-0.7	-0.6	-0.9	-0.8	-0.6	-0.8	-0.6	-0.9	-0.8	-0.8	-0.8	-0.6				
25																										-0.7	-0.5	-0.5	-0.5	-0.8	-0.5	-0.6	-0.9	-0.6	-0.6	-0.7	-0.5	-0.9	-0.7	-0.6	-0.9	-0.5	-0.7	-0.7	-0.8	-0.8	-0.7	-0.6				
26																											-0.5	-0.5	-0.5	-0.8	-0.5	-0.6	-0.8	-0.5	-0.6	-0.7	-0.5	-0.9	-0.8	-0.5	-0.8	-0.5	-0.6	-0.6	-0.8	-0.7	-0.7	-0.5				
27																												-0.4	-0.4	-0.7	-0.4	-0.5	-0.7	-0.5	-0.5	-0.5	-0.4	-0.7	-0.6	-0.4	-0.7	-0.4	-0.6	-0.5	-0.7	-0.6	-0.6	-0.5				
28																													-0.5	-0.7	-0.5	-0.6	-0.7	-0.5	-0.6	-0.6	-0.5	-0.9	-0.8	-0.5	-0.7	-0.5	-0.7	-0.8	-0.8	-0.8	-0.6				
29																														-0.7	-0.6	-0.7	-0.7	-0.6	-0.6	-0.7	-0.6	-0.8	-0.8	-0.6	-0.8	-0.6	-0.8	-0.8	-0.8	-0.8	-0.6				
30																															-0.7	-0.6	-0.7	-0.7	-0.6	-0.7	-0.6	-0.9	-0.8	-0.6	-0.9	-0.6	-0.8	-0.8	-0.8	-0.8	-0.6				
31																																-0.7	-0.6	-0.7	-0.7	-0.6	-0.6	-0.9	-0.8	-0.6	-0.9	-0.6	-0.8	-0.8	-0.8	-0.8	-0.6				
32																																	-0.6	-0.7	-0.6	-0.7	-0.6	-0.9	-0.8	-0.5	-0.9	-0.6	-0.8	-0.8	-0.8	-0.8	-0.6				
33																																		-0.7	-0.8	-0.7	-0.5	-1.1	-0.8	-0.6	-1.1	-0.6	-0.9	-0.9	-0.7	-0.9	-0.9	-0.7				
34																																			-0.6	-0.7	-0.6	-0.9	-0.7	-0.6	-0.9	-0.5	-0.7	-0.7	-0.8	-0.8	-0.6				
35																																				-0.6	-0.8	-0.7	-0.6	-0.8	-0.5	-0.7	-0.6	-0.8	-0.7	-0.7	-0.5				
36																																					-0.5	-0.8	-0.7	-0.5	-0.8	-0.5	-0.6	-0.6	-0.7	-0.7	-0.7	-0.5				
37																																						-1.0	-0.8	-0.6	-1.0	-0.6	-0.8	-0.8	-0.8	-1.5	-0.9	-0.7				
38																																							-1.0	-0.8	-1.1	-0.7	-0.9	-0.9	-1.2	-1.0	-0.8				
39																																								-0.7	-1.1	-0.7	-0.9	-0.9	-1.0	-1.0	-1.0	-0.8				
40																																									-0.8	-0.5	-0.7	-0.7	-0.7	-0.7	-0.7	-0.5				
41																																										-0.6	-0.8	-0.8	-0.9	-0.9	-0.8	-0.7				
42																																											-0.8	-0.7	-0.9	-0.8	-0.8	-0.8				
43																																												-0.8	-1.0	-0.8	-0.8	-0.8				
44																																													-0.8	-0.7	-0.7	-0.8				
45																																														-0.9	-0.9	-0.7				
46																																															-1.4	-1.2			
47																																																-0.8			
48																																																			

Optical Return Loss

Units	dB
Min	65.0

Method: GR1073 4.2.6
Equipment: Luna OBR

Input Port	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
Output Port	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48
ORL	69	66	66	65	68	66	69	70	69	67	68	69	69	71	69	69	69	67	69	68	69	67	69	66

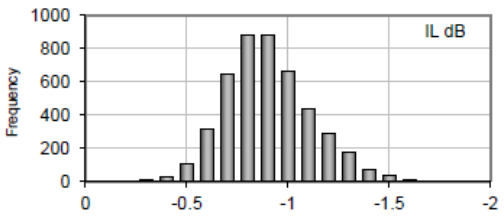
7.2.2 96-fibre OST test results

polatis

OXC test results summary

Wavelength	1550 nm
Temperature	20 °C

Device	OST-96xCC-HU1-MMHNS
Serial	1794
Date	5-Aug-2015
Operator	PK



Insertion Loss

Units	dB
Max	-1.99
Median	-0.93
Min	-0.36

7.2.3 192-fibre OST test results

OXC test results summary

polatis

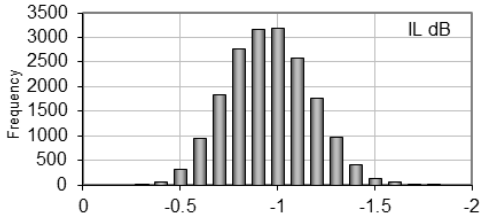
Device	OST-192xCC-LU1_DMHNS
Serial	1500
Date	20-May-2014
Operator	MFG

Wavelength	1550 nm
Temperature	20 °C

Insertion Loss

Units	dB
Max	-1.86
Median	-1.00
Min	-0.38

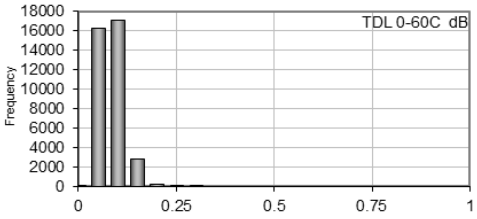
Method: TIA/EIA 526-7 A.3
Equipment: Polatis PMT-32x00-LU1-MSENS
Amnionics ALS-18B-FA



Temperature Dependent Loss 0-60C

Units	dB
Max	0.29
Median	0.06
Min	0.00

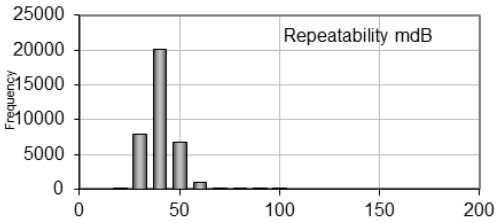
Method: TIA/EIA 526-7 A.3
Equipment: Polatis PMT-32x00-LU1-MSENS
Amnionics ALS-18B-FA



Repeatability

Units	mdB
Max	100
Median	40
Min	20

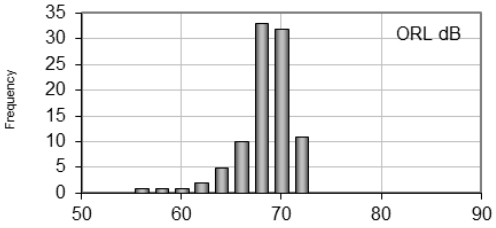
Method: GR1073 4.2.11
Equipment: Polatis PMT-32x00-LU1-MSENS-400
Amnionics ALS-18B-FA



ORL

Units	dB
Min	55

Method: GR1073 4.2.6
Equipment: Agilent 81610A
Agilent 81635A



7.3 Embedded Optical Switch OpenFlow Agent

This section describes the interface to the embedded Polatis OpenFlow agent (POLOA) developed to control the Polatis optical circuit switch in collaboration with the University of Bristol. The OpenFlow agent runs directly on top of the Polatis optical switch application programming interface (API) alongside other user services, as shown in Figure 8 below. The command set is defined by OpenFlow 1.0 and the Circuit Switch Addendum v0.3 [18] together with additional proprietary extensions as detailed below.

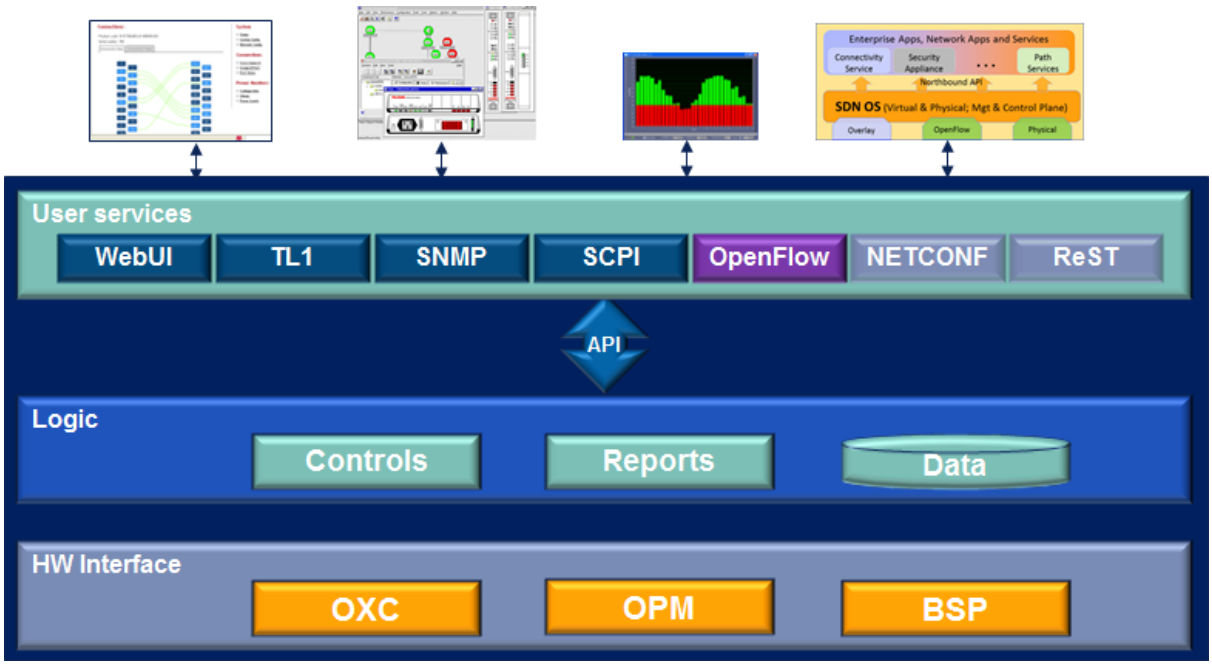


Figure 7-8 Polatis Optical Switch User Interface

7.3.1 Agent configuration

Each OpenFlow switch agent requires a controller IP address along with port and a datapath identifier (DPID). A unique DPID is used to identify its connection to a controller. The DPID is derived from the Polatis unit serial number

7.3.2 OpenFlow Messages

Hello: Hello messages are exchanged between the switch and controller upon connection startup. Hello messages allow negotiation of the latest compatible OpenFlow version supported by both the switch and the controller.

Switch features messages (*SWITCH_FEATURES*): Upon Transport Layer Security (TLS) session establishment, the controller sends a features request message to the switch. The switch must respond with a features reply that specifies the capabilities supported.

Modify state messages (*FLOW_MOD*): Modify-State messages are sent by the controller to manage state on the switches. Their primary purpose is to add/delete and modify flows in the flow tables and to set switch port properties.

Read state messages (*FLOW_STATS*): Read-State messages are used by the controller to collect statistics from the switch's flow-tables, ports and the individual flow entries.

Flow-Removed messages (*FLOW_REMOVED*): When a flow entry is added to the switch by a modify message, an idle timeout value indicates when the entry should be removed due to a lack of activity, as well as a hard timeout value that indicates when the entry should be removed, regardless of activity. The flow modify message also specifies whether the switch should send a flow removed message to the controller when the flow expires.

Port-status messages (*PORT_STATUS*): The switch is expected to send port-status messages to the controller as port configuration state changes. These events include change in port status (for example, if it was brought down directly by a user).

Error messages (*ERROR_MSG*): The switch is able to notify the controller of problems using error messages.

Echo: Echo request/reply messages can be sent from either the switch or the controller, and must return an echo reply. They can be used to indicate the latency, bandwidth, and/or liveness of a controller-switch connection.

Vendor: Vendor messages provide a standard way for OpenFlow switches to offer additional functionality within the OpenFlow message type space. This is a staging area for features meant for future OpenFlow revisions.

7.3.3 OpenFlow protocol extensions:

OpenFlow circuit switching extensions add to the general OpenFlow protocol originally developed for the packet domain. The OpenFlow protocol supports three message types: controller-to-switch, asynchronous and symmetric, each with multiple subtypes. In general, we maintain the same message types for a circuit switch and we change some of the structures and its parameters used in the message.

This section describes the various messages that are implemented using the OpenFlow circuit switching specification in v0.3. The changes that are made to the v0.3 specification to support Polatis switch features are described.

The section is laid out as follows:

- Message type along with the structure name
- Vendor changes for Polatis optical switches
- Structure reference with field descriptions

7.3.3.1 OpenFlow Features Message: ofp_switch_features

In response to an *OFPT_FEATURES_REQUEST* message, the agent sends an *OFPT_FEATURES_REPLY* message with the structure shown below.

7.3.3.1.1 Vendor Changes:

- ***n_cports***

Note that if the port count is more than 256 the agent reports only inports with count greater than 128 and the controller deduces the final port count by multiplying by 2

- ***Capabilities:***
  - *Added to the fibre switch capabilities.*
  - *OFPC_Attenuate = 1 << 24, /*Attenuation monitoring feature available */*
  - *OFPC_Power = 1 << 23, /*Power monitoring feature available */*
  - *OFPC_CPORT_STATS = 1 << 22, /*Power monitoring feature available */*
  - *OFPC_CFLOW_STATS = 1 << 21, /*Circuit Flow monitoring feature available */*

7.3.3.2 Openflow circuit ports: ofp_phy_cports

Physical circuit ports are reported as part of an array of struct ofp_phy_cport in the *OFPT_FEATURES_REPLY* message. In the interest of staying true to the packet switching spec, we retain similar name ofp_phy_cport.

7.3.3.2.1 Vendor changes:

- *Port_no* : Actual fibre/port number as per the front panel numbering on the switch
- *Hardware addr*: mac_address..
- *Name*: adds a predefined string (PF or POL) with the actual fibre/port number
- *Port State*:
  - Current state. It is continuously monitored and changes are informed via cport_status message.
- *Port features and switching type*:
  - *OFPPF_FIBER | OFPPF_X | OFPST_FIBER*
- *Peer DPID & port number*:

Manually provided via the peer configuration file. This file is found in the same location as the agent and takes DPID and port number of the connected nodes.

7.3.3.3 Circuit Flow Modification:

Modify Flow Entry Message modifies the cross connection table from the controller via *OFPT_CFLOW_MOD* message. The *CFLOW_MOD* message contains the *OFP_CONNECT* structure which details the cross connections details. The logical equivalent of the ofp_match structure from the packet switching spec is the ofp_connect structure in the circuit switching addendum. It is used to describe the circuit flow much like the match structure is used to describe the packet flow.

7.3.3.3.1 Vendor Changes:

Flow Command:

OFPFC_ADD, / New flow. */*

OFPFC_MODIFY, / modify flow*/*

OFPFC_DROP / terminate circuit flow*/*

Ignoring ofp_action_header as it is for hybrid switches.

Three options available for adding cross connections via flow mod messages. All of them use the command *OFPFC_ADD*.

1) Single cross connect: Use only the in_port and out_port fields of ofp_connect.
   a. ofp_connect->in_port ←→ ofp_connect->out_port

2) Multiple cross connect Use inport[] and out_port[] as a list of ports
   a. ofp_connect->in_port[x] ←→ ofp_connect->out_port[y]

3) An optional VOA feature can also be used. Use the wave_port description where in the extra uint64_t bandwidth field can be used to define new features such as power attributes etc.

```
struct ofp_wave_port {
  uint16_t wport;     /* in port and out port */
  uint8_t  pad[6];    /* align to 64 bits */
  uint64_t wavelength; /* use of the OFPCBL_* flags if power bit set, 10-25 bits reserved */
 };
```

where *OFPCBL* in ofp_port_lam_bwx:

```
OFPCBL_X          = 1 << 0,/* 1 if fiber switch */
OFPCBL_POW    = 1 << 7,/* 1 if VOA attenuate parameters present */
```

The bits from 10 to 42 define the VoA settings as described in *OFP_CPORT_STATS*. Not currently implemented in the initial release.

4) The same applies to deleting connection with command *OFPFC_DELETE/DROP*.
5) *OFPFC_MODIFY* is same as *ADD* since Polatis switch deletes existing XCs and adds new ones.

7.3.3.4 Port Status message:

As physical ports are added, modified, and removed from the switch, the controller needs to be informed with the *OFPT_CPORT_STATUS* message. Its structure is as follows.

7.3.3.4.1 Vendor changes

Reason for port change remains the same i.e.

```
/* What changed about the physical port */
enum ofp_port_reason
{ OFPPR_ADD, /* The port was added. */
 OFPPR_DELETE, /* The port was removed. */
 OFPPR_MODIFY, /* Some attribute of the port has changed. */
};
```

7.3.3.5 Circuit port stats (Power and Attenuation Statistics): cport_stats message

CPORT_STATS message provides the optical power monitor data for a given port in a given direction. The request for a cport_stats can be issued from the controller. The controller uses the stats_request message with type OFPST_CPORT and the body containing the port number and the direction. The port number can be OFPP_NONE which will return power data for all OPM ports for a given direction i.e. ingress or egress.

7.3.3.5.1 Vendor changes

- *OFP_STATS_REQUEST type = OFPST_CPORT,*
- *OFP_CPORT_STATS reply*
- *MES_PMON*: measured power reports in absolute values. Need to check MSB for negative values.
- *PORT_NO:* the actual port not the OPM index number.

7.3.3.6 Circuit error messages: ofp_error_msg

There are times that the switch needs to notify the controller of a problem. This is done with the *OFPT_ERROR_MSG* message. E.g. If a conflict exists between an existing flow entry and the ADD request, the switch must refuse the addition and respond with an ofp_error_msg with *OFPET_FLOW_MOD_FAILED* type and *OFPFMFC_OVERLAP* code.

7.3.3.6.1 Vendor Changes:

Addition of new reason to ofp_error_type which is

OFPET_CFLOW_MOD_FAILED = 0xffff / Problem modifying circuit flow entry */*

7.4 12,000 port single-sided 3 stage optical switch

7.4.1 Physical design

A pluggable optical switch module concept was described in D6.2 which combines the 192-fibre single sided switch core characterized above with a further level of electronics integration beyond the scope of this project. The pluggable switch module would facilitate hitless pay-as-you-go upgrades and rapid mean times to repair (MTTR). To create a 12,288 port strictly non-blocking single sided optical switch fabric, 319 of these 192-fibre single-sided modules would be required which could be accommodated in 160RU of rack height, or roughly 4 equipment bays as shown in Figure 9 below (from D6.2). For ease of illustration, the edge and centre stages have been divided equally between the 4 racks and external fibre ports have been included on the front panel of edge modules (64 per module) using high density LC connectors. Between each of the 4 racks, a mesh interconnect is required, which can be condensed into 256 12-fibre ribbons per direction.

Total optical loss traversing the fabric was estimated to be 4-5dB median and 6-7dB max when 1-2dB of interconnect losses are included; back reflections assuming APC connectors will be on the order of -45dB. The total number of switch ports employed when fully loaded is 61,056, reflecting the internal:external port ratio of 5:1, which is typical for strictly non-blocking Clos fabrics. The maximum energy consumption was projected to be around 3kW.

Figure 7-9 Illustration of the physical size required for a 12,288 port single-sided optical switch fabric using a 3-stage folded-Clos architecture

Further discussions with potential users have suggested that a distributed optical switch fabric with structured cabling interconnect may be more suited to incremental growth of a

central office or multi-tenant datacentre facility. Figure 10 illustrates one potential topology where each edge cage can scale hitlessly in groups of 64 fibres by adding 192-fibre modules. All switch-to-switch interconnections are made with MTP cables with8 fibers per cable. No interconnections are required between Polatis center stage switches.

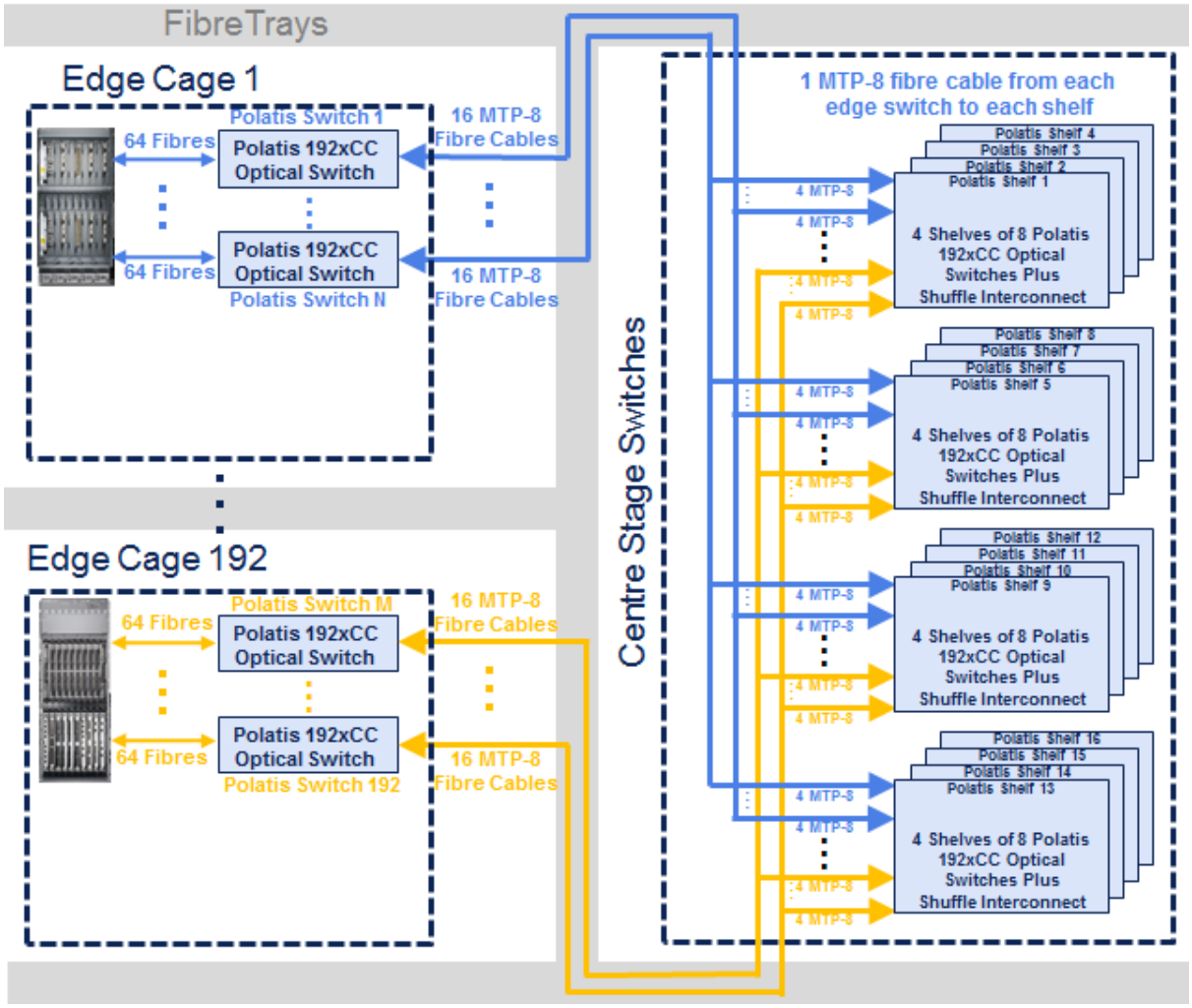


Figure 7-10 A distributed architecture for a 12,288 port optical switch fabric where edge switches are located in associated equipment cages and interconnected using structured cabling

7.4.2 Integration with OpenDaylight SDN controller

In section 3.1.1 we discussed two approaches to control a multi-module optical switch fabric and demonstrated the more straightforward approach using an intermediate switch ("shelf") controller. We also examined the feasibility of controlling multiple optical switches in a pure SDN architecture, using the OpenDaylight SDN controller. A demonstrator was created to co-ordinate both packet and optical switches to schedule the bandwidth allocated between two servers, changing between a 1Gb/s electrical connection to a 10Gb/s optical link.

The Bandwidth on demand/Bandwidth Calendaring application is written in python and uses the REST interface of an OpenDayLight Hydrogen (ODL) controller, which in turn uses the OpenFlow 1.0+ (OF) protocol to talk to the network of optical and electrical switches to install "flows". In the optical switch context, these flows are cross connects between a pair of

ingress and egress ports made using the protocol extensions and embedded OpenFlow agent (POLOA) described in the previous section (7.3) to support circuit switching.

The application featured as part of a remote demonstration during a Cisco/WWT Geek Day in Washington, May 2015, where we had the ODL controller acting as a management control station, communicating with 2 optical (Polatis 48x48, #1708 and Polatis 24x24, #1714) switches and 2 packet switches from NEC (NEC A and NEC B), as shown in the figure 11 below. Two Xen servers were connected to the NEC switches.

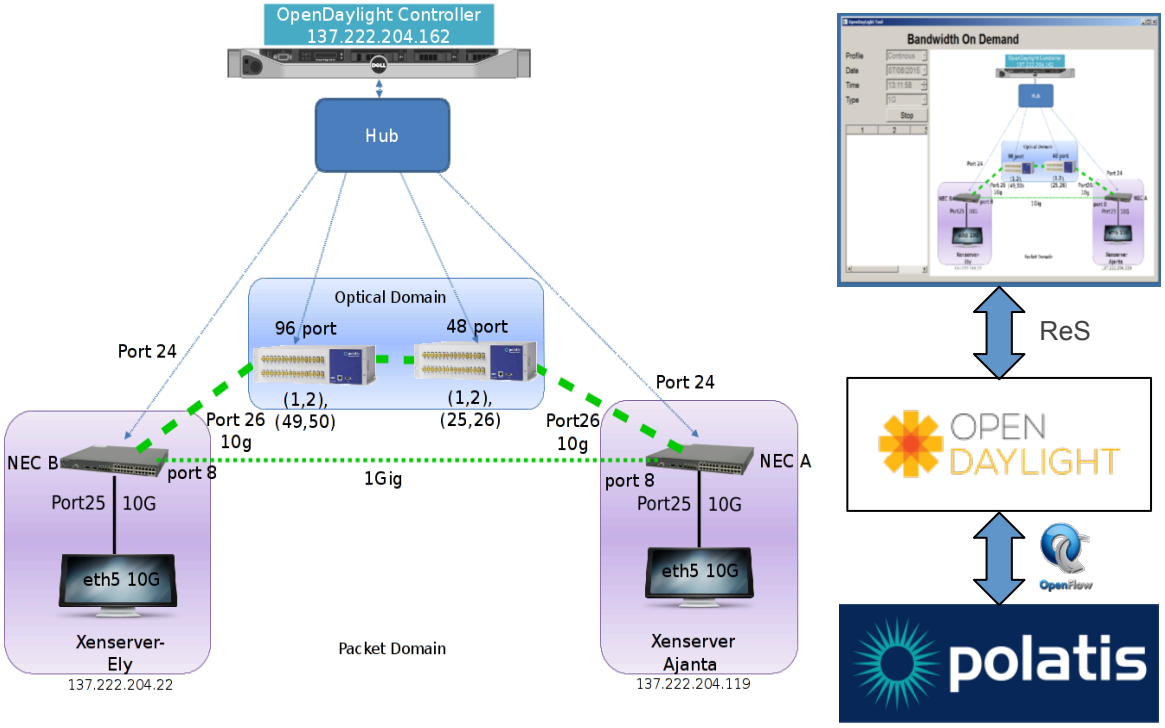


Figure 7-11 a) Hybrid packet-optical bandwidth calendaring experiment using python application on OpenDaylight ReST interface to control multiple OpenFlow-enabled packet and optical switches; b) control structure

The setup represented a "mini data centre" where the NEC devices can be considered as Top of Rack (TOR) switches and the Xen servers represent processing elements in two data centre racks. The black link in Figure 11 between server and NEC switch is a 10Gb/s optical connection to port 25 of the NEC switch. The green (thin) link on port 8 is the electrical 1G connection between the two NEC switches. The green (thick) links are the optical connections between NEC switches and Polatis switches. Port 26 of NEC B switch was connected to port 1 and 49 of Polatis 48x48 switch (#1708) and ports 2 and 50 of this switch was connected to another Polatis 24x24 switch (#1714) on ports 1 and 25. Ports 2 and 26 of switch 1714 were connected in turn to port 26 of NEC A switch. NEC switches had 10G optical transceivers on port 26, so even though Polatis switches are bandwidth agnostic, the maximum bandwidth achievable in the setup was 10G.

To show the actual bandwidth achieved in the network, we ran an iperf (throughput test) client on one Xen server and an iperf server on the other Xen server. The bandwidth calendaring application modified the flows on the optical and electrical switches to achieve a throughput of either 1G or 10G. For 1G throughput the application installed electrical flows

between port 25 and port 8 on both NEC switches. For 10G bandwidth the application installed electrical flows between port 25 and port 26 on NEC switches and cross connects within Polatis switches.

Abbreviations

AMCC	Auxiliary Management and Control Channel
API	Application Programming Interfaces
AWG	Arrayed Waveguide Grating
BAU	Business as usual
BRAS	Broadband Remote Access Server
B&S	Broadcast and Select
BVT	Bandwidth Variable Transponder
CC	Continuity Check
CE	Customer Edge
CIR	Committed Information Rate
CO	Central Office
CPE	Customer Premises Equipment
CV	Connectivity Verification
DBA	Dynamic Bandwidth Allocation
DISCUS	DIStributed Core for unlimited bandwidth supply for all Users and Services
DP-16QAM	Dual Polarization-16-ary Quadrature Amplitude Modulation
DP-BPSK	Dual Polarization-Binary Phase Shift Keying
DP-QPSK	Dual Polarization-Quadrature Phase Shift Keying
DSP	Digital Signal Processing
DWDM	Dense Wavelength Division Multiplexing
EDFA	Erbium Doped Fibre Amplifier
FEC	Forward Error Correction
FFD	Fast Failure Detection
FHD	Full High Definition
FRR	Fast Re-Route
HD	High Definition
H-VPLS	Hierarchical Virtual Private LAN Service
ICT	Information and Communication Technology
IoT	Internet of Things
IP	Internet Protocol
LR-PON	Long Reach-Passive Optical Network

L1	Layer 1
L2	Layer 2
L3	Layer 3
LAN	Local Access Network
LI	Line Interface
LSP	Label Switched Path
LTE	Long-Term Evolution
MAC	Media Access Control
MC	Metro-Core
MDT	Mean Down Time
MNO	Mobile Network Operator
MMS	Multi Media Service
MP2MP	Multi-Point to Multi-Point
MPLS	MultiProtocol Label Switching
MPLS-TP	MultiProtocol Label Switching - Transport Profile
MS	Multicast Swtich
MTBF	Mean Time Between Failures
MTTR	Mean Time To Repair
NGN	Next Generation Network
NIC	Network Interface Card
NNI	Network-to-Network Interface
NO	Network Operator
NP	Network Provider
NT	Network Termination
OAM	Operation Administration and Maintenance
OCh	Optical Channel
ODN	Optical Distribution Network
OLT	Optical Line Terminal
ONT	Optical Network Terminal
ONU	Optical Network Unit
OPM	Optical Performance Monitoring
OPU	Optical channel Payload Unit
OSA	Optical Spectrum Analyzer
OSM	Optical Switch Module
OSNR	Optical Signal Noise Ratio

OSS/NMS	Operation Supporting System/Network Management System
P2MP	Point to MultiPoint
P2P	Point to Point
PA	Public Administration
PC	Private Circuit
PCC	Policy and Charging Control
PCE	Path Computation Element
PCEF	Policy and Charging Enforcement Function
PCR	Program Clock Reference
PCRF	Policy and Charging Rules Function
PDS	Pure Data Service
PE	Provider Edge
PHP	Penultimate Hop Popping
PIP	Physical Infrastructure Provider
PPPoE	Point-to-Point Protocol over Ethernet
PW	Pseudo Wire
QoS	Quality of Service
RBD	Reliability Block Diagram
ROADM	Reconfigurable Optical Add-Drop Multiplexer
RSVP	Resource ReserVation Protocol
S-BVT	Sliceable Bandwidth Variable Transponder
SCh	Super Channel
SD	Software Decision
SDN	Software Defined Networking
SOHO	Small Office/Home Office
SP	Service Provider
SPME	Sub-Path Maintenance Elements
SR	Spectrum Routing
TCM	Tandem Connection Monitoring
TCP	Transmission Control Protocol
TDM	Time Division Multiplexing
TE	Traffic Engineering
T-SDN	Transport Software Defined Networking
TWDM	Time and Wavelength Division Multiplexing
UK	United Kingdom

UMTS	Universal Mobile Telecommunication System
UNI	User Network Interface
US	UpStream
UHD	Ultra High Definition
VCCV	Virtual Circuit Connectivity Verification
VLAN	Virtual Local Access Network
VoIP	Voice over IP
VoD	Video on Demand
VPLS	Virtual Private LAN Service
VPN	Virtual Private Network
VRF	Virtual Routing and Forwarding
VSI	Virtual Service Instance
VULA	Virtual Unbundled Loop Access
WDM	Wavelength Division Multiplexing
WM	Wavelength Mux/demux
WS	Wavelength Selective
WSS	Wavelength Selective Switch

References

[1] A. Buttaboni et al., “Virtual PON Assignment for Fixed-Mobile Convergent Access-Aggregation Networks”, Proc. ONDM’15 , Th9.30, 2015.

[2] BEREC Common Position on best practice in remedies on the market for wholesale (physical) network infrastructure access (including shared or fully unbundled access) at a fixed location imposed as a consequence of a position of significant market power in the relevant market (BoR (12) 127).

[3] Beam Steering Optical Switch, US Patent 7,389,016

[4] Clos, C.,” A Study of Non-Blocking Switching Networks“, BSTJ 32-2, p406, 1953

[5] D.G. Foursa, H.G. Batshon, H. Zhang, M. Mazurczyk, J.-X Cai, O. Sinkin, A. Pilipetskii, G. Mohs, N. S. Bergano, "44.1 Tb/s transmission over 9,100 km using coded modulation based on 16QAM signals at 4.9 bits/s/Hz spectral efficiency," 39th European Conference and Exhibition on Optical Communication (ECOC 2013), pp.1,3, 22-26 Sept. 2013

[6] IDEALIST: http://www.ict-idealist.eu/

[7] H. Hasegawa, "Large Scale Optical Cross-connect: architecture, performance analysis, and feasibility demonstration," in Optical Fiber Communication Conference 2015, OSA Technical Digest, paper W3J.1, Mar. 2015

[8] N. Amaya, G. Zervas, D. Simeonidou, “Architecture on demand for transparent optical networks”, Proc. ICTON, 2011, pp. 1-4.

[9] A. Muhammad, G. Zervas, N. Amaya, D. Simeonidou, and R. Forchheimer, “Introducing flexible and synthetic optical networking: Planning and operation based on network function programmable ROADMs,” J. Opt. Commun. Netw., vol. 6, no. 7, pp. 635–648, 2014.

[10] M. Garrich, N. Amaya, G. Zervas, P. Giaccone, and D. Simeonidou, “Power consumption analysis of architecture on demand,” Proc. ECOC, 2012, paper P5.06.

[11] M. Garrich, N. Amaya, G. Zervas, P. Giaccone, and D. Simeonidou, “Architecture on demand: Synthesis and scalability,” Proc. ONDM, 2012, pp. 1–6.

[12] M. Džanko, M. Furdek, G. Zervas, D. Simeonidou, “Evaluating availability of optical networks based on self-healing network function programmable ROADMs”, IEEE/OSA J. Opt. Comm. Netw., vol. 6, no. 11, pp. 974-987, 2014.

[13] M. Džanko, M. Furdek, B. Mikac, G. Zervas, E. Hugues-Salas, D. Simeonidou, “Synthesis, resiliency and power efficiency of function programmable optical nodes”, Proc. ConTEL, 2015, pp. 1-8.

[14] Ward Van Heddeghem at al, “Power consumption modelling in optical multilayer networks” Photonic Networks Communications, January 2012

[15] Beam Steering Arrangements and Optical Switches, US Patent 7,095,915

[16] A Piezoelectric Actuator, US Patent 7,026,745

[17] Beam Steering Optical Switch, US Patent 7,389,016

[18] Sauras Das, Extensions to the OpenFlow Protocol in support of Circuit Switching. Addendum to OpenFlow Protocol Specification (v 1.0) – Circuit Switch Addendum v0.3. June, 2010.

Document versions

Version[5]	Date submitted	Comments
V1.0	23/09/2015	First version sent to the commissions.

[5] Last row represents the current document version