POLITECNICO DI TORINO

Master's Degree in COMMUNICATIONS AND COMPUTER NETWORK ENGINEERING

Master's Degree Thesis

Optical Network Control Exploiting Open Interfaces

Supervisors

Prof. Vittorio CURRI

Renato AMBROSONE

Emanuele VIRGILLITO

Candidate

Riccardo SCHIPS

July 2024

Abstract

The exponential growth of internet data traffic necessitates a significant evolution in network infrastructure, particularly within optical networks. Modern optical networks are increasingly incorporating software-based devices designed to enhance the management and optimization of data traffic. These devices include transponders, which are responsible for injecting signals at fixed frequencies, and Reconfigurable Add Drop Multiplexers (ROADM), which play a crucial role in managing signal forwarding within the network. Each of these devices operates with specific hardware and runs its own operating system. The passage from static hardware devices to software-based ones represents a critical aspect towards the evolution to the open optical Software Defined Networks (SDN).

An open optical SDN is characterized by a disaggregated network structure controlled by a central software-based entity known as the SDN controller. The SDN controller is designed to manage the reconfigurable devices, by sending towards them specific commands for data exchange.

Before moving towards the proposed activities, it is important to focus on the ROADM entity, analyzing both hardware and software aspects, since it plays a key role in the experiment. ROADMs are devices that enable the dynamic reconfiguration on the handling of the optical signal's forwarding.

The principal aspects studied are the architecture of the ROADM and the signal's handling, focusing in particular on how to create the software entities aimed to route the incoming signals, dividing them in channels according to the Wavelength Division Multiplexing (WDM) rules, to equalize the created channels and to retrieve information of each channel's power.

Typically, networks are oriented towards single-vendor solutions: in particular the devices belong to the same model. A transition to multi-vendor scenario is possible with the transition towards the open optical SDN, and can offer significant advantages: multi-vendor networks enable cost control through diverse market options, allowing for the selection of various device models. The diversification of the devices introduces also a competitive scenario, able to enhance the models. To move towards a multi-vendor scenario, it is necessary to develop standard models to ensure interoperability within a multi-vendor network.

In this context, the interoperability between the controller and a specific ROADM model has been tested to demonstrate the feasibility of multi-vendor communication. This test illustrated how communication happens between the controller, utilizing a specific standard, and a ROADM device with its proprietary language. In particular, there is an interaction between ONOS, an open source optical SDN controller that runs inside the laboratory, and the Adtran model of ROADM, always present in the lab: the selected standard is OpenROADM, whose aim is the control of the ROADMs.

To handle the OpenROADM requests, a driver has been developed and its aim is the conversion of the commands, originally written according to the OpenROADM standard, to the proprietary language supported by the Adtran model. The driver has been written in C-language and collects the most important functions for the ROADM management, related to the creation of optical paths and forwarding of incoming signals.

Acknowledgements

At first, I want to thank PLANET team, in particular prof. Vittorio Curri, who gave me the opportunity to improve my skills and my knowledge in the optical network world during my thesis activity. I want to thank Renato, Emanuele, Stefano and Francesco for the significant help and support they provide me to reach my goals, from the theoretical explanations to the practical activities.

Thanks to my friends, who have always represented a constant guide and continue to give me a significant support thanks to their precious advice, and for all the significant experiences shared during my university years.

My deepest gratitude goes to my entire family, from my uncles to my cousins, and in particular to my parents, my siblings and my grandmother, who taught me the essential values of life, have always been fundamental in my personal growth and for the love they continue to show.

Riccardo

Table of Contents

List of Figures

Acronyms

ASE Amplified Spontaneous Emission

ASK Amplitude-Shift Keying

CLI Command Line Interface

DMX De-multiplexer

EDFA Erbium-Doped Fiber Amplifier

FDM Frequency Division Multiplexing

FSK Frequency-Shift Keying

HTTP Hypertext Transfer Protocol

ILA In-Line Amplifier

IMDD Intensity Modulation Direct Detection

IP Internet Protocol

IS In Service

MUX Multiplexer

OCM Optical Channel Monitor

OLC Optical Line Controller

OLS Optical Line System

OMS Optical Multiplexer Section

ONC Optical Network Controller

ONOS Open Network Operating System

OOK On Off Keying

OOS Out of Service

OSA Optical Spectrum Analyzer

OTSiA Optical Tributary Signal Assembly

PSD Power Spectral Density

PSK Phase-Shift Keying

QAM Quadrature Amplitude Modulation

OADM Optical Add-Drop Multiplexer

ROADM Reconfigurable Optical Add-Drop Multiplexer

RWA Routing Wavelength Assignment

SDN Software Defined Network

SPSLG Spectrum Slot Group

SRG Shared Risk Group

SSH Secure SHell

TRX Transceiver

WDM Wavelength Division Multiplexing

WSS Wavelength Selective Switching

Chapter 1

Introduction

The traffic growth within the networks requires a constant evolution in terms of architecture enhancement, including new devices and new concepts of network. This phenomena represents the starting point towards open and disaggregated networks. To face the internet traffic increment, the introduction of new resources and the redefinition of the network strategies are essential.

This work starts with the introduction of the general concepts that govern the optical domain, focusing on the optical network components and different network architectures. In the context of disaggregated networks, it is possible to adopt an heterogeneous scenario, with the aim of guarantee an interoperability among the devices within the optical network. For this purpose, standards have been developed to face with a multi-vendor scenario, where new infrastructures must be inserted to allow the communication between the diverse languages of the devices.

In this context a specific device, a ROADM, provided by a specific vendor, Adtran, has been studied: the characteristics of the ROADM are explained considering both hardware and software. ROADMs are characterized by a specific hardware and operating system able to compute the general aims of the device, but with different approaches that depends on the model.

After the presentation of the device in use, it is possible to focus on the main proposed activity: the introduction of an interface able to guarantee the interoperability between the ROADM and the network. For this reason, a middleware based on the features of the model has been realized in order to insert the examined ROADM within an optical disaggregated network. The aim of the middleware is the translation of the commands written according to a standard model directed to the ROADM towards the proprietary language.

The collected results are presented to show at first the characteristics of the specific ROADM, focusing on internal losses and important features that can be exploited. Finally, results to demonstrate the feasibility of the middleware insertion are proposed.

Chapter 2

Fundamental Concepts

Communication has always been fundamental for humans and has become especially crucial for those needing to communicate remotely. To facilitate this, improvements in information exchange methods are necessary to ensure effective communication. Over the decades, communication has evolved to meet people's needs: it began with simple one-to-one voice calls and has advanced to a modern context where remote group calls and internet data exchange are essential.

In the optical network world, it is possible to distinguish two main network structures: closed networks and SDN networks. The first family can be considered as a traditional internet approach, while SDN networks represent an evolution of the network paradigm to face with the increment of the internet traffic.

This chapter explores the principal characteristics of these distinct network solutions and focuses on the key components within the network, analyzing their general and fundamental aspects.

2.1 Optical Networks

2.1.1 Closed networks

Closed networks, Figure 2.1, are composed by devices able to make decisions by exchanging information with the neighbour devices. In particular, each network device is characterized by a proprietary operating system, used to directly communicate with its neighbours to handle network functions, such as routing decisions. These entities are also characterized by their own hardware. The devices that populate this type of network are able to communicate among each other since they are provided by the same vendor. On the contrary, these entities are featured by complex infrastructures to handle control, data and management planes.

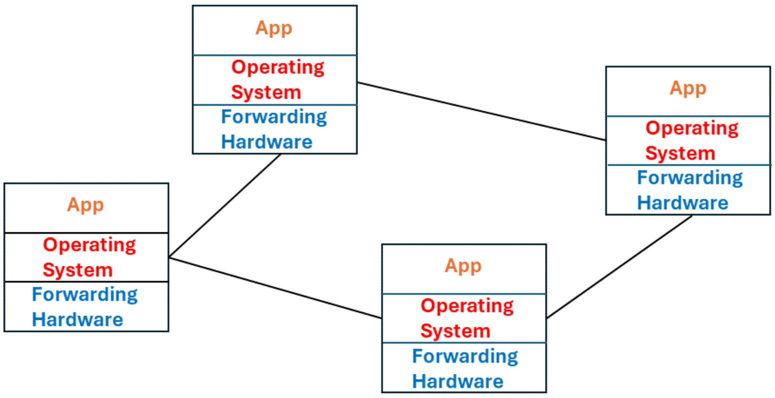


Figure 2.1: Scheme of a closed network

2.1.2 SDN networks

SDN [1] is a new model of network aimed to deal with the internet data traffic growth[2], based on a diverse paradigm: the main goal is to simplify the general architecture with the separation of the functions.

Within a SDN, there are different software-based components: the principal entity is the controller, a central unit aimed to manage the entire network, in particular focusing on the traffic management. This entity is responsible to handle the devices, in particular exploiting their operating systems to configure the necessary resources to forward the data traffic. As a consequence, the devices are now provided by a simplified architecture that comprehends the hardware and a simple operating system aimed to configure the parameters for the data forwarding, according to the decisions made by the controller.

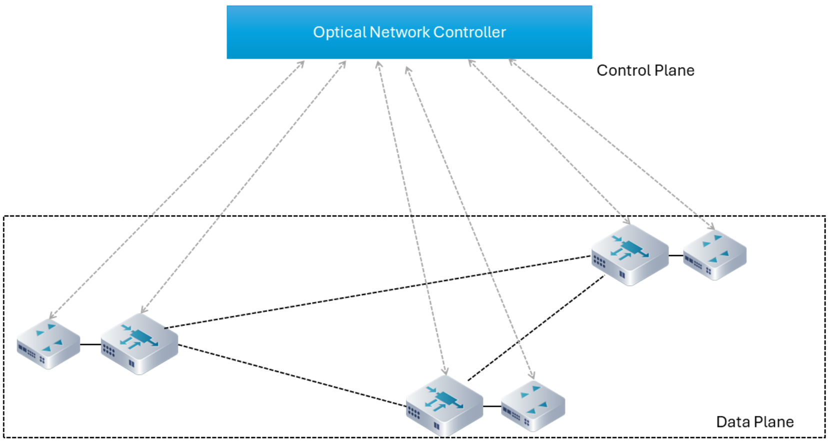


Figure 2.2: Simplified scheme of a SDN

The SDN optical networks can be divided in three main families: aggregated networks, partly disaggregated networks and fully disaggregated networks [3].

2.1.3 Aggregated Networks

Aggregated networks, shown in Figure 2.3, represent the traditional networks composed by several devices provided by the same vendor, i.e., they are single vendor oriented. The main principle is the same within the closed networks, Section 2.1.1, due to their single-vendor orientation, but in this case the control and management plane are moved towards a controller. The main advantage of this solution is the guaranteed interoperability among all the devices within the network: they are designed to work together and use proprietary interfaces for the communication. The main disadvantage is the limited flexibility to integrate different vendor devices, that could be able to enhance the network performance.

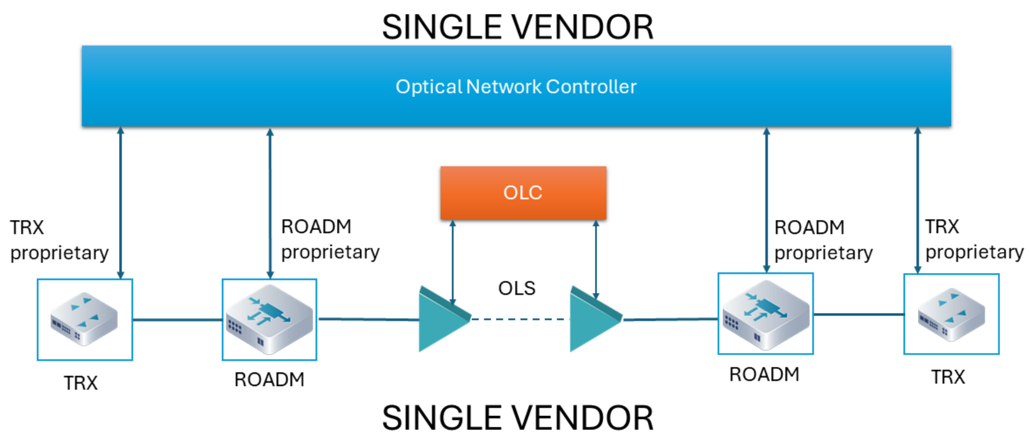


Figure 2.3: Representation of an aggregated network topology

2.1.4 Disaggregated Networks

Partly Disaggregated Networks

Partly disaggregated networks are composed by different parts: in this context only a portion of the devices can be provided by different vendors. This architecture represents an intermediate step between aggregated and fully disaggregated networks.

In general, within a partly disaggregated network the two main blocks are the transponders and the entire optical line system.

Fully Disaggregated Networks

Figure 2.4 shows a scheme of a disaggregated network, where each element can be produced by different vendors: this represents the basis of a multi-vendor network. Within a disaggregated network, it is fundamental to deal with the interoperability among the different models: additional entities are required to guarantee the feasibility of the system, since it is potentially multi-vendor oriented [4]. The advantages of this solution are the cost control due to the competitive market and the faster upgrade of the components.

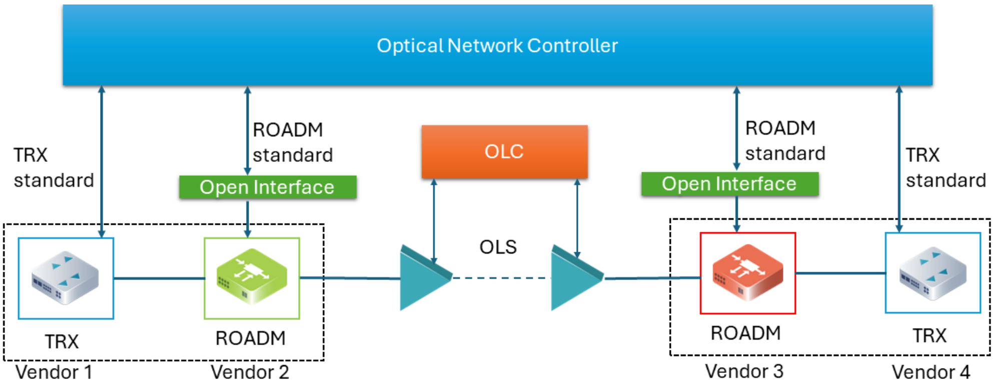


Figure 2.4: Complete scheme of a SDN disaggregated network

2.1.5 Translucent Optical Network

A translucent optical network [5] is characterized by the regeneration of the signal, passing from the optical to the electrical and again to the optical domain. The regeneration happens only in intermediate nodes. This solution reduces the latency introduced by the continuous regeneration of the signal, mitigating the power consumption.

2.1.6 Opaque Optical Network

This network is the extreme case of a translucent optical network: here the signal is regenerated at each node. This strategy avoids the signal degradation, but requires a huge power consumption [6].

2.2 Optical Network Elements

An optical network is composed by several elements able to manage the optical signals. In general, the optical signals follow determined optical paths oriented by the switching nodes that populate the network; at each node, it is possible to add new signals and drop incoming ones, using the transceivers attached to the nodes. All the switching nodes are directly or indirectly connected among each other by an optical line, as shown in Figure 2.5.

An optical network is composed by an optical line system that connects two optical network nodes. Figure 2.5 shows the general structure of an optical network, highlighting how the nodes and the lines are connected.

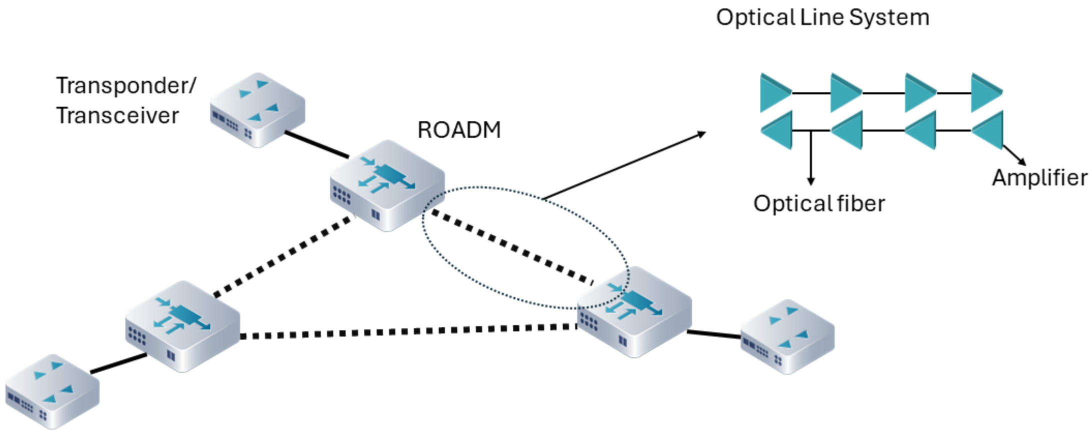


Figure 2.5: Each node is connected by an optical line composed by fiber spans and amplifiers.

2.2.1 Optical Fiber

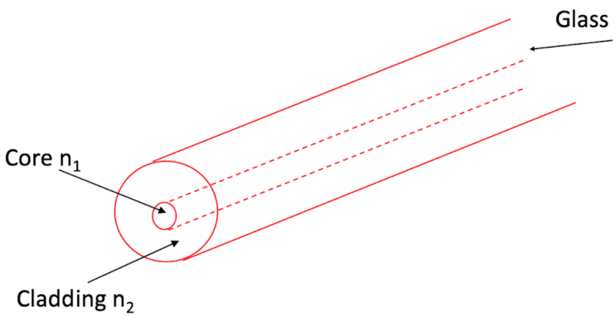


Figure 2.6: Theoretical representation of a fiber.

An optical fiber is a thin glass cylinder [7], with two parts characterized by a diverse refractive index: the internal part is called core, with index n_1, while the external part is the cladding, with index n_2; n_1 is higher than n_2. If the light collides inside the core with an angle of incidence larger than a critical value, θ_c calculated with the Equation 2.1, there will be a total internal reflection of the light that avoids the fiber dispersion and allows a propagation of the signal through the fiber [8].

$$\theta_c = \arcsin \frac{n_2}{n_2} \tag{2.1}$$

A glass material allows very low attenuation values at optical frequencies, as shown in Figure 2.7.

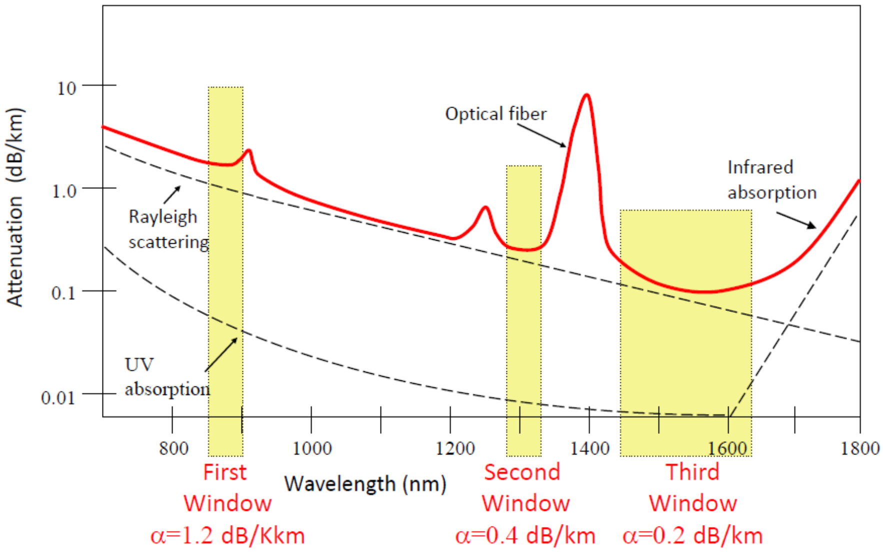


Figure 2.7: Fiber attenuation.

In general, the optical frequencies are between 191 THz and 196 THz, hence the wavelengths are between 1520 nm and 1570 nm: in this region the fiber attenuation reaches the minimum value, around 0.2 dB/km.

2.2.2 Optical Amplifier

An optical amplifier aims to counteract the fiber attenuation. The optical amplification consists in a energy transfer from a laser that pumps optical power to a propagating optical signal. A typical model of amplifier is the EDFA: in this case, the fiber is characterized by a core that contains erbium ions. The signal is sent into the fiber with the pump laser aimed to amplify the signal. The erbium fibers absorb the energy of the pump laser, reaching an higher energy level: this extra energy is transferred to the optical signal[9]. This process is summarized in Figure 2.8.

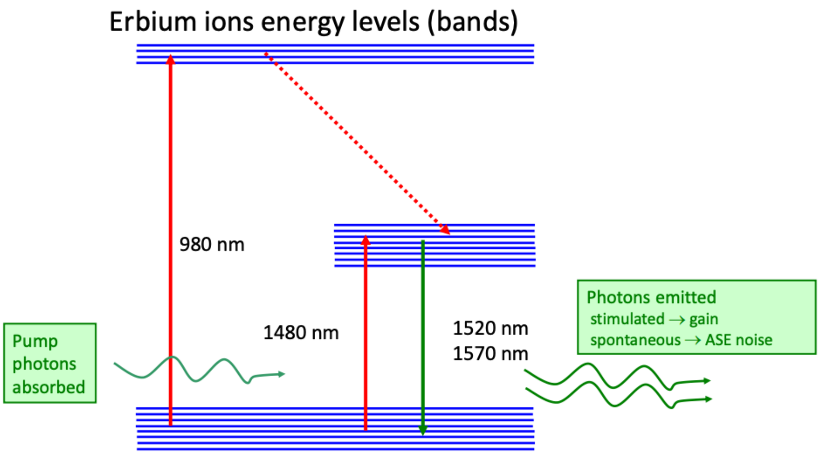


Figure 2.8: Erbium behaviour

The disadvantage is the ASE[10] generation: the excited erbium atoms return to their original power level spontaneously, generating a spontaneous emission. This light can travel along the fiber and it can also be amplified. This behaviour generates a noise to add at the optical signal, as shown in Figure 2.9.

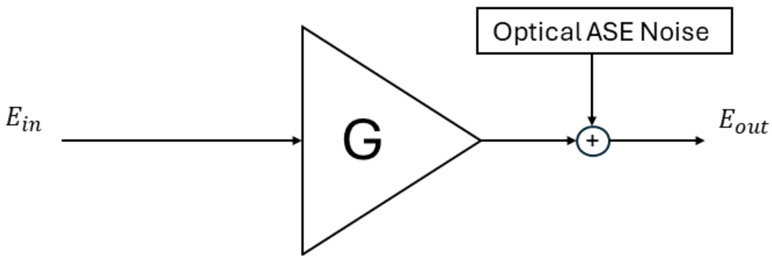


Figure 2.9: Scheme that represents the general effect of an EDFA: it amplifies the signal, but generates an additive noise (ASE)

2.2.3 Optical Line System

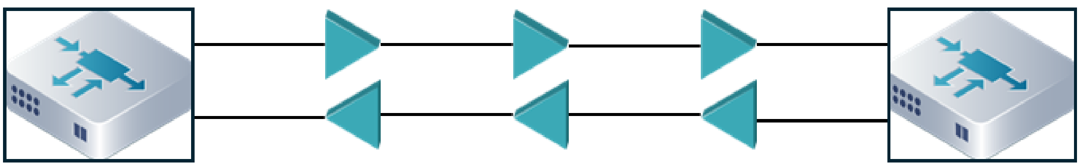


Figure 2.10: Optical Line System

An optical line system is composed by several spans of fibers, that depend on the physical distance between two nodes: each span is composed by a couple of fibers, one for each direction. Each span terminates with the in-line amplifiers, where each ILA is composed by two amplifiers to handle both fiber directions. There are also other two types of amplifiers: the booster amplifier and the preamp amplifier, located respectively at the output and at the input of a switching node.

2.2.4 Transponder and Transceiver

The transceiver is the optical component able to transmit and receive an optical signal, working on a selected frequency, bandwidth and modulation: each transceiver can only transmit or receive one signal at a time. It is connected to a fiber which typically reaches the ROADM for the forwarding of the signal within the network.

Figure 2.11: Examples of transceivers

The transponder is a device that hosts the transceivers: it is composed by many slots; each of them is able to handle a transceiver. It is a software based device and its operating system can be exploited to assign a frequency and a modulation to a specific transceiver [11].

Figure 2.12: Example of transponder

2.2.5 ROADM

ROADMs[12] are the optical entities aimed to forward the optical signals within the network: they are connected to the OLS and transceivers. In particular, these devices are able to apply switching operations according to the WDM rules: a signal is forwarded to a specific destination depending on its central frequency. Due to its reconfigurability, the ROADM can forward each wavelength, or frequency, to any output.

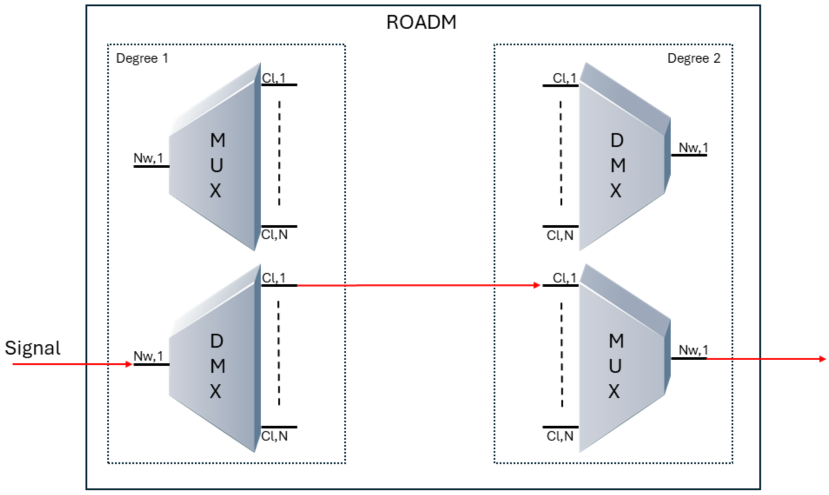


Figure 2.13: General ROADM structure with a signal forwarded from the receiving side to the transmitting one

ROADMs play a key role in the proposed activity, and a more detailed description will be reported in Chapter 3.

2.3 Optical Signal

2.3.1 Optical Transmission Techniques

There are two main approaches to transmit optical signals: IMDD [13] and coherent modulation [14].

Intensity Modulation Direct Detection

This modulation is the simplest technique to transmit signals: it is based on OOK and here the light is switched on to transmit a 1 bit, otherwise off to transmit 0. There is a threshold to detect the transmitted bit: if the power is higher in respect to the threshold, a 1 is transmitted, on the contrary 0 is sent. In this scenario, only the amplitude of the signal is exploited.

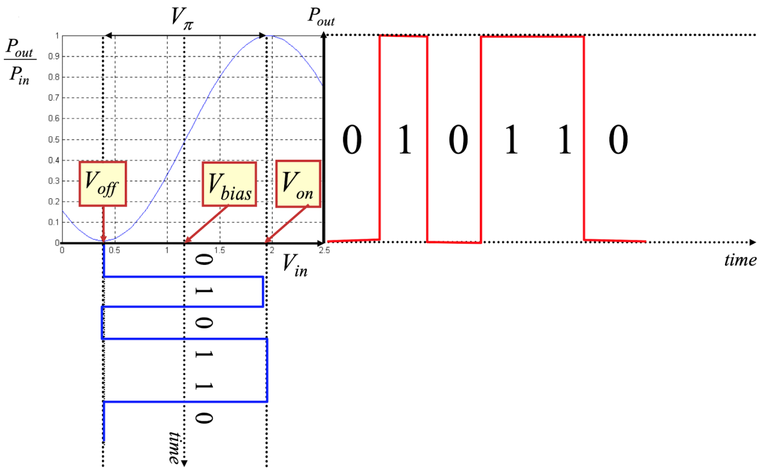


Figure 2.14: IMDD scheme

Coherent Modulation

The coherent communication introduces a multilevel modulation format based not only on the amplitude of the signal, but also on its phase. This modulation increase the data rate, allowing the symbol transmission that contains more data per bandwidth unit.

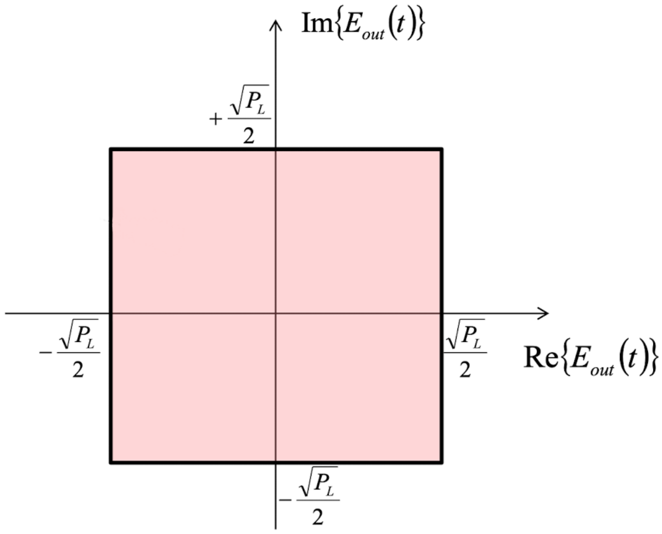


Figure 2.15: Coherent modulation representation

2.3.2 Multiplexing Techniques

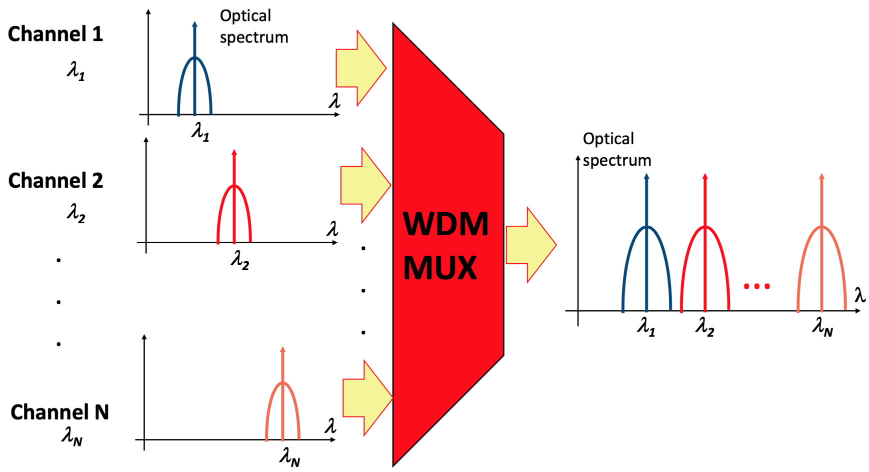


Figure 2.16: WDM representation

The technique used to transmit data in the optical domain is called Wavelength Division Multiplexing[15], which follows the same principle used in wireless communication (FDM based). With this strategy, the optical signals use an allocated portion of the spectrum. Figure 2.16 shows how it is possible to converge different optical signals within a single fiber: each signal is characterized by its own central frequency and channel width, which does not overlap the other channels transmitted towards the same fiber. With this technique, the multiplexing of the optical signals keeps the separation of the different channels, which can be retrieved at the receiver. ROADMs use the WDM technique to forward traffic towards specific directions according to the wavelength that characterizes the optical signal.

2.3.3 Modulation Format

The modulation is the method of encoding information in an optical signal to transmit, exploiting the concept of "symbol". Each symbol is the unit of information and represents the combination of the signal properties included in the selected modulation.

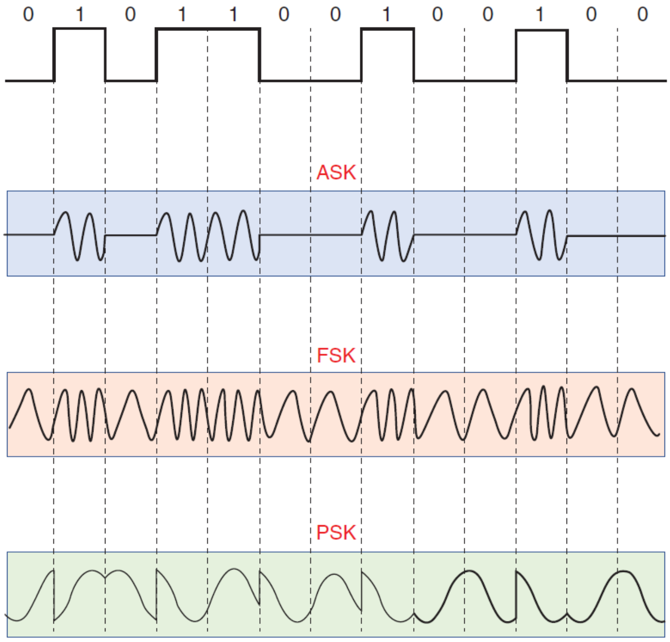


Figure 2.17: An electrical bit stream and the resulting electric field patterns when it is converted to optical domain using ASK, FSK, PSK modulation formats[16].

Several modulation formats can be exploited in optical communications [17]:

- ASK: the information is encoded within the amplitude of the signal;

- PSK: the information is encoded within the phase of the signal. It is possible to distinguish two cases: Binary PSK (BPSK) and Quadrature PSK (QPSK), where the former exploits only two phase values, while the latter is based on four possible phase values. BPSK is translated to 1 bit per symbol, while QPSK to 2 bits per symbol;

- QAM: the information is encoded exploiting both amplitude and phase modulation to increase the data rate. In general, this modulation is referred with the nomenclature M-QAM, where M indicates the number of unique states: the number of bits per symbol can be obtained with the equation $n_{bit} = \log_2(M)$;

- FSK: the information is encoded by varying the frequency of the optical signal.

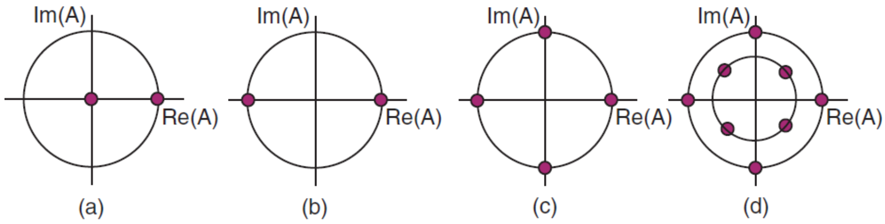


Figure 2.18: Constellation diagrams for (a) ASK, (b) PSK, (c) QPSK, and (d) multilevel QPSK formats.[16].

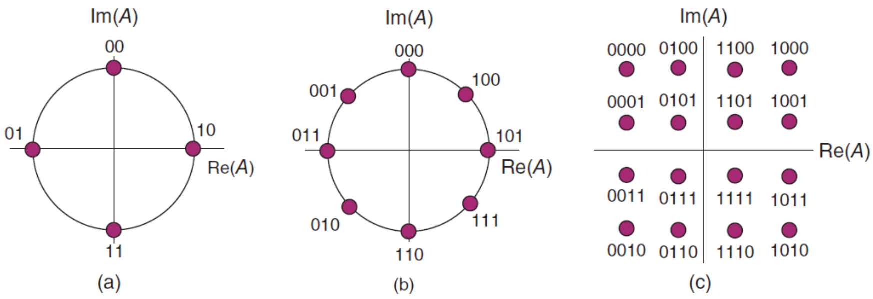


Figure 2.19: Constellation diagrams for (a) QPSK, (b) 8-PSK, and (c) 16-QAM modulation formats showing how multibit combinations are assigned to different symbols[16].

2.4 Open Optical SDN

The open and disaggregated networks represent the next step of the optical network evolution and are potentially characterized by devices provided by different vendors: each device runs its own operating system, so, due to a potential multi-vendor scenario, interoperability must be ensured. To guarantee the interoperability, as mentioned in 2.1.4, standard models and open interfaces must be developed. The open SDN are scalable, due to the independence from the vendors: in this context each device exposes a compatible interface to communicate within the network. The strength of this solution is the following: if the device is changed, the interface remains the same.

The fundamental planes, the control plane, the data plane and the management plane, that characterize the network are managed by separate entities[18].

The control plane aims to perform routing actions: it determines which is the path that each signal must follow. For this purpose, information between neighbour devices are exchanged in order to create optimized routing tables to allow the signals to follow the best paths in terms of costs, latency, link's failure, and so on.

The data plane is responsible of the effective transmission of the data inside the network, according to the forwarding rules, that include the check of the packing headers to retrieve the essential information to route data.

The management plane provides an interface to handle the network infrastructure: thanks to this entity, it is possible to configure the software based devices within the controlled network.

In the SDN networks there is a transition towards a centralized scenario, where a centralized controller is introduced to handle the network devices; in this context, the control and data planes, previously managed within single devices, are now managed by separated entities, as mentioned in Section 2.1.2. In particular, the control plane is managed by the Optical Network Controller, while the devices only aim to forward packets according to the rules imposed by the controller.

2.4.1 Optical Network Controller

Inside an optical SDN, the ONC is the key element that manages the network components, since it handles the behaviour of the devices in order to route the traffic through a determined optical path [19]. This entity is able to manage several devices within the network, such as transponders and ROADMs, using appropriated commands to drive these entities. After setting up a connection towards the devices, the ONC is able to handle the traffic requests by allocating the optical resources, according to the WDM rules. Each established connection is identified by the name "lightpath" [20], characterized by an allocated spectrum portion.

At each optical line, the coexistence of more than one lightpath within the same fiber is guaranteed only if the allocated resources for each OLS are not overlapping, i.e., each lightpath must have a diverse and not already used central frequency, or wavelength. Lightpaths go from a source to a destination, that coincide with two transceivers configured with the same frequency, bandwidth and modulation of the signal: to reach the destination, they may traverse several optical lines, confined between two ROADMs. Since the lightpaths cannot change their parameters, the wavelength continuity must be ensured from the source to the destination: a checking phase must be performed before allocating a lightpath to find an available wavelength. If there is no availability, blocking event occurs, otherwise it is possible to allocate an optical path form the source to the destination [21].

The ONC is able to communicate with the different devices within the network through standard models aimed to control the network entities.

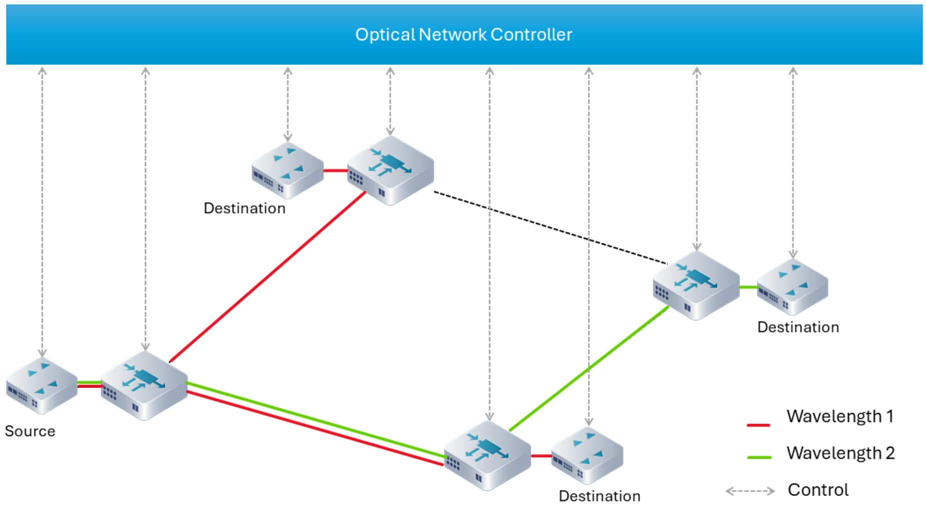


Figure 2.20: Optical network controller

Figure 2.20 represents how the optical network controller interfaces the different software based devices within the network. In this example, three different lightpaths are allocated: two are characterized by the same wavelength, red, while the third is characterized by a different wavelength, green, in respect to the others. In this example, two lightpaths don't share any optical lines to reach their destination, so the reuse of the same wavelengths is possible. On the other hand, two lightpaths (red and green) share one optical line: here it is mandatory to avoid the resource overlapping, i.e., the two lightpaths must be characterized by a different wavelength, according to the WDM and RWA rules.

2.4.2 Optical Line Controller

The OLC is a part of the control plane dedicated to the management of all the amplifiers. In particular, its aim is the setting of each amplifier's gain within an OLS and also the power per channel at the input of each fiber span [22]. An OLC communicates with ILAs, booster and pre-amplifier.

An OLC is a proprietary software aimed to manage an OLS: each OLS is generally associated to an optical line controller. Also in this context there can be different models of OLCs: for this reason, between proprietary controller software and others control plane components, standard interfaces must be implemented to guarantee interoperability.

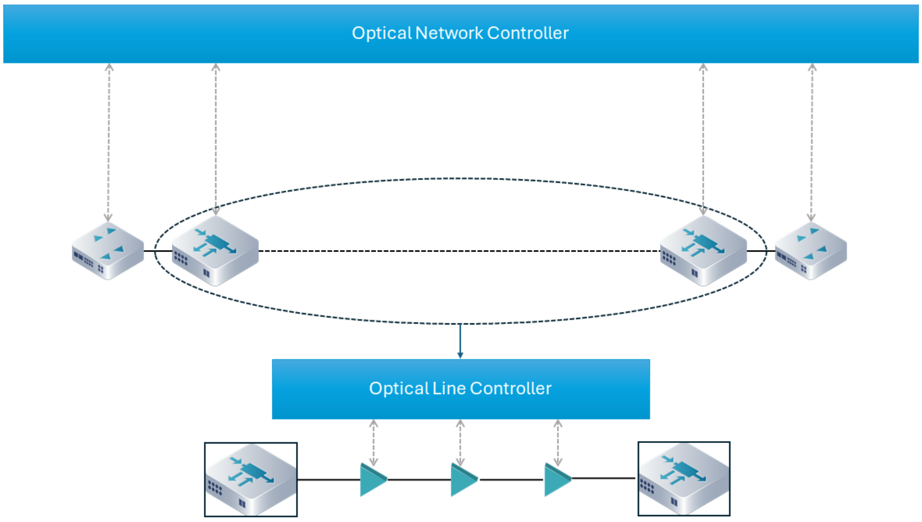


Figure 2.21: The optical line system is managed by the OLC

2.4.3 Interfaces and Standards

The open interfaces represent additional and fundamental resources within the disaggregated open optical networks [23]. They aim to guarantee a feasible interoperability among all the open optical network components, that include ROADMs, transponders and amplifiers. These additional layers are able to expose the characteristics of the devices and guarantee communication with the controller, which works using standard models. Furthermore, these interfaces allow the substitution and the enhancement of the network infrastructures without the redefinition of the controller's behaviour.

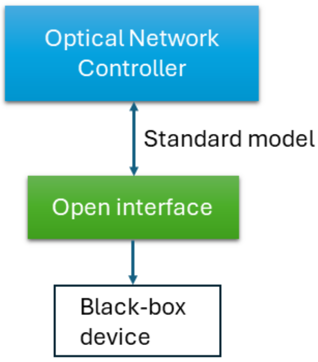


Figure 2.22: An open interface inserted between the controller and the device

A black-box device is a non accessible entity within a disaggregated network since it does not support the standard model used by the optical network controller to control the network devices. The insertion of an open interface makes the entity a white-box device [24], accessible by the optical network controller.

Figure 2.23: With the introduction of an open interface, a black-box device becomes accessible by the ONC, now able to use a standard model.

The open interface aims to collect functions retrieved by the ONC written in the standard model and translate them according to the proprietary model of the entities to manage. Furthermore, its role is crucial to expose the architectural features of the devices: the ONC is able to allocate the necessary resources for the optical signal, according to the architecture of the software based entities.

Chapter 3

Reconfigurable Optical Add Drop Multiplexer

The ROADM is a device used to forward optical signals within an optical network. There are different models, produced by different vendors, such as CISCO or Adtran, and each of them has its own hardware and software, but the general functions are the same: each ROADM, independently from the model, is able to forward the incoming signals, is characterized by the same components, that can differ in quantity, and is characterized by a software that can let the device reconfigurable. In this chapter, the general model will be described at first, and then, a focus on the Adtran model is presented, since it has been used in laboratory for the experiments.

3.1 General ROADM description

3.1.1 The Degree

The core of the ROADM is an entity called degree, and each ROADM is composed by a single degree or a repetition of them: each degree is a potential direction that an incoming signal can follow [25]. Degrees are composed by a couple of multiplexer and de-multiplexer: the former is used to converge all the internal signals to the external network, while the latter is used to switch an incoming signal from the optical line to the internal side of the degree.

At each multiplexer, two types of ports are present: the network port and the client port. Each port is bidirectional, so it is able to transmit and receive signals; typically, there is a single network port and several client ports. These two ports are used to manage and forward the signals, and each port is physically accessed by a couple of connectors, one dedicated to the transmission (labeled with TX) and the other to the reception (labeled with RX) of the signals; at the connectors the fibers must be inserted.

The network and the client ports work mostly in the same way, but their aim is different: the network port is used to communicate directly with the optical network, in both directions, while the client ports are used for the communication with a transmitter or a receiver, in general a transceiver, where the former usually sends a signal to a receiving client port, while the latter usually receives a signal from a transmitting client port. The second function of the client ports is to route the signals between different degrees which belong to the same ROADM.

Within the MUX/DMX, the pair of ports can be connected by the operating system, and the connection can be unidirectional or bidirectional.

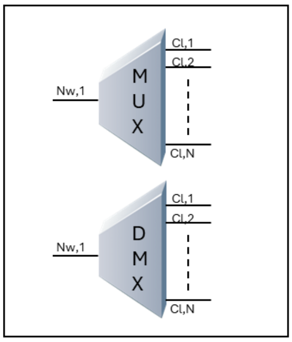


Figure 3.1: This figure represents a simplified scheme of a degree, focusing on the MUX/DMX and the ports.

Figure 3.1 shows the essential elements that compose a degree inside a ROADM: it is possible to observe the presence of the couple MUX/DMX, at which the ports are connected; the network port is located to the optical line side, while the client ports face towards the other degrees of the same ROADM or the transceivers that transmit or receive the signals.

Within ROADMs, internal amplifiers interfaced directly with the degrees can be added: they act as pre and booster amplifiers to compensate the internal losses introduced by MUX/DMX.

3.1.2 Add/Drop

A ROADM module is able to forward the signals coming from the network: it can receive and re-transmit a signal, otherwise it can establish a connection with the optical transceiver to transmit new signals through the network (add) or forward incoming signals to the receivers (drop); in the following there are explained these two functions, both related to the general concept of the routing functions.

The principle of add and drop is the opposite: the adding function is associated to the action of the ROADM to collect new signals coming from the transceivers, and these signals are added to the network; on the opposite, the dropping function is the action that the ROADM performs in order to route a signal, or a group of signals, characterized by specific frequencies, to the proper receiving transceivers.

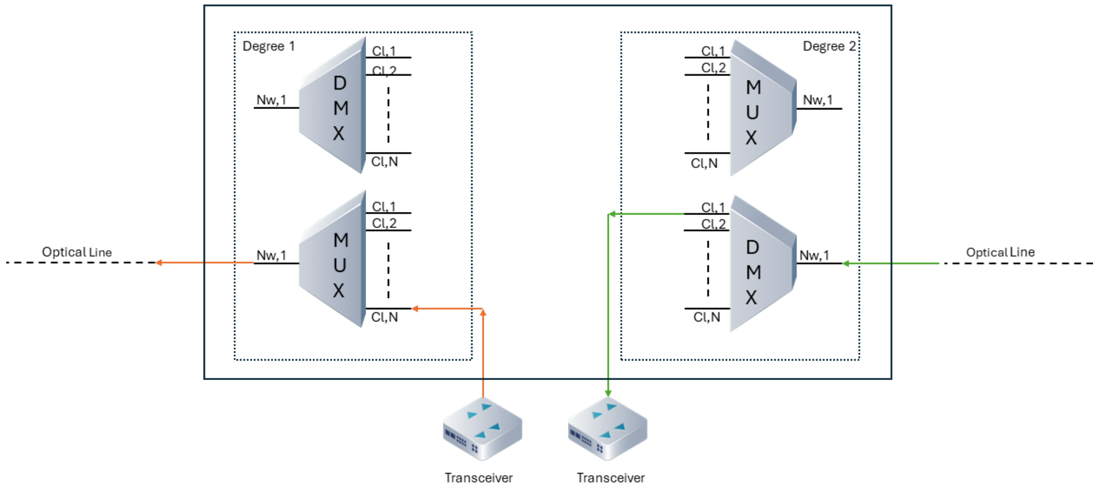


Figure 3.2: Add and drop functions

Figure 3.2 represents a scenario where both add and drop situations are tested: the left part shows the adding scenario, while the right side is related to the dropping one. Starting from the left side, the transceiver wants to transmit a signal characterized by a fixed central frequency: the signal is sent to a receiving client port and then to the transmitting network port, to be finally forwarded towards the optical line. On the other hand, as represented at the right side, a signal that arrives from the optical line can be forwarded to the transceiver: in this case the signal is dropped from the network.

To handle the add and drop functions, additional supports can be found within ROADMs: transceivers are not directly connected to the client ports, but they communicate through an intermediate interface.

3.1.3 The Concept of Reconfigurability

ROADMs are characterized by an internal operating system that can be used for different tasks, where the most important ones are related to principle aim of the

ROADM: the forwarding of the signals. In particular, the integrated operating system is helpful for the implementation of the WDM rules for the management of the incoming signals, from the transceivers or from the optical lines, which can be switched according to their central frequency. Other functions can be exploited for retrieving useful information, such as the power level or the status of the channels.

In the following, it is explained the potentiality of a reconfigurable device if compared to a non-reconfigurable device.

ROADM vs Non-Reconfigurable Multiplexer

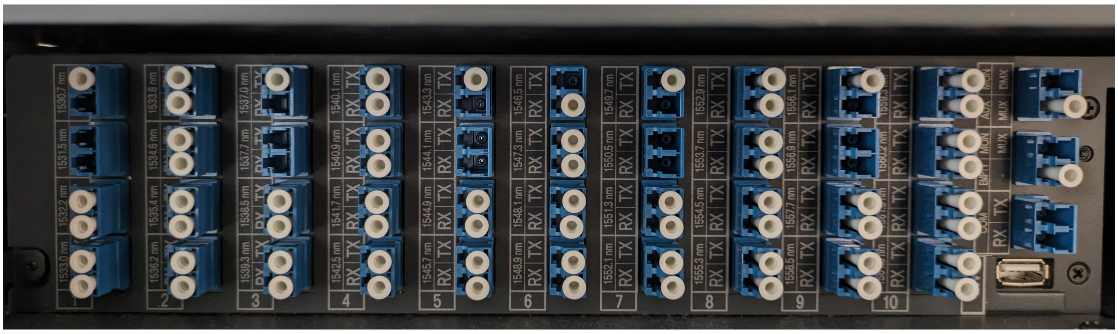


Figure 3.3: This figure represents a non-reconfigurable multiplexer provided by CISCO.

The difference between a non-reconfigurable add-drop multiplexer (OADM [26]) and a ROADM must be found in the WDM implementation: the former cannot change the routing strategies, while the latter is able to change the way to switch the signals.

Starting from a non-reconfigurable multiplexer, the rule used to forward an incoming signal is fixed. Given an incoming signal characterized by a specific central frequency, the way to route it is predefined: the signal arrives to the receiving network port and, according to its wavelength, and consequently to its frequency, is forwarded to a specific client port, which is not possible to be changed. In fact, every port is characterized by a specific wavelength, so it is not possible to change the switching rules, since they are fixed for each port (Fig. 3.3).

On the other hand, a ROADM allows the redefinition of the switching rules: using the proper commands, it is possible to decide how to manage the signals, and according to their central frequency, how to route them; it is also possible to change the switching rules in a second moment. This is possible for the presence of the integrated operating system: it allows the possibility to decide which are the central frequencies that each client port has to manage. It is possible to select a specific central frequency, and each signal characterized by that frequency that arrives at a specific network port is automatically switched to the selected client port. Then, if it is necessary to change the switching rules, it is possible to select another destination port. Then, it is also possible to forward different signals at different wavelength to the same client ports if a ROADM is used: the unique rule that must be respected is the avoidance of overlapping of the channels, due to the WDM rules.

3.2 Adtran Model

The vendor of the ROADMs present in laboratory is Adtran, and, in particular, the model is the *FSP3000C*.

Figure 3.4: FSP3000C model of Adtran

3.2.1 Adtran's Hardware

The *FSP3000C* is composed by 7 slots, which are the physical positions of the degrees inside the ROADM, and they could be compared to drawers: each drawer is a different slot within the single ROADM, and inside it, it is possible to physically locate a degree. In this specific case, the slots used are the first, the third and the fifth: this means that only three slots are busy and they host a single degree; as a consequence there are in total three degrees. It is important to mention the slots, since they are used by the operating system to access the degrees: for example, if a user wants to reach the ports of the second degree, it must refer to the third slot.

All the degrees are characterized by thirteen ports: a single network port and 12 client ports, and each of them can be used for transmitting and receiving the signals. For each port, in fact, there are two other ports, which can be distinguished using the labels, that describe their purpose: TX or RX.

Figure 3.4 shows the Adtran model: the three degrees are clearly visible and it is possible to observe the 13 ports for each degree. In the right part it is possible to identify the 7 slots in which degrees must be located.

3.2.2 Adtran's Operating System

A ROADM can be controlled by an internal software, characterized by several features that can be exploited in order to use the device for different purposes. The most important are:

- the creation of connections to switch the optical signals within a MUX/DMX;
- the equalization of the channels;
- the power value retrievement.

There are different ways to interact with the device, since the operating system of the ROADM can be accessed via SSH, RESTCONF[27] or NETCONF[28] protocols: the way to manage the device is almost similar between the diverse approaches, while the syntax of the commands changes in respect to the chosen protocol.

RESTCONF

RESTCONF is a protocol based on HTTP and is used in this context to communicate with the operating system of the ROADM. RESTCONF requests and responses are structured in a header field and a body. The body is formatted according to a JSON structure and contains the parameters and the action that the ROADM must follow, while the header contains different fields in which information, (e.g. return type, authorization token, ..) must be inserted. The frequently used actions are:

- GET: to retrieve information;
- POST: to create new structures;
- PATCH: to modify already existing structures;
- DELETE: to delete already existing structures.

Within the response's header field it is reported the status code of the operation to check if a specific request has been satisfied: in general, as happens for HTTP, status codes with the format 2xx indicate the success of an operation, while 4xx a failure (x can be substituted with integer numbers).

The following string represents the endpoint of the network port within the degree 1 of a ROADM:

```
/mit/me/1/eqh/shelf,1/eqh/slot,1/eq/card/ptp/nw,1
```

The most important information to extract from the endpoint are:

- *shelf,1*: identifier reported in the display in the upper left of the ROADM (Fig. 3.4);
- *slot,1*: physical position of the degree within the ROADM (slot 1 is the identifier of the first degree);
- *nw,1*: the selected port within the degree; the direction is not considered.

3.3 Actions and Features

3.3.1 Creation of Connections

In this paragraph, it is described the procedure to create a connection between two ports, exploiting the MUX/DMX inside a degree. This procedure represents the starting point for the creation of connections between degrees and the injection of the signals towards the optical lines.

Several steps must be computed to manage the incoming signals, from the optical lines or transceivers, and they are collected in the following list:

1. creation of a SPSLG;
2. creation of an OTSiA associated to a SPSLG;
3. creation of the connection between two ports (client and network) within a MUX/DMX;
4. setting of the OTSiA state to *In-Service*.

If it is necessary to connect two degrees, these steps must be performed for each one, otherwise, such as in the case of add or drop functions, it is sufficient to execute these functions only within the interested degree.

SPSLG creation

The first step is the creation of a spectrum slot group: it is an ensemble of one or more spectrum slots. In general, a single spectrum slot is a continuous spectrum, characterized by a central frequency and a width[29]. These two parameters must respect the following rules: the central frequency must fall in the C-Band [30], from 191.3 THz to 196.1 THz, while the width must be included between 37.5 GHz and around 5 THz; in particular, the width must follow the equation $W = K \cdot 6.25 + 37.5$, where K is an integer number that defines the increment of the width from the lowest bound. In each port, the created spectrum slots cannot overlap the others. Another important parameter that characterizes a spectrum slot is the direction: it can be unidirectional or bidirectional. In particular, if a slot is unidirectional, it can only transmit or receive a signal, so, as a consequence, the direction (tx or rx) must be specified. If a slot is bidirectional, the signal characterized by the same central frequency of the slot can be transmitted and received, so it will be propagated in both directions. By default, the direction is set as bidirectional.

The purpose of the creation of a spectrum slot is related to a filtering action made on an incoming signal: thanks to the division in slots, it is possible to divide the entire stream in specific slots, characterized, as said before, by a central frequency and a width; the stream is divided in channels that can be managed separately. The signal inside each channel can be managed separately, according to the rules specified with the help of the operating system: some channels can be dropped, others can be forwarded.

Here follows an example of the executed command within the network port:

```
curl -d '{"tpdir":"unirx","sm":{"admin":"is","isst":["ains"]},"usrlbl":"","spslg":{"spsl":[{"slotfrq":193500000,"slotwdth":50000,"id":1}]}}' \
-H "X-Auth-Token: TOKEN" \
-k -X POST -i https://IP_ADDR/mit/me/1/eqh/shelf,1/eqh/slot,1/eq/card/ptp/nw,1/ctp/oms/ctp/spslg-1
```

The same command must be repeated to the client port at the other side of the MUX/DMX.

Within the body, it is potentially possible to define more spectrum slots, which must be indicated by different integer identifiers.

OTSiA creation

The second step is the creation of the OTSiA, which is the carrier of the signal: its role is related to the allocation of the resources used in order to transport the signal contained by the spectrum slot at which the tributary signal belongs to. This operation must be performed for each port that will be crossed by the signal: in particular, the OTSiA is associated to a specific spectrum slot of a specific port, and it must have the same central frequency, the same width and the same direction of the associated slot.

Here follows an example of the executed command within the network port:

```
curl -d '{"tpdir":"unirx","sm":{"admin":"oos"},"usrlbl":"","otsia":{"
    otsiacar":{"nodeinp":1},"otsi":[{"id":2,"freq":193500000,"optbw
    ":50000,"cpmgt":{"sptdev":0}}]}}' \
-H "X-Auth-Token: TOKEN" \
-k -X POST -i https://IP_ADDR/mit/me/1/eqh/shelf,1/eqh/slot,1/eq/card
    /ptp/nw,1/ctp/oms/ctp/spslg-1/ctp/otsia
```

The same command must be repeated to the client port at the other side of the MUX/DMX.

Media Connection

The third step allows the ROADM to connect two spectrum slots which are physically connected by a MUX or DMUX: this operation is executed within the degree, and its goal is the creation of a connection between a client port and a network port. The command used to create this connection requires the specification of both the endpoints, which are the spectrum slots: there must be a frequency matching between the slots, so the central frequency and the width of the channel must coincide. If the connection between the two slots is bidirectional, the order of the ports is not important. On the other hand, if the connection is unidirectional it is mandatory to specify at first the receiving slot and then the transmitting one.

```
curl -d '{"aendlist":["/mit/me/1/eqh/shelf,1/eqh/slot,1/eq/card/ptp/
    nw,1/ctp/oms/ctp/spslg-1"],"zendlist":["/mit/me/1/eqh/shelf,1/eqh/
    slot,1/eq/card/ptp/cl,1/ctp/oms/ctp/spslg-1"],"entname":"1"}' \
-H "X-Auth-Token: TOKEN" \
-k -X POST -i https://IP_ADDR/mit/me/1/eqh/shelf,1/eqh/slot,1/eq/card
    /sn/media/snc
```

OTSiA status

Finally, the status of the OTSiA of each port must be set to *in-service* in order to start the forwarding of the signal through all the interested ports. By default, the status is set to *out of service* and if it is not switched to IS, the channel is inactive and not observable with the instruments used to represent the power spectrum.

```
curl -d '[{"op":"replace","path":"/sm/admin","value":"is"}]' \
-H "X-Auth-Token: TOKEN" \
-X PATCH -k -i https://TOKEN/mit/me/1/eqh/shelf,1/eqh/slot,1/eq/card/
    ptp/nw,1/ctp/oms/ctp/spslg-1/ctp/otsia
```

The same command must be repeated to the client port at the other side of the MUX/DMX.

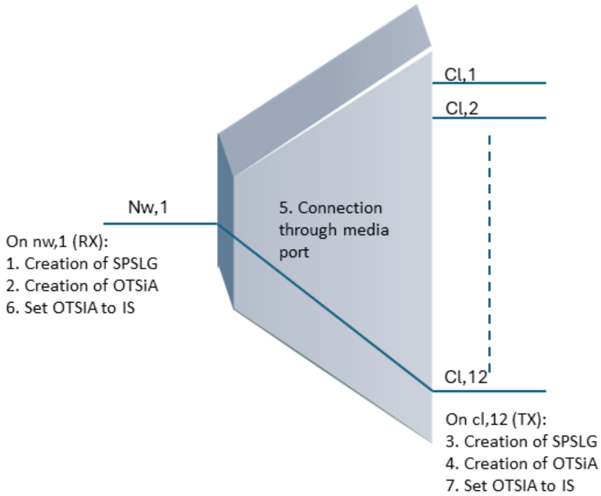


Figure 3.5: This figure represents the steps performed inside a degree for the connection between the network port and a client port.

Figure 3.5 shows a graphical representation of the steps described in this section.

3.3.2 Equalization

The equalization, also named as control output power, is an important feature that can be exploited. This function is used to control the output power of each created channel, by manipulating a specific parameter called setpoint PSD: this parameter is expressed in dBm/GHz, and its value is included in a specific range that varies according to the selected port (from -32 dBm/GHz to -39 dBm/GHz for the network port and from -24 dBm/GHz to -35 dBm/GHZ for the client ports).

The equalization can be enabled or disabled: when it is enabled, the PSD of the channels is scaled to the setpoint value, if the input power is higher than the setpoint, otherwise, if the equalization is disabled, each channel does not receive any extra attenuation. It is important to specify that the equalization works only inside the transmitting ports, and it is not possible to disable it channel per channel, but its status is referred to the port, e.g., if the equalization status of the network port of a specific degree is enabled, all the created channels will receive the extra attenuation. Although all the channel must receive an extra attenuation if the equalization is enabled, there is another parameter, called setpoint delta, that allows to play with the attenuation value for the specific channels: this parameter has a selectable range from -6 dBm/GHz to 6 dBm/GHz, and is added to the setpoint PSD. By default, it is set to 0 dBm/GHz, so the attenuation is only introduced with the equalization.

3.3.3 Power Monitoring

The ROADM is equipped by a power monitor, called OCM [31], an internal component able to give information related to the total transmitted and received power of each port. It is also able to show the total power of each channel. This instrument is useful because allows to analyze the power evolution at each ROADM, equipped of an OCM, within the optical line. Due to its structure, it is not possible to retrieve the power in respect to a specific frequency, but a single value that represents the integral power in respect to a channel is shown: there is an association between the central frequency of a channel and the total power that falls into that specific channel. Analyzing all the power information of the crossed ports, it is possible to understand if a power reduction is experienced, due to the presence of internal losses and if an extra attenuation that comes from the equalization is applied.

To conclude, it is possible to assert that this instrument can be very useful to study the behaviour of the ROADM and retrieve information related to the characteristics of the device, such as the internal losses. If the values of the output power per channel, which comprehends both internal losses and attenuation, are distant from what is expected, it is possible to affirm that something is not working in the proper way, so this instrument is also fundamental for the debugging of the system when there are differences between what is read and what is theoretically expected. On the other hand, the resolution is not very high, since it is possible to obtain only one value of power per channel, so the visualization of the spectrum is not so precise as an external device, such as an OSA [32].

Chapter 4

Disaggregated Multi-Vendor Networks

The majority of the optical networks are single-vendor oriented: the controller and the devices are provided by the same vendor, so the interoperability among the components is guaranteed. On the opposite, a transition to multi-vendor networks could offer several advantages: the costs can be controlled through the diverse market options and it is possible to easily enhance the network infrastructure. In order to achieve a multi-vendor scenario, the interoperability between the diverse vendor devices must be guaranteed: standard models are developed to let the ONC the communication towards the devices, without caring about their proprietary language, and an open interface must be inserted to allow the communication between the ONC and the diverse devices, if they don't support the standard model used by the controller [33].

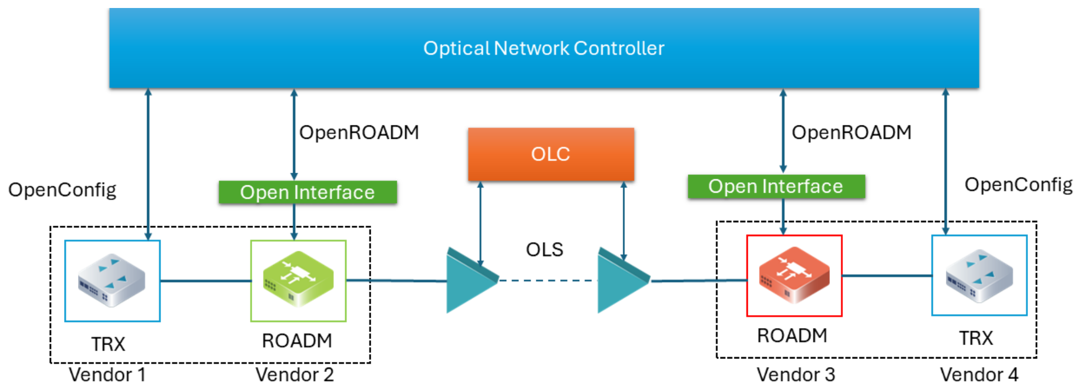


Figure 4.1: General strategy on how to approach interoperability on multi-vendor devices

Figure 4.1 represents an example of a multi-vendor network: each device is potentially provided by a different vendor and the optical network controller communicates towards the ROADM devices and the transponders exploiting the proper standards. The standard commands are then translated with the help of an open interface: its aim is the conversion of the commands sent by the controller from the chosen standard to the proprietary language supported by the specific device (Section 2.4.3).

In this context, the insertion of proprietary ROADMs within a multi-vendor network has been tested: as explained in Chapter 3, they are characterized by their own operating system and the commands used to create and delete the necessary connections for the data handling are run according to the proprietary language. To achieve the interoperability among different vendors, a driver able to convert the requests to the proprietary language must be implemented and inserted between the ONC and the black-box devices.

The main goal of the following activity is to illustrate how a communication between a controller and a commercial ROADM can be established.

4.1 Setup

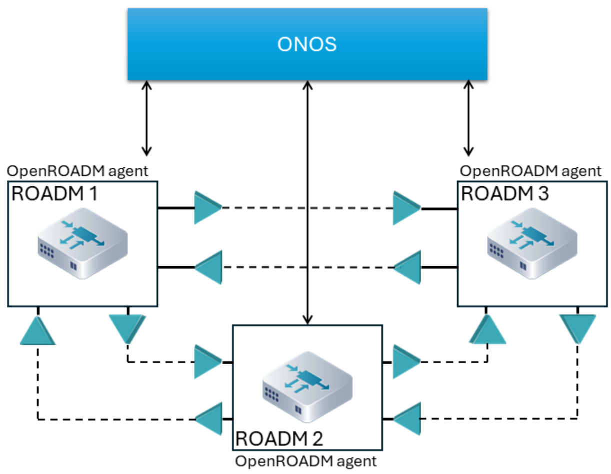


Figure 4.2: This scheme represents how the three nodes are connected in the lab

The setup in the laboratory is composed by three ROADMs, produced by Adtran, which are connected by bidirectional optical lines: here several EDFA amplifiers are located to compensate the loss introduced by the fibers. Figure 4.2 shows how the ROADMs are connected among each others.

Within the same figure, three OpenROADM agents are reported: they work in the same way, but each of them is associated to a specific ROADM.

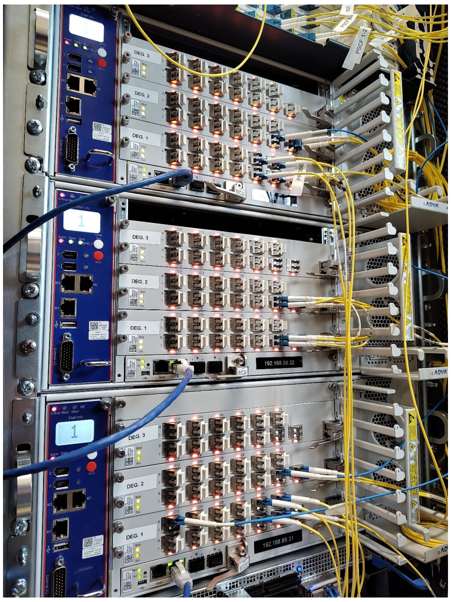


Figure 4.3: Setup in the laboratory

4.1.1 ONOS and Adtran Connection

The controller of the open optical SDN taken in exam is ONOS[34], an open source software able to control the devices within the network. It supports the standard model OpenROADM [35] in order to control the ROADMs present in the network; this system is helpful because it allows the usage of a generic set of functions to interface all the ROADMs, independently from the model. On the other hand, this standard is not recognized by the devices, because their operating system can only accepts specific formats of instructions. As a consequence, a middleware must be introduced between the controller and the ROADMs, in order to translate the OpenROADM requests to the proper model language.

The middleware is called OpenROADM agent, a docker container [36] that includes a software, written in C language, able to collect the requests sent by ONOS and translate them in RESTCONF commands that must be directed to the ROADMs; each ROADM is associated to a specific agent.

The communication between ONOS and the ROADMs is split in two parts: at first, the controller of the network sends the commands to the OpenROADM agent, and then, the commands are translated in a ROADM readable language, thanks to the script contained inside the middleware.

4.1.2 OpenROADM

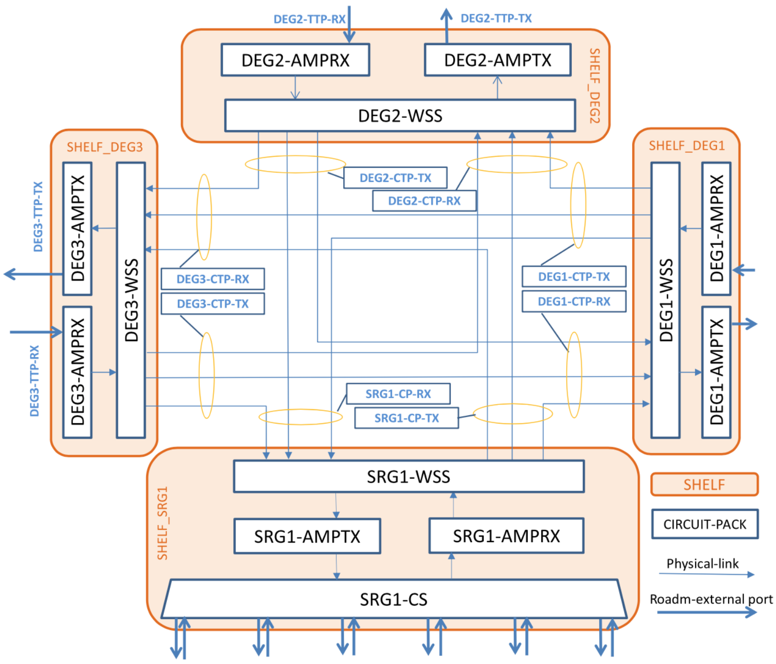


Figure 4.4: OpenROADM scheme with nomenclature

OpenROADM is a standard aimed to guarantee interoperability within a multi-vendor network, in particular operating with ROADMs and transponders. In this context it is used for the creation of express-connections within the ROADM device and cross-connections between two ROADM ports: an express-connection, Figure 4.6, defines a connection between two degrees of the same ROADM, while a cross-connection, Figure 4.5 is related to the connection between two ports physically connected at the same MUX/DMX of a specific degrees. Although the two types of connections involves different entities, the way they work is the same. Furthermore, the cross-connection between two ports represents also the basis of the signal forwarding towards a ROADM at the end of an optical line (Figure 4.7).

Figure 4.4 represents the adopted nomenclature linked to the theoretical hardware infrastructure of the ROADMs. Within the structure, also the amplifiers at the input/output of the degrees and the SRG modules, used for the add/drop actions, are reported, but in the laboratory the ROADMs are not equipped by these features: transceivers are directly connected to the client ports of the ROADMs and the the booster and pre-amplifier are the first and the last amplifiers of each optical line.

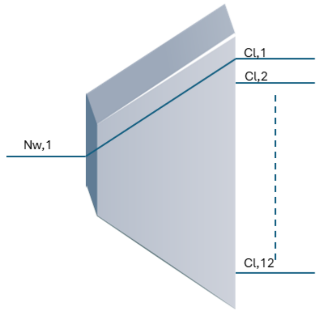


Figure 4.5: Simplified representation of a cross connection between two ports within the same MUX/DMX

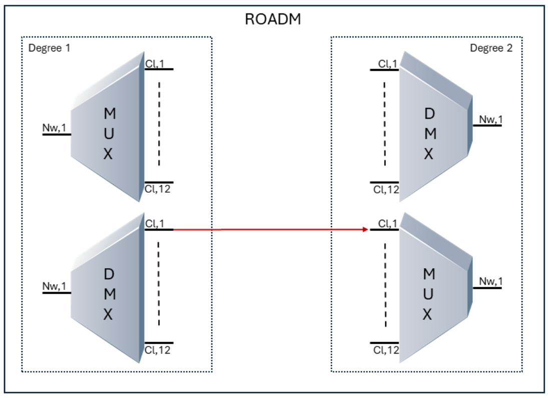


Figure 4.6: Simplified representation of an express connection between two degrees within the same ROADM

Figure 4.7: Simplified representation of a connection between two ROADMs within the same optical line

Nomenclature

Here it is reported the mapping of the most important entities from the Open-ROADM standard to the Adtran's proprietary language:

- DEGx-WSS → position of degree x among the slots;

- DEGx-WSS-RX → receiving *nw,1*
- DEGx-WSS-TX → transmitting *nw,1*
- DEGx-WSS-INy → receiving *cl,y*
- DEGx-WSS-OUTy → transmitting *cl,y*

OpenROADM identifies both transmitting and receiving directions for each port, considering a specific port as two different entities. On the other hand, Adtran does not provide this seperation, but, as mentioned in Section 3.3.1, it is possible to indicate the channel's direction when the resources to handle the signal are allocated.

4.1.3 The OpenROADM agent

The OpenROADM agent [37] is a docker container and includes several files, in particular: the compiled version of the ROADM's driver, which implements the functions listed and explained in Section 4.1.3, and three datastores that describe the structure of the ROADMs.

The first two datastores are: *i) config.xml*, which contains a detailed description of the ROADM associated to the specific agent, including the degrees, the ports and eventually the amplifiers, *ii) status.xml* which contains information of the ROADM's model and its components, such as the direction of each port. The last datastore is named *circuit_packs.xml*, in which there are contained the information related to the initialization of the involved entities, in this case the three degrees. The three datastores use the OpenROADM's nomenclature.

```xml
<circuit-pack>
    <name>DEG1-WSS</name>
    <connection type="rest">
      <address>IP_ADDR</address>
      <port>PORT</port>
      <username>username</username>
      <password>password</password>
    </connection>
  </circuit-pack>
```

Listing 4.1: circuit_packs.xml

4.1 shows an example of a circuit pack inserted within the datastore *circuit_packs.xml*: in particular, all the necessary parameters to connect to the ROADM's degree 1 are listed. In this case, this datastore contains the same parameters for the other two degrees, since the operating system considers the three degrees under the same entity.

```
[...]
  <info>
     <node-id>ROADM-Torino</node-id>
     <node-number>1</node-number>
     <node-type>rdm</node-type>
[...]
<circuit-packs>
    <circuit-pack-name>DEG1-WSS</circuit-pack-name>
    <circuit-pack-type>circuit-pack</circuit-pack-type>
[...]
<ports>
  <port-name>DEG1-WSS-IN1</port-name>
  <port-qual>roadm-internal</port-qual>
  <logical-connection-point>DEG1-CTP-RX</logical-connection-point>
[...]
<ports>
  <port-name>DEG1-WSS-RX</port-name>
  <port-qual>roadm-internal</port-qual>
  <logical-connection-point/>
</ports>
[...]
</circuit-packs>
[...]
<physical-link>
<physical-link-name>ExpLink12</physical-link-name>
<source>
  <circuit-pack-name>DEG1-WSS</circuit-pack-name>
  <port-name>DEG1-WSS-OUT1</port-name>
</source>
<destination>
  <circuit-pack-name>DEG2-WSS</circuit-pack-name>
  <port-name>DEG2-WSS-IN1</port-name>
</destination>
</physical-link>
[...]
<degree>
<degree-number>1</degree-number>
[...]
<circuit-packs>
  <index>2</index>
  <circuit-pack-name>DEG1-WSS</circuit-pack-name>
</circuit-packs>
</degree>
[...]
```

Listing 4.2: config.xml

4.2 contains details related to the first degree circuit pack of a ROADM within the datastore *config.xml*:

- network and client ports, where their two directions are considered as separated entities;
- the linked circuit pack name, where the details are reported within the datastore *circuit_packs.xml*;
- the hardware position, i.e., the shelf and the slot at which the circuit pack belong;
- the ports involved in the express connections between two degrees.

Also an accurate hardware description is reported, that includes the structure of the ROADM, considering the number of degrees, the presence of amplifiers and add/drop modules.

```
 [...]
<info>
  <vendor>Adtran</vendor>
  <model>RDM12</model>
  <serial-id>LBADVA71234100274</serial-id>
   [...]
  <max-degrees>3</max-degrees>
   [...]
</info>
 [...]
<circuit-packs>
  <circuit-pack-name>DEG1-WSS</circuit-pack-name>
  <administrative-state>inService</administrative-state>
  <circuit-pack-category>
      <type>circuitPack</type>
  </circuit-pack-category>
  <ports>
      <port-name>DEG1-WSS-IN1</port-name>
      <port-wavelength-type>wavelength</port-wavelength-type>
      <port-direction>rx</port-direction>
       [...]
  </ports>
   [...]
  <ports>
      <port-name>DEG1-WSS-RX</port-name>
      <port-wavelength-type>multi-wavelength</port-wavelength-type>
      <port-direction>rx</port-direction>
       [...]
  </ports>
  <ports>
      <port-name>DEG1-WSS-TX</port-name>
      <port-wavelength-type>multi-wavelength</port-wavelength-type>
      <port-direction>tx</port-direction>
       [...]
```

```
  </ports>
</circuit-packs>
 [...]
<!-- DEGREE -->
<degree>
  <degree-number>1</degree-number>
    [...]
</degree>
 [...]
```

Listing 4.3: status.xml

4.2 lists some details within the datastore *status.xml*: the ROADM vendor and model, the features of each circuit pack, specifying the ports and their direction, TX or RX, and also in this case the feetures of the ROADM, including the number of degrees.

Driver's Functions

Four main functions have been developed: *i) init* function for the initialization of the environment; *ii) close* function used to free the allocated resources; *iii) make_connection* function to create a connection; *iv) delete_connection* function to close an already created connection.

Init is the first executed function when the OpenROADM agent is run: its aim is the retrieving of the general information of the degrees, in particular the IP address, the protocol port and the credentials to generate the token, which is a temporary string that must be specified in the header section of the RESTCONF requests sent to the ROADM; without this header, the user cannot be authorized to send any request to the ROADM. When the OpenROADM agent is run, *init* function is called to set-up the environment and retrieve the information (4.1).

Make_connection function is used in order to connect two ports within a degree: this function executes all the steps described in Section 3.3.1. The most important parameters for the execution of the function are: the degree and the two ports of interest, the central frequency and the channel bandwidth. It is mandatory to respect the presence of one network port and one client port, specifying at first the receiving and then the transmitting port. Also the central frequency and the channel bandwidth must follow the rules reported in Section 3.3.1.

Delete_connection function aims to delete the already created connections inside the ROADMs: in this case, it is sufficient to specify the degree and the two ports.

When *make_connection* and *delete_connection* are executed, at first they generate the token string, if necessary: a check is performed to the previously created token and, if it has expired, a new one is created. Then, the parameters passed according to the OpenROADM nomenclature are mapped according to the proprietary language.

If called properly, all the functions produce the expected results: the *init* function retrieves the parameters to generate the token string passed to the header of the following requests to the ROADM; *make_connection* and *delete_connection* results can be verified by observing the power spectrum, since the presence of the created channels, with the specified characteristics, and consequently their elimination, allows the check of the result's correctness.

On the contrary, if the specified parameters are wrong, the operating system is able to recognize and communicate the occurred error. In particular, the system is able to detect the following errors:

- the specified frequency does not fall in the C-band;
- some channels are overlapping in frequency within the same port;
- the two provided ports to connect through media layer are not within the same degree;
- the user wants to connect two same port type through the media layer;
- within the unidirectional connection, the TX slot is specified before the RX one.

4.1.4 ONOS configuration

ONOS is able to establish a connection towards all the devices within the controlled network, thanks to the supported standard models. In this context, each ROADM is reachable since the OpenROADM agent is used as an additional support for the ROADM device. As a consequence, ONOS identifies each ROADM as an OpenROADM model, without considering the real entity accessed through the middleware.

The first action performed by ONOS is the initial NETCONF handshake with the agent: in this step, the hardware features of the ROADM are retrieved. When the initial setup is completed, ONOS is able to manage the connections within the network; a user can directly handle the lightpaths using ONOS, or a software can be associated to the network controller to manage the connections that must be created. To execute a function, a command must be sent from the ONOS console according to the standard model, specifying the parameters that must be passed to the function.

ONOS is not able to check if the wavelength continuity can be guaranteed, as explained in Section 2.20, but it will execute the functions where it is possible to allocate the necessary resources. For this reason, the user or a software must keep track of the already used frequencies to ensure the wavelength continuity from the source to the destination, according to the RWA rules.

4.2 Theoretical Example

In this section an example will be presented to show how *make_connection* works: this function can be called after the initialization of the environment and the generation of the token.

An incoming signal arrives from the optical line at a central frequency of 193 THz, characterized by a channel width of 50 GHz: it arrives at the receiving network port of the first degree and must be forwarded to the transmitting network port of the second degree; the involved client ports are `cl,1` in both degrees.

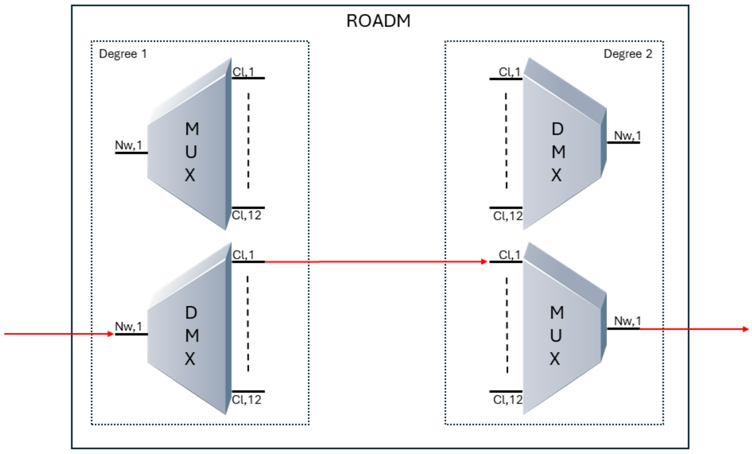


Figure 4.8: Representation of the example: connection between two degrees

Figure 4.8 offers a graphical representation of the scenario. In this example, *make_connection* must be called two times: at first, a cross-connection within the first degree must be created, then the same operation must be repeated at the second degree. Between the two client ports, a fiber must be properly inserted to allow the signal to follow the optical path.

Cross-Connection within Degree 1

The parameters to specify in this call are:

- degree: DEG1-WSS;
- RX port: DEG1-WSS-RX;
- TX port: DEG1-WSS-OUT1;
- central frequency = 193000000 MHz;
- channel bandwidth: 50000 MHz.

The result of the first call of *make_connection* is shown in Figure 4.9: here the functions acts inside the first degree, creating the necessary structures to connect the receiving network port and the transmitting client port 1. It is visible the created optical path that the signal must follow.

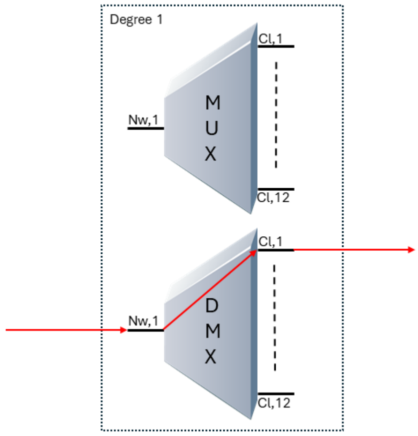


Figure 4.9: Cross-connection within Degree 1

Cross-Connection within Degree 2

The parameters to specify in this call are:

- degree: DEG2-WSS;
- RX port: DEG2-WSS-IN1;
- TX port: DEG2-WSS-TX;
- central frequency = 193000000 MHz;
- channel bandwidth: 50000 MHz.

The result of the second call of *make_connection* is shown in Figure 4.10: it is analog to what happens within the first degree. The signal now is ready to be forwarded towards the optical line: inside the ROADM that closes the line, the same function (*make_connection*) must called to direct the signal.

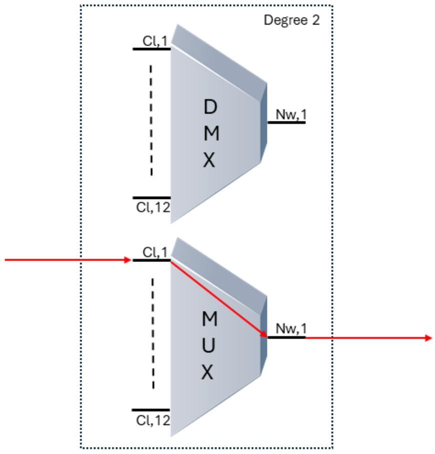


Figure 4.10: Cross-connection within Degree 1

Chapter 5

Results

The first results to analyze are related to the power evolution at each port of the ROADM, performing some tests and retrieving the power values from the integrated OCM: this must be the first step to understand if the device is working correctly. In this way, it is also possible retrieve the general characteristics of the device, including the introduced penalties and eventual extra attenuation. In the following, several plots related to two different scenarios in which there are tested the properties and functions of the ROADM are shown.

5.1 Scenario 1

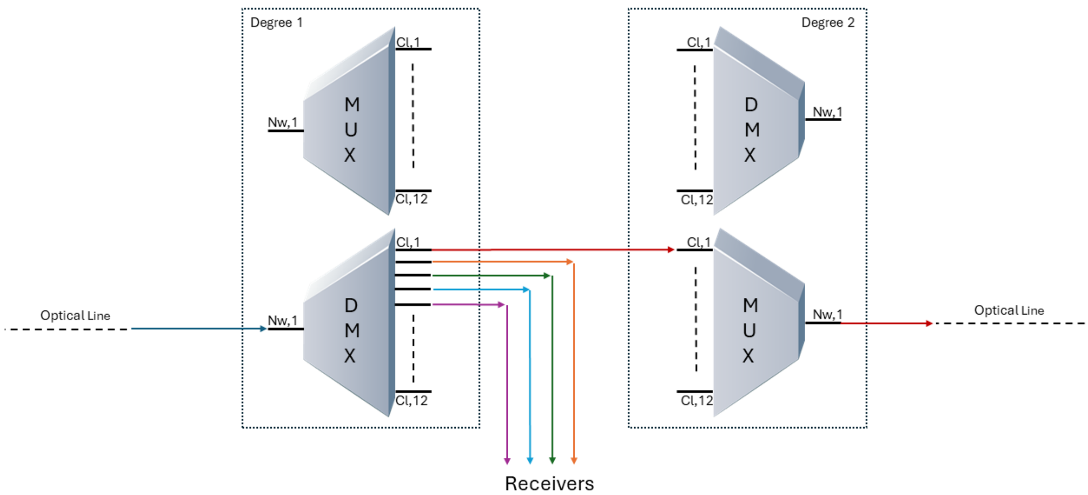
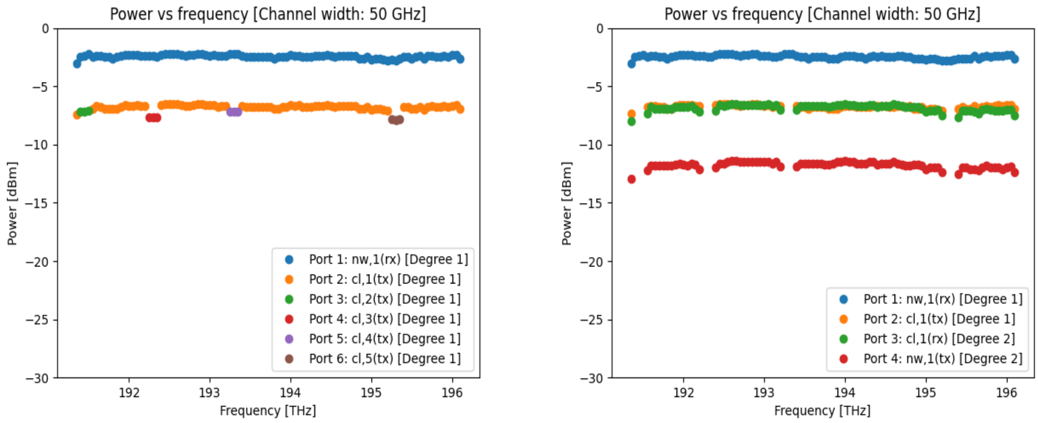


Figure 5.1: This scheme represents a simplified environment of the first scenario: each arrow represents an ensemble of channels forwarded to separated receivers.

Figure 5.1 is helpful to understand what happens inside the ROADM: a signal arrives at the network port of the degree 1, where it is divided in 96 channels, and is forwarded to the transmitting client ports from 1 to 5. The first client port forwards a huge portion (84 channels) of the incoming signal to the second degree, while the others are dropping ports, so they theoretically send channels that they receive (3 channels per port) to the eventual transceivers. Once the incoming signal is forwarded to the network transmitting port of the second degree, it is ready to be forwarded towards the optical line.

5.1.1 No Equalization



(a) Drop on degree 1 **(b)** Complete connection

Figure 5.2: Drop and forwarding of scenario 1

Figure 5.2a shows what happens inside the degree 1 of the ROADM: a full spectrum signal arrives at the receiving network port and is split in 96 different channels of 50 GHz each. At this point each channel is forwarded towards the successive transmitting client ports of the same degree; in particular, some channels are destined to eventual transceivers, and they are the ones that arrive at the ports `cl,2`, `cl,3`, `cl,4` and `cl,5`. On the other hand, the remaining channels are forwarded to the `cl,1` port, and then forwarded to the following degree, which is the degree 2, as it is possible to observe in Figure 5.2b; the holes in the figure are the consequence of the fact that some channels have been dropped in other ports, while the plot shows what is happening at the `cl,1` port.

In this case, the equalization has been disabled, so the power loss that can be observed in both plots is introduced by the MUX/DMX crossed by the signal: in particular, as it is possible to notice, the loss is about 5 dBm, but this is an average.

The real loss depends on the specific port and on the specific degree. This aspect is observable in Figure 5.2a: even if the loss is similar, it is possible to notice that the dots that characterize the plot are not all at the same level. This behaviour is justified by the fact that, as mentioned before, the introduced loss is different between each port.

Within Figure 5.2b, it is possible to notice that the penalty of about 5 dBm is present only between the network and the client ports. On the other hand, the power level remains almost constant when the signal passes from the client port of the first degree to the client port of the degree 2: to connect these two ports it is sufficient to use a really short fiber, so no significant attenuation is introduced, and the observed power level is practically the same at both sides.

Introduced Losses

The following list collects the losses introduced within the MUX/DMX of the degree 1: the calculated loss is about 5 dB as expected, but it is not constant.

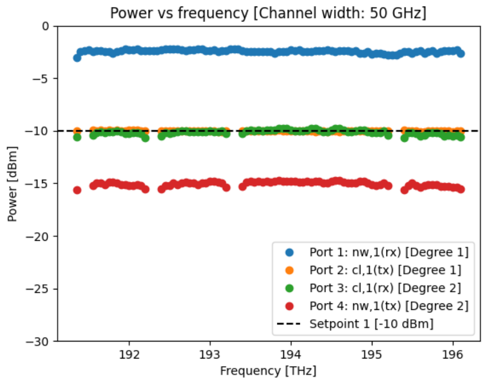
- cl,1 : 4.3 dBm;
- cl,2 : 4.8 dBm;
- cl,3 : 5.1 dBm;
- cl,4 : 5.0 dBm;
- cl,5 : 5.0 dBm;

No losses are experienced between the degrees 1 and 2, except for the negligible effect of the fiber attenuation.

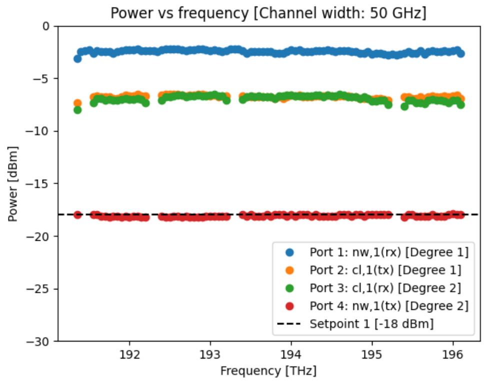
In degree 2, the measured penalty between the client port 1 and the network port is 4.9 dB.

5.1.2 The Effect of the Equalization

The equalization is an important feature that can be exploited for the management of the channels: it is able to influence the power of the channels after they cross a MUX/DMX. The user can impose a threshold: if the channel PSD is higher in respect to the threshold, it will be scaled at the level of the threshold, otherwise it remains the same. The equalization can be useful in order to have all the channels at the same power. In the following, there are shown some plots in which the effect of the equalization on the power of the channels is represented.



(a) cl,1 - degree 1



(b) nw,1 - degree 2

Figure 5.3: Equalization on one side

Figures 5.3a and 5.3b show the effects of the equalization when enabled in only one side: in the first figure, the equalization has been enabled only on the client port of the first degree, while in the second plot the equalization has been enabled on the network port of the degree 2; an horizontal line has been added in both cases in order to show at which target the channel power must be adapted when the equalization is enabled.

From an analytical point of view, the calculation that must be performed in order to predict the power of each channel is the following: $10 \cdot \log_{10}(B_{ch}) - \text{setpoint}$; where B_{ch} is the width of the channel, in this case 50 GHz, and the setpoint is the target PSD, expressed in dBm/GHz, and its value can be chosen in a specific interval, mentioned in Section 3.3.2.

In Figure 5.3a, the chosen setpoint is -24 dBm/GHz, so it has been scaled to -10 dBm, since the channel width is 50 GHz: as it is possible to observe, the power of each channel at the client port is coincident to the threshold. On the other hand, Figure 5.3b represents the case in which the equalization is enabled only on the network port: the selected setpoint is equal to -35 dBm/GHz, which has been subsequently scaled to -18 dBm. As a consequence, when the equalization is enabled, the output power is not only reduced by the internal loss of the MUX/DMX, but is also influenced by an extra attenuation that depends on the selected setpoint.

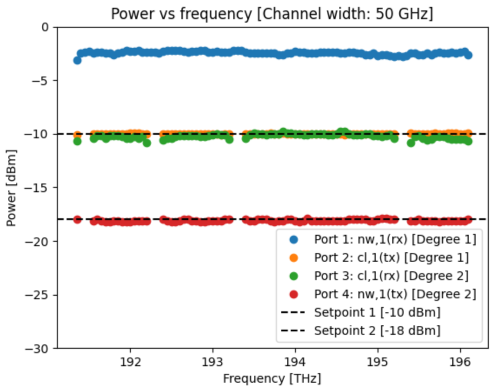


Figure 5.4: complete case

Figure 5.4 represents the combination of Figures 5.3a and 5.3b, with the equalization enabled in both sides.

In the previous cases there have been shown conditions in which the extra attenuation could be added to the output power, since for each channel it was always higher in respect to the threshold, but there can be situations in which the output power is lower; this situation is tested in the following.

In order to test what happens when the power is lower in respect to the scaled setpoint, the power of each channel has been manipulated; this operation has been performed with the help of the parameter setpoint delta, mentioned in Section 3.3.2. The power at the output of the client port of the first degree has been modified in order to have a ramp, with some channel's power above the threshold while the others below.

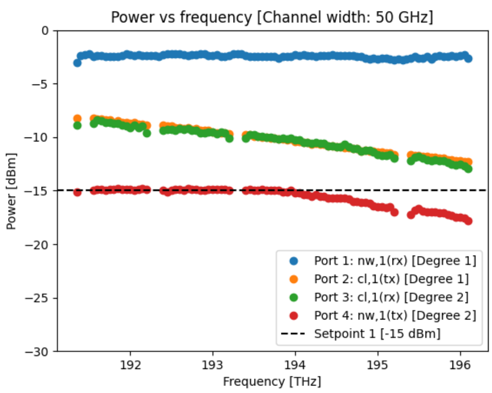
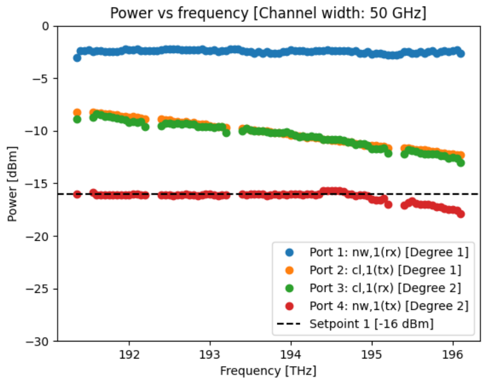
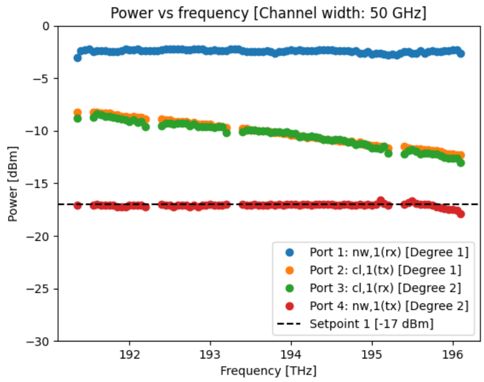


Figure 5.5: Setpoint $= -15$dBm



(a) Setpoint = −16dBm



(b) Setpoint = −17dBm

Figure 5.6: Setpoint increment

Figures 5.5, 5.6a and 5.6b show the same scenario, but a different setpoint has been selected on the network port; the oblique signal has been obtained using the parameter delta, which could be set when the equalization is enabled: the received signal at the network port of the degree 1 has been reduced by 5 dB at the port `cl,1`, always due to the internal loss, and then an increasing extra attenuation has been applied at each channel thanks to the parameter delta. As a consequence, the resulting signal has an oblique shape: by observing the three pictures, it is possible to notice that the first channels from the left side have an integral power value around -7 dBm, while the last channels have a value around -13 dBm. Finally, at the network port on degree 2, there must be applied again 5 dB of penalty, so the channels have an integral power value from about -12 dBm to about -18 dBm. Now it is possible to analyse what happens at the variation of the setpoint.

As it is possible to observe, the increment of the absolute value of the setpoint is translated to an higher number of channels that can be equalized: in fact, in Figure 5.5 only the first half of the channels has been attenuated to the setpoint value, due to the fact that only the first half of the channels has a power value higher in respect to the threshold. By increasing the setpoint, as in Figures 5.6a and 5.6b, the number of channels that are higher than the threshold increases: as it is possible to notice, only a small portion on the right side has not been attenuated.

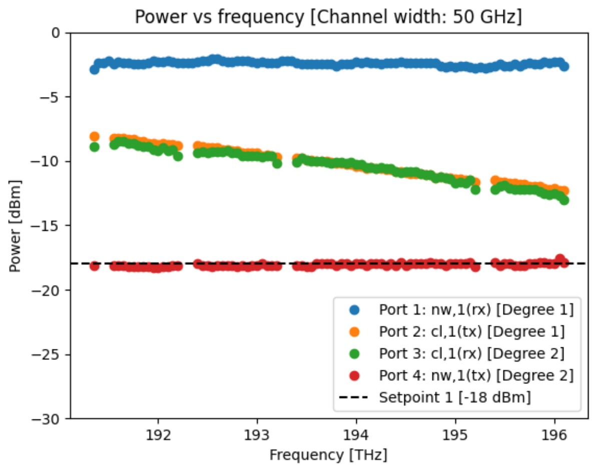


Figure 5.7: Setpoint = −18dBm

When the setpoint is set to a value that is sufficiently low, or high if the absolute value is considered, all the channels can be attenuated: this is the case represented in Figure 5.7, where all the channels have the same power value.

5.2 Scenario 2

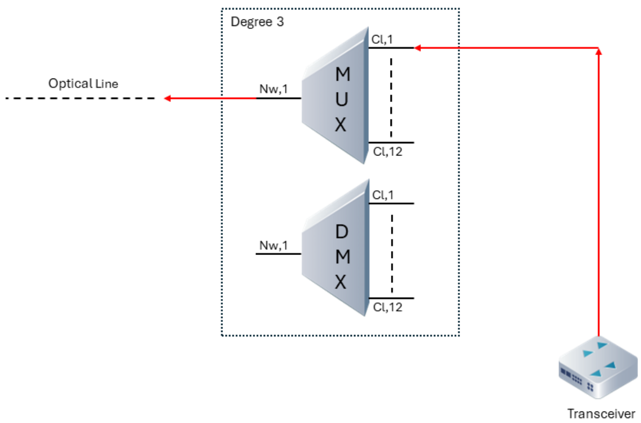
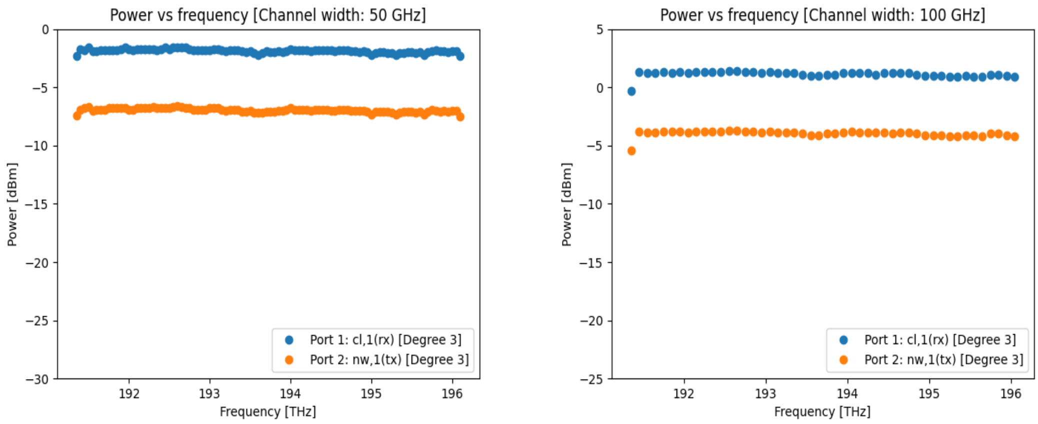


Figure 5.8: This scheme represents a simplified environment of the second scenario: each arrow represents the entire signal forwarded to the optical line.

Figure 5.8 shows the environment tested in the second scenario: here the signal arrives to the `cl,1` port of the third degree and is forwarded to the transmitting network port of the same degree. In this case it has been simulated a scenario where the adding function is used.

As it is possible to observe from Figure 5.9, also in this case a 5 dB loss has been applied at each created channel, but no extra attenuation has been added, since in this case the equalization was kept disabled.



(a) **(b)**

Figure 5.9: Scenario 2

Figure 5.9 also compares the same scenario with the variation of the channel width value: Figure 5.9a represents the incoming signal divided in 96 channels characterized by a bandwidth of 50 GHz, while Figure 5.9b shows the same signal, but now divided in 48 channels that have a bandwidth of 100 GHz.

By observing the two plots, it is possible to notice that the increase of the bandwidth can be translated in an increase of the power value per channel, and vice versa; in particular, doubling the channel width, in this case from 50 GHz to 100 GHz, each power channel has an increment of 3 dBm. This behaviour can be justified by the fact that doubling the channel width, it is also doubled the power within the channel, so the OCM will measure an higher power level. Any eventual extra attenuation or the loss remain the same, independently from the channel width.

As mentioned in Section 3.3.1, the smallest channel width is 37,5 GHz: in that case, more channels can be created and the resolution of the power monitor will be the highest one. On the other hand, using the largest channel width, so the entire bandwidth, it is possible to observe only one point: as a consequence, the creation of larger channel will be translated to a lower number of channels, but they will be characterized by higher power values, due to the fact that they collect a larger portion of the spectrum.

The following sections wrap the results obtained with the interaction between ONOS and the ROADMs.

5.3 Interaction with ONOS

To start the communication between ONOS and Adtran, the OpenROADM agent is run and the initial setup is performed to retrieve the information of the controlled ROADM: all the entities (circuit-packs) are listed during the initial handshake.

```
netopeer-server[1]: *************************************
netopeer-server[1]: *** Initializing OpenROADM device ***
netopeer-server[1]: *************************************
netopeer-server[1]: Transapi calling callback /A:org-openroadm-device/A
netopeer-server[1]: *** ROADM Device ***
```

Figure 5.10: Setup initialization: when the OpenROADM agent is run, all the ROADM features are collected

Init is automatically called for each circuit-pack entity contained within *circuit_packs.xml* to retrieve the relative information.

At this point it is possible to test *make_connection*: ONOS is used to create a connection between two degrees of a ROADM. To fulfil the request sent by the controller, *make_connection* is executed two times: at first within the receiving degree (in this case degree 2), while the second call acts within the transmitting degree (degree 3). Figure 5.12 shows the details and the parameters of the execution: the created channels are characterized by a specific spectrum portion and are instantiated within the degrees 2 and 3 of the selected ROADM. It is also observable the choice of the client ports: within the datastore *config.xml*, the instructions to connect two specific degrees are reported (Fig. 5.11).

```
<physical-link>
  <physical-link-name>ExpLink23</physical-link-name>
  <source>
    <circuit-pack-name>DEG2-WSS</circuit-pack-name>
    <port-name>DEG2-WSS-OUT12</port-name>
  </source>
  <destination>
    <circuit-pack-name>DEG3-WSS</circuit-pack-name>
    <port-name>DEG3-WSS-IN12</port-name>
  </destination>
</physical-link>
```

Figure 5.11: Instructions to instantiate an express connection between degrees 2 and 3 of the selected ROADM

In this test, the involved ports to connect degrees 2 and 3 are DEG2-WSS-OUT12 (transmitting client port 12 of degree 2) and DEG3-WSS-IN12 (receiving client port 12 of degree 3).

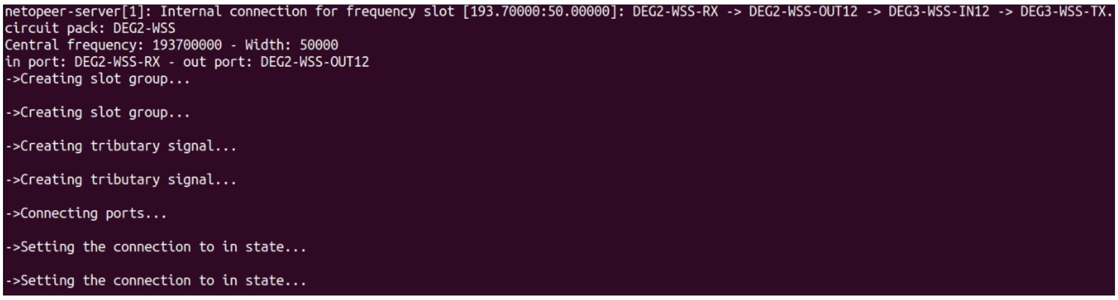


Figure 5.12: *Make_connection* executed within the second degree

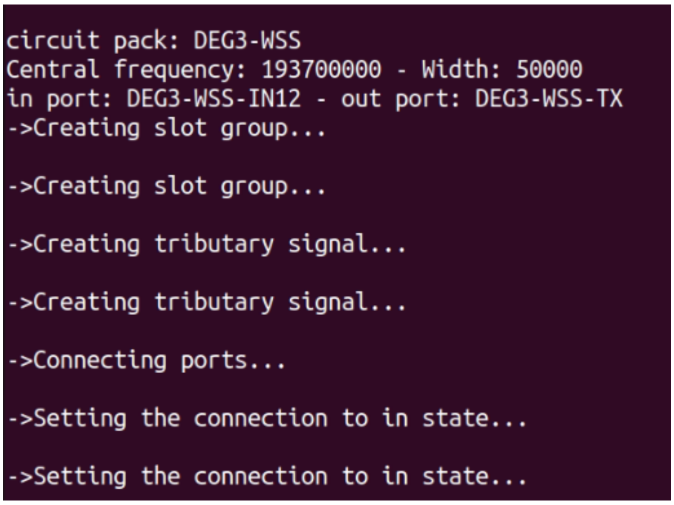


Figure 5.13: *Make_connection* executed within the third degree

Figures 5.14, 5.15, 5.16 and 5.17 show useful screenshots of the information retrieved using the CLI: the necessary resources are correctly allocated. Each channel is characterized by a central frequency of 193.7 GHz and 50 Ghz as a channel bandwidth: these values are chosen at the launch of *make_connection*.

```
admin@FSP3000C> show interface 1/3/n/oms/spslg-1/otsia otsia otsi 1

otsi 1:
  id: 1
  center-frequency: 193.700000 THz
  bandwidth: 50.000 GHz
```

Figure 5.14: Channel created within the network port of the second degree: it is characterized by the central frequency and the channel width selected by ONOS.

```
admin@FSP3000C> show interface 1/3/c12/oms/spslg-1/otsia otsia otsi 1

otsi 1:
  id: 1
  center-frequency: 193.700000 THz
  bandwidth: 50.000 GHz
```

Figure 5.15: Channel created within the network port of the second degree: it is characterized by the central frequency and the channel width selected by ONOS.

```
admin@FSP3000C> show interface 1/5/n/oms/spslg-2/otsia otsia otsi 1

otsi 1:
  id: 1
  center-frequency: 193.700000 THz
  bandwidth: 50.000 GHz
```

Figure 5.16: Channel created within the network port of the third degree: it is characterized by the central frequency and the channel width selected by ONOS.

```
admin@FSP3000C> show interface 1/5/c12/oms/spslg-2/otsia otsia otsi 1

otsi 1:
  id: 1
  center-frequency: 193.700000 THz
  bandwidth: 50.000 GHz
```

Figure 5.17: Channel created within the client port 12 of the third degree: it is characterized by the central frequency and the channel width selected by ONOS.

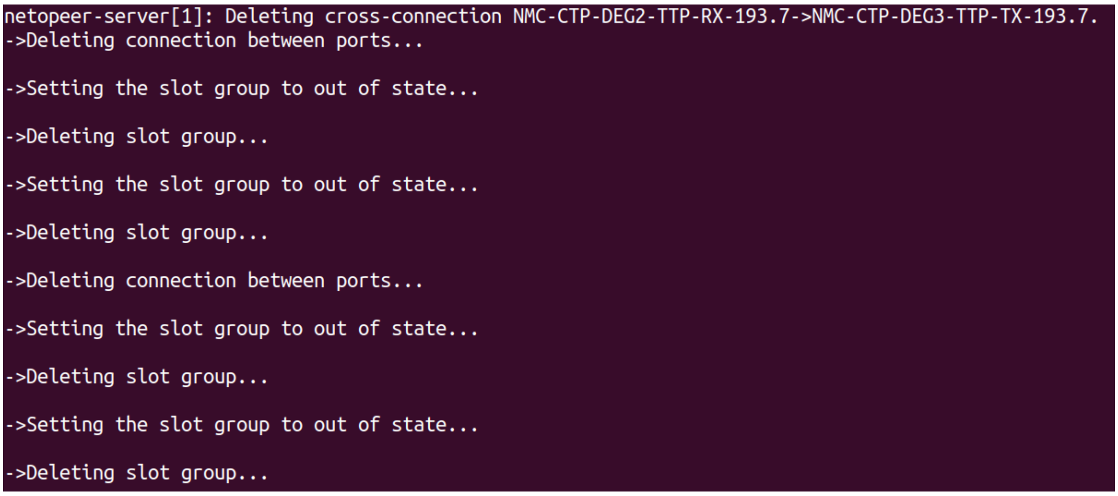


Figure 5.18: *Delete_connection* test

Figure 5.18 shows the *delete_connection* in action: from the optical network controller (ONOS), it is ordered to eliminate an already created connection, which is the one created previously.

The first visible line shows details of the called function. In particular, the entity NMC-CTP-DEG2-TTP-RX-193.7->NMC-CTP-DEG3-TTP-TX-193.7 refers to the created connection between degrees 2 and 3 of the involved ROADM:

- NMC-CTP-DEG2-TTP-RX: second degree. In this case it is the one that receives a signal;
- NMC-CTP-DEG3-TTP-TX: third degree. In this case it is the one that transmits a signal;
- 193.7: the central frequency that characterizes the connection between the two entities.

To delete a connection, at first the operating system must act on the media port: it has to delete the connection between the spectrum slot groups of the two ports within each degree. Then, to delete the spectrum slot groups, it is mandatory to set the entity to eliminate in *out of state*. Finally, the SPSLG can be deleted, so the free resources can be reused for the creation of new connections. These steps are repeated at both degrees (at first at the receiving degree, then at the transmitting degree).

5.4 Optical Network Control

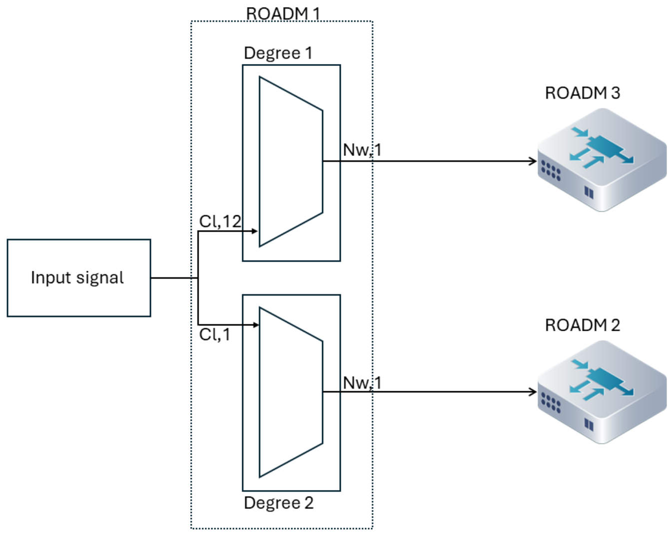


Figure 5.19: A full band input signal is split between the first two degrees of ROADM 1: each network port propagates the filtered channels.

To simulate a controlling action within the network, *make_connection* is now sent several times: 6 channels have been created within ROADM 1, Fig. 4.2, and are propagated towards ROADM 3. ONOS calls *make_connection* 6 times: at each execution one channel is created; the channels are characterized by the same width (62.5 GHz), but with an increasing central frequency, in order to avoid the channel overlapping. These channels have been propagated through the optical line, composed by EDFA amplifiers, as described in Section 4.1, up to ROADM 3, where 64 equally spaced channels have been created (62.5 GHz of channel width). The 64 channels have been created in order to visualize the entire C-band spectrum with the help of the OCM.

Figure 5.20: This scheme represents where data have been collected.

The following figures represent the power curve in respect to the frequency, retrieved as shown in Fig. 5.20.

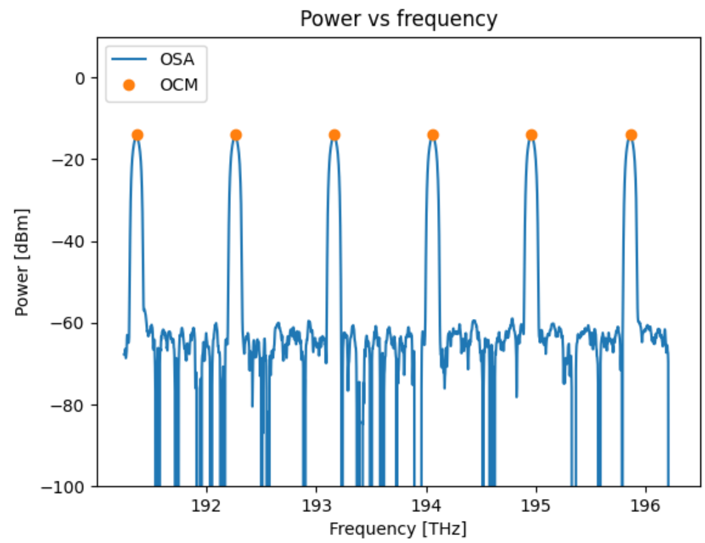


Figure 5.21: Plot retrieved at transmitting `nw,1` of ROADM 1

Figure 5.21 represents the combination of the plots retrieved with the help of the integrated OCM compared to the power spectrum obtained with the OSA connected at the network port of ROADM 1. As it is possible to observe, the dots coincide with the peaks of the channels: this aspect affirm that the OCM is a good instrument to evaluate if the channels are created and the power values are at the correct level. The presence of the channels indicates the success of the connection creation: as said previously, *make_connection* has been called 6 times, and at each call, one connection has been set up and it is possible to observe the correctness of the function's behaviour, since 6 channels are visible. Between the channels, the noise of the OSA is observable: in these regions nothing can be measured, so the OSA represents this condition with low power lines. Each channel's peak is at the same level, near -14.0 dBm: the equalization is enabled at the network port, so an extra attenuation has been added in order to reach the setpoint value.

Figure 5.22: Power of the client port 12 within the first ROADM.

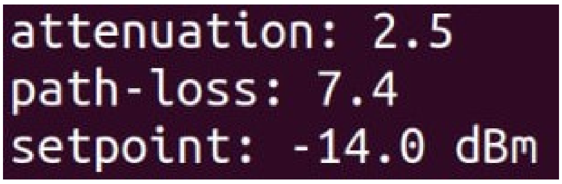


Figure 5.23: Useful values at the network port within the first ROADM.

Figures 5.22 and 5.23 show how the CLI represents values connected to the power of the channels: the former reports directly the power of the first channel, with the lowest central frequency, created at the client port 12, the latter represents the total path loss, that includes the extra attenuation, measured at the network port connected to `cl,12`; if the path loss is subtracted to the power measured at the client port, it is obtained the peak of the first channel's power. This value must be near the chosen setpoint.

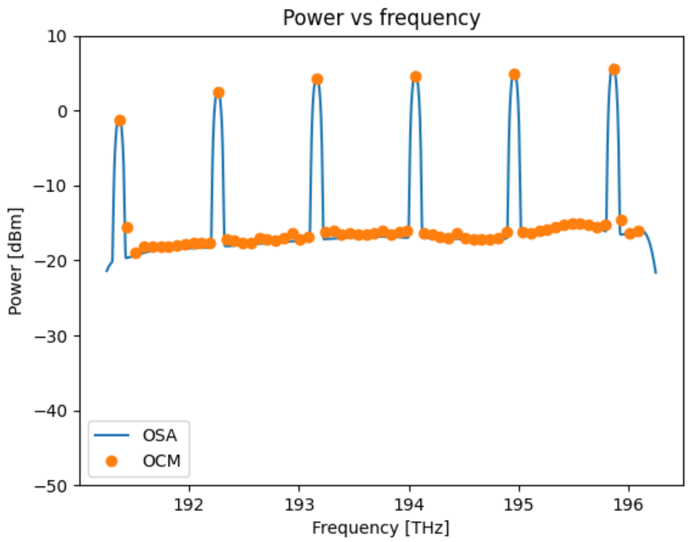


Figure 5.24: Plot retrieved at receiving `nw,1` of ROADM 3

Figure 5.24 shows the power at the end of the optical line, inside the receiving network port of ROADM 3. The power values are higher in respect to the transmitting side: here the power has been measured after the last Juniper EDFA amplifier. The peaks present different values and are not equalized: this behaviour is a combined effect of the gain, that depends on the imposed parameters, and fiber loss. In particular, due to the settings of the EDFA, the gain increases with the frequency [38], while the attenuation is almost flat in the C-band, except for the edges, where its value is a little bit higher. These two effects are well visible:

increasing the frequency, the channels present higher peaks; the first channel is affected by lower gain and higher attenuation, so it is the most penalized. The retrieved total received power is 12.5 dBm.

The same six channels are also created on the degree 2 of ROADM 1, Figure 5.19, and routed towards ROADM 2 within another optical line, where CISCO EDFA amplifiers are used to amplify the signal.

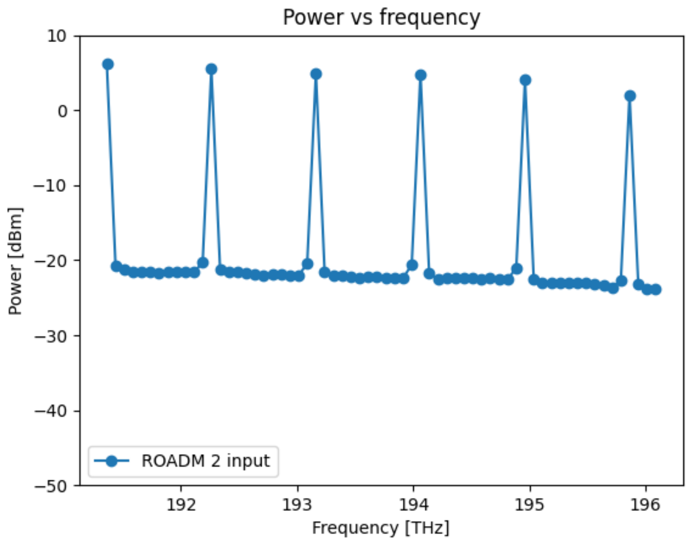


Figure 5.25: Plot retrieved at receiving `nw,1` of ROADM 2

Figure 5.25 shows the six channels at the end of the optical line, within the receiving network port of ROADM 2: also in this case, 64 channels characterized by a width of 62.5 GHz and equally spaced are created to collect the signal arriving from the optical line. In this context, due to the EDFA settings, the highest channels are the ones characterized by a lower frequency, and go from 6 dBm to 2 dBm. Considering the channel peaks at the receiving network port, they are similar and mirrored to the ones retrieved at the input of ROADM 3 (Figure 5.24).

In general, the gain (G) is set to fully compensate the fiber loss (L), i.e., $G = L$. If the gain is set to a different value, the retrieved optical power will be higher for $G > L$, and lower for $G < L$.

In Figure 5.21, it is shown the case in which the transparency ($G = L$) is not respected, but the gain is higher than the fiber loss.

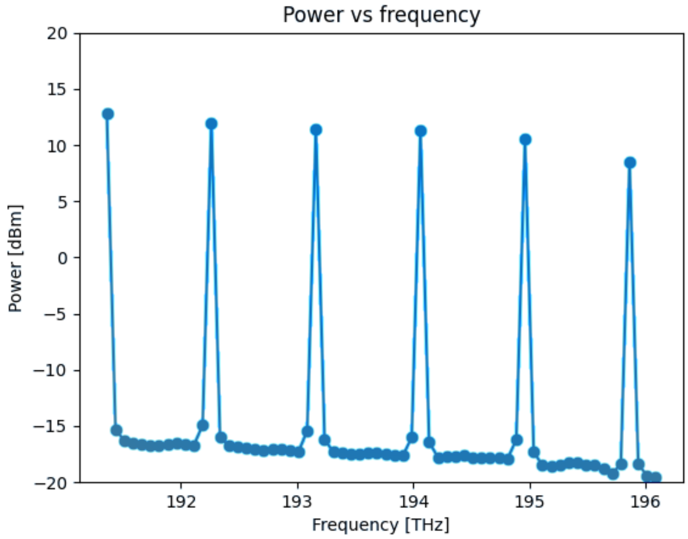


Figure 5.26: Plot retrieved at receiving `nw,1` of ROADM 2 (no transparency)

The channels received at the input of ROADM 2, Figure 5.25, are then rotated towards the output of the same device to reach ROADM 3 using another optical line.

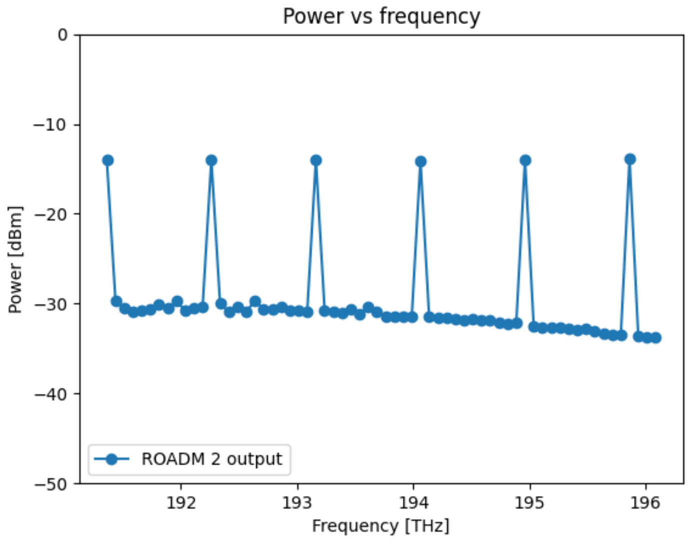


Figure 5.27: Plot retrieved at transmitting `nw,1` of ROADM 2

Figure 5.27 shows the power spectrum of the ROADM 2 output: the two equalization steps, performed within the receiving degree (`cl,1` TX) and at the output of ROADM 2 (`nw,1` TX), allow the channels to obtain the same power measured at the output of ROADM 1, Figure 5.21, where all the peaks are around -14 dBm.

The optical line that connects ROADM 2 and ROADM 3 is characterized by Juniper EDFA ampifiers, as the line between ROADM 1 and ROADM 2. At the end of the optical line, i.e., at the receiving network port of ROADM 3, the six created channels are clearly visible and reported in Figure 5.28. In this case, the EDFA settings influence the optical signal mostly penalizing the highest frequency channels, as happens at the input of ROADM 2 (Figure 5.25).

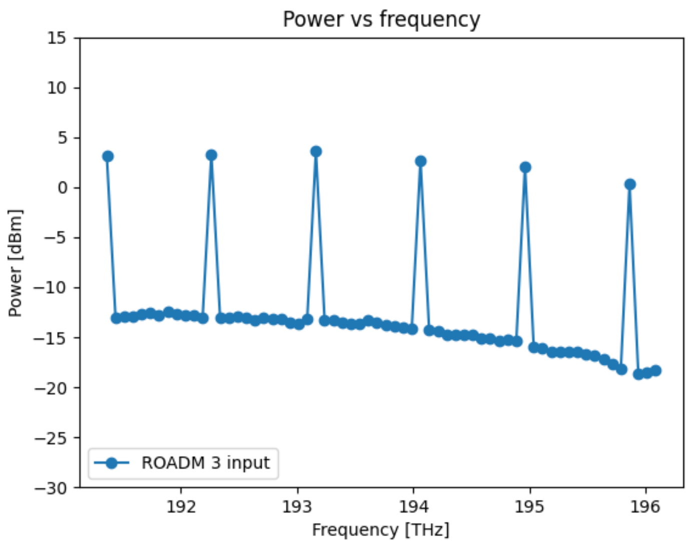


Figure 5.28: Plot retrieved at receiving `nw,1` of ROADM 3

At the input of ROADM 2, the total received optical power is 12.7 dBm, while the total power at the input of ROADM 3 is 11.3 dBm, although the launched power per channel is the same. To obtain the same input power at each ROADM of the selected path, the gain of the last ROADM's pre-amplifier has been increased to perfectly match the 12.7 dBm measured at the input of ROADM 2.

Figure 5.29 shows the input of ROADM 3: each channel presents a slightly increment of the peak power value, due to the action performed at the pre-amplifier settings.

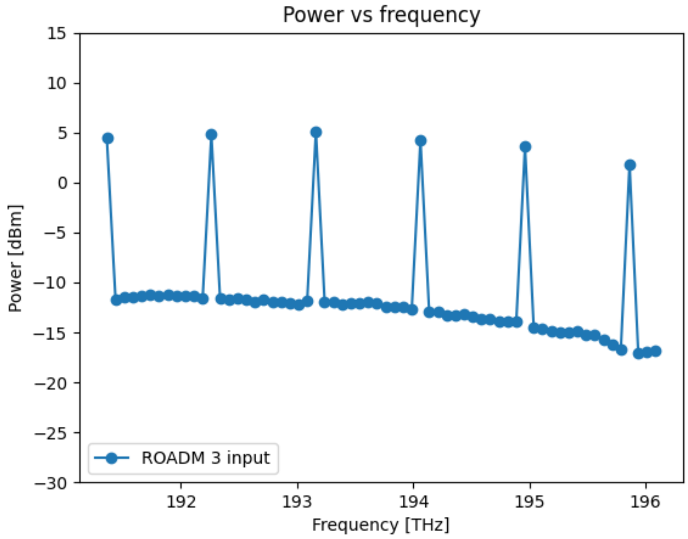


Figure 5.29: Plot retrieved at receiving `nw,1` of ROADM 3 acting on the preamp

Chapter 6

Conclusions and Comments

Sections 5.3 and 5.4 demonstrate the feasibility of establishing a connection between a controller and a device produced by Adtran. This result is significant as it supports the possibility to develop multi-vendor networks. In this scenario, the device successfully executes commands delivered by the controller, facilitated by the open interface, the OpenROADM agent, that contains the necessary instruments capable of translating from a standard to a proprietary model.

The development of a multi-vendor network necessitates a comprehensive understanding of various device models, that comprehends both hardware and software perspectives. To create code that effectively manages the device under examination, it is essential to study the supported protocols and determine the commands, written in the proper syntax, that the device must execute. Knowledge of the hardware is also crucial, as it provides insight into the level at which the developed software operates. For instance, understanding the physical structure of the Adtran ROADMs used in the setup is necessary for configuring commands appropriately. Additionally, familiarity with the selected controller and the chosen standard is fundamental.

Thus, the establishment of a multi-vendor network is achieved with the development of a standard able to communicate with all the different vendor devices and the accurate studying of the specific devices within the network. In this context, the development of code for translating OpenROADM commands to the proprietary model necessitates an in-depth understanding of managing the Adtran ROADM model, particularly regarding the essential commands required to fulfill OpenROADM requests delivered by the ONOS controller.

The introduction of an additional layer may costs in terms of latency and needs additional resources to operate with a controller, but it is a simple way to avoid the redefinition of the controller's behaviour, based in this context on the delivery of OpenROADM requests. Without introducing this additional layer, the controller must adapt its communication depending on all the devices, increasing the complexity of the system. With the help of an interface, the controller is programmed to support only the standard models, and each device will expose an ad hoc interface to communicate with the controller of the network.

This activity can be helpful to extend the introduction of open interfaces towards all the devices, making feasible a multi-vendor context, which is one of the main targets of the open optical network world.

At this point these network devices can be controlled to propagate optical signals towards the optical lines within the network: thanks to this activity, it is possible to characterize the behaviour of the network. In particular, it is possible to find the best paths from a source to a destination, evaluating different terms. It is also possible to find the best parameters to apply at the devices within the optical line to obtain the best network performances.

Bibliography

[1] Wenfeng Xia, Yonggang Wen, Chuan Heng Foh, Dusit Niyato, and Haiyong Xie. «A survey on software-defined networking». In: *IEEE Communications Surveys & Tutorials* 17.1 (2014), pp. 27–51 (cit. on p. 3).

[2] Lawrence G Roberts. «Beyond Moore's law: Internet growth trends». In: *Computer* 33.1 (2000), pp. 117–119 (cit. on p. 3).

[3] Luis Velasco, Andrea Sgambelluri, Ramon Casellas, Lluis Gifre, Jose-Luis Izquierdo-Zaragoza, Francesco Fresi, Francesco Paolucci, Ricardo Martínez, and Emilio Riccardi. «Building autonomic optical whitebox-based networks». In: *Journal of Lightwave Technology* 36.15 (2018), pp. 3097–3104 (cit. on p. 4).

[4] Mark Filer, Mattia Cantono, Alessio Ferrari, Gert Grammel, Gabriele Galimberti, and Vittorio Curri. «Multi-vendor experimental validation of an open source QoT estimator for optical networks». In: *Journal of Lightwave Technology* 36.15 (2018), pp. 3073–3082 (cit. on p. 5).

[5] Gangxiang Shen and Rodney S Tucker. «Translucent optical networks: the way forward [topics in optical communications]». In: *IEEE Communications Magazine* 45.2 (2007), pp. 48–54 (cit. on p. 5).

[6] K Bala, RR Cordell, and EL Goldstein. «The case for opaque multiwavelength optical networks». In: *IEEE/LEOS 1995 Digest of the LEOS Summer Topical Meetings. Flat Panel Display Technology*. IEEE. 1995, pp. 58–59 (cit. on p. 5).

[7] Robert D Maurer. «Glass fibers for optical communications». In: *Proceedings of the IEEE* 61.4 (1973), pp. 452–462 (cit. on p. 6).

[8] Robert J Potter. «Transmission properties of optical fibers». In: *JOSA* 51.10 (1961), pp. 1079–1089 (cit. on p. 7).

[9] AW Naji et al. «Review of Erbium-doped fiber amplifier». In: *International Journal of the Physical Sciences* 6.20 (2011), pp. 4674–4689 (cit. on p. 7).

[10] Uri Ganiel, A Hardy, G Neumann, and D Treves. «Amplified spontaneous emission and signal amplification in dye-laser systems». In: *IEEE journal of quantum electronics* 11.11 (1975), pp. 881–892 (cit. on p. 8).

[11] Masahiko Jinno, Hidehiko Takara, Yoshiaki Sone, Kazushige Yonenaga, and Akira Hirano. «Multiflow optical transponder for efficient multilayer optical networking». In: *IEEE Communications Magazine* 50.5 (2012), pp. 56–65 (cit. on p. 9).

[12] Jonathan Homa and Krishna Bala. «ROADM architectures and their enabling WSS technology». In: *IEEE Communications Magazine* 46.7 (2008), pp. 150–154 (cit. on p. 9).

[13] Anas Chaaban, Zouheir Rezki, and Mohamed-Slim Alouini. «On the capacity of intensity-modulation direct-detection Gaussian optical wireless communication channels: A tutorial». In: *IEEE Communications Surveys & Tutorials* 24.1 (2021), pp. 455–491 (cit. on p. 10).

[14] Marvin K Simon and M-S Alouini. «A unified approach to the probability of error for noncoherent and differentially coherent modulations over generalized fading channels». In: *IEEE Transactions on Communications* 46.12 (1998), pp. 1625–1638 (cit. on p. 10).

[15] Hideki Ishio, Junichiro Minowa, and Kiyoshi Nosu. «Review and status of wavelength-division-multiplexing technology and its application». In: *Journal of lightwave technology* 2.4 (1984), pp. 448–463 (cit. on p. 12).

[16] Govind P Agrawal. *Fiber-optic communication systems*. John Wiley & Sons, 2012 (cit. on pp. 13, 14).

[17] Peter J Winzer. «High-spectral-efficiency optical modulation formats». In: *Journal of Lightwave Technology* 30.24 (2012), pp. 3824–3835 (cit. on p. 13).

[18] Michael Jarschel, Thomas Zinner, Tobias Hoßfeld, Phuoc Tran-Gia, and Wolfgang Kellerer. «Interfaces, attributes, and use cases: A compass for SDN». In: *IEEE Communications Magazine* 52.6 (2014), pp. 210–217 (cit. on p. 14).

[19] Nicola Andriolli et al. «Optical networks management and control: A review and recent challenges». In: *Optical Switching and Networking* 44 (2022), p. 100652 (cit. on p. 15).

[20] Imrich Chlamtac, András Faragó, and Tao Zhang. «Lightpath (wavelength) routing in large WDM networks». In: *IEEE Journal on selected areas in communications* 14.5 (1996), pp. 909–913 (cit. on p. 15).

[21] Josué Kuri, Nicolas Puech, Maurice Gagnaire, Emmanuel Dotaro, and Richard Douville. «Routing and wavelength assignment of scheduled lightpath demands». In: *IEEE Journal on Selected Areas in Communications* 21.8 (2003), pp. 1231–1240 (cit. on p. 15).

[22] Giacomo Borraccini et al. «Cognitive and autonomous QoT-driven optical line controller». In: *Journal of Optical Communications and Networking* 13.10 (2021), E23–E31 (cit. on p. 16).

[23] Andrea Sgambelluri, Alessio Giorgetti, Davide Scano, Filippo Cugini, and Francesco Paolucci. «OpenConfig and OpenROADM automation of operational modes in disaggregated optical networks». In: *IEEE Access* 8 (2020), pp. 190094–190107 (cit. on p. 17).

[24] Emilio Riccardi, Paul Gunning, Óscar González de Dios, Marco Quagliotti, Víctor López, and Andrew Lord. «An operator view on the introduction of white boxes into optical networks». In: *Journal of Lightwave Technology* 36.15 (2018), pp. 3062–3072 (cit. on p. 18).

[25] Yongcheng Li, Jieming Lin, Liangjia Zong, Sanjay K Bose, Biswanath Mukherjee, and Gangxiang Shen. «Colorless, partially directionless, and contentionless architecture for high-degree ROADMs». In: *Journal of Optical Communications and Networking* 14.6 (2022), pp. 481–492 (cit. on p. 19).

[26] June-Koo Rhee, Ioannis Tomkos, and Ming-Jun Li. «A broadcast-and-select OADM optical network with dedicated optical-channel protection». In: *Journal of lightwave technology* 21.1 (2003), p. 25 (cit. on p. 22).

[27] Andy Bierman, Martin Bjorklund, and Kent Watsen. *RESTCONF protocol*. Tech. rep. 2017 (cit. on p. 24).

[28] Rob Enns. *NETCONF configuration protocol*. Tech. rep. 2006 (cit. on p. 24).

[29] Bijoy Chand Chatterjee, Nityananda Sarma, and Eiji Oki. «Routing and spectrum allocation in elastic optical networks: A tutorial». In: *IEEE Communications Surveys & Tutorials* 17.3 (2015), pp. 1776–1800 (cit. on p. 25).

[30] Nicola Sambo, Alessio Ferrari, Antonio Napoli, Nelson Costa, João Pedro, Bernd Sommerkorn-Krombholz, Piero Castoldi, and Vittorio Curri. «Provisioning in multi-band optical networks». In: *Journal of Lightwave Technology* 38.9 (2020), pp. 2598–2605 (cit. on p. 26).

[31] Valery I Tolstikhin, Adam Densmore, Kirill Pimenov, Yury Logvin, Fang Wu, Sylvain Laframboise, and Serge Grabtchak. «Monolithically integrated optical channel monitor for DWDM transmission systems». In: *Journal of lightwave technology* 22.1 (2004), p. 146 (cit. on p. 29).

[32] Douglas M Baney, Bogdan Szafraniec, and Ali Motamedi. «Coherent optical spectrum analyzer». In: *Ieee Photonics Technology Letters* 14.3 (2002), pp. 355–357 (cit. on p. 29).

[33] Anny Martinez, Marcelo Yannuzzi, Víctor López, Diego López, Wilson Ramírez, René Serral-Gracià, Xavi Masip-Bruin, Maciej Maciejewski, and Jörn Altmann. «Network management challenges and trends in multi-layer and multi-vendor settings for carrier-grade networks». In: *IEEE Communications Surveys & Tutorials* 16.4 (2014), pp. 2207–2230 (cit. on p. 30).

[34] Alessio Giorgetti, Andrea Sgambelluri, Ramon Casellas, Roberto Morro, Andrea Campanella, and Piero Castoldi. «Control of open and disaggregated transport networks using the Open Network Operating System (ONOS)». In: *Journal of Optical Communications and Networking* 12.2 (2020), A171–A181 (cit. on p. 32).

[35] *http://www.openroadm.org* (cit. on p. 32).

[36] Arif Ahmed and Guillaume Pierre. «Docker container deployment in fog computing infrastructures». In: *2018 IEEE International Conference on Edge Computing (EDGE)*. IEEE. 2018, pp. 1–8 (cit. on p. 32).

[37] Ramon Casellas, Ricardo Martínez, Ricard Vilalta, and Raül Muñoz. «Abstraction and control of multi-domain disaggregated optical networks with OpenROADM device models». In: *Journal of Lightwave Technology* 38.9 (2020), pp. 2606–2615 (cit. on p. 35).

[38] Ankush Mahajan, Konstantinos Christodoulopoulos, Ricardo Martínez, Salvatore Spadaro, and Raul Muñoz. «Modeling EDFA gain ripple and filter penalties with machine learning for accurate QoT estimation». In: *Journal of Lightwave Technology* 38.9 (2020), pp. 2616–2629 (cit. on p. 57).