

Title	Packetponder: a multi-domain, software-defined, packet-optical convergence node at the edge
Authors	Raulin, Julie
Publication date	2025-09-01
Original Citation	Raulin, J. 2025. Packetponder: a multi-domain, software-defined, packet-optical convergence node at the edge. PhD Thesis, University College Cork.
Type of publication	Doctoral thesis
Rights	© 2025, Julie Raulin. - https://creativecommons.org/licenses/by-nc-sa/4.0/
Download date	2026-10-31 16:56:30
Item downloaded from	https://hdl.handle.net/10468/18855

Packetponder: a multi-domain, software-defined, packet-optical convergence node at the edge

Julie Raulin

M.ENG.SC

119225871

**Thesis submitted for the degree of
Doctor of Philosophy**



NATIONAL UNIVERSITY OF IRELAND, CORK

COLLEGE OF SCIENCE, ENGINEERING AND FOOD SCIENCE

SCHOOL OF COMPUTER SCIENCE & INFORMATION TECHNOLOGY

TYNDALL NATIONAL INSTITUTE

PHOTONICS SYSTEMS GROUP

September 2025

Head of School: Prof. Dirk Pesch

Supervisors: Dr. Fatima Gunning
Prof. Cormac Sreenan
Dr. Paul Gunning

Contents

List of Figures	v
List of Tables	viii
Acknowledgements	x
Abstract	xii
List of Publications	xiv
1 Introduction	1
1.1 Motivation	1
1.1.1 The growing need for flexibility	1
1.1.2 The rise of power consumption	3
1.1.3 Software-Defined Networking for convergence, simplification and increase in energy efficiency	4
1.2 Challenges	5
1.3 Thesis statement	7
1.4 Contribution	7
1.5 Thesis outline	11
2 Background	13
2.1 Origins of optical communications	13
2.1.1 Historical milestones in visual signalling	13
2.1.2 Pioneering long-distance electrical networks	14
2.1.3 Modern optical communication systems	15
2.2 The Software-Defined Networking paradigm	17
2.2.1 The concept	17
2.2.2 SDN whitebox solutions for the packet layer	19
2.2.3 Advancements in open Control Planes	20
2.2.4 SDN paradigm for optical networks	21
2.2.5 SDN whitebox solutions for the physical layer	22
2.2.6 Advancements in the optical Control Plane standards	22
2.3 IP-over-DWDM	23
2.3.1 Transport technologies evolution	23
2.3.2 Origins of the IP-over-DWDM concept	25
2.3.3 The evolution of data transmission systems	26
2.3.3.1 SERDES evolution	26
2.3.3.2 Optical transmission evolution with coherent pluggables	27
2.3.3.3 The miniaturisation of Digital Signal Processors enabling power-efficient pluggables	28
2.3.4 Power savings and economical benefits	29
2.4 Interoperability	30
2.4.1 YANG models for device and configuration standardisation	30
2.4.1.1 Optical domain YANG models	31
2.4.1.2 IP domain YANG models	31
2.4.1.3 Packet-optical multi-domain YANG models	31
2.4.2 Packet-over-optics standardisation efforts before 2021	32

2.4.3	Interoperability developments from 2022 onwards . . .	33
2.5	Conclusion	34
3	Packet-optical convergence with <i>packetponders</i>	36
3.1	Challenges in pluggable optics integration and management . .	36
3.2	Packetponder concept with common form factor pluggables . .	38
3.2.1	Concept and name	38
3.2.2	Client-side and Line-side definitions	39
3.2.3	<i>Packetponder</i> : an enhanced Transponder and Muxponder	40
3.2.4	Wavelength conversion and 3R regeneration	41
3.2.5	Plug-and-play approach with legacy and new pluggables	42
3.3	Experimental software stack and hardware	43
3.3.1	Hardware components	44
3.3.2	Network Operating System for our <i>packetponder</i>	44
3.4	Pluggables integration in open NOS	46
3.4.1	I ² C bus and pluggable registers	46
3.4.2	MUTEX	48
3.4.3	Frequency change for SFF interfaces	50
3.4.4	Frequency change for CMIS interfaces	50
3.4.5	Abstraction layers in Stratum	51
3.4.5.1	Southbound interface of Stratum: <i>onlp_wrapper</i>	52
3.4.5.2	Northbound interface of Stratum: <i>gNMI server</i>	53
3.5	Pluggables remote management with open SDN controller . . .	55
3.6	Experimental proof-of-concept	58
3.6.1	Wavelength conversion of SFF and CMIS-compliant plug- gable from ONOS controller	59
3.6.2	Reconfiguration time of the 100ZR pluggable transceiver in a ROADM scenario	60
3.7	Implementation discussion with current state-of-the-art	64
3.8	Conclusion	66
4	Wavelength provisioning in open packet-optical SDN controller	68
4.1	Current packet-optical SDN controller: wavelength assignment algorithm and failure recovery strategy	69
4.1.1	Controllers coordination in traditional packet-optical net- works with transponders	69
4.1.2	Hierarchical controller and abstraction interfaces for packet- optical node with pluggables	70
4.1.3	Failure recovery state-of-the art in SDN controllers . . .	72
4.1.4	Routing and Wavelength Assignment (RWA) in SDN con- trollers	73
4.1.4.1	RWA studies over the years	73
4.1.4.2	RWA in current SDN controllers	75
4.2	Proposed multi-layer controller and routing decision	76
4.3	Open-source multi-layer simulator and emulator: Mininet-optical	78
4.3.1	Packetponder node within Mininet-optical	79

4.3.2	Software-defined control of the <i>packetponder</i> element with Mininet-optical	81
4.4	Failure recovery scenario with packetponders in a 9-ROADMs mesh topology	82
4.4.1	Emulated testbed components	83
4.4.2	Network configuration and load	84
4.4.3	Fault recovery scenario	85
4.4.4	SDN controller for emulated network	87
4.4.5	Random ROADM-mesh network results	88
4.4.5.1	Analysis of Recovery Attempts for Link Failure r8–r9	88
4.4.5.2	Analysis of Recovery Attempts for Link Failure r3–r4	90
4.4.5.3	Analysis of Recovery Attempts for Link Failure r6–r7	91
4.4.5.4	QoT and latency measurements	92
4.5	GEANT network topology experiment	93
4.6	Discussion and conclusion	96
5	Packetponder for access, metro and core networks	99
5.1	Edge and services consolidation	100
5.1.1	Software-Defined Passive Optical Networks (SD-PON)	100
5.1.1.1	SDN-controlled OLTs	101
5.1.1.2	MicroPlug OLT and control	103
5.1.2	Converged Metro-Access Platforms: Universal Edge Devices	104
5.1.3	Alternative solutions for aggregation and consolidation	106
5.2	Packetponders for access/metro consolidation	108
5.2.1	Hardware consolidation with <i>packetponders</i> at the edge	108
5.2.2	Multi-services and aggregation node	110
5.2.3	Interoperability and management framework	111
5.3	Multi-services aggregation demonstration with a <i>packetponder</i>	112
5.3.1	Hardware testbed	112
5.3.2	PON integration and management	114
5.3.3	Traffic aggregation from different services	119
5.4	Impact on network topologies	123
5.4.1	BT network use-case	123
5.4.2	<i>Packetponders</i> for more simplified network, more flexibility and better management	125
5.4.3	Preliminary results on potential energy savings by using the <i>packetponder</i>	126
5.4.3.1	Typical architecture	126
5.4.3.2	Architecture with <i>packetponders</i>	128
5.5	Conclusion and Discussion	129
6	Conclusion	131
	Acronyms	137

Appendix	142
--------------------	-----

List of Figures

1.1	Traditional network architecture with transponders as packet and optical layer separator.	6
1.2	Development timeline of packet-over-DWDM technologies. . . .	10
2.1	Chappe optical telegraph in Saverne, France: last remaining vestige of the connection Paris/Strasbourg from 1798 to 1852. . . .	14
2.2	(a) Landing point in Valentia Island, Ireland, of the first transatlantic cable attempt; (b) Actual vestige of the 1866 cable exposed in the Valentia Island museum, that successfully linked Ireland and Canada.	15
2.3	Current network topology from access to core. Figure extracted from [19].	16
2.4	SDN layer architecture from [24].	18
2.5	Timeline of packet-over-DWDM developments during my PhD research.	32
3.1	<i>Packetponder's</i> concept.	39
3.2	IP-over-DWDM concept with coherent pluggables, including SERDES electrical connections within the Transponder.	40
3.3	<i>Packetponder's</i> functionality	41
3.4	<i>Packetponder</i> in metro/core networks	42
3.5	Conceptual layered software architecture of a disaggregated switch.	45
3.6	I ² C block diagram in our Ethernet switch to access the ports. . . .	48
3.7	Block diagram to set new frequency for SFF pluggable transceivers (10G/25G).	51
3.8	Block diagram to set new frequency for CMIS pluggable transceivers (100/400G ZR/ZR+).	52
3.9	Initial chassis configuration text file with the added frequency parameter.	54
3.10	Simplified architecture of Stratum [85].	54
3.11	SDN management stack of our <i>packetponder</i>	55
3.12	ONOS simplified architecture for the frequency tuning function. . . .	58
3.13	Remote frequency configuration using ONOS, OpenConfig, YANG and gNMI	59
3.14	Stratum logs for 100ZR wavelength reconfiguration.	60
3.15	Stratum logs for 10G/25G wavelength reconfiguration.	61
3.16	Experimental setup for proof-of-concept and packet loss measurements.	61
3.17	ROADM MUX rules.	62
3.18	Spectrum showing wavelength reconfiguration of the 100ZR	62
3.19	<i>IxNetwroks</i> data rate statistics	64
4.1	Traditional network architecture with SDN control.	70
4.2	Control conflict (red dotted line) between packet and optical layers when using switch+pluggable optics.	71

4.3	Network architecture and SDN control with <i>packetponders</i>	77
4.4	Virtual testbed: Mininet and Mininet-optical.	78
4.5	3-nodes topology to test the <i>packetponder</i> 's implementation in mininet-optical. "h": host, "t": terminal, or <i>packetponder</i>	80
4.6	<i>packetponder</i> control using REST in mininet-optical.	81
4.7	Hairpin experiment results showing the successful implementation of the <i>packetponder</i> element into mininet-optical	82
4.8	9-POP topology for failure recovery experiment. POP1-POP3 is an example of link configuration and the same applied between every POP connected to each other.	83
4.9	Network configuration and failure recovery steps during the experiment.	85
4.10	Emulated topology in mininet-optical with failure example r8-r9.	85
4.11	eRWA pseudo code.	86
4.12	New rules to be installed for the ROADMs r2,r3,r4,r9 and the <i>packetponder</i> t3.	87
4.13	Failure recovery analysis for link failure r8-r9	89
4.14	Failure recovery analysis for link failure r3-r4	90
4.15	Failure recovery analysis for link failure r6-r7	91
4.16	PING test from h2 $\rightarrow$ h9 before and after reconfiguration, showing latency measurements and successfully end-to-end connectivity of the new lightpath t2 $\rightarrow$ t9 using λ conversion.	93
4.17	Simplified version of the GEANT optical network, linking European cities together.	94
4.18	GEANT network: failure recovery analysis for link failure r8-r9	95
4.19	GEANT network: failure recovery analysis for link failure r18-r20	95
5.1	Typical PON architecture.	100
5.2	Network element consolidation: a) Typical network; b) Packetponder implementation at the metro/core edge; c) Proposed Packetponder implementation for Access/Metro edge with pluggable OLT and DWDM transceivers.	109
5.3	Packetponder at the network edge: supporting 5G fronthaul, point-to-multipoint PON and point-to-point leased line Access services; and coherent optical DWDM back-haul and long-haul transmission across the <i>Metro/Core</i>	110
5.4	<i>Packetponder</i> with PON and aggregation functionalities.	111
5.5	Proof-of-concept demonstration: SDN software control of the <i>packetponder</i> to support access/metro services.	113
5.6	Experimental arrangement: Packet flow to/from TTG via DUT & ROADM.	113
5.7	PON experimental setup to test MicroPlug Tibit inside the <i>packetponder</i>	114
5.8	Software architecture to control the Tibit using MCMS software.	116
5.9	MCMS dashboard, showing the integration of the OLT pluggable and the two ONUs.	117

5.10	Traffic items sent during the experiment, showing no loss of packets, with associated latency measurements.	119
5.11	Upstream traffic rate (Rx) of ONU1 and ONU2 during the experiment.	119
5.12	Common-form-factor pluggable identification, port-labelling & connectivity mapping scheme between TTG, DUT and ROADM.	121
5.13	Number of frames sent and received by the traffic generator during a 10-hour run.	121
5.14	Throughput — dynamic, sequential addition of traffic.	122
5.15	Snapshot of the throughput during the experiment with IxNetwork software, showing L1 data rate and the number of packets traversing.	122
5.16	BT Topology: (a) Current architecture with Core, Metro, Tier 1 and Multi-Service Access Node (MSAN) nodes; (b) Hypothetical 2040+ network architecture showing a simplified topology without MSAN nodes. While not fully representative of real deployment constraints, it illustrates the potential site reduction achievable through network consolidation, retaining only Core, Metro, and Tier 1 nodes.	124
5.17	Power estimation in the typical architecture use-case.	127
5.18	Power estimation in the packetponder use-case.	129

List of Tables

2.1 YANG encoding method with NETCONF, RESTCONF and gNMI protocol [63].	31
--	----

I, Julie Raulin, certify that this thesis is my own work and has not been submitted for another degree at University College Cork or elsewhere.

Julie Raulin

Acknowledgements

Writing the acknowledgements section has always been my favourite part of the thesis. Through these lines I want to thank everyone who has been part of this journey and who has turned my PhD experience into something far more joyful and enriching than I could ever have imagined.

First and foremost, my deepest gratitude goes to Dr Fatima Gunning, my main supervisor. You have been not only an excellent supervisor but an even greater mentor. Thank you for knowing when to support and when to challenge me, for your patience, your trust, and your constant encouragement. Against all odds and contrary to many stories I had heard about the PhD life, you made this journey feel more like an adventure. I have learnt so much from you, not only technically, but also how to grow as a researcher and as a person. I could not have asked for a better guide.

To my external supervisor, Dr Paul Gunning, thank you for being a living, breathing encyclopaedia, seemingly knowing (almost) everything and everyone. Thank you for believing in me, trusting me, and for always having my back.

And completing this dream supervisory team, Prof Cormac Sreenan, thank you for your sharp insights, wise advice, and constant support. Every contribution of yours mattered more than I can put into words.

I am also grateful to all the people I had the pleasure to collaborate with. To the Mininet-optical team at Trinity College Dublin and in the US, and in particular Dr Alan Diaz-Montiel, whose help in the early stages of my PhD was invaluable, and Dr Bob Lantz, whose expertise in software was extremely helpful, and whose sharp comments were always on point and pushed me to see things more clearly. To Gawen Davey, who probably saved me months of unravelling coding mysteries. Thank you for being available just when I needed it (I know I am lucky) and for spending countless hours debugging and troubleshooting with me. To Dr Jim Zou, who has supported me unfailingly since the day we met. Your constant support, trust, and generosity with your time and energy have meant more than I can say. Thank you for teaching me all the things the books don't cover about navigating a PhD and beyond, and for making all the conferences so much fun. And to Dr Kaida Kaival, with whom I have not had the pleasure to work with (yet) but whose advice, support, and generous introductions to so many people in our field (along with plenty of laughs) have been truly invaluable.

To all the people and mentors I have met along the way, thank you for sharing your experiences and advice. This PhD journey would not have been the same if I hadn't met all of you in conference corridors and exhibitions. Thank you for taking me under your wing and always bringing some fun to the work.

To the Photonics Systems Group, thank you for being not only brilliant minds but also brilliant company. From sharing experiences over far-too-long coffee breaks to hours spent hunting for mysterious devices in the lab, you made even

the most frustrating days bearable.

To my dear friends in Cork — Eoin, John, Conor, Salam, Dario, Pasha — thank you for every coffee break and after-work pint, for the laughs, the adventures, and the countless memories. For always reminding me what life is about, and for making everything about Cork more fun. For making me stronger and also weaker at the same time (you know what I mean). For all the days and nights, I am eternally grateful to have spend those few years with all of you around. To Rachel and Camille, for the much-needed reminders of home comforts and especially how a cheese night can save a long and exhausting day. To my French crew (FierENSEA#), the entire AA group, and everyone I met in Cork, you know who you are, and you made every day brighter.

To my housemates through the years, Veronica, Samira, Giulia, Rachel, and Mehran; thank you for living through every high and low with me. From late-night chats to early-morning grumpy hellos, from listening and advising to laughing and sharing the same PhD roller-coaster, you made the last five years so much more enjoyable.

To my friends back in France, who have seen me leaving France wondering why on earth I chose to live in a country with such questionable weather and food (they have yet to taste the beers, though). Your humour and your reminders that I made the right choice kept me grounded. And to Héloïse, my dearest friend. You have been one of my greatest supports throughout this PhD. Thank you for understanding everything without needing explanation, for sharing the ups and downs, the doubts and the successes, and for always being there despite the distance. I have learnt so much from your own experiences, which helped me to grow, avoid mistakes, and do things better along the way. I'm incredibly lucky to have you in my life.

Last but not least, my biggest thanks to my family. To my mom, for your unconditional love and support, for being proud of me every single day, and for teaching me the value of building a network long before I knew what it meant. To my dad, for always reminding me that nothing will ever be as stressful as the “khôles”, and that if I survived those, I can survive anything. And to my brother, for the quiet but immense support I've always felt.

Abstract

The exponential growth of traffic from cloud services, Artificial Intelligence (AI) workloads, and latency-sensitive edge applications (such as those running close to end-users or devices) is placing unprecedented demands on the underpinning telecommunications infrastructure. Supporting such use cases requires networks that not only can provide for high-capacity, but are also dynamic, resilient, and more efficient. Yet, despite progress, development and research in disaggregation and open networking, today's infrastructure remains highly fragmented, with separate hardware platforms and control systems for different network domains — access, metro and core — leading to operational silos, complicated end-to-end orchestration, duplicated functions, and underused capabilities.

To answer these challenges, this thesis introduces the *packetponder* concept: it relies on a fully open and programmable whitebox switch capable of accommodating a wide range of modular, common-form-factor pluggable transceivers. While traditionally designed to support electronic switching functions, the *packetponder*'s open software stack and programmable hardware enable not only layer 1 and layer 2 convergence, but also position it as a strong candidate for broader applications spanning access, metro, and core networks, while enabling unified management across network layers. Indeed, by consolidating packet and optical functions into a single node, the *packetponder* reduces the need for multiple dedicated devices such as Optical Line Terminals (OLTs), transponders, and aggregation switches. This consolidation not only simplifies network architectures but also lowers operational complexity and, in turn, energy consumption. Moreover, by exposing all functions to a centralised controller through open and standardised APIs, the *packetponder* allows end-to-end visibility and control, enabling more flexible, resilient, and vendor-agnostic networks.

As a proof of concept, this thesis demonstrates, for the first time, the feasibility of the *packetponder* across multiple network domains. In the metro/core domain, it bridges the gap between packet and optical layers, a convergence that gained industry traction during the course of this PhD with the rise of coherent pluggable transceivers. However, managing such a device remains challenging, as operational responsibility are traditionally split between packet and optical teams. With a *packetponder*, the thesis proposes one approach based on common, standardised APIs and open-source software, to enable vendor-independent integration. Specifically, an open-source Network Operating System (NOS) originally designed for packet switching was extended to support the optical-layer control,

including dynamic configuration of tunable and coherent pluggable transceivers. This enhancement provides a foundation for multi-domain, multi-vendor interoperability. Consequently, the control plane was re-designed to reflect the *packetponder*'s hybrid role: an advanced Routing and Wavelength Assignment (RWA) algorithm was developed and embedded in an SDN controller, leveraging the wavelength tunability of *packetponder* nodes to improve resource utilisation and fault recovery. Simulations over realistic topologies demonstrated the ability to identify and configure additional lightpaths in failure scenarios, even after conventional recovery strategies were applied. Finally, the feasibility of extending the *packetponder* to the network edge was experimentally validated, integrating mixed access-domain traffic via standard form-factor pluggables and open software interfaces where possible, and successfully transmitting an aggregated high-capacity signal of 99.56 Gbits/s over metro-scale distances. By co-locating packet switching with 3R optical-electrical-optical wavelength conversion, the *packetponder* introduces a new architectural paradigm with significant implications for the evolution of TELCO networks.

Through these contributions, this thesis shows that the *packetponder* is a strong candidate to enable true node consolidation and end-to-end simplification of fragmented infrastructures, delivering vendor-agnostic flexibility under a unified SDN control framework. As data growth accelerates with AI and other emerging applications, such convergence is not just desirable but essential to ensure networks remain scalable, efficient, and resilient in the years ahead.

List of Publications

Peer-reviewed published paper:

- [JR4] J. Raulin, J. Zou, S. Hill, J. Griffiths, G. Davey, A. Jeffries, I. Cooper, P. Gunning, C. J. Sreenan and F. C. Garcia Gunning, "Simplification of Edge Network Architecture with Open Packet Switches and Pluggable Transceivers," 2024 IEEE Photonics Society Summer Topicals Meeting Series (SUM), Bridgetown, Barbados, 2024, pp. 1-2, doi: 10.1109/SUM60964.2024.10614540.
- [JR3] J. Raulin, J. Zou, S. Hill, J. Griffiths, G. Davey, A. Jeffries, I. Cooper, P. Gunning, C. J. Sreenan and F. C. Garcia Gunning, "Single Open Programmable Packetponder Element for Access/Metro Aggregation and Network Consolidation," In Proceedings of 2024 International Conference on Optical Network Design and Modeling (ONDM), Madrid, Spain, 2024, ISBN: 978-3-903176-62-1.
- [JR2] J. Raulin, G. Davey, Y. Verbishchuk, A. Jeffries, C. J. Sreenan, F. C. Garcia Gunning, "Packetponder with open-source management system for future packet-optical networks", In Proceedings of 2023 Optica Advanced Photonics Congress (APC), doi: 10.1364/NETWORKS.2023.NeM3B.3.
- [JR1] J. Raulin, G. Davey, Y. Verbishchuk, A. Jeffries, C. J. Sreenan, F. C. Garcia Gunning, "A programmable Ethernet transport Packetponder using common compact form factor pluggable tunable transceivers to support novel DWDM architectures," In Proceedings of 2023 Optical Fiber Communications Conference and Exhibition (OFC), San Diego, CA, USA, 2023, pp. 1-3, doi: 10.1364/OFC.2023.Tu3D.6.
- [JR5] F. C. Garcia Gunning, A. Kaur, N. C. Estrada, J. Raulin, Y. Verbishchuk and C. J. Sreenan, "Enabling High Capacity Flexible Optical Networks," In Proceedings of 2021 International Conference on Optical Network Design and Modeling (ONDM), Gothenburg, Sweden, 2021, pp. 1-3, doi: 10.23919/ONDM51796.2021.9492503.

Paper [JR1] introduces the *packetponder* concept for consolidating and simplifying metro and core networks. It demonstrates the integration of tunable transceivers into disaggregated switches using open-source software and open APIs, as described in Chapter 3. Paper [JR2] details the open-source software stack and the method for controlling tunable pluggables via a remote controller, proposing a uniform control approach for operators regardless of vendor equipment. This work is also presented in Chapter 3, with further analysis of the controller aspects provided in Chapter 4. Finally, Papers [JR3] and [JR4], described in Chapter 5, extend the *packetponder* concept to the network edge, enabling the consolidation of access, aggregation, metro, and core networks within a single open and disaggregated platform.

Author's contribution to the publications:

For the papers mentioned above, my contributions as first author included developing the research concept, designing and implementing the laboratory test setup, creating the use cases, executing the experiments, collecting and analysing the data, and drafting the full initial manuscript.

The co-authors contributed by providing the equipment required for the demonstrations, offering hands-on guidance or training for the devices or control software when necessary, participating in discussions on the initial structure and flow of the papers, and proofreading the work.

Other contribution:

- B. Lantz, *et al.*, "SDN-controlled Dynamic Front-haul Provisioning, Emulated on Hardware and Virtual COSMOS Optical x-Haul Testbeds," 2021 Optical Fiber Communications Conference and Exhibition (OFC), San Francisco, CA, USA, 2021, pp. 1-3.

I helped with the demo: debugging and creating test files on mininet-optical emulation tool, and creating a visualisation code in python for the demo.

- J. Raulin and A. Raj, "Mininet-Optical tutorial," ACM SIGCOMM 2022, Tutorial, Amsterdam, Netherlands, 2022.

I gave a talk and tutorial on mininet-optical as part of the COSMOS tutorial session in ACM SIGCOMM conference.

Open-source code:

The software framework used in this work comprises Stratum [JRc1] and ONL [JRc2] as the NOS, and ONOS [JRc3] as the controller. All three components are open-source and have been extended in this work to support the frequency tuning of various tunable and coherent pluggable transceivers. While ONL code is inherently platform-specific (with ONLP which provide hardware-specific data and control functions), both Stratum and ONOS added code employ generic functions to ensure broad compatibility. [JRc4] provides the code for using Mininet-optical to create packet-optical network topologies and connect them to an SDN controller. It also includes implementations of routing and wavelength assignment algorithms as described in Chapter 4, along with an evaluation of a failure recovery use case leveraging *packetponder* nodes.

The code can be found:

- [JRc1] Stratum: 10.5281/zenodo.10119264
- [JRc2] Open Network Linux (ONL): 10.5281/zenodo.10119254
- [JRc3] ONOS: 10.5281/zenodo.10119275
- [JRc4] Mininet-optical experiment for *packetponder* control: 10.5281/zenodo.17105578

Chapter 1

Introduction

In this chapter, the motivation for the research is introduced, presenting the key research question of how Software-Defined Networking (SDN) paradigm could address network complexity challenges while offering potential energy savings.

1.1 Motivation

1.1.1 The growing need for flexibility

Subsea cables are the invisible backbone of the global economy, carrying around 99% of all intercontinental Internet traffic. While satellite communication often captures the public imagination, it represents only a very small fraction of the world's data transmission. These vast networks of fibre optic cables, stretching across oceans, enable everything from financial transactions and cloud services to video streaming and global business operations. Without them, modern life as we know it — instant communication, seamless trade, and global connectivity — would simply not exist.

However, their indispensability also makes them highly vulnerable to both accidental damage and deliberate attacks. One such incident occurred in January 2022, when the single undersea cable connecting Tonga to the rest of the world was severed following a massive volcanic eruption. The island nation was abruptly plunged into digital isolation for weeks, with serious humanitarian consequences. At the same time, satellite services were disrupted by the volcanic plume, leaving citizens unable to call for help, access weather forecasts,

or receive tsunami warnings. Emergency services such as hospitals and fire departments were unreachable, highlighting the extreme vulnerability of isolated regions dependent on limited communication infrastructure.

Another striking example occurred in late 2024, when a critical Baltic subsea communication cable was severed, leaving thousands without Internet for several days and causing two months of limited connectivity in affected regions [1]. Whether caused by an accident, such as an anchor dragging across the seabed, or a more covert act of sabotage, the consequences were severe. This incident highlighted how fragile these connections are, unforeseeable weather events are, and how geopolitical tensions have the potential to disrupt entire economies, financial markets, and national security operations.

As the primary arteries of the Internet, subsea cables represent a major point of failure and expose a critical vulnerability in global communications, emphasising the urgent need for greater network flexibility. If not carefully designed with flexibility, resilience and redundancy, their destruction could disable global connectivity. This is especially critical today, as we rely on low-latency applications such as remote surgery, autonomous vehicles, and real-time financial trading, which cannot tolerate network outages. A lack of connectivity, even for a few seconds, could mean catastrophic failures in these time-sensitive applications. These cables carry enormous amounts of data, and when one is damaged, rerouting traffic is not always straightforward due to limited available capacity on alternative paths. In a traditional static provisioning, which means only pre-configured backup paths can be used, only little room is left for dynamic adjustments based on real-time network conditions. More adaptive solutions (i.e. highly flexible network devices and intelligent management systems) need to be found, that could enable a more efficient, on-demand reallocation of bandwidth, ensuring that priority traffic is preserved while making the best use of available resources.

The issue is not limited to subsea networks; terrestrial infrastructure faces similar challenges. Data centres and backbone networks must be capable of handling sudden shifts in traffic loads without excessive delays or over-provisioning. A flexible, software-driven approach is needed in the network to dynamically redistribute bandwidth, reconfigure optical links, and adjust network policies based on real-time demands, which is what this thesis will investigate.

1.1.2 The rise of power consumption

At the same time, the rapid rise of Artificial Intelligence (AI) applications has led to the emergence of hyperscale data centres, which are notably power-demanding. For instance, a ChatGPT search consumes ten times the electrical energy of a standard Google search [2]. Locally, some power grids are already on the verge of saturation. In Ireland, for example, the proliferation of data centres has significantly impacted national energy consumption: they collectively consumed 11% of the national electricity supply in 2020, compared to 22% for all urban domestic dwellings and 12% for rural domestic dwellings [3]. By 2023, this picture had changed significantly: data centres consumed 21% of the national electricity supply, nearly double their share from three years earlier. This was more than the consumption of all urban domestic dwellings (18%) and twice the consumption of all rural domestic dwellings (10%) within the state [4]. It is anticipated that electricity consumption will increase to approach 30% of the Irish power grid by 2030.

This surge challenges the nation's commitment to renewable energy, as the sudden and substantial demand may necessitate reverting to fossil fuel-based power sources to bridge the gap [5]. Without strategic interventions, the increasing energy requirements of AI and data centres could lead to increased carbon emissions, affecting global sustainability efforts and preventing the European Union from meeting its climate neutrality targets under the European Green Deal [6], which is achieving climate neutrality by 2050. Implementing energy-efficient and flexible network solutions is therefore crucial to balance technological advancement with environmental responsibility.

Hyperscaler cloud providers and telecommunications network operators (TELCOs) recognise this challenge, making the reduction of energy consumption across compute, storage, and network resources a target commitment in their sustainability efforts [7]. Hyperscalers concentrate resources regionally into few, massive data centres — often exceeding 10,000 square meters in area — of common modular design [8] scattered across their global infrastructure [9]. Their respective energy usage is appreciable: Amazon web services (AWS) consumed 24TWh of electricity in 2020 while Google consumed 15.4TWh in 2021 [10]. Nevertheless, they continue to work towards improving energy efficiency, which is commonly expressed by the power usage effectiveness (PUE) metric [11]. Telecom operators also consume substantial amounts of energy: AT&T (17.1 TWh), Telefónica (6.1 TWh), Vodafone (5.8 TWh), and BT (3.3 TWh) [10]; primarily

driven by the demands of their mobile and fixed networks. To face this challenge, hyperscalers and TELCOs have committed to migrate to renewable, low-carbon energy built atop technical innovations that promote power efficiencies [7]. At BT, one approach involves dynamically switching interfaces, ASICs, processors, or transmission links on and off based on traffic demand [12], which allows networks to scale down power usage during periods of low utilisation. Other options range from simple-to-implement routing policy changes that re-direct traffic to traverse lower-carbon routes [13], to opportunities to rationalise and consolidate functions within fewer hardware devices by harnessing Moore's Law. This thesis investigates this approach to consolidation.

1.1.3 Software-Defined Networking for convergence, simplification and increase in energy efficiency

One promising technology to succeed the consolidation and thus simplification in today's networks is the use of the Software-Defined Networking (SDN) paradigm. By enabling the decoupling of functions from closed, proprietary systems through open hardware/software interfaces and Application Programming Interfaces (APIs), SDN has the potential to be transformative. Organisations such as the Open Compute Project (OCP) [14], Open Networking Foundation (ONF) [15] and Telecom Infra Project (TIP) [16] are actively developing open specifications that provide alternatives to proprietary blackboxes, so called *whiteboxes*. Such whiteboxes provide opportunities for technology substitution, elimination and consolidation that can drive profound changes. This shift toward open, programmable networks benefits both TELCOs and hyperscalers by simplifying the infrastructure layer and enabling automated provisioning of network elements and service functions they support.

SDN has been widely adopted across various network domains and layers, demonstrating its effectiveness in packet, wireless, and, more recently, optical networks. It has driven innovation, eliminated vendor lock-in, and can provide operators with greater control. While SDN was not originally designed for energy efficiency, extending its scope to unify network domains and streamline operations could potentially enhance flexibility, lower costs, and significantly reduce power consumption.

But for that, achieving a simpler and more adaptable network requires a converged network layer architecture with a multi-domain control plane. At the hardware level, this means deploying an open and disaggregated platform that

would support diverse pluggable transceivers, facilitating both packet-optical convergence, and the integration of access and metro networks into a unified, programmable system. On the software side, a multi-domain control plane is essential, providing a centralised, SDN-based framework that seamlessly manages all network layers and domains. By adopting an open architecture, interoperability between vendors is ensured, allowing operators to break free from vendor lock-in while optimising network flexibility, scalability, and efficiency for example through multi-source agreements.

To align with this vision, this thesis introduces the *packetponder*, an open and programmable platform designed to fit this converged architecture. Rather than being tied to specific hardware or software, the *packetponder* embraces standardised, common-form-factor pluggables and open APIs. This fosters innovation at both the hardware and software levels, enabling operators to mix and match solutions from different vendors while maintaining a highly adaptable and future-proof network infrastructure.

1.2 Challenges

Networks evolved from very simple point-to-point transmission to complex mesh networks supporting different services and technology. As demand for connectivity grew, networks expanded by adding more hardware, leading to increased complexity and power consumption. Each new device required additional space, power, and cooling, making large-scale networks expensive to operate. Managing this infrastructure became challenging, as any configuration change or troubleshooting required coordination across multiple teams with different expertise. This is because each network domain — access, metro, and core — has historically been developed independently, leading to whitebox hardware and software solutions that were specifically tailored to function only within their respective domains. As a result, there has been no unified platform capable of supporting applications and services across multiple network domains in an open and seamless manner, and still today, telecommunications are deployed over a heterogeneous infrastructure in terms of domains, technologies, and management platforms, making the implementation of a multi-layer, multi-domain, and centralised controller solution challenging.

As will be mentioned in Chapter 3, the metro and core domains have received significant attention in recent years for merging the packet and optical layers.

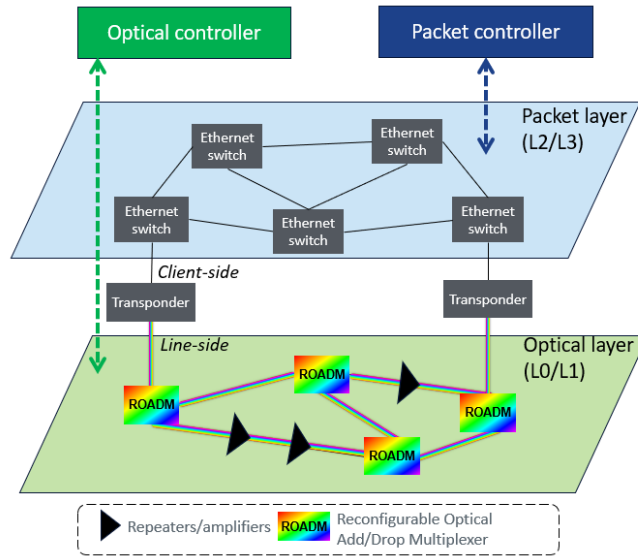


Figure 1.1: Traditional network architecture with transponders as packet and optical layer separator.

Fig. 1.1 represents the traditional network architecture with the use of transponders but coherent pluggable transceivers are emerging as energy-efficient alternatives to traditional transponders and a promising solution to simplify metro/core networks. A coherent pluggable transceiver is a compact optical module that uses coherent detection and advanced digital signal processing to transmit and receive high-capacity data over long distances, and can be inserted into standard switch or router ports.

This evolution occurred in parallel with the exponential growth of Ethernet traffic, driven largely by AI and enabled by those advances in pluggable optics. However, integrating these transceivers into switches and routers presents challenges, particularly on the software side for their management. While both academia and industry are actively working on solutions, a fully open and comprehensive approach that seamlessly integrates tunable and coherent pluggables is still not quite there yet. The 400ZR transceivers [17], as an example, began to gain traction at the beginning of my PhD research and have since become critical components in modern network architecture, highlighting both the relevance and the timely nature of this thesis work.

On the access/aggregation side, explored in Chapter 5, pluggable technologies such as MicroPlug Optical Line Terminals (OLTs) and XR optics are gaining traction, offering infrastructure simplification, cost reduction, and power savings. Yet, similar challenges persist in managing these pluggables within generic switches

and routers.

Despite these advancements, a truly unified hardware platform with an open software stack, capable of seamlessly supporting and managing all these technologies across network domains, has yet to emerge.

Additionally, domain convergence also requires organisational alignment, as it brings together separate teams in charge of managing different network layers. For example, as depicted in Fig. 1.1, the optical controller and the packet controller are separate entities, each managed independently by different teams. These teams traditionally rely on distinct protocols, APIs, and software solutions, making the adoption of a unified control framework a significant challenge. Current architectures lack the centralised intelligence needed to optimise resources and provide end-to-end visibility across domains and layers, highlighting the importance of continued research and development in this space. A converged solution would necessitate consensus on standardised control mechanisms, defining who has write or read-only permissions over various devices. Considerable work is being undertaken in this area, with academia and industry collectively collaborating to develop standardised and interoperable approaches.

1.3 Thesis statement

This thesis proposes the *packetponder* as a novel approach to overcome the operational and architectural fragmentation that has historically separated network layers. While inherently more energy-efficient than traditional architectures, its main contribution lies in consolidating diverse functions into an open, programmable, and disaggregated platform. The *packetponder* enables architectural simplification and unifies management control across domains, supporting the convergence of packet and optical layers.

1.4 Contribution

The proposed *packetponder* architecture directly addresses key challenges in modern telecommunications networks as depicted in section 1.1. By consolidating packet and optical functions into a single programmable node, this approach reduces the number of physical devices and inter-domain interactions, which can contribute to lower end-to-end latency. The consolidation of multiple network elements, such as transponders, aggregation switches, and access equipment,

combined with improved utilisation of shared hardware resources, also supports enhanced energy efficiency, partially illustrated through the analysis presented in Chapter 5. Furthermore, the use of an SDN-based unified control across layers, together with open hardware, open-source software, and standardised APIs, enables more flexible and vendor-agnostic network operation, which provides the foundations for fast and adaptive reconfiguration in response to failures, as explored in Chapter 4. Finally, the proposed approach is compatible with ongoing advances in coherent pluggable technologies as demonstrated in Chapter 3, and can scale with increasing interface capacities (e.g., 400 Gbit/s, 800 Gbit/s, and beyond), making it applicable to future high-capacity network infrastructures.

Specifically, this thesis addresses key challenges in multi-layer, multi-domain convergence, highlighted in section 1.2, and the main technical contributions are as follows:

- **Enhancement of an open-source Network Operating System (NOS):** To address the challenge of managing coherent and tunable pluggable transceivers in a unified, open, and vendor-agnostic manner, this thesis enhances an existing open-source NOS that was originally designed for packet-based applications. The extended NOS introduces support for optical-layer parameters, including the dynamic configuration of frequency settings for tunable and coherent pluggables, directly from the same control environment used for packet switching. By embedding optical management into the packet NOS, the work facilitates a truly disaggregated and open approach to control packet-optical devices, streamlining deployments, reducing operational complexity, and paving the way for more energy-efficient and flexible metro/core architectures.

[JR1] J. Raulin et al., "A programmable Ethernet transport Packetponder using common compact form factor pluggable tunable transceivers to support novel DWDM architectures," 2023 Optical Fiber Communications Conference and Exhibition (OFC), San Diego, CA, USA, 2023, pp. 1-3, doi: 10.1364/OFC.2023.Tu3D.6.

- **Remote control of tunable/coherent pluggable transceivers:** This thesis demonstrates the ability to remotely configure and reconfigure the frequency of tunable and coherent transceivers using an open-source SDN controller and standardised protocols such as gNMI. The experimental results show a reconfiguration time of a few seconds for three different transceivers' type in a ROADM-based network. This integration into the

SDN controller removes the historical separation between packet and optical management systems, allowing coherent/tunable pluggables to be managed as seamlessly as traditional Ethernet interfaces. The novelty lies in bridging the operational gap between Layer 2/3 switch/router environments and the optical transport layer without requiring proprietary vendor software, thereby enabling a unified control plane for metro and core networks.

[JR2] J. Raulin et al., "Packetponder with open-source management system for future packet-optical networks", 2023 Optica Advanced Photonics Congress (APC), doi: 10.1364/NETWORKS.2023.NeM3B.3.

- **Development of an advanced Routing and Wavelength Assignment (RWA) algorithm within an SDN controller:** To address the architectural challenge of domain convergence where packet and optical teams operate with separate controllers, protocols, and operational practices, this thesis develops an advanced RWA algorithm implemented within an SDN controller. Unlike traditional approaches, which often overlook the full potential of converged packet-optical nodes, the proposed solution explicitly leverages the wavelength tunability of the *packetponder* to enable dynamic reconfiguration of optical resources, and hence improve network flexibility and resilience. Simulations on realistic topologies, emulated in mininet-optical, demonstrate that the controller can rapidly reconfigure lightpaths in response to faults, thereby reducing service disruption and improving resource utilisation. This work showcases how a centralised, open, and programmable framework can extract greater operational value from converged network nodes.
- **Experimental validation of the *packetponder* at the network edge:** To address the lack of a truly unified, open, and domain-spanning hardware platform, this thesis presents the experimental validation of the *packetponder* at the network edge. The demonstration evaluates the feasibility of using the same open, disaggregated platform for edge aggregation, integrating mixed traffic from the access domain through standard form-factor pluggables under an open software and API framework wherever possible. While certain components employed in the proof-of-concept are not yet fully open, the architecture is designed with the ultimate goal of achieving complete openness and centralised control. The system successfully aggregated and transmitted a 99.56 Gbit/s signal over 80 km of single-mode

fibre, validating the performance of the approach. By extending the *packet-ponder* concept beyond metro and core into the access/aggregation layers, this work demonstrates a concrete step towards an open, programmable, and truly end-to-end converged network architecture.

[JR3] J. Raulin et al., "Single Open Programmable Packetponder Element for Access/Metro Aggregation and Network Consolidation," 2024 International Conference on Optical Network Design and Modeling (ONDM), Madrid, Spain, 2024, ISBN: 978-3-903176-62-1.

[JR4] J. Raulin et al., "Simplification of Edge Network Architecture with Open Packet Switches and Pluggable Transceivers," 2024 IEEE Photonics Society Summer Topicals Meeting Series (SUM), Bridgetown, Barbados, 2024, pp. 1-2, doi: 10.1109/SUM60964.2024.10614540.

The foundations of this thesis rely on a set of key enabling technologies, including coherent pluggable transceivers, open-source software platforms, and standardised interfaces, which are discussed in more detail in Chapter 2. These technologies have not emerged simultaneously, as shown in Fig.1.2: some were already established prior to this work, while others have matured during the course of this PhD, reflecting the rapid and continuous evolution of the field. Overall, this evolution reflects a highly competitive, industry-driven field, where technological directions are largely defined by industrial players.

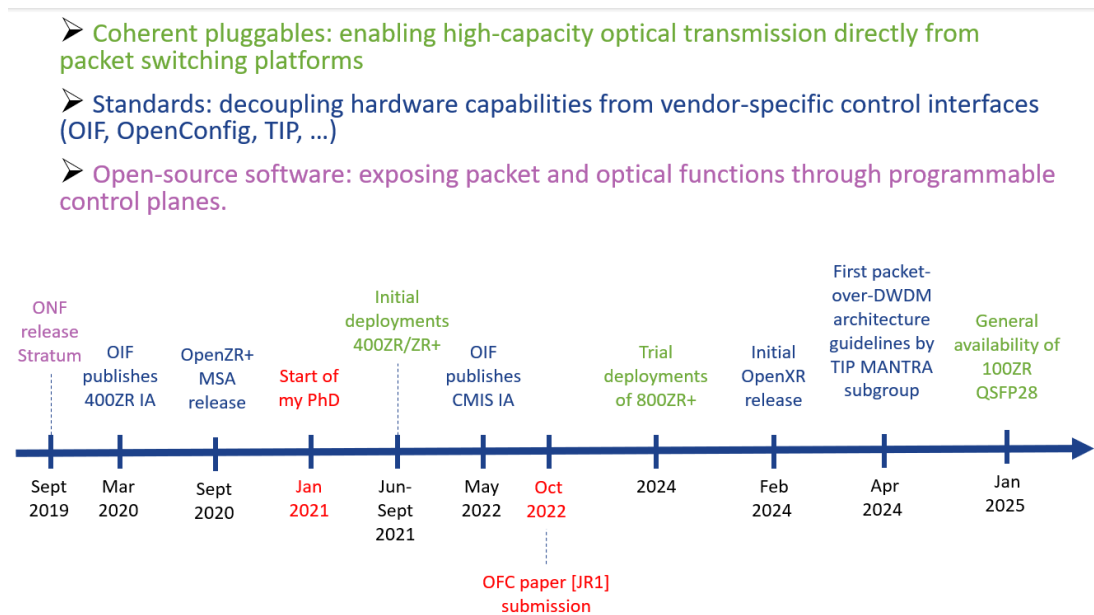


Figure 1.2: Development timeline of packet-over-DWDM technologies.

1.5 Thesis outline

In this chapter, we stated the motivation for this work with the main contributions, as well as the thesis outline.

Chapter 2 provides the foundational background for this work, exploring the SDN paradigm in both packet and optical networks. It delves into the evolution of IP-over-DWDM evolving to general packet-over-optics, covering advancements in both electrical and optical transmission technologies. This chapter also highlights important efforts toward interoperability to support packet-optical applications (IP-over-DWDM) that could reduce power consumption and improve network efficiency.

The following chapters constitute the core of this thesis, each combining the presentation of original contributions with a review of related work and relevant literature.

Chapter 3 focuses the integration challenges faced in today’s packet-optical networks, with the lack of NOS capable of integrating the coherent and tunable transceivers into a switch/router. It also addresses the lack of direct exposure of optical parameters to the packet controller, which limits full SDN control. To address these issues, the chapter introduces the *packetponder* concept and demonstrates how an off-the-shelf programmable whitebox, equipped with an open-source NOS and open APIs, can be enhanced to support and dynamically manage optical transceivers remotely. The chapter details the entire software stack, illustrating how a request from a remote controller is processed by the switch to modify the optical parameters of pluggable transceivers by accessing their registers. The experimental demonstration confirms a fast reconfiguration time for wavelength tuning, proving the seamless integration of optical transceivers into the switch.

Chapter 4 covers the management challenges from a controller perspective, when multiple control instances (packet and optical controllers) are attempting to access and manage pluggable transceivers. While the coordination of both controllers is out of the scope of this thesis, the chapter advocates for tighter integration between packet and optical controllers, particularly by incorporating wavelength conversion — a feature now feasible at minimal additional cost within the *packetponder* — into Routing and Wavelength Assignment (RWA) algorithms. The effectiveness of this approach is evaluated using mininet-optical, an emulation tool for packet and optical networks, demonstrating improved channel

availability during failure recovery scenarios. The evaluation includes multiple link failures scenarios across two different topologies, showing the *packetponder*'s advantages when intelligently managed. Additionally, the chapter highlights the value of emulation tools in scaling experiments and testing SDN controllers in realistic network conditions.

Chapter 5 explores key innovations in the access domain and the technologies driving edge and access network consolidation, particularly in optical networks. These advancements focus on simplifying infrastructure, reducing hardware equipment, and improving overall efficiency. The chapter then introduces the application of the *packetponder* at the network edge, proposing a novel architecture for the convergence of metro and access network services, including point-to-point leased lines, point-to-multipoint Passive Optical Networks (PONs), and 5G fronthaul transport. An experimental demonstration shows the *packetponder*'s ability to aggregate diverse access domain signals using common-form-factor pluggables and open software/APIs, successfully transmitting an aggregated 99.56 Gbit/s signal over 80 km of single-mode fibre. Finally, the chapter provides insights into the potential real-world impact of this solution, including a small analysis of energy savings in a large-scale deployment, using the BT network in the UK as a case study.

Chapter 6 concludes this thesis, with a summary of the thesis contributions and achievements of the work, and discusses possibilities for further research.

Chapter 2

Background

Chapter 2 introduces the concept of Software-Defined Networking (SDN) in both the packet and optical domains, and provides essential background on the evolution of IP-over-DWDM which is a fast evolving research to commercial implementation since the beginning of my PhD in 2021. Although SDN emerged in the late 2000s, advancements in technology have only recently made it a practical reality. The chapter also explores key efforts toward interoperability, which have contributed to multi-vendor IP-over-DWDM deployments.

2.1 Origins of optical communications

2.1.1 Historical milestones in visual signalling

While optical communication existed for millennia, the Chappe telegraph was the first structured (in the sense of creating a network) system capable of sending detailed, coded messages, making it the direct ancestor of modern optical telecommunication networks [18]. Invented by Claude Chappe, the first optical telegraph line was installed between Paris and Lille in the early 1790s. The system used relay towers, like the one shown in Fig. 2.1, positioned no more than 30 kilometres apart to ensure visibility through a telescope. Each tower featured a large horizontal bar with two smaller adjustable arms at either end. Using a gear and pulley mechanism, operators could position the arms at different angles, with each configuration representing a letter, number, syllable, symbol, or common word (which is equal to the baud rate). However, the operators of each

tower were not given access to the message key, and their role was simply to relay signals to the next station, which is commonly called "regeneration" today. Only those who possessed the codebook, could decode the messages and translate it into words. This encryption was a closely guarded military secret, as the system was initially dedicated exclusively to military and government communications. Despite its low effective baud rate — less than 1 symbol per second — the system enabled the transmission of entirely new, unplanned messages to distant locations, a groundbreaking advancement in long-distance optical communication.



Figure 2.1: Chappe optical telegraph in Saverne, France: last remaining vestige of the connection Paris/Strasbourg from 1798 to 1852.

2.1.2 Pioneering long-distance electrical networks

Following the optical telegraph was the electrical telegraph in the early 19th century. Early experiments by Francis Ronalds (1816) and later practical implementations by Samuel Morse and Alfred Vail (1830s-1840s) led to the widespread adoption of telegraphy by transmitting electrical signals over a wire cable encoded in Morse code, which allowed terrestrial communication at the speed of

light in the medium. Later, as the need for global communications grew, attention turned to undersea telegraph cables, with Valentia Island in Ireland, chosen as the starting point for the first transatlantic telegraph cable. Fig. 2.2(a) shows the landing point on Valentia Island from the initial attempt in 1857, which failed when the cable snapped. A second attempt in 1858 briefly succeeded, allowing the first transatlantic message to be sent, but the cable failed within weeks. Renewed efforts in 1865 also ended in failure, but in 1866, a new, stronger cable was successfully laid between Valentia Island, Ireland and Newfoundland, Canada, finally establishing a permanent telegraphic link between Europe and North America. The original cable, shown in Fig. 2.2(b), is now preserved in the Valentia Island museum. This revolutionised global communication, reducing message transmission times from weeks to a few minutes.



Figure 2.2: (a) Landing point in Valentia Island, Ireland, of the first transatlantic cable attempt; (b) Actual vestige of the 1866 cable exposed in the Valentia Island museum, that successfully linked Ireland and Canada.

2.1.3 Modern optical communication systems

As we explained in the last sections, in the early days of communications, networks were simple, dominated by point-to-point systems using a single technology, such as telegraph wires or early telephone lines. However, as global communication needs exploded, networks became far more complex due to growing traffic demands, diverse applications, and the need for specialised infrastructure. Today's Internet must handle massive information volumes and low latency, driven by streaming, cloud computing, and Internet of Things (IoT), where billions of devices — from smart sensors to autonomous vehicles — constantly exchange information. To manage this, networks have deliberately evolved into hierarchical structures leading to multi-layered architectures, with distinct roles

and devices for the current access, metro, and core networks as shown in Fig. 2.3.

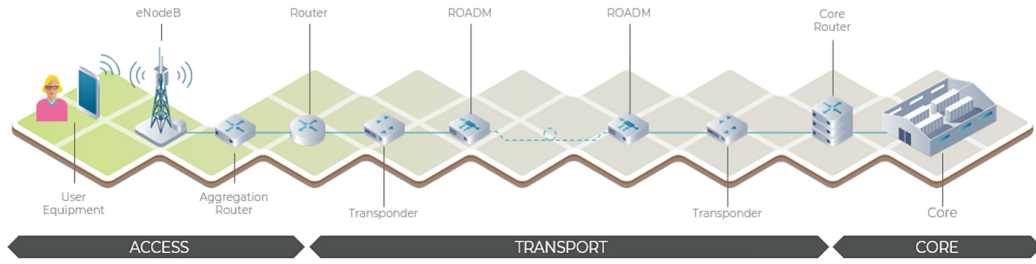


Figure 2.3: Current network topology from access to core. Figure extracted from [19].

In simple terms, the *access network* connects end-users and devices to the broader telecommunication infrastructure; the *metro network* (which could be referred to as the "transport" in Fig. 2.3) aggregates data traffic from multiple access networks and forwards it to the core; and the *core network*, often called the backbone, provides high-speed, long-haul connectivity across regions or countries [20]. Each of these layers requires different technologies and specific hardware (fibre optics, 5G devices, routers, switches, and optical transport systems) leading to a fragmented ecosystem of physical devices and control systems. This fragmentation, combined with continuous innovation and the need for scalability, has made modern networks incredibly complex compared to the simpler, single-technology origins of the Chappe optical telegraph.

Moreover, networks have become very tightly coupled with proprietary hardware and vendor-specific control software (which is the part of the system responsible for deciding how traffic should be handled and routed), mainly because vendors sought to differentiate their solutions, in addition to providing resilience, security, and customer support. When issues in the network arose, operators had to rely on its vendors to troubleshoot and fix problems, as the control and forwarding functions were tightly coupled within the hardware, often within thousands of lines of vendor-specific code embedded in the devices. This lack of openness led to vendor lock-in, slowing innovation and making network upgrades, troubleshooting, and scaling increasingly difficult [21]. It also made the management and control more complex and costly, albeit with significant customer support.

The need for a more open and flexible, software-driven approach to networking — where control and data planes could be separated, procured separately, and managed centrally — ultimately led to the development of SDN, enabling greater automation, programmability, and interoperability across competitive multi-vendor environments. The initial SDN movement for centralised control

systems and the use of whiteboxes appeared within data centres because traditional networks were rigid, making them poorly suited to the rapid scaling and dynamic requirements of modern cloud environments. The idea was then adapted for optical networks to achieve similar benefits.

2.2 The Software-Defined Networking paradigm

2.2.1 The concept

Software-Defined Networking (SDN) is a paradigm that separates the control plane (which makes decisions about where traffic should go) from the data plane (which forwards traffic based on a set of logical rules). Traditionally, networking devices such as routers and switches were closed proprietary systems where both planes were tightly integrated, leading to limited flexibility. In addition, SDN enables a centralised control architecture, using an SDN controller that programs the network dynamically via open, non-proprietary application programming interface (APIs).

The evolution of SDN began with early research projects such as Ethane [22], which explored centralised security and policy enforcement in networks. Ethane, developed at Stanford University, introduced the idea of a logically centralised controller managing forwarding rules in a switch, laying the foundations for SDN's control and data plane separation. This led to the creation of OpenFlow [23], the first widely adopted SDN protocol, which allowed controllers to program network devices via standardised flow forwarding rules. OpenFlow gained significant traction in academia and industry, demonstrating the power of centralised network control and dynamic policy enforcement.

A good description of the SDN layer architecture can be found in [24], illustrating the widely adopted concept and structure of an SDN network system. Figure 2.4 [24], represents the standard layers and planes that define the SDN architecture. It differentiates between: i) the *Forwarding Plane*, which is responsible for handling packets in the data path based on the instructions received from the control plane (forwarding, dropping, modifying, etc.); ii) the *Operational Plane*, which is responsible for managing the operational state of the network device, e.g., whether the device is active or inactive, the number of ports available, the status of each port, etc. ; iii) the *Control Plane*, which is responsible for making decisions on how packets are traversing the data path and how they should be

forwarded from one network node to another; iv) the *Management Plane*, which is responsible for monitoring, configuring, and maintaining network devices, e.g., making decisions regarding the resources of the hardware itself and the state of the network device; and v) the *Application Plane*, which is where network behaviours are defined.

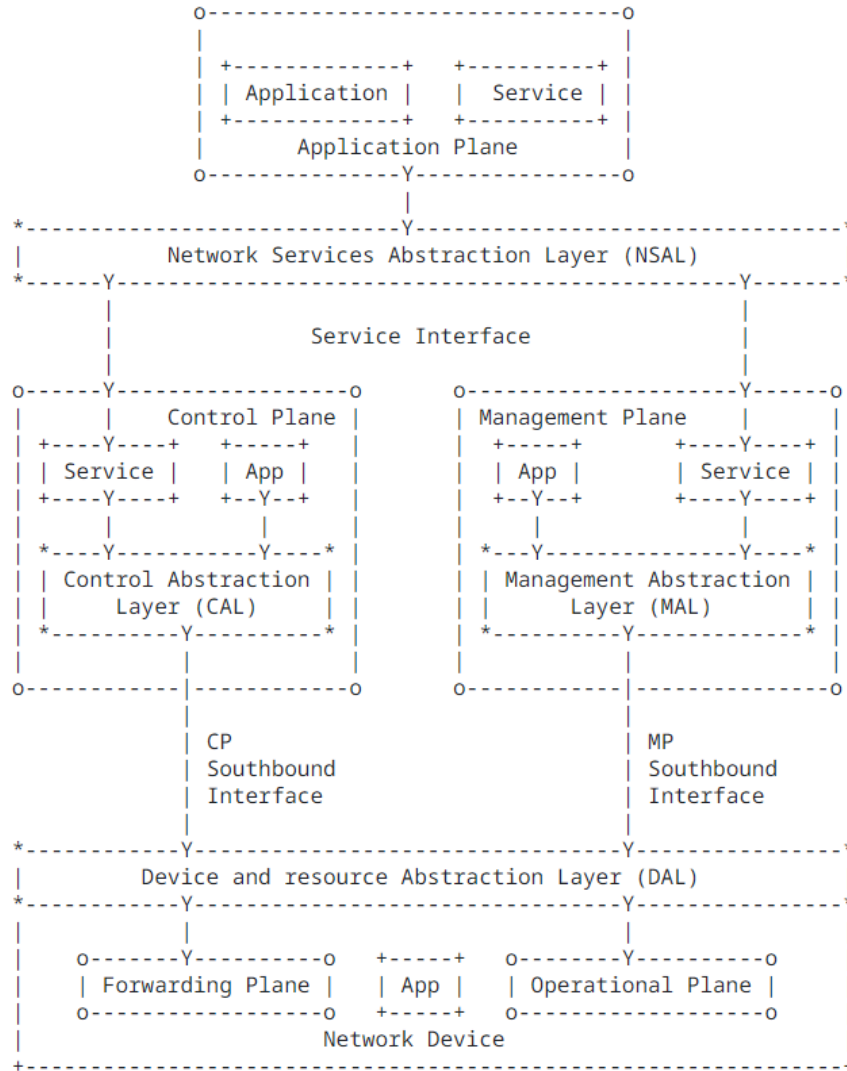


Figure 2.4: SDN layer architecture from [24].

While the Forwarding Plane and the Operational Plane serve different functions and are controlled through different mechanisms and protocols, they originate from the same physical hardware. Therefore, in this thesis, we will refer to both the Forwarding Plane and the Operational Plane as the "Data Plane", without distinguishing between them unless otherwise stated. Similarly, we will use the term "Control Plane" to encompass both the traditional Control Plane and Management Plane when discussing network functions responsible for traffic

forwarding decisions, monitoring, and configuration. However, when referring specifically to the system or software that implements the control logic in an SDN architecture — whether local (within the whitebox) or remote (centralised) — we will use the term "controller."

Communication between the control and data planes is achieved through various service interfaces, implemented using protocols tailored to specific functions such as traffic configuration and state management. Examples of such protocols include NETCONF, RESTCONF or gRPC [25]. These protocols enable the SDN controller to program and monitor network devices dynamically, reinforcing the separation of the control plane (decision-making logic) from the data plane (packet forwarding) while ensuring efficient and automated network management.

2.2.2 SDN whitebox solutions for the packet layer

In the context of Software-Defined Networks, a whitebox refers to a disaggregated, vendor-neutral network device such as a switch or a router, built with commodity hardware and running an open or third-party Network Operating System (NOS). Over the years, various whiteboxes have emerged, offering alternatives to proprietary hardware. Original Design Manufacturer (ODM) switches from companies such as Quanta, Accton, and Delta, running third-party NOS like Cumulus Linux, were among the first SDN-ready whitebox switches, an idea that was originally proposed by the company Big Switch Networks. Facebook's Wedge [26] then came along as part of the Open Compute Project (OCP) to enable SDN programmability and disaggregation. More recently, companies such as Edge-core Networks and Dell released more modern SDN-capable whitebox switches for open networking, as today, whiteboxes have become a must-have due to the increasing need for rapid innovation and fast deployment.

For even more programmability and flexibility into whiteboxes, Barefoot Tofino, now part of Intel, was a major breakthrough as the first to create an open P4 programmable (Programming Protocol-independent Packet Processors) switching ASIC, allowing deep customisation of packet processing. While traditional SDN whiteboxes enabled network programmability through open interfaces and protocols such as OpenFlow, their underlying hardware still restricted how packets were processed. Hence, unlike OpenFlow, P4 allowed developers to define how packets are processed at the hardware level, allowing operators to define custom packet-processing behaviour without waiting for fixed-function ASIC up-

dates. This made P4 a valuable feature into whiteboxes, allowing networks to evolve beyond predefined protocols and adapt dynamically to new requirements in areas such as network security, telemetry, and 5G slicing.

Despite the fact that Intel announced that they were going to halt development on Tofino in 2024 [27], meaning that whitebox solutions relying on the Tofino chip will no longer receive official support and no further development on a next-generation Tofino chip will be made, the core concept of programmable data planes remains highly relevant and "P4 itself is not going anywhere" [28]. P4 introduced an unprecedented level of flexibility in whiteboxes, and new chips supporting P4 are emerging. For example, Cisco's Silicon One ASICs [29] offer P4 programmability, though they remain proprietary with no open APIs, meaning that while programmability exists, it is restricted to Cisco's ecosystem rather than an open, vendor-neutral approach.

The P4 language remains active and continues to be developed and used in various networking projects. In fact, P4 has recently been made open-source, which is a significant win for the networking community. This ensures that developers, researchers, and vendors can continue innovating with programmable data planes. The open-source future of P4 means that the spirit of network programmability and whitebox flexibility will continue to evolve, potentially influencing new architectures and next-generation programmable ASICs beyond Tofino.

Another major player in this space is Broadcom, whose chips — Trident, Tomahawk, and Jericho — support SDN programmability through open APIs, despite not being P4-programmable. In 2019, Broadcom introduced its own network programming language, aiming to counter the growing competition from Barefoot Networks' P4 and reinforce its position in the programmable networking market.

In this thesis, the research used the Barefoot (Intel) Tofino switch to explore the packet-optical layer consolidation as per Chapter 3.

2.2.3 Advancements in open Control Planes

To fully support SDN whitebox switches, significant advancements have been made in open control plane software. Open SDN controllers come in various forms, differing in programming language, performance, ease of development, supported southbound interfaces, or intended use cases. Each controller is de-

signed with specific strengths, whether optimising for scalability, flexibility, security or ease of integration [30], [31]. Notable examples include ONOS (Open Network Operating System), OpenDaylight (ODL) and RYU, and have been developed as open-source SDN controllers capable of managing large-scale whitebox deployments, supporting a large set of interfaces such as OpenFlow or NETCONF. A comprehensive overview of the ONOS control plane is provided in [32], serving as a reference implementation for control plane architecture. The discussion highlights how the core abstractions can implement diverse control applications (Northbound interface, i.e., how it connects to the Application Plane) and also managing a broad range of network devices (Southbound interface and the connection to the Data Plane as shown at the bottom of Fig. 2.4 [24]).

Such control planes, as well as many other proprietary solutions, have pushed SDN whiteboxes further into mainstream adoption.

2.2.4 SDN paradigm for optical networks

The physical layer in optical networks is built on Optical Line Systems (OLSs), which typically consist of transponders (or line terminals) for optical signal transmission and reception, Reconfigurable Optical Add/Drop Multiplexers (ROADMs) with Wavelength-Selective Switches (WSSs) for signal multiplexing and demultiplexing, and Erbium Doped Fibre Amplifiers (EDFAs) for boosting, in-line amplification, and pre-amplification. These components are managed by an OLS control system, which is traditionally supplied by a single vendor or manufacturer. This vendor lock-in restricts network operators from integrating equipment from multiple vendors which limits operators' choice of equipment to a single supplier.

Hence, similar challenges faced in the packet domain have emerged in the optical domain, leading to the adoption of the same SDN principles (open disaggregation and centralised control). In the optical domain, the SDN paradigm focuses on dynamic wavelength and bandwidth allocation, enhancing the flexibility of optical transport networks [33].

Furthermore, traditional Wavelength Division Multiplexing (WDM) networks relied on static configurations, requiring manual provisioning for each new connection. With SDN-driven optical networks, the control plane can potentially dynamically adjust wavelengths, modulation, and paths based on network conditions and traffic demands.

2.2.5 SDN whitebox solutions for the physical layer

Several SDN whitebox solutions have been developed for transponders and ROADMs, enabling disaggregated optical networking. Voyager [34], an open optical transponder introduced by Facebook (now Meta) and Telecom Infra Project (TIP) [16], was one of the first efforts to bring SDN control to optical transport. Cassini [35], another TIP project, appeared the same year in 2016 and provided a modular, SDN-programmable transponder supporting both packet switching and optical transport, integrating routing solutions with its switching ASIC. More recently, the company Edge-core Networks has released the Phoenix Transponder/Muxponder AMX3200, a "disaggregated system to offer flexibility with different NOS options, [...] designed for building high-capacity optical transport networks" [36].

On the ROADM side, the Czech Light ROADM [37] is a fully open and programmable optical switching solution designed for research and flexible optical networking. Additionally, Lumentum's ROADM Graybox [38] solution provides vendor-agnostic optical switching while maintaining SDN-based control capabilities. These open optical solutions allow operators to move away from monolithic, vendor-locked WDM systems, enabling flexible, software-driven control of optical transport networks.

In this thesis, we used the aforementioned Lumentum ROADM Graybox to emulate a ROADM-based mesh network and create a packet-optical network.

2.2.6 Advancements in the optical Control Plane standards

To enable a control plane to manage optical devices, the first step is to identify a suitable control protocol to access the device. While early efforts attempted to extend OpenFlow for optical devices [39]–[41], it became clear that optical transport networks required more flexible configuration mechanisms than the OpenFlow's model [42]. As a result, the industry shifted towards using the NETCONF protocol with the YANG data modelling language for transponders and ROADMs control, allowing more granular configuration and telemetry retrieval [43], [44]. A more detailed discussion about unified information and YANG data modelling activities for an interoperable ecosystem is described in Section 2.4.1.

More recently, Campanella et al. [45] outlined the objectives and architectural framework of the Open and Disaggregated Transport Network (ODTN) [46] project by the Open Networking Foundation (ONF) [15]. This initiative lever-

ages and extends the ONOS SDN controller to control OLS (transponders and ROADMs) using NETCONF/RESTCONF protocols. In the same direction, the TIP Open Optical & Packet Transport (OOPT) [19] is also a project that promotes the development of disaggregated optical transport solutions, including whitebox transponders and open ROADM architectures, via open interfaces and APIs.

While SDN has been extensively explored in both the packet and optical domains, bringing flexibility, automation, and disaggregation to each layer, its application to the integration of packet and optical transport remains less mature. The concept of IP-over-DWDM predates SDN, but only recent advancements in SDN frameworks, standardisation efforts, and interoperable transceivers have made it more practical. The next section explores how SDN principles are being applied to IP-over-DWDM, enabling greater convergence between packet and optical networks.

2.3 IP-over-DWDM

To understand the reason why IP-over-DWDM was ever created, we need to understand how data traffic was historically carried through the network.

2.3.1 Transport technologies evolution

SONET/SDH

Optical transport networks emerged with Synchronous Optical Networking and Synchronous Digital Hierarchy (SONET/SDH), standardised in the late 1980s and widely deployed in the 1990s for voice-centric applications. SONET/SDH mapped Time-Division Multiplexing (TDM)-encoded voice directly onto fiber optical transport. Then packet-switched data services came along that did not behave like voice. Still, SONET/SDH was able to carry multiple data protocols, like Asynchronous Transfer Mode (ATM) or Ethernet, by converting data from different sources and protocols into a common format for transmission over the network (packet-over-SDH schemes). It could also directly extract low-speed signals from multiplexed high-speed traffic, thereby increasing the efficiency and utilisation of the network. The transport oriented features and the capability to carry multiple protocol types, very quickly made SONET/SDH the de facto standard for voice, data and streaming media applications. However, the global demand for connectivity that increased consistently afterward proved SONET

to be inefficient and expensive due to its fixed frame rate and restrictive ring topologies.

OTN

Eventually, optical transport transitioned from SONET/SDH to optical transport network (OTN) a.k.a. “digital wrappers” to support an array of cell-, frame-, or packet-based Layer 2/Layer 3 (L2/L3) clients that included ATM, Frame Relay and Ethernet [47]. Unlike its predecessor SONET/SDH, OTN was created as a carrier grade technology to work in a multi-vendor and multi domain environment with enhanced transparency through the digital wrapper, and enhances Operations, Administration and Maintenance (OAM) with end-to-end monitoring features [48]. OTN also provided greater reach, increased scalability and better error correction abilities as it supported Forward Error Correction (FEC). As the time progressed, OTN supported higher bit rate (40 Gbit/s and 100 Gbit/s) wrapper specifications and also adopted more granularity [introduction of Optical Channel Data Unit (ODU); ODU0 and ODU Flex] which, overall, made OTN more flexible [49]. Originally a point-to-point transport technique which supported the ring topologies of SDH/SONET, the introduction of ODUs also added switching capability OTN cross-connects (XCs) devices and the ability to support mesh network topologies.

However Ethernet continued to scale, dominated commercially and eventually prevailed. So OTN, with generic framing procedure (GFP), was extended to encapsulate/multiplex Ethernet client frames too [50]. In practice, fixed-port linecards incorporating OTN framers, surrounded a fixed-cell “universal switching” fabric to form OTN XCs [51].

Ethernet

At this point the Ethernet/IP community strongly opposed this approach of encapsulating Ethernet packets into OTN frames. Their service networks were built on fundamentally different principles, and they were frustrated by the need to rely on a separate network — and even worse, a separate organisation — for cost-effective optical transport [52]. This reflected the longstanding Nethead vs. Bellhead division: the Netheads favoured decentralised, packet-based networks (e.g. Ethernet, IP, TCP) while the Bellhead (Bell Labs, teclos) favoured centralised control, circuit-switched networks and invested into SONET/SDH. Consequently, this drove advancements in transforming Ethernet into a carrier-

grade technology.

This thought was also pushed by the fact that all fixed-line data traffic originates and terminates as Ethernet frames in home routers and data centre servers alike. Ethernet has become the dominant *de facto* link-layer technology, and is the Layer 2 underlay in IP-based networks. Ethernet deployment is likely unsuitable for the support of cloud-services within hyperscale data centres; and end-to-end services in next generation networks for telecommunications carriers. Hence, attempts were made to extend the use of Ethernet technologies beyond the traditional any-to-any connectivity that was confined to local area and access networks, most notably with Provider Backbone Bridging Traffic Engineering (PBB-TE). This was the first attempt to adopt Ethernet as an alternative choice as a carrier grade technology for very wide area transport networks [53]. PBB-TE [IEEE 802.1Qay-2009], in practice, re-purposed a standard Ethernet switch chassis to support point-to-point, MAC-in-MAC multiplexing/encapsulation across an Ethernet switch fabric. PBB-TE disabled spanning tree protocol, address learning and flooding to provide connection-oriented, point-to-point deterministic forwarding behaviour offering circuit-like transport services akin to OTN.

Although PBB-TE has not been adopted by TELCOs, it showed the potential of using Layer 2 Ethernet to support multiple, higher-layer services (packet and circuit services) while being a low-cost and service friendly technology [54]. This is paving the way, among other aspects, to the concept and feasibility of IP-over-DWDM, where OTN transponders would be integrated directly into Layer 3 (L3) routers and Layer 2 (L2) switches, leveraging OTN's optimised transmission capabilities while discarding its multiservice and electrical switching functions. It also showed how Ethernet switching ASIC technologies can be harnessed, especially for OTN cross-connect, as SERDES (Serializer/Deserializer) speeds from 100Gbit/s and above.

2.3.2 Origins of the IP-over-DWDM concept

IP-over-DWDM (IPoDWDM) was introduced in the late 1990s as a response to the rapid growth of Internet traffic and the inefficiencies of traditional transport networks. As we just discussed, at the time, the standard approach for carrying IP traffic over optical networks involved multiple intermediate layers, such as SONET/SDH or OTN, which added latency, complexity, and cost. The idea behind IPoDWDM was to bypass these intermediate layers and run IP traffic directly over optical wavelengths, simplifying the network architecture and improving

bandwidth efficiency.

Another driving factor was the increasing demand for high-speed data transmission due to the rise of the World Wide Web, data centres, and early cloud computing. Service providers needed a way to scale their networks efficiently while reducing the operational and capital expenses associated with legacy transport systems. Directly mapping IP onto DWDM wavelengths promised higher throughput, reduced hardware costs, and more flexible network management. Early discussions and implementations of IPoDWDM can be traced back to this period, with companies such as Cisco dominating the integration of IP and optical technologies.

Despite its potential benefits, early adoption of IPoDWDM faced significant challenges. The technology of the time required large, power-hungry optical components, which, when integrated into IP routers, led to issues with port density and increased costs. Additionally, the lack of standardisation and interoperability between vendors hindered widespread implementation. The turning point came with advancements in silicon photonics and the miniaturisation of optical components, often attributed to Moore's Law. These developments led to the creation of compact, energy-efficient pluggable coherent optics with standard form factor, such as the 400ZR modules, which could be directly inserted into router interfaces. This innovation addressed previous limitations, making IPoDWDM a viable and attractive solution for modern high-capacity networks.

2.3.3 The evolution of data transmission systems

The evolution of IPoDWDM was driven by significant hardware and software advancements. Electrical improvements were related to router/switch ASICs, SERDES, and Ethernet speeds (electrical side); while optical improvements were related to coherent optics, modulation formats (optical side), and Digital Signal Processors (DSP).

2.3.3.1 SERDES evolution

At the core of high-speed electrical signalling in routers, switches, and pluggable optics lies SERDES technology. Initially, ASICs and electrical interfaces relied on 10 Gbit/s SERDES with simple Non-Return-to-Zero (NRZ) modulation (1 bit/symbol). This required 10×10 Gbit/s lanes to support a 100 Gbit/s optical signal, leading to large C form-factor pluggable (CFP) in the past. The transition

to 25 Gbit/s SERDES (still using NRZ) enabled 4 lanes at 25 Gbit/s to enable 100 Gbit/s, paving the way for the more compact and widely used QSFP28 form factor. More recently, the adoption of PAM4 modulation (2 bits/symbol) has significantly increased lane capacity, allowing 25 Gbaud PAM4 SERDES (50 Gbit/s per lane) to drive 400 Gbit/s optics using 8 lanes. This typically requires double-density form factors such as QSFP-DD or OSFP. The latest generation features 100 Gbit/s PAM4 SERDES (50 Gbaud), enabling 800G Ethernet (e.g., 8×100 Gbit/s PAM4 lanes per transceiver) and even 1.6 TbE with 16 lanes. Looking ahead, 224 Gbit/s PAM4 SERDES is expected to support 3.2 TbE and beyond. A summary table is provided in the Appendix, mapping transceiver technologies to their typical SERDES lane configurations and speeds.

The evolution of SERDES is closely tied to ASIC advancements, as these chips determine the total switching capacity of a router or switch. Modern ASICs — such as Broadcom’s Tomahawk 5, Cisco’s Silicon One, and Intel’s Tofino — now support multi-terabit per second (Tbit/s) processing, making them critical enablers of high-speed network architectures.

2.3.3.2 Optical transmission evolution with coherent pluggables

The evolution of optical transmission in the Dense Wavelength Division Multiplexing (DWDM) context has been driven by advancements in modulation formats and baud rates, enabling ever-increasing line-side data rates. At the start, DWDM systems relied on NRZ modulation (1 bit/symbol), limiting per-channel data rates to 10 Gbit/s per wavelength in early deployments. To increase capacity without consuming excessive spectrum, the industry transitioned to coherent detection in the early 2010s, allowing for advanced modulation formats like Dual-Polarisation Quadrature Phase-Shift Keying (DP-QPSK). With DP-QPSK, each symbol carried 4 bits, enabling 100 Gbit/s per wavelength.

Further innovations led to higher-order modulation formats such as 16QAM (4 bits per symbol) and 64QAM (6 bits per symbol), significantly improving spectral efficiency. Higher modulation formats means higher data rates, however, they required increasingly sophisticated DSP to mitigate transmission impairments, as higher modulation introduce greater sensitivity to noise and signal distortions.

As the industry moves towards ultra-high-speed transmission rates such as 800 Gbit/s, 1.6 Tbit/s, and beyond, simply increasing modulation complexity is not always feasible. Instead, increasing the baud rate (the number of symbols transmitted per second) has emerged as a more balanced approach to achieving higher

capacities while keeping modulation complexity manageable. However, higher baud rate means that more spectrum per wavelength is required, leading to challenges in DWDM channel spacing and optical filter design. Consequently, components with wider bandwidth such as tunable lasers across the full optical C-band and photodetectors with broader electrical bandwidths, are being suggested to support higher baud rate transmission signals [55].

2.3.3.3 The miniaturisation of Digital Signal Processors enabling power-efficient pluggables

The electrical host side and optical line side of a transceiver are tightly coupled, as SERDES speeds on the electrical side directly influences the evolution of pluggable optics, allowing higher transmission capacities as per section 2.3.3.2 and 2.3.3.3. The relationship between both sides is governed by the DSP, which maps the electrical input into an optical signal while compensating for impairments such as dispersion, noise, and sometimes non-linearities.

The evolution of DSP has been significant in the integration of packet and optical layers, largely driven by Moore's Law and advances in semiconductor manufacturing. Early coherent DSP, introduced in the 2010s, were large, power-hungry, and typically found in bulkier transponders or line cards within big transport chassis. Consequently, their high power consumption and size made them impractical for direct router integration. Over time, the semiconductor industry reduced DSP process nodes from 90nm to 7nm and below, leading to a drastic reduction in power consumption, heat dissipation, and physical footprint [56].

This miniaturisation allowed coherent DSP to be integrated into pluggable modules of smaller common form factor such as Quad Small Form-factor Pluggable Double Density (QSFP-DD) or Octal Small Form-factor Pluggable (OSFP), which could fit directly into modern router and switch port cages. As a result, IP-over-DWDM became more viable, eliminating the need for external transponders and enabling direct integration of optical transmission into IP routers.

Smaller DSP ASICs have enabled support for higher baud rates (e.g., 120+ GBaud per channel) and more sophisticated signal processing algorithms, enhancing performance while maintaining power efficiency. Higher baud rates require processing more symbols per second, which increases the computational load on the DSP. Advancements in transistor miniaturisation allow for higher clock frequencies and greater parallelism, making it possible for DSP to keep up with these demands.

2.3.4 Power savings and economical benefits

Since 2021-2022, the adoption of coherent pluggables (e.g., 400ZR/ZR+) in optical network architectures has led to significant power consumption savings compared to traditional transponder-based or muxponder-based transport networks. In traditional architectures, separate DWDM transponders or muxponders housed in big chassis are required to convert client-side data traffic into an optical signal for line-side transmission. These standalone transport platforms consume high power per transmitted bit, often in the range of 20–40 W per 100 Gbit/s. In contrast, coherent pluggables integrate the DSP and optics within the pluggable package, eliminating the need for external transport equipment and reducing power consumption to 5–10 W per 100 Gbit/s — a power saving of up to 50% [57], depending on deployment conditions.

However, the actual power savings vary based on network architecture [58]. In a hop-by-hop (also called opaque) IPoDWDM deployment, where optical-electrical-optical (OEO) conversion occurs at every intermediate node, the power savings may be modest, as two "peer"¹ pluggables are required per hop [59]. In a fully transparent IPoDWDM architecture using ROADMs-based optical transport, power savings are maximised because signals can bypass intermediate routers, eliminating unnecessary OEO conversions. Indeed, the authors in [60] and [61] indicate that as bandwidth demands increase, "optical express"/transparent architectures become more cost-effective than hop-by-hop/opaque approaches. However, since optical signals are subject to relevant physical layer impairments when traversing core network devices, signal regeneration is often required, which has to be accomplished either at the electronic layer by routers/digital cross connects or directly at the optical layer through 3R-regenerators. If regeneration is needed, the power efficiency depends on the number of regenerations and whether routers handle the regeneration more efficiently than the optical transport devices [58], [62]. Also, factors such as network topology, reach requirements, regeneration needs, and traffic patterns play a crucial role in determining the overall energy efficiency gains.

However, deploying transponders in ROADMs-based networks has one advantage as it creates a clear separation between switches/routers and optical transmission, simplifying operations and allowing transmission vendors to guarantee optical reach. This demarcation is also advantageous when offering external optical

¹"peer" pluggable means two pluggables with the same specifications at the transmitter and receiver.

services, as it provides a well-defined boundary for service agreements [61]. Also, eventually integrating 400ZR+ transceivers directly into routers requires multi-vendor agreements to establish optical reach expectations and necessitates robust operational tools for network planning and troubleshooting. As networks evolve to 800 Gbit/s and beyond, similarly complex architectural trade-offs will need to be addressed.

Although the concept of IP-over-DWDM existed over 30 years, with early attempts in the past, it is only since the 2020s that it is becoming a reality due to availability of advanced switches and pluggables capable of handling physical layer demands and layer 2/3 demands. As such, it may now be more appropriate to speak more broadly of "packet-over-DWDM" or even "packet-over-optics" to reflect the wider range of supported protocols beyond just IP.

2.4 Interoperability

2.4.1 YANG models for device and configuration standardisation

Yet Another Next Generation (YANG) is a data modelling language used for defining and describing the structure and behaviour of network devices and configurations in a standardised way. It was developed by the Internet Engineering Task Force (IETF) to provide a vendor-neutral and human-readable format for network management.

YANG models represent data in a hierarchical tree structure, similar to how XML or JSON structures data. This tree consists of nodes, which define various aspects of the element being modelled. In a simplified way, the tree is composed of several container nodes (high-level nodes) which groups related data together (e.g., a container for all interface configurations) and at the end of those are "leaf nodes" which represents individual data point (e.g., an IP address). These are the lowest-level elements and cannot contain other nodes.

To interact with network devices using YANG-defined data, network management protocols such as NETCONF, RESTCONF, and gNMI are used. A YANG data model is not tied to any particular protocol or encoding method as shown in Table 2.1. These protocols define how YANG-modelled data are sent, retrieved, modified, and managed. Each protocol supports different encoding formats, which define how the data is serialised for transmission. The common encoding

formats include XML (Extensible Markup Language); JSON (JavaScript Object Notation); and Protobuf (Protocol Buffers), and the choice of protocol depends on the device and applications: while NETCONF is mostly used for optical components, gNMI and RESTFUL are more used for IP layer equipment.

Table 2.1: YANG encoding method with NETCONF, RESTCONF and gNMI protocol [63].

Protocol	NETCONF	RESTCONF	gNMI
YANG data encoding	XML	JSON/XML	protobuf/JSON

2.4.1.1 Optical domain YANG models

Several groups and Multi-Source Agreements (MSA) are defining common YANG models to be used by every vendor for devices that serve the same network function. In the optical domain, OpenROADM [64] is one of the key MSAs enabling interoperability between transponders, ROADMs, and optical line systems from different vendors. OpenROADM defines not only hardware interoperability but also YANG models for SDN-based control of optical networks, allowing seamless multi-vendor integration in disaggregated optical transport networks, with other major initiatives including the TIP OOPT group.

2.4.1.2 IP domain YANG models

In the IP layer, the equivalent of OpenROADM is OpenConfig [65]. OpenConfig is a vendor-neutral initiative defining YANG models for IP and transport network elements, including routers, switches, and optical transceivers. It enables SDN-based control of IP networks, providing a standardised way to configure and monitor routing protocols, Border Gateway Protocol (BGP), Multiprotocol label switching (MPLS), interfaces, and more.

2.4.1.3 Packet-optical multi-domain YANG models

It is evident that the same way it applies to optical and packet networks, YANG models can be usefully used for packet-over-optics architecture and devices. Indeed, they play a crucial role in enabling vendor interoperability by providing a standardised way to model and manage coherent pluggables. As packet-over-optics deployments grow, network operators need automated, scalable, and vendor-neutral ways to configure, monitor, and troubleshoot coherent pluggables directly from routers or centralised controllers.

While OpenROADM and OpenConfig provide well-defined YANG models for the optical and packet layers, respectively, an open question remains: which YANG model should be used to define packet-optical nodes with pluggables? These nodes, which integrate both packet and optical transport functions, require a unified approach for configuration and management, as existing models largely remain layer-specific. Industry bodies and operators are still working toward a solution that enables end-to-end programmability, telemetry, and control across both domains in a vendor-neutral way, and will be discussed in more detailed in Chapter 3.

YANG primarily facilitates remote device management by enabling a single controller to manage devices from multiple vendors without requiring vendor-specific adaptations. This plays a crucial role in reducing vendor lock-in. However, achieving full interoperability also requires devices from different vendors to communicate seamlessly with each other — especially pluggable transceivers as they are at both end of a transmission. Addressing this challenge requires collaborative efforts from both academia and industry to define, develop and adopt useful standards.

2.4.2 Packet-over-optics standardisation efforts before 2021

Interoperability between vendors is a key requirement for the successful deployment of packet-over-optics architectures using tunable and coherent pluggables within routers and switches. While the need for interoperability became especially clear with the emergence of coherent pluggables, these transceivers only became available and widely deployed after I started my PhD as shown in Fig. 2.5.

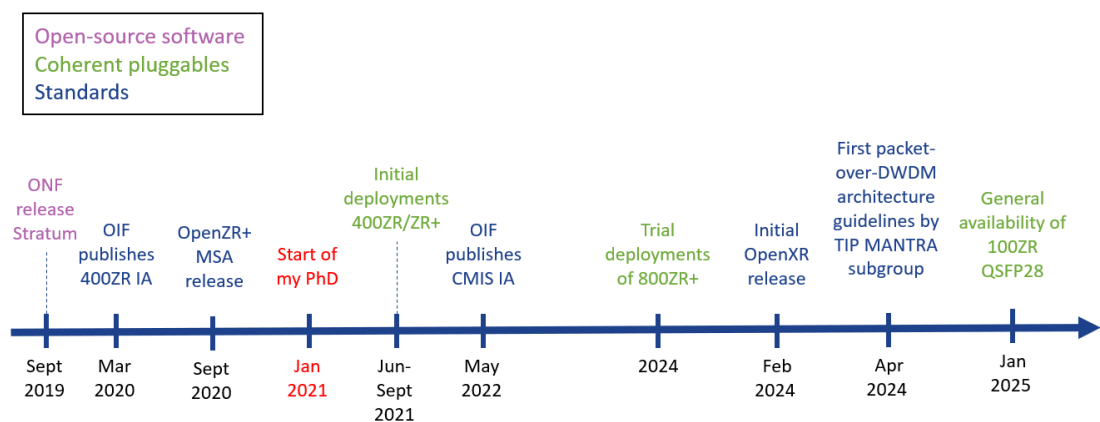


Figure 2.5: Timeline of packet-over-DWDM developments during my PhD research.

Indeed, at the beginning of my PhD in 2021, coherent pluggable transceivers such as the 400ZR were still in their early stage and had not yet been widely deployed in operational networks. At that time, operators were largely relying on fixed-wavelength or tunable pluggable transceivers, typically in SFP/SFP+ or QSFP28 form factors supporting up to 100 Gbit/s. These transceivers adhered to the SFF (Small Form Factor) MSA, as the Common Management Interface Specification (CMIS) implementation agreement [66] was not yet fully established, and in any case, CMIS was specifically designed for coherent pluggables and more advanced modules, which were only just beginning to emerge.

At that time, the optical and packet layers were still handled separately, both in terms of hardware and software control. While QSFP28 was the predominant pluggable form factor for switches, coherent optics in transponders typically used the CFP2 form factor. Standardisation efforts, such as those led by the TIP, focused mainly on enabling interoperability for coherent CFP2 pluggables within traditional transponder boxes, rather than switches or routers. Consequently, although important progress was being made in disaggregating the optical transport layer, these advances did not immediately translate to converged packet-optical systems. As a result, bridging that gap required us to work with the tools that were available (SFP/SFP+ modules), and to focus on the integration of SFF pluggables into programmable switches (that were originally designed for Layer 2/3 packet switching only) as a first step toward packet-over-optics convergence.

2.4.3 Interoperability developments from 2022 onwards

Since then, the area has received enormous attention, and coherent pluggables have become a central enabler for packet-over-optics architectures. Industry efforts have rapidly matured to not only deploy coherent pluggables at scale, but also to standardise their integration into routers and switches.

With the emergence of 400Gbit/s pluggables in QSFP-DD/OSFP form factors, the 400ZR [17], and OpenZR+ [67] have been established. The 400ZR implementation agreement, which was the first effort for coherent pluggables to promote interoperability among vendors and characterise a common interface, was created by the Optical Internetworking Forum (OIF). 400ZR targets the edge and relatively short reach, up to 120km Data Center Interconnect (DCI) applications with amplification, and 80km without amplification, but does not support deployment over ROADM networks. With the increasing demand to extend reach capabilities and the enhanced performance achieved through pluggable trans-

ceivers, 400ZR+ pluggable appeared with higher performance, notably using better open Forward Error Correction (oFEC) and greater tolerance to chromatic dispersion [61] and non-linearities. The OpenZR+ MSA [67] was then introduced in September 2020. This agreement supports link lengths of up to 500km, addressing the evolving requirements of inter-city network infrastructure. Currently, 400ZR and OpenZR+ specifications support Ethernet clients only, but some "ZR+" pluggables comply with the OpenROADM standard and allow not only Ethernet clients but also OTN transport technologies to be supported over ROADM mesh networks. Often, those pluggables are closed and include a proprietary FEC and/or vendor specific DSP features that do not lend themselves to multi-vendor interoperability, such as the newly announced 400G QSFP-DD High-Power (Bright) Optical Module [68]. Proprietary FEC schemes contribute to vendor lock-in because transceivers from different vendors cannot interoperate unless they support the same FEC encoding and decoding. Sometimes, this problem exists even with the same vendor but with different transponders! Since FEC is essential for correcting errors in high-speed optical links, a mismatch prevents seamless communication. This is why open and standardised FECs are critical for building open, disaggregated networks, where components from different vendors must interoperate reliably. Additionally, the OIF continues to work on next-generation interoperability standards for 800G and beyond, ensuring that coherent pluggables can interoperate across diverse network environments.

Similarly, another industry consortium called OpenXR Forum [69] was created to focus on driving the adoption and interoperability of XR (Extended Reach) coherent pluggable optics, particularly in metro, regional, and long-haul networks. XR optics are coherent pluggable transceivers that use point-to-multipoint optical networking instead of traditional point-to-point solutions.

In addition to interoperability, a unified software driver for managing pluggables from different vendors is essential. The OIF's CMIS [66] addresses this by defining a standard protocol over the I²C bus for communication between host devices and coherent pluggables. CMIS enables NOS integration and gives operators visibility into physical layer parameters.

2.5 Conclusion

The world has acknowledged the need for Software-Defined Networking in both the packet and the optical domain, and a large amount of effort is being under-

taken to create a standardised architecture or model to allow Software-Defined packet-over-DWDM networks to flourish. Indeed, the future for packet-over-optics is more promising than ever, far beyond the limited traction it initially gained when first introduced. For this, global consortia have emerged, such as 400ZR, OpenZR+ or OpenXR, to promote interoperability between vendor and allow control planes to manage a wide variety of pluggables without having to change the main core functions. In recent years, record-breaking live demonstrations have showcased multi vendor, interoperable packet-optical solutions at major conferences, such as the Optical Fiber Communication (OFC) conference and the European Conference on Optical Communication (ECOC). Notably, at OFC 2024, the OIF's interoperability demonstration brought together forty-seven member companies, highlighting cutting-edge advancements in interoperability but also in speed, power efficiency, and density to address the increasing demands of next-generation data centre networking [70]. This reflects the current direction of the industry.

However, key elements such as the pluggable integration into switches and their control still need to be investigated. In this thesis, we will investigate how to effectively merge the packet and optical layers by integrating tunable and coherent pluggables directly into switches, enabling remote and centralised control through standardised, vendor-agnostic APIs (Chapter 3). This chapter highlights the crucial role of standardisation bodies that we just described, and demonstrates how open-source software plays a crucial role to support the seamless integration of optical transceivers into programmable switches.

Beyond physical integration, this thesis also explores the architectural implications of packet-over-DWDM and the operational benefits it offers. While power savings have already been outlined in Section 2.3.4, Chapter 4 focuses on how this convergence improves flexibility and resilience (e.g., in fault recovery scenarios) by allowing dynamic reconfiguration across layers and domains.

Although packet-over-optics was initially developed for point-to-point data centre interconnects, it is now being adopted in more complex metro and core networks to reduce hardware complexity and improve scalability. This same philosophy can now be extended to the access domain, where multiple legacy systems and service-specific devices can potentially be replaced by a simplified, pluggable-based architecture closer to the edge. Chapter 5 explores this emerging use case, demonstrating how coherent and access pluggables can be integrated into a unified platform to consolidate access, aggregation, and metro functions.

Chapter 3

Packet-optical convergence with *packetponders*

Chapter 3 introduces packet-optical convergence with our novel concept: the packet-ponder. This chapter illustrates a seamless integration of pluggable transceivers into an open-source Network Operating System (NOS) and highlights how the frequency parameters of tunable modules can be made accessible to a remote, open control plane, thereby improving the flexibility and effectiveness of managing packet-over-DWDM networks. This chapter is based on published work at Optical Fiber Conference (OFC) 2023 [JR1] and Optical Advanced Photonics Congress (APC) 2023 [JR2].

3.1 Challenges in pluggable optics integration and management

As just seen in Chapter 2, packet-over-optics is increasingly becoming a reality, driven by advancements in pluggable optics. However, managing a switch integrated with pluggable transceivers presents significant challenges that have drawn attention from the networking community in recent years. Beyond the issues of controller coordination — which will be discussed in detail in Chapter 4 — one major challenge is the integration of pluggable optics within the switch [71]. Traditional Layer 2 switches are not designed to incorporate Layer 1 functionalities into their control frameworks, creating a gap that needs to be addressed for seamless operation.

Transponders, which still currently play a predominant role in packet-optical network architectures, have garnered significant attention in advancing open management solutions. The Open Optical and Packet Transport (OOPT) [19] group released in 2017 their first specification for transponders: the transponder abstraction interface (TAI) [72], that provided independent, uniform, multi-vendor control of pluggable transceivers (CFP2/CFP4/CFP8 form factor). Community group such as the Telecom Infra Project (TIP) [16] or the Open Networking Foundation (ONF) [15] were actively focused on implementing pluggable optics as part of the TAI architecture in Transport Network Operating Systems (NOS) such as Goldstone [73] or Stratum [74]. Cassini, from Edgcore Networks; Voyager, from Facebook; and Galileo, from Wistron; are three examples of transponders where TAI was implemented and enabled a vast set of layer-1 performance monitoring metrics and provisioning parameters to be exposed to the controller. However, TAI is not implemented into open, disaggregated packet-optical L2 switches and NOS have historically lacked support for implementing flexible tunable/coherent pluggables within switches.

To achieve this, an open switch NOS must be enhanced to support pluggable transceivers. It also needs to facilitate seamless interoperability between the switch and pluggables from different vendors. For this to work, the pluggables must adhere to common standards, enabling the host operating system (OS) to access pluggable information through generic, modular functions. We discussed in Chapter 2 the OIF [75] effort introducing the Common Management Interface Specification (CMIS) implementation agreement [66], which is the key driver that should facilitate the implementation of different vendors' pluggables into NOS and provide visibility of the physical layer parameters required by operators, as it helps to pass instructions between the module and the host.

At the time of this work, the only available open NOS for disaggregated L2 switches that could integrate the pluggable optics, due to their open-source nature, modularity and support for open APIs; were Stratum and SONiC [76]. A first attempt to control pluggable modules within packet-optical nodes was made in [77] using an enhanced version of the open-source SONiC NOS, but the experiment was using an external muxponder with 100 Gbit/s coherent line cards *in lieu* of pluggable coherent module inside the switch. Since then, the authors of this work have integrated some optical parameters of the coherent pluggable optics and a comparison of our two methods will be discussed in section 3.7. CMIS was used in [77], but CMIS can only apply to newer coherent pluggables such as 400ZR/ZR+ or more recently to the 100ZR, but it does not

support legacy tunable transceivers with lower data rate. At the start of this thesis, as coherent pluggables were not yet commercially available, the goal was to integrate 10 Gbit/s and 25 Gbit/s tunable transceivers; not only because they were more accessible, but also because they remained widely deployed in access and metro networks and to ensure backward compatibility.

Efforts were underway, but, to the best of our knowledge, there was no open NOS for open switches and routers that fully incorporated pluggable optics parameters in a manner similar to transponders, which could be exposed to any controller via open interfaces and models such as OpenConfig [65] and YANG [78]. This highlighted the need for a fully open and disaggregated platform that is flexible enough to support any form-factor transceiver management with a plug-and-play approach in a multi-vendor environment.

3.2 Packetponder concept with common form factor pluggables

3.2.1 Concept and name

This thesis defines the concept of the *packetponder* [JR1] as an off-the-shelf open, programmable and disaggregated Ethernet whitebox switch capable of hosting a wide range of standard form-factor pluggable transceivers, thereby integrating packet and optical functionalities within a single platform. Innovations such as P4 (Programming Protocol-independent Packet Processors), while not extensively researched in this thesis, enable flexibility in packet handling and is consequently a useful feature that compose the *packetponder*, as depicted in Fig. 3.1. Beyond its hardware realisation, the *packetponder* embodies a design paradigm in which openness, programmability, and adherence to standard interfaces enable unified and vendor-agnostic control of both packet and optical resources. By exposing these capabilities through a centralised SDN control framework, the *packetponder* allows flexible configuration and dynamic reconfiguration of network functions, including optical parameters such as wavelength selection and conversion, while supporting multi-vendor interoperability. Leveraging open-source Network Operating Systems and control frameworks, it can be seamlessly integrated into existing and future network architectures, enabling both point-to-point deployments and more complex mesh topologies, and providing a foundation for more flexible, efficient, and simplified network operation.

In essence, the *packetponder* combines disaggregated and programmable hardware, open and standardised software interfaces, and common form-factor optical pluggable transceivers, enabling a unified and flexible integration of packet and optical functionalities (Fig. 3.1).

We chose to name our solution a *packetponder*, although others might refer to such a device as a "IP-over-DWDM switch". The distinction lies in two key areas: first, our *packetponder* is fully programmable and open, using open APIs, as opposed to most IP-over-DWDM switches with pluggables, which are typically proprietary. Second, the term *packetponder* reflects its ability to perform packet-over-DWDM functionality: thanks to its P4-programmable ASIC, the *packetponder* can process and forward packets based on an extensive range of header fields, as opposed to IP-over-DWDM switches, which suggests forwarding decisions based on IP headers alone. While "IP-over-DWDM" is a well-known and widely adopted concept, we believe the name *packetponder* better captures the enhanced functionality and flexibility of our solution, which extends beyond the traditional scope of IP-over-DWDM applications and systems.

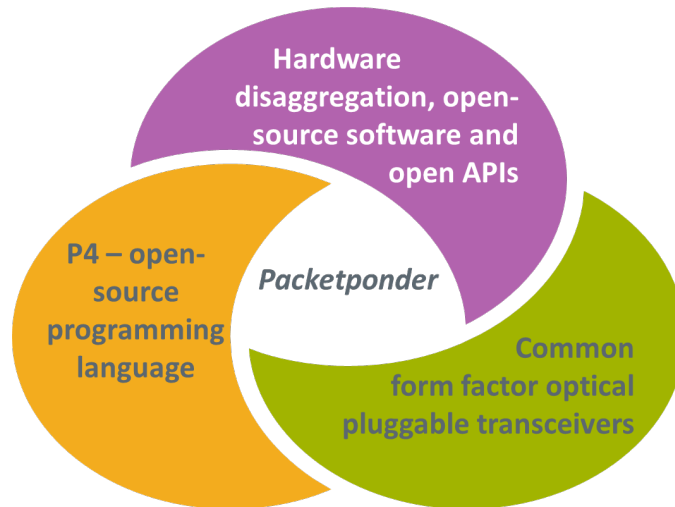


Figure 3.1: *Packetponder's* concept.

3.2.2 Client-side and Line-side definitions

In the context of IP-over-DWDM, the "client-side" normally refers to the interface where data enters or exits a device from the IP or Ethernet network. Typically, those are lower speed interfaces such as 10 Gbit/s or 100 Gbit/s and relatively short distances, and are connected to routers and switches.

The "line-side", refers to the optical interface used for long-haul or metro trans-

mission over the Dense Wavelength Division Multiplexing (DWDM) network. Line-side optics often use coherent transmission, advanced modulation formats (e.g., QPSK, 16QAM, etc.), and FEC, where several signals are transported over a single fibre.

In this Thesis, both client-side and line-side will be used as defined above and illustrated in Fig. 3.2. The client-side refers to Ethernet traffic transmitted over a grey interface, meaning one or more point-to-point optical lines over short distances where the specific wavelength is not significant, or alternatively, one or more copper cables connection. The line-side encompasses coherent/tunable pluggables over DWDM transmission, where multiple carrier wavelengths can share the same optical link for more efficient long-distance transport.

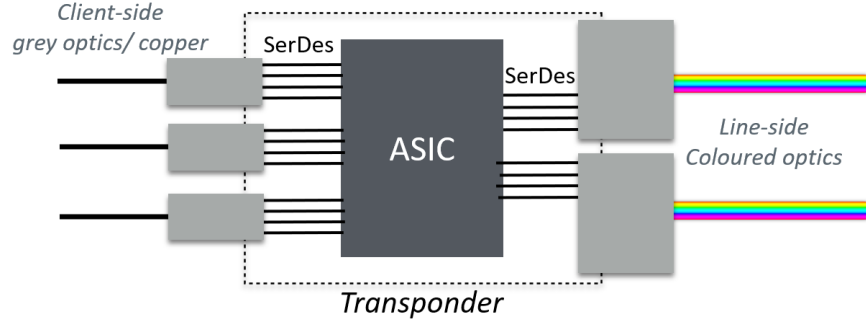


Figure 3.2: IP-over-DWDM concept with coherent pluggables, including SERDES electrical connections within the Transponder.

3.2.3 Packetponder: an enhanced Transponder and Muxponder

Integrating pluggable optics into a programmable L2 switch not only reduces CAPEX but also significantly enhances flexibility in network operations. In essence, the *packetponder* leverages open APIs and a programmable ASIC, that is P4 programmable, to enable a dynamic and flexible mapping between grey and coloured optics, as opposed to a traditional transponder which maps client interfaces to DWDM line-side. Indeed, a traditional transponder enable point-to-point forwarding between grey optics and coloured optics ports, which is shown by mode ① in Fig. 3.3(a). On top of that, the *packetponder* enables client-side to client-side forwarding as a L2 switch (mode ②), and a novel programmable line-side to line-side reconfiguration (mode ③ in Fig. 3.3(a)).

In addition to providing a straightforward 1:1 mapping function, as seen in traditional transponders and switches, the *packetponder* offers the advanced capability

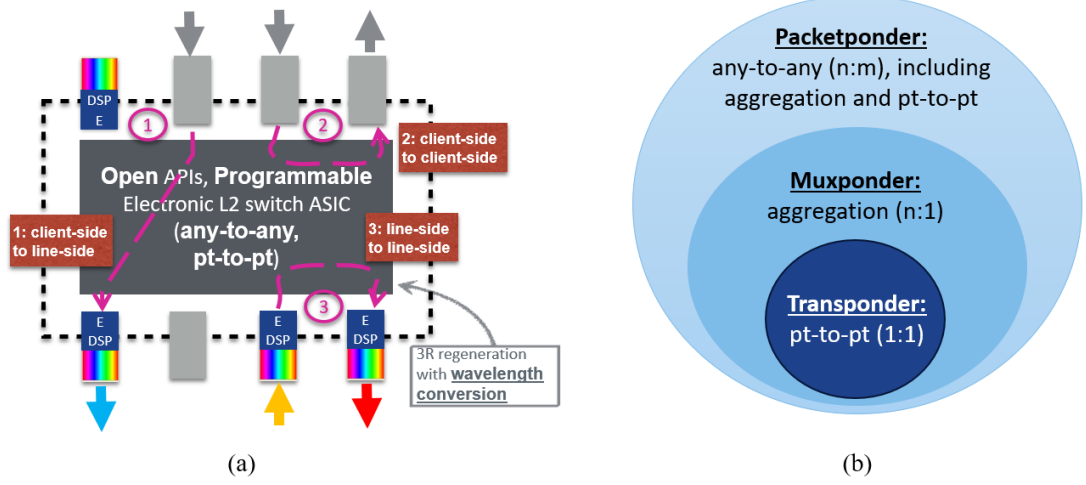


Figure 3.3: (a) *packetponder*'s functionality: ① client-side to line-side - a transponder; ② client-side to client-side - Ethernet cross-connect; ③ line-side to line-side - a P4 programmable opaque optical cross-connect + 3R OEO regeneration with wavelength conversion; (b) Transponder vs. Muxponder vs. *Packetponder*.

of acting as a muxponder. It can perform $n:1$ mapping, enabling the aggregation of multiple high-radix, lower-speed access terminations into a smaller number of low-radix, high-bandwidth metro/core terminations within a single hardware platform. This functionality will be demonstrated in Chapter 5. Furthermore, the P4-programmable ASIC enhances the *packetponder*'s flexibility by enabling packet cloning [79], which allows for $1:m$ or $n:m$ mapping. This capability significantly broadens its forwarding behaviour, making the *packetponder* an adaptable solution for diverse network requirements, as shown in Fig. 3.3(b).

3.2.4 Wavelength conversion and 3R regeneration

By incorporating the *packetponder* concept, we also gain important functionalities that can transform optical network architectures and management. Indeed, mode ③ from Fig. 3.3(a) has useful implications since it introduces wavelength conversion and 3R (re-amplification, re-shaping, and re-timing) regeneration that relaxes the requirement for end-to-end wavelength continuity across multiple ROADM hops, as shown in Fig. 3.4. The reason is because the signal from one part goes through the ASIC enabling regeneration. In contrast to conventional transponders, which typically originate or terminate an optical signal and statically forward protocol data units (PDUs) either from the client-side to the line-side or vice versa, the *packetponder* offers greater flexibility: it can be de-

ployed at any location within the access, metro or core nodes and facilitates forwarding from any port to any other port. This means the wavelength can be assigned and/or re-assigned locally, therefore preventing wavelength blockage situations. The wavelength provisioning use-case with *packetponders* and wavelength conversion will be discussed in Chapter 4.

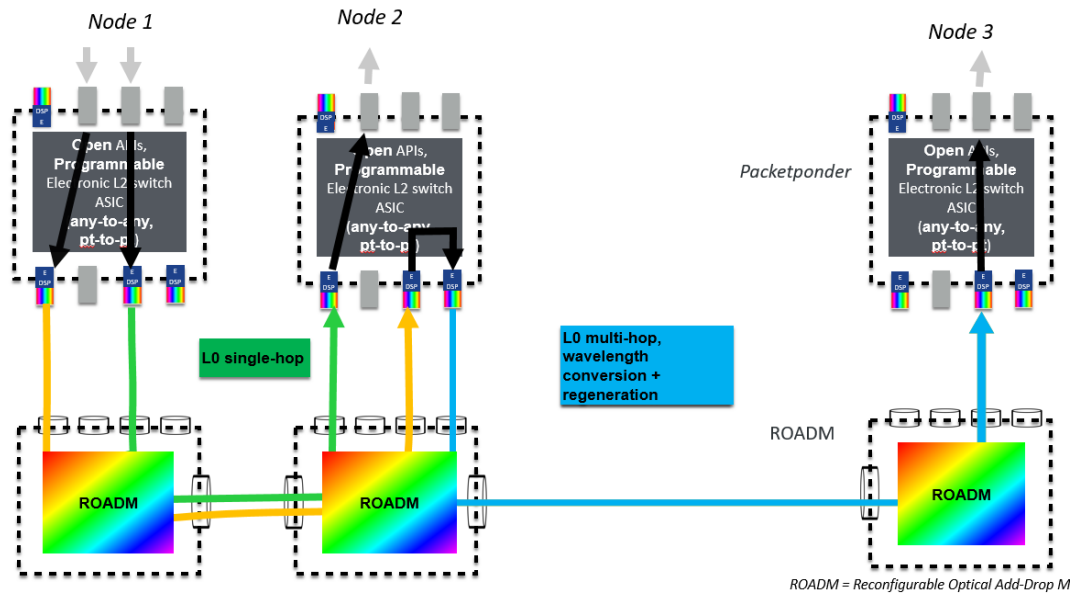


Figure 3.4: *Packetponder* in metro/core networks enabling wavelength conversion for fault recovery or maximise the use of the bandwidth.

Wavelength conversion can also be used to promote interoperability and flexibility. This is very relevant between adjacent sub-network domains, where each domain is managed by an independent entity, for example when crossing countries where networks are managed by different operators. In this scenario, wavelength assignment within one domain is independent of the wavelength assignment within an adjacent domain [80], as shown in Fig. 3.4, where nodes 1, 2 and 3 are located in different domains and a different wavelength is set for packets going from node 1 to node 3. Indeed, when the signal is regenerated at node 2, the wavelength is tuned from yellow to blue. In this scenario, node 2 - a tandem, or transit node - also performs high-speed optoelectronic 3R regeneration, which is a key function to scale-up all-optical networks.

3.2.5 Plug-and-play approach with legacy and new pluggables

The flexibility of pluggable modules within the cages of a *packetponder* is a key advantage, offering operators the freedom to select modules based on their specific needs. Our solution is designed with uniform form factor cages and

specifically our experiments used QSFP28, into which various types of pluggable transceivers can be inserted. These include options like 100GBASE-CR4 (direct attach copper), 100GBASE-LR4, or 100GBASE-CWDM4, ensuring compatibility with a wide range of network configurations. The *packetponder* concept can also adopt switches that can accommodate QSFP-DD or OSFP cages to support next-generation pluggables, such as 800G, 1.2T, and 1.6T modules, ensuring scalability, as long as a multi-source agreement exists.

This work also demonstrates how legacy devices, optics, and existing ROADM infrastructure can be leveraged to implement the *packetponder* concept, simplifying network design while maintaining compatibility with current technologies. The *packetponder* supports a wide variety of pluggables for different services and rates, enabling a plug-and-play approach with standardised client-side and line-side modules. It provides the capability to map diverse line rates while maintaining a uniform cage format across the front panel of the switch. One potential application of the *packetponder* is at the network edge, where multi-service client traffic is aggregated into higher-capacity pluggable transceivers. Unlike traditional architectures that differentiate between client-side and line-side optics, our approach seamlessly integrates all pluggable types (as long as it is compatible with the switch's cages, including the use of adaptors), enabling flexible traffic forwarding from any client or line interface to any other. This adaptability to various line rates ensures efficient utilisation of resources and greater versatility in handling diverse network demands. For example, our solution supports advanced pluggables such as the MicroPlug TiBiT module for Passive Optical Network (PON) applications, which will be explored in detail in Chapter 5.

3.3 Experimental software stack and hardware

In this thesis, prior to the experimental investigations, and to enable the *packetponder* concept, it was essential to integrate pluggable transceivers into the NOS of the programmable Ethernet switch. This required software adaptation to the switch NOS and the YANG interface model to support the relative parameters of transceivers (e.g. the frequency) compliant with both the Small Form Factor (SFF) and Common Management Interface Specification (CMIS) standards, thereby enabling tuning functionalities. Notably, any physical layer parameter exposed by the pluggable — provided the corresponding register supports READ and/or WRITE operations — can be similarly integrated into the system. The validation encompassed 10G Tunable SFP+, 25G Tunable SFP28, and coherent

100ZR transceivers, with flexibility for substitution with higher capacity tunable or coherent transceivers (e.g., 400ZR) as needed.

This section provides an overview of the software stack and hardware devices used in this thesis. The remainder of this chapter outlines the steps taken to implement the *packetponder* proof-of-concept. While the methodology is described in the context of our specific software stack and hardware, the same methodology can be adapted to other disaggregated hardware and open software platforms, since it builds on generic functions, standard interfaces, and adopted abstractions rather than vendor-specific implementations.

3.3.1 Hardware components

Our demonstration used the programmable APS-Networks BF6064X-T Ethernet switch - with 65 x QSFP28 cages arranged regularly on its external frontplate - as the *packetponder*. The whitebox contained a P4 programmable Intel Tofino 1 Intelligent Fabric Processor ASIC with 260 x 25G/10G Ethernet SerDes I/O blocks housed within a 2RU. Wavelength provisioning and pluggable integration within the switch was demonstrated with commercial Lumentum 10G SFP+ tunable transceivers (gray and tunable); a Adtran tunable 25G SFP28 transceiver; and a coherent QSFP28 100ZR from Adtran, performing 100 Gbit/s dQPSK at 27.952 GBaud. Both 10 and 25 Gbit/s transceivers used a Mellanox adaptor (Mellanox MAM1Q00A-QSA28 Compatible QSFP28 100G to SFP28 25G Ethernet Adapter Converter Module)

3.3.2 Network Operating System for our *packetponder*

Our testbed used open management software and open standard interfaces to integrate a wide selection of tunable transceivers into the NOS. The SDN stack used for our experimental validation was determined by the capabilities of our Ethernet switch. In practice, most disaggregated switches can relate to the simplified layered architecture shown in Fig. 3.5. The switch is composed of an operating system stack that can either consist of a base operating system with a dedicated switch OS layered on top, or a standalone switch OS (also referred to as a Network Operating System, NOS). This software stack typically accesses hardware peripherals through platform APIs, which are hardware-specific (green box in Fig. 3.5). At the time, Stratum [74] and SONiC [76] were the most promising open-source NOS due to their modularity and support for open APIs and hardware platforms. We ultimately selected Stratum, combined with Open Network

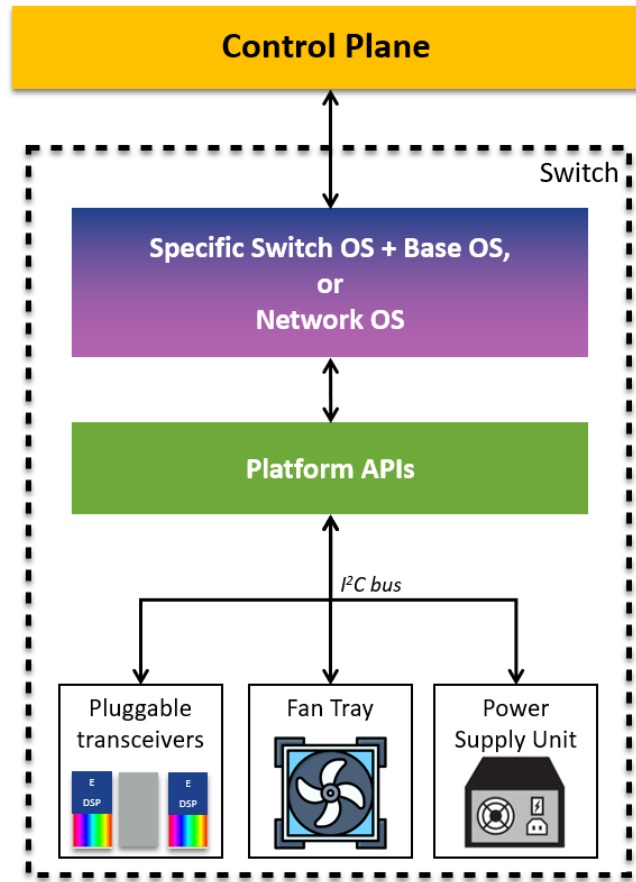


Figure 3.5: Conceptual layered software architecture of a disaggregated switch.

Linux (ONL) as a base OS and its ONL Platform (ONLP) as the platform APIs, because it was a compatible NOS for our APS-Networks BF6064X-T switch. Stratum provided the ability to control packet forwarding through P4 programmability while also incorporating a Physical Hardware Access Layer (PHAL), which allowed us to extend ONLP to access and manage the pluggable optics. At the time of our selection, the Stratum project was led by ONF and supported by a strong community of contributors, which comforted our decision to adopt Stratum for this work. Its combination of programmability and hardware abstraction made it an ideal choice for our experimental setup.

Currently, Stratum only retrieves a small subset of the information available from the registers of the pluggable unit, such as the vendors information or transceiver's type, which are READ only for the user. This is because Stratum's original focus lied on managing layer 2 and above, but its open-source nature made it an ideal NOS candidate to run on packet/optical platforms, as it can incorporate optical layer parameters and enhance management capabilities within the packet controller. While our solution doesn't comprehensively control all pluggable pa-

rameters, it shows how an open switch NOS can usefully integrate and control tunable pluggable optics, which is a functionality that was lacking in all open-source NOS at the start of this work. We demonstrated how to READ and WRITE to the registers that control the carrier frequency (wavelength) across a selection of tunable pluggable transceivers that are contained within the interfaces cages of the P4-programmable switch. This exemplifies our wavelength interrogation and provisioning use-case.

3.4 Pluggables integration in open NOS

3.4.1 I²C bus and pluggable registers

In order to control the carrier frequency of the pluggables, the Network Operating System needs to access a selection of the registers of each pluggable. These registers can be accessed from the switch using the electrical data Inter-Integrated Circuit (I²C) bus, as shown in Fig. 3.5. The I²C bus is the copper traces, multi-master and multi-slave serial communication protocol, that form a control/management bus to allow peripherals to communicate with each other within a device. In our demonstration the pluggables and their registers are addressable slave peripherals. Most Linux-based OS include kernel drivers that enable bi-directional communications over the I²C bus, allowing the host to read and write diagnostic information of the pluggables. Information is stored within a 256-byte addressable memory window register, and are standardised based on the Small Form Factor (SFF) Management Interface MSA for the low-capacity transceivers (SFP, SFP+ and QSFP+ pluggable from factor [81]), and the CMIS Implementation Agreement (IA) for the more recent high-capacity coherent pluggables (QSFP28, QSFP-DD, OSFP [82]). The difference between the two is that historically with SFF, each form factor and each generation had a different management interface specification, which is not scalable given the fast evolution of pluggables and the complexity of their management. Moreover, the SFF management interface defines basic management interfaces, covering only essential monitoring and diagnostics. In contrast, the CMIS IA is scalable and provides a more advanced and flexible management interface for modern, high-speed optical modules, supporting complex configurations, dynamic control, and richer diagnostics. Consequently, SFF and CMIS differ significantly in terms of the structure of their memory maps, specifically regarding the number of pages and banks used for organising and accessing data. SFF-compliant modules organise their

memory into two distinct 'fields' located at addresses 0x50 (or A0h) and 0x51 (or A2h), with a limited number of pages (typically four). In contrast, CMIS uses a single 'field' at address 0x50 while supporting a broader range of pages. Additionally, CMIS introduces the concept of 'banks', enabling efficient management of modules with more than eight lanes, offering greater scalability for advanced optical transceivers, including coherent pluggables, programmable DSPs, and multi-rate or multi-lane configurations, requiring a more scalable and modular memory structure.

In either case, an ordered sequence of READs/WRITEs to the register memory of the pluggable is required [83]: first to set the correct page and do the necessary checks, and then to change the value of the frequency's register. This can be done using simple Linux functions from the I²C driver such as `readb` and `writb`. The completed procedure should operate for both SFF-based and CMIS-based transceivers and irrespectively of the transceiver vendor.

To select a particular switch port where the plugable's frequency needs to change, the I²C bus is also used. The circuit board of the APS-Networks BF6064T-X switch incorporates a "tree" of I²C multiplexers that connects to various peripheral devices. In this arrangement, the single I²C master controller "root" has a transmission path to the "leaves" of peripheral devices. These peripheral devices include the interface cages that accommodate the pluggable transceivers and the registers therein. We specify a particular arrangement of I²C multiplexers to address and READ/WRITE the registers of each pluggable transceiver, as shown in Fig. 3.6. In our switch architecture, accessing a specific port requires configuring a sequence of two multiplexers. The MUX (multiplexer) has its own address, and writing to it selects the channel in use through which the second MUX is connected. For instance, to access port 1, the first multiplexer at address 0x70 must be set to channel 0x4, while the second multiplexer at address 0x74 must be set to channel 0x1. Once these two multiplexers are correctly configured, the registers of the pluggable module inserted in cage 1 become accessible.

The sequence of actions looks like the following:

- Open "i2c-0" bus;
- Select address of primary MUX for writing: 0x70;
- Write channel number containing second MUX device: 0x4;
- Select address of second MUX for writing: 0x74;

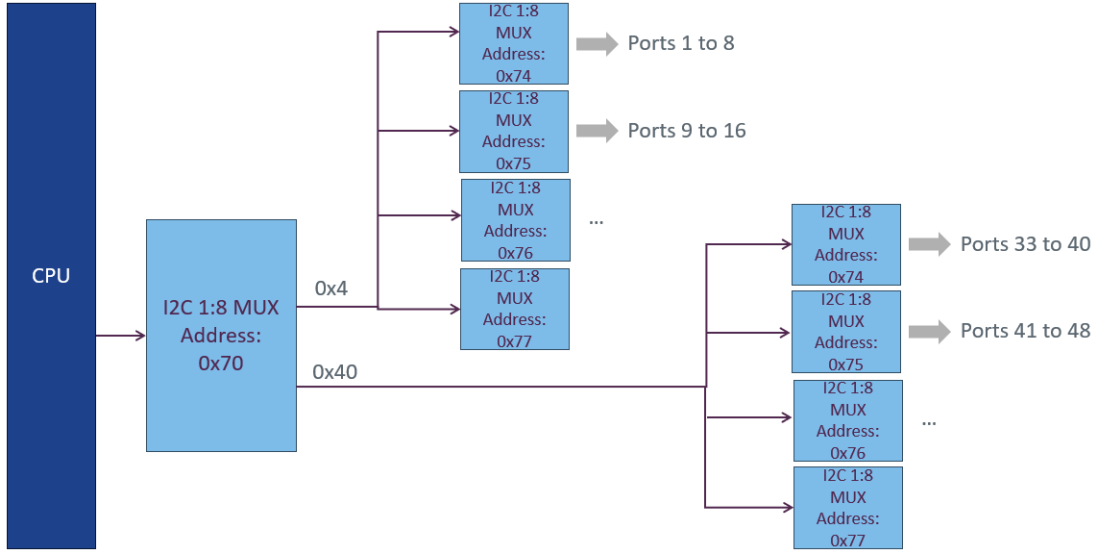


Figure 3.6: I²C block diagram in our Ethernet switch to access the ports.

- Write channel number containing selected port: 0x1 for port 1;
- Read from device: address 0x50 and 0x51 for SFF-based and 0x50 for CMIS-based pluggables;
- Reset secondary MUX;
- Reset primary MUX.

A significant challenge arises from the potential for multiple programs to access the bus concurrently, with examples being the Barefoot Runtime components (e.g., the Software Development Environment) and ONLP. Both aim to query data from pluggables; ONLP might attempt to read a sensor while the Software Development Environment (SDE) seeks pluggables' information simultaneously. Each request is the product of multiple commands, the quantity of which depends on the number of multiplexers involved. In our scenario, selecting more than one transceiver simultaneously was impossible as they all share the same address to access their registers (0x50/0x51). Additionally, these requests can be interleaved with one another.

3.4.2 MUTEX

The kernel I²C API provides a way to lock access to devices on the I²C bus. This prevents conflicts when multiple processes or drivers try to communicate with the same device simultaneously. The locking mechanism works if each of the devices on the bus have their own dedicated kernel-level device drivers. However, both

the SDE and ONLP do not use kernel-level drivers. Instead, they use user-space drivers, which do not rely on the kernel's I²C driver framework. This means the kernel cannot manage access to the I²C bus for these devices, and ultimately we cannot lock the bus itself.

However, we can control all the user space drivers so that they agree on a method for handing control. To protect the I²C bus and manage control effectively, we therefore implemented file locks (or MUTEX) to ensure mutual exclusion. Since locks cannot be shared among multiple threads or processes, this approach ensures that all tasks can safely access a specific resource, while other processes wait for the file lock release. Using locks facilitates secure configuration of multiplexers to access a designated port and the transceiver's registers. Locks are applied for all devices using the I²C bus, including the transceivers, the PSU, the LEDs or the fans.

To implement this, we collaborated closely with Gawen Davey, a Software Engineer at APS-Networks. Locking mechanisms are now integrated into the source code of the operating system across all APS-Networks programmable switches that use ONL as their base OS.

One example of the lock implementation to read the EEPROM of the transceivers was implemented as follows:

```
bf6064x_lock_acquire();  
int rv = dpapi_sfp_eeprom_read(id, dst);  
bf6064x_lock_release();
```

In this case, before a process begins reading the transceiver in a specific port, it first locks the file to ensure exclusive access. Once the task is completed, the lock is released. If the process unexpectedly terminates, the file is automatically released to prevent other applications from hanging or encountering access issues. In our frequency reconfiguration use-case, whenever the frequency function is called, a mutex is used to lock the I²C bus ensuring exclusive access. The mutex is released before the function exits, regardless of whether the frequency was successfully changed or an error occurred.

Once the bus is locked, the two multiplexers are configured to map to the transceiver's registers in the requested port. However, there was no existing functionality to configure the multiplexers or change the frequency of tunable transceivers, despite the SFF standard providing guidelines for doing so. Therefore, both the multiplexer 'mapping' functionality and the frequency 'write' operation had to be

created as per section 3.4.1, using the APIs available through ONLP. The function, called:

```
onlp_i2c_mux_mapping(int port_number, int reset)
```

was created in the user-space driver "i2c" in the ONLP library of ONL [JRc2], and either set the multiplexers to map the requested port, or reset the multiplexers.

Another function was then created and called:

```
set_sfp_frequency(int port_number, int frequency)
```

which allowed the frequency at the given port_number to be tuned. In this function, the module type identifier is first read (given by SFF-8024 A0h. Byte 0) to determine the form factor in use and, consequently, the management interface implemented for the pluggable.

3.4.3 Frequency change for SFF interfaces

For modules using the SFF MSA interface, i.e., our 10G and 25G transceivers, the function starts by locking the file, map the corresponding MUX to access the transceiver's register, and then follows a succession of read and write to configure the new frequency as shown in Fig. 3.7.

After setting page 2, the function reads the spacing grid and the initial frequency, and finally assigns the channel number as per the formula given by the SFF-8690 [84]. Each time the reading/writing is not successful, the MUX is released and the mapping is set back to its initial state (registers back to 0x0). In this case, setting page 2 was enough to assess the tunability of the transceiver. If the function is applied to a non-tunable transceiver, an error is raised and the function terminates by resetting the MUX and releasing the lock.

3.4.4 Frequency change for CMIS interfaces

For CMIS-compliant modules, the function evaluates the module's tunability in page 1, verifies the channel number's validity based on the grid spacing, and sets the channel number in the frequency registers located on page 12 of the memory map as shown in Fig. 3.8[82].

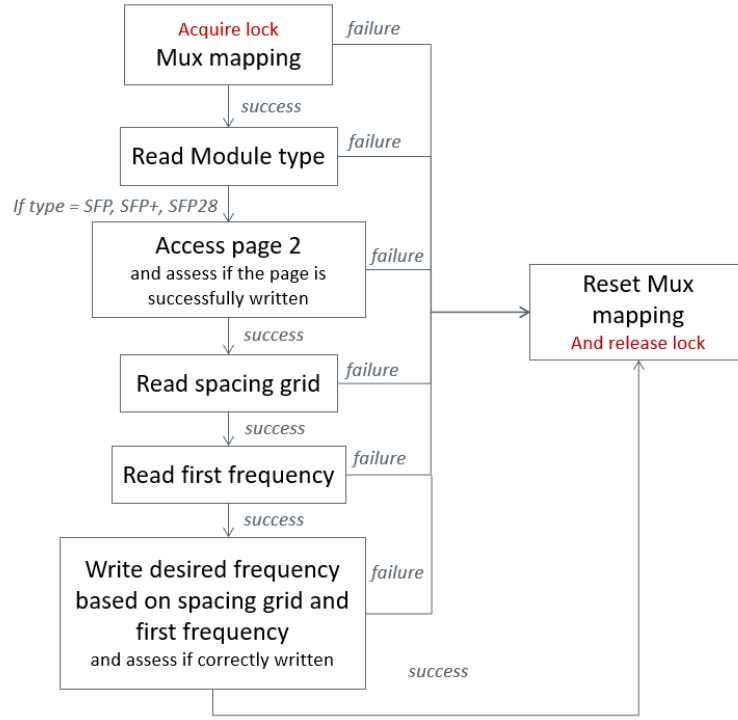


Figure 3.7: Block diagram to set new frequency for SFF pluggable transceivers (10G/25G).

3.4.5 Abstraction layers in Stratum

Initial discussions with the Stratum core team — specifically Brian O’Connor and Max Pudelko — led to the identification of three potential approaches for integrating the frequency tuning into the Stratum software stack.

Option (i) involved reusing the well-defined structures within the TAI (Transponder Abstraction Interface) architecture, which already supported frequency tuning for coherent pluggables. The idea was to reuse its transceivers’ architecture and implement it into the SAI (Switch Abstraction Interface), which is what Stratum is using. However, this approach proved challenging, as the coherent-specific data structures were tightly coupled with the TAI context and would only be invoked if the hardware was recognised as a transponder, not as a switch.

Option (ii) consisted of manually setting the frequency during device boot-up using an initial configuration text file. This involved populating the ONLP database within Stratum, with the desired frequency values and writing directly to the transceiver registers via the functions defined in section 3.4.2. This approach was adopted in the initial phase of the experiment to validate the basic functionality and confirm low-level register access (see section 3.4.5.2).

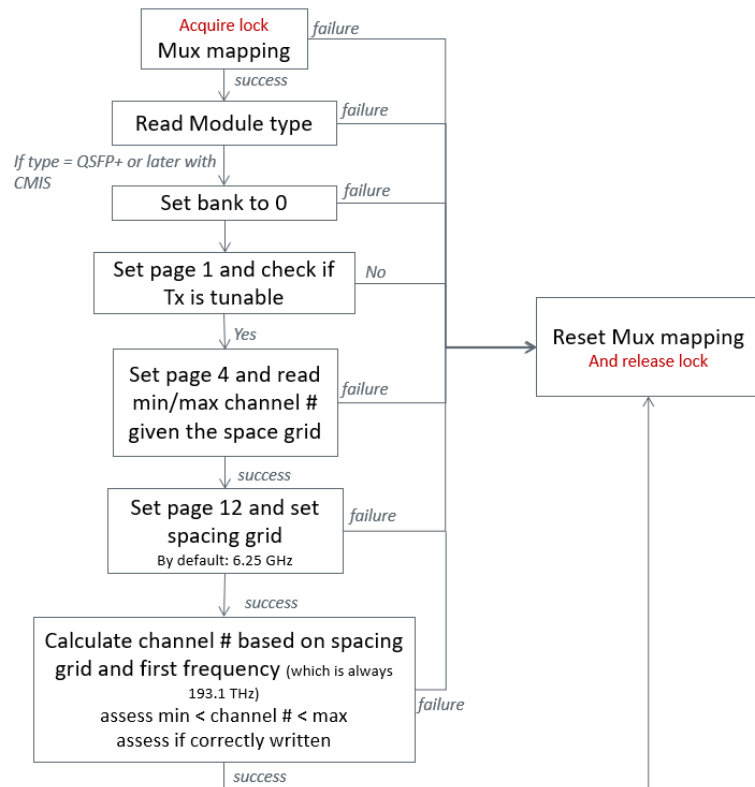


Figure 3.8: Block diagram to set new frequency for CMIS pluggable transceivers (100/400G ZR/ZR+).

Option (iii) proposed extending the existing transceiver model within SAI to include frequency parameters, similar to how interface speed or loopback modes are currently defined. In this way, the gNMI protocol could also be used to configure the frequency, in the same way that it does to change the speed of a transceiver.

Each option has its own trade-offs, which are discussed in more detail in section 3.7. But given that coherent pluggables were not yet available at the time of development, we, the Stratum team and us, opted for option (iii). This allowed us to integrate frequency tuning into the standard SAI transceiver definition, enabling the feature to be accessed and configured by the packet controller, without requiring transponder-specific infrastructure.

3.4.5.1 Southbound interface of Stratum: onlp_wrapper

Stratum loads several libraries during the boot process, including common and platform-specific libraries with different functions. As our frequency function contains the MUX mapping function that is specific to our switch, we included it in the onlplib platform-specific library. Ideally, only the MUX mapping is

platform-specific and the frequency tuning function is loaded by default on all platform.

Stratum does not directly communicate with the hardware but uses abstraction layers like `onlp_wrapper` to translate platform-independent operations into ONLP APIs, enabling tasks such as reading from and writing to tunable pluggable optics. By standardising access to hardware components, the `onlp_wrapper` abstracts vendor-specific hardware implementations, allowing Stratum to interact with a wide variety of network switches in a consistent and uniform manner.

We introduced an `sfp-frequency` function within the `onlp_wrapper` [JRc1], which interacts with the ONLP API to tune the frequency of the pluggable optics (if available). This implementation enables the core of Stratum to make direct calls to the `onlp_wrapper` for managing optical parameters.

3.4.5.2 Northbound interface of Stratum: gNMI server

At the northbound, Stratum includes a gNMI server, which listens for incoming requests from external network management systems. gNMI, which stands for gRPC Network Management Interface, is a control data streaming protocol used for network management and configuration that is designed to support model-driven management of network devices, allowing for more modern and efficient interaction with network infrastructure. Alternatively, other protocols like NETCONF can also be used, depending on what the NOS provides as its northbound interface.

Once the gNMI server receives a request specifying which data is being queried or modified, Stratum uses its internal database (e.g., *Chassis Config* Database or persistent state database) to check if the requested information is already cached or preconfigured. If the data is available in the database and is up to date, the database responds directly, avoiding a direct hardware query. This reduces latency and avoids unnecessary hardware polling for frequently accessed information. If the information is not available in the database or the request requires real-time data, Stratum translates the gNMI request into a platform-independent internal operation. This involves identifying which hardware component (e.g., fan, transceiver) is being queried or controlled. Stratum then delegates the platform-independent operation to `onlp_wrapper`, which acts as a middleware layer.

To populate the *Chassis Config* database, which deals with platform-related

```
singleton_ports {
  id: 4
  name: "4/0"
  slot: 1
  port: 4
  speed_bps: 10000000000
  config_params {
    admin_state: ADMIN_STATE_ENABLED
    frequency: 193400
  }
  node: 1
}
```

Figure 3.9: Initial chassis configuration text file with the added frequency parameter.

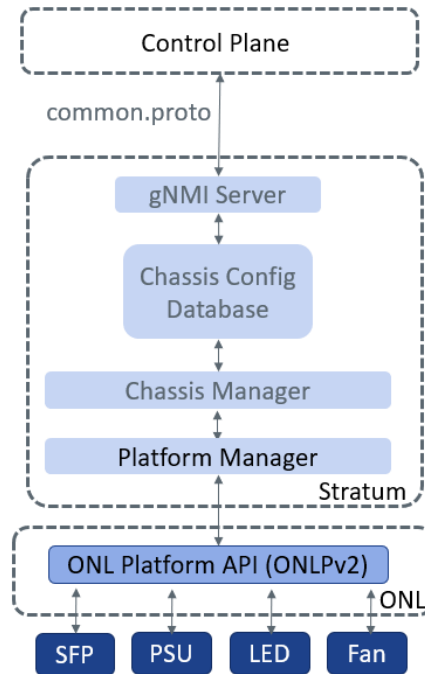


Figure 3.10: Simplified architecture of Stratum [85].

components, a text file can be used in which each ports are configured using `singleton_ports` abstraction. As the frequency parameter was not part of any abstraction definition, we extended the `singleton_ports` configuration to support the frequency parameters of the tunable pluggable, as per Fig. 3.9, so when a new frequency is sent from the controller and parsed by the gNMI server, it is stored in the chassis configuration of the Stratum database, and calls are made towards the Chassis Manager to push the new configuration all the way down to the registers of the optical module. The platform manager at the bottom of Fig. 3.10, will use `onlp_wrapper` to load and call `set_sfp_frequency` to tune

the frequency SFF or CMIS transceivers. The initial frequency of a tunable transceiver in a particular port can also be set during the boot-up process of Stratum by uploading the previously mentioned text file (`chassis-config.pb.txt`), which contains the entire structure and values of the chassis configuration at initialisation. The `chassis-config.pb.txt` file will populate the database in a similar manner to how a gNMI request operates.

This section demonstrated how the frequency of a tunable transceiver (coherent or not) can be configured either during the initial loading of Stratum (option ii) or through a gNMI request (option iii). The next step involves connecting Stratum's northbound interface (gNMI) to a remote open controller.

3.5 Pluggables remote management with open SDN controller

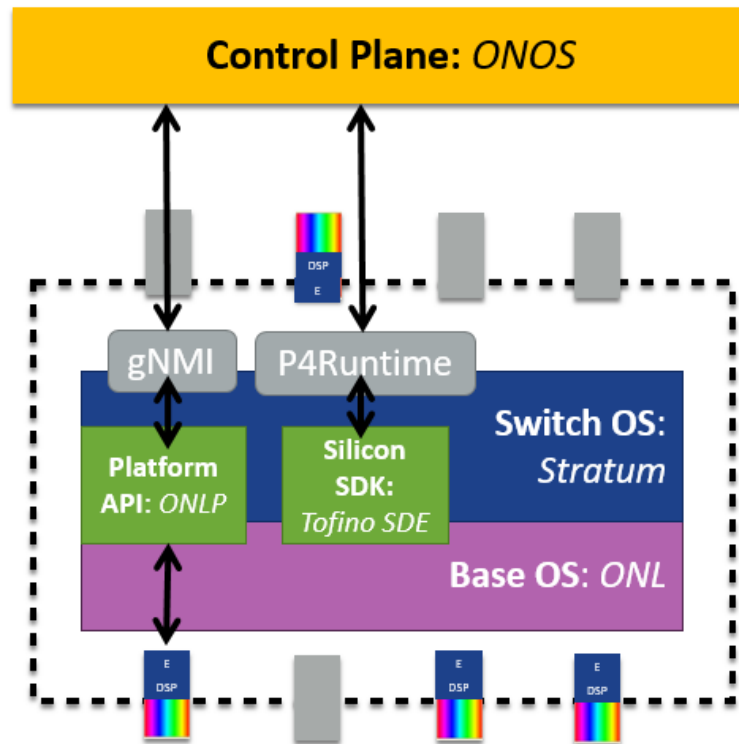


Figure 3.11: SDN management stack of our *packetponder*.

Now that the gNMI server is linked to the ONLP APIs and hence the pluggable's registers in each port, we want to access them at runtime and dynamically change the frequency. For that, we used the vendor agnostic Open Network Operating System (ONOS)[86] remote controller designed and implemented by ONF, and

Fig. 3.11 outlines the full programmable SDN stack used for the experimental validation. ONOS was chosen because we were already co-developing with the ONF ecosystem, which made it a logical and seamless choice for our work. ONOS was originally developed for packet networks but rapidly addressed control of optical disaggregated systems with the Open Disaggregated Transport Network (OTDN) activity which successfully developed a framework to control Open Line Systems and Transponders [87]. ONOS works particularly well with data plane devices that are defined by standard abstraction, which is the case with Stratum. ONOS uses YANG models, a leading choice to configure networking devices, to express the structure of data by using "tree structure". In association to YANG, various open working groups, including OpenConfig, are gaining importance by providing vendor-neutral data models, written in YANG. Each model in the OpenConfig hierarchy defines a combination of a configuration state that can be both read and written by the user. The configuration requests from ONOS leverage protobuf for inter-plane message exchange, while gNMI (Network Management Interface) serves as the runtime protocol for remote configuration and monitoring.

The most logical choice for configuring coherent pluggables is the OpenConfig platform YANG model [88], as it is commonly used to characterise pluggable optics within transponders. This model has been naturally adopted for ZR/ZR+ configuration because it provides comprehensive access to all optical parameters of the pluggable. However, its scope is limited to coherent pluggables and does not extend to other tunable transceivers. Currently, no YANG model includes the frequency parameter with both read and write permissions for tunable transceivers. To address this gap, we incorporated the frequency parameter into the OpenConfig interfaces YANG model [89], enabling read and write permissions (with write allowed only if the transceiver is tunable, verified via the `set_sfp_frequency` function). The OpenConfig interfaces manages the configuration and operational state of physical and logical network interfaces, including attributes like interface type, speed or admin status. Therefore, we extended the OpenConfig interfaces tree structure, in both the "state" (with read only permissions - ro) and "config" (with read/write permissions - rw) branch, as shown below, by attaching the frequency parameter directly to the host interface name rather than to the logical interface or optical channel, as done in OpenConfig platform. The following tree is obtained by applying the `pyang` [90] function to generate a tree representation of the YANG model supported by Stratum.

```
module: openconfig-interfaces
```

```

+--rw interfaces
  +--rw interface* [name]
    +--rw name                -> ../config/name
    +--rw config
      | +--rw loopback-mode?   boolean
      | +--rw enabled?         boolean
      | +--rw health-indicator string
      | +--rw sfp-frequency    uint64
    +--ro state
      | +--ro name?            string
      | +--ro loopback-mode?   boolean
      | +--ro ifindex?         uint32
      | +--ro admin-status     enumeration
      | +--ro oper-status      enumeration
      | +--ro last-change?     oc-types:timeticks64
      | +--ro id               uint32
      | +--ro hardware-port    string
      | +--ro health-indicator string
      | +--ro sfp-frequency    uint64

```

This approach simplifies access for the packet controller, allowing it to retrieve port frequency in the same way it retrieves other attributes such as port speed. Additionally, we extended the PortConfigParams protobuf internal message to enable the Stratum gNMI server to handle the new parameter and gNMI path, and store it in the database [JRc3].

Within ONOS, we created a Command Line Interface (CLI) function to either set or get the frequency of a particular switch port [JRc3], which is:

```
frequency-config set/get <device-id> <port-number> <frequency>
```

This CLI command triggers ONOS to parse the input and convert it into a higher-level request to the system. As depicted in Fig. 3.12, ONOS then maps the CLI command to the corresponding behaviour interface (called FrequencyConfig in our case) in its control framework. Behaviours are abstract definitions of capabilities that a device or driver should support. It defines what the system can do, not how it is implemented for a specific device. The behaviour then implements a driver that is specific to the device (in our case, a driver that uses the gNMI protocol and that can be applied for Tofino switches). That driver trans-

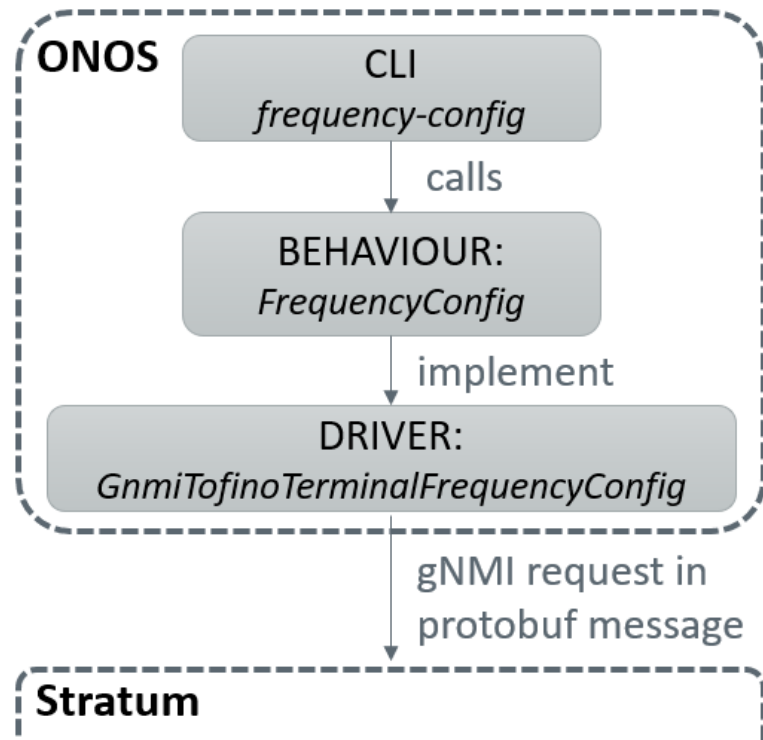


Figure 3.12: ONOS simplified architecture for the frequency tuning function.

late the behaviour request, the new frequency, into a concrete action the device can understand. The driver then sends a gNMI message containing the new frequency value (in GHz) and a specific path that identifies the configuration node. This path adheres to the YANG interface model and aligns with the hierarchical structure of the Stratum database. In our case, the gNMI path is:

```
/interfaces/interface[name=<port>]/config/sfp-frequency
```

When Stratum receives the request with a new configuration state, it updates the database which will then successively call the Chassis Manager and the Platform Manager to push the new chassis configuration and the new frequency (or wavelength) to the related transceiver.

3.6 Experimental proof-of-concept

The software stack described earlier supports both CMIS-compliant and SFF-compliant pluggable modules by using generic functions where necessary. For the proof-of-concept of the *packetponder*, our testbed demonstrated wavelength provision by the integration and control of wavelength tunable pluggable transceivers of various formats, from different vendors, and supporting different stan-

dards, all within the same box. For non-QSFP28 form factors, we employed a QSFP28-to-SFP+ Mellanox converter to adapt them to our 64xQSFP28 cages. The demonstrated use case showed wavelength assignment by the remote controller, ONOS.

3.6.1 Wavelength conversion of SFF and CMIS-compliant pluggable from ONOS controller

Our network scenario comprised our APS-Networks programmable switch as the *packetponder*: the tunable 10G SFP+ was inserted into port 4/0, the tunable 25G SFP28 was in port 8/0, and the 100ZR coherent pluggable was in port 30/0 of the *packetponder*, and all pluggables were configured in a loopback to themselves. The first part of the experiment pushed in-situ a selected wavelength (or frequency) down the full software stack, from controller to the pluggable tunable 10G SFP+, tunable 25G SFP28, and coherent 100ZR transceiver via the I²C bus. This independently set/reset the wavelength of each individual line-side tunable transceiver - without the need for a transponder.

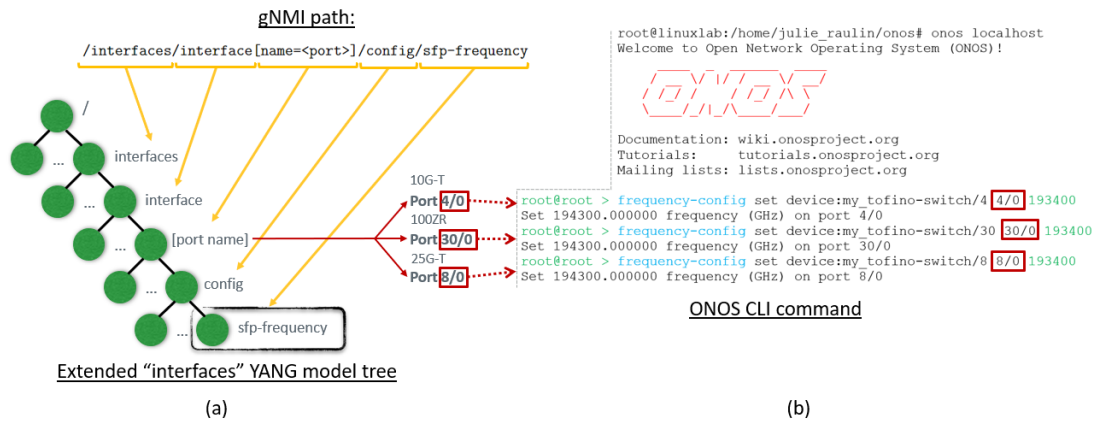


Figure 3.13: (a) gNMI path and new YANG tree node to access the frequency; (b) ONOS Command Line Interface (CLI) to reconfigure the frequency of tunable transceivers.

Initially, all transceivers were set to a frequency of 194,300 GHz. Using ONOS, we reconfigured all transceivers to 193,400 GHz, as illustrated in Fig. 3.13(b). This was achieved by executing the ONOS command, which included the gNMI path aligned with the extended Openconfig interface YANG tree depicted in Fig. 3.13(a). The command was run successively to adjust the frequency on port 4/0 (10G tunable), port 30/0 (100ZR), and port 8/0 (25G tunable).

Fig. 3.14 depicts Stratum log after the 100ZR was reconfigured. The first red box

```
bf-sde.pm> I20231026 13:25:42.483511 11305 bfrt_switch.cc:249] BfrtSwitch::SetValue is not implemented yet,
but changes will be performed when ChassisConfig is pushed again.
requests { port { node_id: 1 port_id: 30 frequency { frequency: 193400 } } }.
...
I20231026 13:25:45.548467 11305 bfrt_switch.cc:61] Chassis config pushed successfully.
I20231026 13:25:50.036953 11285 bf_chassis_manager.cc:1289] State of port 30 in node 1 (SDK port 292): UP.

lowest_n_hexa for 6.25 GHz spacing 0xfffffee8highest_n_hexa for 6.25 GHz spacing 0x1e0
lowest_n for 6.25 GHz spacing -280highest_n for 6.25 GHz spacing 480grid_spacing_hexa 0x10
channel_number: 0x30 = channel 48 = 193400 GHz (CMIS 1A)
```

New gNMI request with new frequency

Extra print information about the transceiver's registers

Figure 3.14: Stratum logs for 100ZR wavelength reconfiguration.

corresponds to Stratum processing the gNMI request, and pushing the chassis configuration with the new frequency value. The last few lines are prints statement for monitoring purposes, and show that the implemented value into the channel number register is 0x30, which corresponds to channel 48. Given that we set our coherent pluggable to operate with 6.25 GHz grid spacing, and that the first frequency is 193,100 GHz, the CMIS documentation specifies that $\text{frequency} = \text{first_frequency} + \text{grid_spacing} * n$, which is our case, $193,400 = 193,100 + 6.25 \times 48$, hence 48 is the channel number wanted.

Likewise, for the 10G tunable SFP+ in port 4/0 and the 25G tunable SFP28 in port 8/0, the Stratum logs are depicted in Fig. 3.15 and show the successful change of frequency after a few seconds with the ports transitioning to an UP state [JR2]. Even though the same frequency is requested, the channel number for each differs. This discrepancy is due to the distinct initial frequencies and grid spacings of both transceivers. Indeed, our frequency setting function meticulously considers all parameters before adjusting the frequency, consistent with the guidelines outlined in the SFF and CMIS documentation. This wavelength conversion using the same command from a remote controller demonstrates the benefit of standardisation, open source software and open APIs, which plays a key role in this work.

3.6.2 Reconfiguration time of the 100ZR pluggable transceiver in a ROADM scenario

While the Stratum logs in Fig. 3.14 and Fig. 3.15 provide an approximate reconfiguration time — from when the request is received by the switch to when the port reaches the UP state — they measure this time at the Stratum level. According to the logs, the reconfiguration takes about 8 seconds for the 100ZR and 2 seconds for the 10G and 25G tunable transceivers. To obtain a more accurate estimate of the frequency tuning configuration time at the physical layer, we generated traffic to the system and measured packet loss while the system is

```
bf-sde.pm> I20231019 14:36:09.624068 11305 bfrt_switch.cc:249] BfrtSwitch::SetValue is not implemented yet,
but changes will be performed when ChassisConfig is pushed again.
requests { port { node_id: 1 port_id: 4 frequency { frequency: 193400 } } } 10G SFP+
...
I20231019 14:36:09.719377 11305 bfrt_switch.cc:61] Chassis config pushed successfully.
I20231019 14:36:11.398473 11285 bf_chassis_manager.cc:1289] State of port 4 in node 1 (SDK port 440): UP.

grid_spacing_hexa 0x3e8 grid_spacing 100 first_frequency_THz 0xc0 first_frequency_GHz 0x3e8 first_frequency 192100
channel_number: 0xe = channel 14 = 193400 GHz (SFF MSA)

bf-sde.pm> I20231019 14:39:09.389220 11306 bfrt_switch.cc:249] BfrtSwitch::SetValue is not implemented yet,
but changes will be performed when ChassisConfig is pushed again.
requests { port { node_id: 1 port_id: 8 frequency { frequency: 193400 } } } 25G SFP28
...
I20231019 14:39:09.423218 11306 bfrt_switch.cc:61] Chassis config pushed successfully.
I20231019 14:39:11.309118 11285 bf_chassis_manager.cc:1289] State of port 8 in node 1 (SDK port 408): UP.

grid_spacing_hexa 0x1f4 grid_spacing 50 first_frequency_THz 0xbff first_frequency_GHz 0x3e8 first_frequency 191100
channel_number: 0x2f = channel 47 = 193400 GHz (SFF MSA)
```

Figure 3.15: Stratum logs for 10G/25G wavelength reconfiguration.

reconfiguring. By correlating the packet loss with the packet rate, we derived a more precise reconfiguration time, which is expected to be faster than the Stratum-level log results.

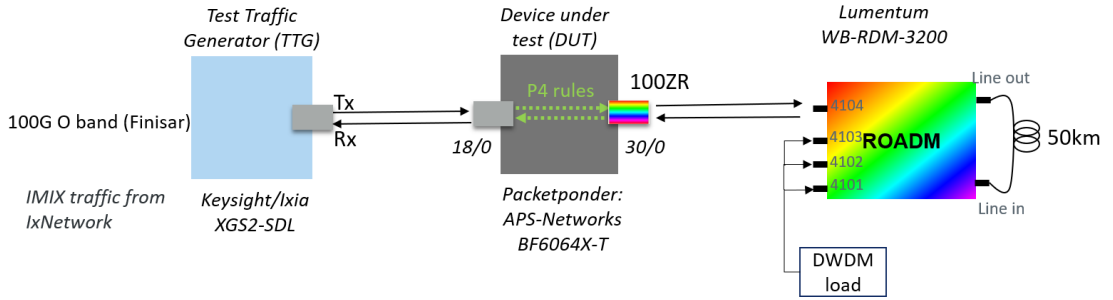


Figure 3.16: Experimental setup for proof-of-concept and packet loss measurements.

Moreover, for a more realistic scenario, we introduced a Lumentum Reconfigurable Optical Add/Drop Multiplexer (ROADM) into the setup as shown in Fig. 3.16. The 100ZR transceiver was connected to the ADD/DROP port 4104 of the ROADM, with the ROADM's line out port looped back to its line in port via a 50 km fibre spool, simulating two half ROADMs in a network. To emulate a fully loaded DWDM system, we used a filtered Amplified Spontaneous Emission (ASE) source (ASE load) passing through the ROADM, representing 80×50 GHz grid-spaced data channels. Four bandwidth slots (equivalent to 2×100 GHz slots) were left vacant to simulate available bandwidth. The three distinct bandwidths generated using the ASE source to simulate client traffic were routed through the ADD/DROP ports 4101, 4102, and 4103 of the ROADM, as configured in Fig. 3.17. The ROADM's demultiplexer was set to drop these channels since they were not relevant. As the DWDM load emulation power was not equalised, the

table in Fig. 3.17 shows uneven input and output powers, and this is also evident in Fig. 3.18, showing ASE slope in the spectrum.

Mux Connections									
Connec...	Name	Start Frequency (G...	End Frequency (GHz)	Blocked	Maintenance State	Attenuation (...)	Connected P...	Input Power (d...	Output Power (...)
1	port3	194350.00	196125.00	No	In Service	0.0	4103	2.30	-1.90
3	port1	191325.00	193350.00	No	In Service	0.0	4101	8.50	4.50
4	port2	193450.00	194250.00	No	In Service	0.0	4102	1.70	-2.40

Figure 3.17: ROADM MUX rules.

Initially, as in the first experiment, Stratum loaded a chassis configuration file with the 100G ZR transceiver set to an initial frequency of 193.400 GHz (1550.12 nm), corresponding to the blue data channel in Fig. 3.18 (top diagram). The ROADM was preconfigured with an initial rule for both the MUX and DEMUX, allowing the passage of the 193.400 GHz frequency channel from the ADD/DROP port 4104 to the Line IN/OUT. Ports 18/0 of the switch was a 100G O-band transceiver, non-tunable, and connected to a packet generator to generate traffic at 100 Gbit/s. The packets were forwarded from port 18/0 to 30/0 and vice versa, using a simple P4 program implemented into the *packetponder*.

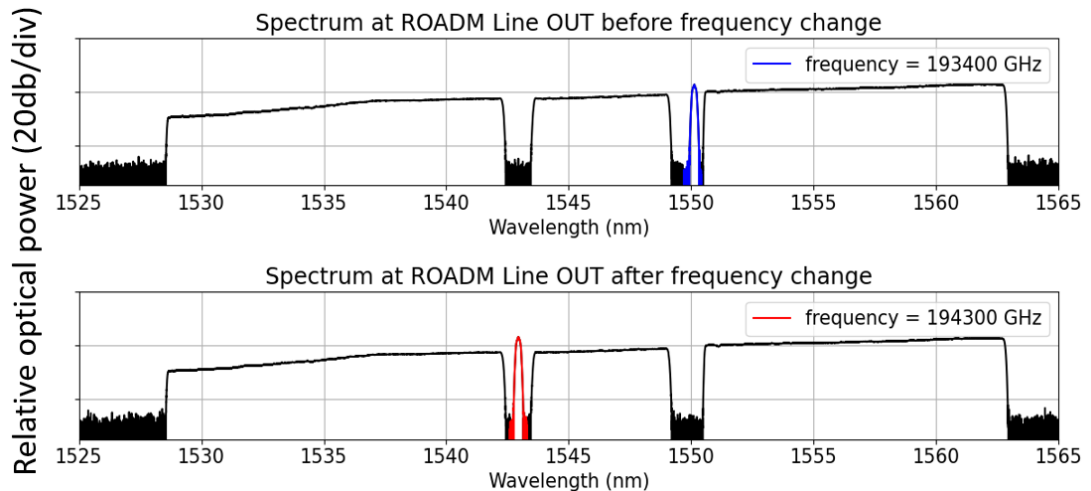


Figure 3.18: Line-side spectrum at the initial state (top trace) and after reconfiguration (bottom trace), showing wavelength switching between dis-contiguous wavelength spectrum slots.

Ethernet test packets were generated by the Keysight/Ixia XGS2-SDL test traffic generator (TTG). The Keysight TTG's chassis contained: I) Novus-S: QSFP-28 port cages supporting 100/50/25/10 GbE; and II) Novus-NP: SFP+ and 10GBase-T port cages supporting 10GbE/1GbE/100MbE load modules. It emitted Ethernet packets of different format, and used the *IxNetwork* software for building test scenarios and viewing statistics. We used Internet MIX (IMIX) of packets for each traffic to resemble a real-world scenario, with 58.33% of the packet distribution

being small token size (66-70bytes, such as Session Initiation Protocol traffic), 33.33% of medium size (570 bytes, representing web traffic with http meta data), and 8.33% of size 1518 bytes, illustrating video traffic for example. Packets generated had a unique IPv4 source address, which was used by the P4 switch to forward the traffic to the egress port connected to the ROADM at a rate of 100 Gbit/s. The optical data was then sent through the ROADM and looped-back with a patchcord to the receiver port of the 100ZR transceiver, and subsequently captured by the test generator and Wireshark for measuring packet loss and the reconfiguration time during the experiment.

While the system was running, similarly to a real network with packets transmitted through the optical path, we then ran an ONOS CLI script to simultaneously adjust the start and end frequency values of the ROADM MUX and DEMUX (using the existing `wavelength-config` command in ONOS [91]) and the tunable transceiver centre frequency (with `frequency-config` from Fig. 3.12) using NETCONF and gNMI respectively. The sequence of activation was as follows: the ROADM rules were changed first, and then the frequency of the pluggable within the switch. In our experiment, we used the same instance of ONOS to control both the ROADM and Stratum switch because our devices were located on the same local area network. If this concept is to be scaled, a multi-instance architecture of ONOS is needed, notably to fulfil bandwidth and latency requirements [77]. The new centre frequency was set to 194,300GHz (= 1542.9nm), and Fig. 3.18 shows the line-side optical spectrum, showing the emulated DWDM channels (or ASE load), and wavelength switching from 1550.1nm (top diagram) to 1542.9nm (bottom diagram).

The wavelength switching process required hardware changes both in the transceiver and the ROADM, temporarily interrupting traffic as expected. To evaluate the reconfiguration time of the transceiver itself, we sent 1 billion frames in total with *IxNetworks*. We monitored the Layer 2 frame rate (resulting in a speed slightly below 100 Gbit/s) and observed packet loss, as depicted in Fig. 3.19. Losing 222,508,645 frames, at a frame rate of around 33,028,592 frames/s (value given by *IxNetworks* while the experiment is running), indicates an approximate wavelength reconfiguration of 100ZR transceiver of 6.7s, which is faster than current proposed solution [92].

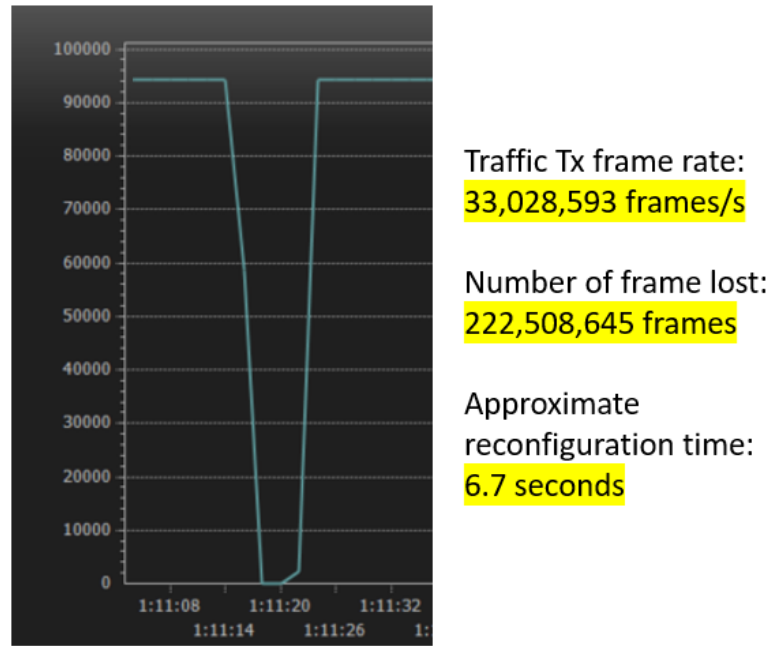


Figure 3.19: *IxNetwroks* data rate statistics, showing traffic interruption during the 100ZR pluggable's frequency reconfiguration.

3.7 Implementation discussion with current state-of-the-art

We presented a potential solution to control pluggable optical modules. While our demonstration primarily addressed the frequency parameter, it supports different standardised modules compliant with both SFF and CMIS specifications. This capability enables wavelength provisioning and offers greater flexibility in spectrum management, particularly as networks approach capacity limits. The practical application of this will be discussed further in the next chapter.

At the start of this work, it became clear that the industry was moving in the same direction [61], with growing interest from companies in addressing these challenges. Since then, several projects and community groups have tackled the integration of pluggables in switches, including the MANTRA (Mandatory Transceiver Requirements Agreement) project under the Telecom Infra Project (TIP). MANTRA focuses on improving the integration of optical pluggable transceivers into switches, routers, and other network devices, especially in disaggregated and open networks.

Phase 1 of the MANTRA project focused on integrating coherent pluggable modules with router platforms using partially standardised or proprietary interfaces

and primarily focused on the operational aspect. Phase 2 expanded the scope and aimed to provide a vendor-agnostic management of pluggable modules. This phase also introduced OpenConfig as a standardised management interface to reduce reliance on proprietary solutions, enhance interoperability, and simplify the integration of new components [93].

While our solution addresses some of the issues raised by the MANTRA project, it may not represent the most flexible abstraction layer for next-generation pluggable transceivers and architectures. At the start, our solution was well-suited to the low-speed tunable pluggables we worked with. However, the industry's focus has since shifted to coherent pluggables, such as 400ZR/ZR+ and 100ZR modules, and the emergence of 800G and soon 1.6Tb/s coherent pluggable modules has highlighted the limitations of our approach. For instance, our choice to include the frequency parameter in the interface OpenConfig YANG module was motivated by the assumption that an IP controller would primarily use the interface YANG model to define transceivers, making an optional frequency parameter the most practical solution for the IP controller to have access to optical parameters. This decision was also driven by the simplicity of managing 10/25G tunable transceivers, which are not coherent and therefore have less complex management requirements. While we still believe that associating the frequency parameter with the interface model was appropriate at the time, we acknowledge that scaling to support coherent pluggables requires a different approach. The adoption of the component YANG model, commonly used for transponder devices, is more suitable for managing next-generation coherent pluggables. This is the option (ii) that we mentioned at the beginning of section 3.4.5. Despite this, many existing infrastructures still rely on lower-capacity tunable transceivers, making our solution relevant for facilitating a smooth transition between legacy hardware and next-generation devices entering the market.

Similarly, the choice of management software such as Stratum and ONOS was highly supported during the initial phase of this work and showed significant promise. However, the industry has since shifted its focus toward other solutions, such as SONiC. SONiC now supports ZR/ZR+ pluggables and benefits from contributions by major companies actively improving its codebase. Nonetheless, many advanced features in SONiC remain part of proprietary NOS implementations developed by individual vendors, and that SONiC does not support the frequency tuning of tunable transceiver if they are not coherent. To this date, the only demonstration of ZR/ZR+ management with open-source software and disaggregated hardware was done by R. Morro, et al. [94], where they augmented

the SONiC version with a coherent manager - that includes a CMIS application - to access the coherent pluggables.

This implementation required a containerised application/agent running on the switch to manage the optical parameters of the pluggable transceiver, rather than integrating this functionality directly within the NOS. This method has the benefit of decoupling the coherent optics management from the NOS itself, which makes it easier to update or modify the container independently of the NOS. It is also easier to deploy this containerised solution across different NOS, making it adaptable to a wide range of use cases and offers a practical path for early adoption and experimentation. However, this approach introduces an additional layer of interaction between the NOS, the container, and the controller, which increases latency and potential for failure. Managing the optical layer separately from the packet layer can result in fragmented decision-making. The controller might need to reconcile data from multiple sources to make holistic decisions, which could be more complex and error-prone.

On the other side, while integrating the optical management directly into the NOS requires more development effort upfront, Stratum's open and modular architecture allows direct access to optical parameters alongside packet-forwarding parameters via the same northbound gNMI interface. This makes it simpler for controllers to interact with the network. For future-proofing, having a single, unified interface for both optical and packet layers (as in the Stratum approach) may ultimately result in better synergy between the optical and packet layers, aligning with trends like SDN-driven automation.

3.8 Conclusion

This chapter demonstrated how pluggable transceivers can be effectively integrated into an open-source Network Operating System. We illustrated how the frequency parameters of tunable modules can be exposed to a remote, open control plane, enhancing the flexibility and efficiency of managing packet-over-DWDM architectures and networks. By bridging the gap between the packet and optical layers, this integration enables seamless management and control across both domains.

While we have successfully made optical and IP parameters accessible through the same framework, the next step is to enhance the controller's decision-making capabilities. This involves integrating insights from both the optical and packet

layers into a unified control mechanism. Currently, these two layers are managed independently, often by separate teams, which can limit overall network efficiency and coordination. Achieving a cohesive management approach will unlock the full potential of combined optical and packet-layer orchestration.

Chapter 4

Wavelength provisioning in open packet-optical SDN controller

Chapter 4 explores the state-of-the-art packet-optical management approaches within the SDN framework. It then demonstrates how controllers can leverage packetponders with wavelength conversion to enhance Routing and Wavelength Assignment (RWA) algorithms and argues for its integration into existing SDN controllers. Additionally, the chapter highlights the use of packet-optical emulation tools for testing and scaling SDN controllers, ensuring their effectiveness in larger network environments. As part of this work, the packet-optical emulation software was adapted and extended to support the experimental evaluation. In this chapter, we then evaluated an SDN controller with an enhanced RWA algorithm across two different network topologies, each subject to various link failures: a random complex mesh topology; and a topology representing GEANT — Europe’s leading high-speed research and education network.

A unified platform that integrates pluggable optics into routers and switches is the first step toward seamless packet-optical network convergence. However, to match or exceed the efficiency of traditional architectures with transponders and muxponders, it must also be intelligently managed and controlled. This requires a controller that is aware of both the packet and optical layers, and makes decisions that fully leverage the capabilities of optical transceivers, including coherent pluggables. Achieving this level of optimisation relies on tight coordination between the packet and optical layer controllers, ensuring efficient resource allocation and dynamic network adaptability.

4.1 Current packet-optical SDN controller: wavelength assignment algorithm and failure recovery strategy

4.1.1 Controllers coordination in traditional packet-optical networks with transponders

Historically, the packet and optical domains operated separately, relying on distinct frameworks and being managed by separate teams with different expertise. As illustrated in Fig. 4.1, the packet SDN controller typically controls switches using protocols such as gNMI, gNOI, P4Runtime, or OpenFlow, while the optical SDN controller manages elements like transponders, Open Line Systems (OLS), Reconfigurable Optical Add/Drop Multiplexers (ROADMs), and amplifiers. To provide network operators a comprehensive view of the entire network topology, a higher-layer orchestrator needs to be used to aggregate data from both controllers, facilitating service provisioning, automation, and cross-layer optimisation; for instance, ensuring that an IP path has a valid optical path beneath it.

The need for multi-layer controllers in packet-optical networks has been widely studied to determine the most effective way for operators to gain full network visibility. I. Maor *et al.* [95] were the first to demonstrate a multi-layer SDN controller using commercial packet switches and optical transponders. While each device was managed by its respective vendor-specific controller, they developed a higher-level orchestrator capable of overseeing the entire testbed through a REST-based interface. In [96], researchers leveraged protocols such as NETCONF and OpenFlow, extending OpenFlow into OpenFlow+ to incorporate optical layer resources. More recently, in [97], authors adopted more modern and widely used standard interfaces, employing NETCONF/OpenConfig for transponder management and the ONF Transport API (TAPI) for OLS configuration. Their demonstration showed real-time IP and optical layer reconfiguration using a hierarchical controller with commercial network equipment, highlighting the feasibility of unified, software-driven control across both domains. These works successfully coordinate packet and optical layers but their topology rely on the use of transponders.

Today, the introduction of packet-optical nodes with pluggable transceivers (such as the *packetponder*) as transponder replacements, necessitates a redesign of

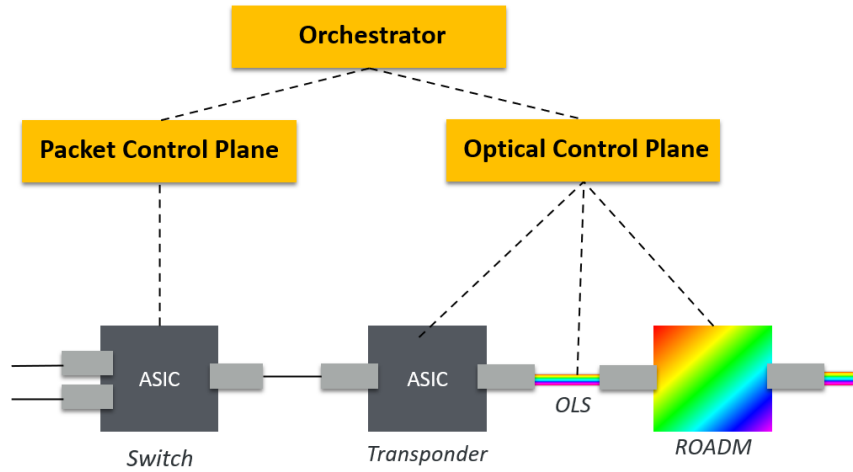


Figure 4.1: Traditional network architecture with SDN control.

the SDN control architecture. In the traditional approach, optical and packet devices operate as separate entities, making their control straightforward, with the orchestrator simply retrieving information from both layers. However, with the introduction of packet-optical node and pluggables as discussed in Chapter 3, a single device now belongs to both the optical and packet layers, requiring a more integrated and coordinated control framework.

As depicted in Fig. 4.2 with the red dotted lines, a key challenge emerges: who controls the pluggables — and by extension, the Layer 2 switch? More specifically, who has read access to the pluggables, and who has write access? Since both the packet and optical teams require visibility and control over the system, a well-coordinated control framework is essential to ensure seamless management, prevent conflicts, and optimise network operations across both layers.

4.1.2 Hierarchical controller and abstraction interfaces for packet-optical node with pluggables

This key challenge of the coexistence and coordination of both packet and optical controllers to effectively manage the pluggables inside the router has received some attention recently. E. Riccardi *et al.* [98] have evaluated the potential of a hierarchical controller (a parent controller on top of both layer's controller) against a join management between the packet controller and the optical controller (no third controller instances), and while the hierarchical solution seems to simplify the workflow between the controllers, it might add some latency due

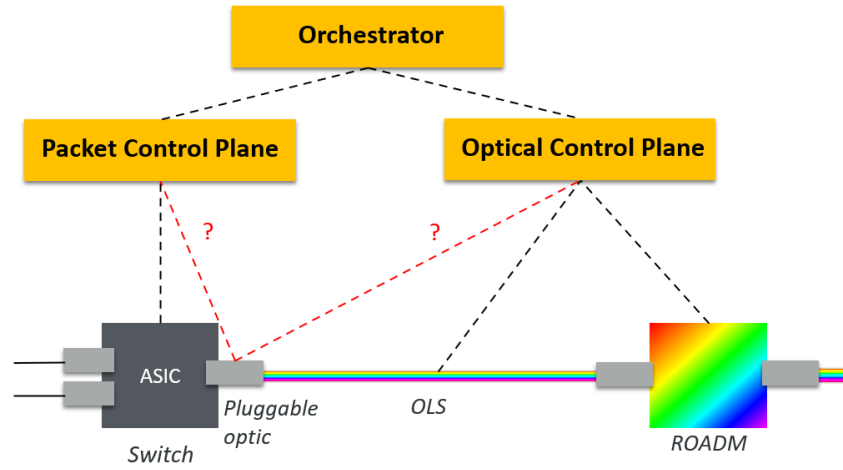


Figure 4.2: Control conflict (red dotted line) between packet and optical layers when using switch+pluggable optics.

to the adoption of a third instance of controller. On the other hand, the use of a join management implies that both packet and optical controllers communicate with the packet-optical device, which can lead to complex coordination workflows and potential issues during maintenance with firmware/software updates. D. Scano, A. Giorgetti and O. Gerstel [77], [99], [100] have acknowledged the difficulty of synchronising the router controller with pluggable transceivers and the OLS, and have provided a solution based on a hierarchical controller communicating to both packet and optical control-plane. A. Giorgetti *et al.* [99] have successfully demonstrated such a controller using open-source software such as SONiC and ONOS.

The authors in [100] highlight a critical aspect of how the hierarchical controller should interact with packet-optical nodes. The key aspect being discussed is not only who (which controller) manages the pluggables but also how they are controlled, particularly through which APIs/data models (OpenConfig, TAPI, ...). Indeed, TAPI abstracts transport network elements (such as ROADMs, transponders, and optical links) and provides APIs for SDN controllers to manage and provision end-to-end services in optical transport networks. It is therefore well used in optical SDN controllers and is worth considering to use for packet-optical node management. However, TAPI is designed for transport (optical) network elements, whereas packet-based switches (Layer 2/3) are typically managed using OpenConfig models, through P4Runtime, or gNMI protocols. That is why O. Gerstel argues against integrating TAPI into the packet controller framework, as introducing it at the packet level could unnecessarily complicate the architecture.

While [77] proposes a dual approach, where both the packet and optical controllers have direct access to the pluggables, O. Gerstel *et al.* [100] point out that this could lead to dual-write conflicts, where multiple controllers attempt to modify the same device simultaneously. The author concludes by giving his preferred solution which is: the hierarchical controller communicates with the packet controller, which then interfaces with the routers using OpenConfig, which avoids the dual write problem. We agree with this perspective and believe that the packet controller should independently manage both packet-related configurations and pluggable transceiver settings. This can be efficiently achieved using OpenConfig, the interface we implemented and demonstrated in Chapter 3, ensuring a streamlined, vendor-agnostic approach to managing pluggables within packet-optical switches.

4.1.3 Failure recovery state-of-the art in SDN controllers

In SDN-controlled optical networks, robust recovery strategies are necessary to maintain service continuity and minimise disruptions. These strategies are generally classified into two main categories: proactive and reactive approaches [101].

In proactive methods, backup paths and resources are precomputed and reserved before any failure occurs, which means a pre-provisioned optical backup path is established in advance, including spare pluggables in IP-over-DWDM nodes. This pre-planning allows for immediate switchover to backup paths when a failure is detected, resulting in minimal service disruption. However, this approach can lead to under-utilisation of network resources, as the reserved backup capacity remains idle during normal operations.

Reactive methods involve computing alternative paths and reallocating resources in response to a failure. Either performing an optical restoration: the recovery is handled entirely within the optical domain by reconfiguring ROADMs, without modifying coherent pluggables or involving the packet controller; or by performing a hybrid restoration: both the optical and packet controller cooperate, allowing changes to the coherent pluggables' frequency slot and modulation format [92]. While the latest approach can be more resource-efficient, it may result in longer recovery times due to the need for on-the-fly computation and controller coordination.

4.1.4 Routing and Wavelength Assignment (RWA) in SDN controllers

Effective coordination is also crucial for wavelength assignment in an optical network. The approaches discussed in Section 4.1.2 effectively provide a global network view and enable the provisioning of optical pluggables with the appropriate wavelengths, but we believe that the full potential of packet-optical nodes is not yet fully leveraged within the SDN controller.

Indeed, while open SDN controllers can incorporate Optical Signal-to-Noise Ratio (OSNR) for quality-of-transmission estimation (QoT-E) within their algorithms [102], such controllers often lack full visibility of available resources and may not exploit *all* possible packet-optical lightpath alternatives. This deficiency is evident in RWA algorithms that mandate end-to-end wavelength continuity between packet-optical end-point nodes, and can result in restricted bandwidth availability, especially when nearing capacity.

Although RWA are critical in flexible networks, in this thesis a simple RWA was implemented to enable comparative simulation experiments when using a *packet-ponder*. The development of novel RWA algorithms was therefore beyond the scope of this work. However, a solid understanding of RWA principles remains essential and is reviewed in the following section.

4.1.4.1 RWA studies over the years

RWA algorithms for optical networks have been extensively studied over the years, with numerous comparative analyses conducted to evaluate their performance. Researchers have simulated various network topologies, applying different routing and wavelength assignment strategies to assess factors such as blocking probability, spectrum efficiency, and network adaptability. These studies have provided valuable insights into the strengths and limitations of different RWA approaches, contributing to the development of more efficient and dynamic wavelength allocation methods.

RWA algorithms can be divided in two parts: the routing policy and the resource (wavelength) assignment logic. The most common used path computation is the Shortest Path routing (e.g., Dijkstra [103], Bellman-Ford [104]), which is very simple but does not consider wavelength availability, leading to possible blocking. Some other example are Load-Balancing routing (or Least Congested Path) [105]–[107] that routes connections through paths with the most avail-

able wavelengths to avoid congestion, or Constraint-Based routing [108], [109], which uses additional constraints such as link utilisation, latency, or signal quality to determine the best path. Once the route is selected, the resource (here the frequency/wavelength) must be assigned while considering network constraints. The most common and simple algorithm for this is the First-Fit algorithm which selects the lowest-index available wavelength along the computed path. Random-Fit algorithm is another one that randomly selects an available wavelength from the set of free wavelengths, and the Least-Loaded algorithm selects the wavelength with the least overall usage across the network to balance load. Newer version of RWA have been studied, notably hybrid RWA that uses real-time monitoring and feedback from the SDN controller to adjust routes and wavelengths dynamically [110], or machine learning-based RWA [111] to predict traffic patterns and optimise wavelength assignment. Numerous highly effective RWA algorithms have been developed for packet-optical networks, but our primary focus is on the impact of wavelength conversion and its potential to enhance resource utilisation.

Wavelength conversion is a technique used in RWA algorithms to enhance flexibility in optical networks by allowing an optical signal to change wavelengths at intermediate nodes rather than maintaining the same wavelength end-to-end. This helps alleviate wavelength continuity constraints, reduces blocking probability, and improves network resource utilisation. The concept was extensively studied in the 1990s, when researchers explored all-optical wavelength conversion to optimise wavelength reuse in WDM networks. It proved effective in freeing bandwidth, reducing blocking probability and minimising wavelength conflicts [80], [112], [113]. However, its implementation in reality was commercially challenging whether based on O/E/O or O/O/O (all-optical), due to the cost of wavelength converter devices. The prevailing view historically and currently, is that a standalone wavelength converter device represents an additional CapEX and OpEX expense. It is also an active device and so is power-hungry to cater for events that rarely occur in a network. Moreover, since wavelength conversion did not occur frequently enough to justify the cost, operators preferred simpler wavelength routing strategies that did not rely on conversion. As a result, early WDM networks were designed with strict wavelength continuity constraints, meaning a signal had to remain on the same wavelength from source to destination.

With the advent of pluggable coherent transceivers and software-defined networking, wavelength conversion has regained interest as a cost-effective way to improve spectral efficiency, reduce blocking, and enable more dynamic and

flexible optical network architectures [114]. Our *packetponder* concept is capable of O/E/O wavelength conversion and 3R regeneration using open-source software control, common YANG data models, and standardised protocols. The wavelength conversion can be performed at marginal, incremental cost inherent in the *packetponder*, with no additional equipment or severe downtime, so the approach warrants reconsideration [115].

4.1.4.2 RWA in current SDN controllers

In open-source SDN controllers like ONOS and OpenDaylight, RWA algorithms are typically customisable, meaning that network operators or developers can implement different RWA strategies depending on their needs. The controllers provide modular architectures where custom path computation logic can be integrated into the control plane. ONOS, as an example, offers support for optical networks through applications such as the Optical Disaggregated Transport Network (ODTN) [46]. This application facilitates the management of multi-vendor optical equipment by providing a platform for implementing RWA algorithms.

However, in most of the experimental demonstration of SDN packet-optical controller, wavelength conversion is not applied. Typically, when a new service request is made, the hierarchical controller interacts with the optical controller (and optical controller only) to compute the routing and wavelength assignment [116], meaning that only the optical devices are considered to create a route. The hierarchical controller then uses the returned optical parameters to instruct the IP controller to configure the pluggable transceivers on the router. In this model, the pluggables are treated merely as end-points, and the RWA algorithm considers only the optical layer, without accounting for packet layer considerations. As a result, within the core intelligence of the SDN controller, the packet and optical layers remain fundamentally separate, rather than being tightly integrated. This limits the ability to fully exploit the flexibility and new functionalities that packet-optical nodes offer, such as dynamic wavelength conversion, in-switch optical regeneration, and cross-layer optimisation — all of which could enhance network efficiency and resilience.

To the best of our knowledge, no existing SDN control framework has been demonstrated using switches and pluggables where packet-optical nodes function as intermediate (tandem) nodes rather than end-points. Current SDN implementations typically treat optical pluggables as fixed transponders at the network edge, enforcing wavelength continuity along each path between end-points. This

approach limits network flexibility by requiring the same wavelength to be maintained across the entire optical route.

In contrast, our packet-optical node introduces wavelength conversion capabilities, allowing it to serve as an intermediate node within the optical path. This enables more efficient spectrum utilisation, dynamic re-routing, and improved fault recovery, as wavelengths can be reassigned at various points instead of being constrained by end-to-end continuity. Such an approach enhances the scalability and adaptability of packet-optical networks, yet has not been fully explored in existing SDN architectures.

4.2 Proposed multi-layer controller and routing decision

In this Chapter, we contend that the *packetponder* ameliorates the CapEx/Opex within a device that includes programmable forwarding and transponder functionality. In essence, wavelength-tunability provides greater flexibility — less end-to-end wavelength blocking — to free-up stranded wavelength channel resources. For example, in cases where insufficient "free" wavelength channels are available e.g. due to end-to-end wavelength continuity, *packetponders* could liberate, on a hop-by-hop basis, dis-contiguous end-to-end optical channels by reconsolidating several discrete wavelength channels of lower bandwidth into a singular discrete wavelength channel and so maximise the bandwidth utilisation [117]. This ability to dynamically re-route analogue carriers, by tuning wavelengths can provide relief to excessively occupied access/metro/core networks requirements and improve spectral utilisation.

While in section 4.1.2 we discussed how packet and optical controllers should be coordinated, here, we suggest a hierarchical controller where the packet controller manages pluggable transceivers, as shown in Fig. 4.3 (red dotted line). The pluggables are accessed via the gNMI interface using OpenConfig models. While we do not rule out dual access or more advanced coordinated control mechanisms, we emphasise that network-wide decisions must be made with a comprehensive, multi-layer view. For this reason, we refer to our controller as a multi-layer controller, ensuring a tight integration between the packet and optical domains.

To enhance RWA, we introduce an approach that accounts for both switches

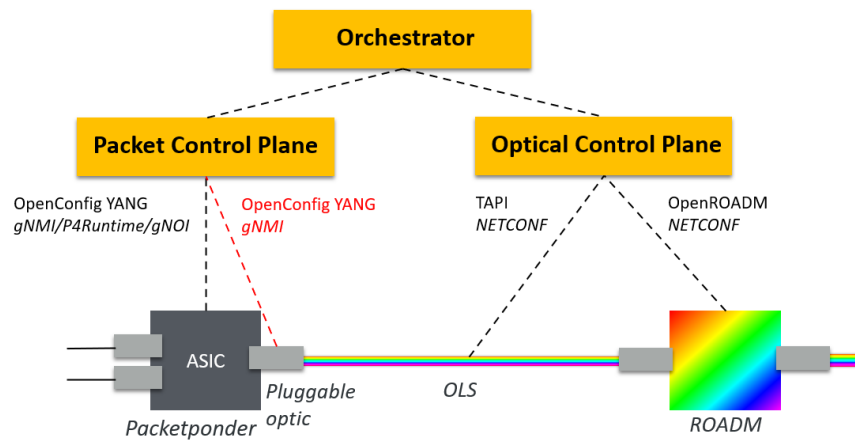


Figure 4.3: Network architecture and SDN control with *packetponders*.

and ROADMs, allowing the packet-optical node to function as a tandem node, which means it can act as a network element that is neither the source nor the destination of a signal but, instead, processes and forwards it along the optical path. Unlike traditional end-point transponders, a tandem node can perform intermediate processing such as wavelength conversion and optical regeneration.

While our proof-of-concept and use case focused on failure recovery, the same principles apply to wavelength provisioning for new service requests. The following sections will demonstrate how integrating both layers into the controller’s decision-making allows for more efficient lightpath setup. However, due to the limited physical resources in our lab — a single packet-optical node and one ROADM — we used Mininet-optical, an emulation tool, to further evaluate the benefits of wavelength conversion and, more broadly, the *packetponder*’s role in software-defined packet-optical networks.

In the following demonstration, we present a failure recovery scenario where wavelength conversion enhances the number of lightpaths that can be restored. The experiment was conducted on two different network topologies emulated in the mininet-optical [118] software tool: a random ROADM-based mesh network, and a close representation of the GÉANT network [119], which connects national research and education networks across Europe and providing high-capacity, low-latency connectivity for scientific collaboration, research, and education.

4.3 Open-source multi-layer simulator and emulator: Mininet-optical

Mininet-optical [118] extends Mininet [120] by integrating optical network modelling with existing IP and Ethernet emulation, providing a cross-layer testing platform for SDN control planes [121]. Mininet itself is a widely used, open-source network emulator that creates realistic virtual networks on a single machine, supporting features such as switches, hosts, links, and controllers for rapid prototyping and experimentation. While other SDN emulation and simulation tools exist, Mininet has become the de facto standard in research and education due to its simplicity, flexibility, and broad community support. Building on this foundation and with the same goal in mind, Mininet-optical was created to simulate physical optical impairments, emulate packet and optical data planes, and expose SDN control APIs for seamless network management. It provides a software-based environment for testing and developing packet-optical SDN architectures, as shown in Fig. 4.4.

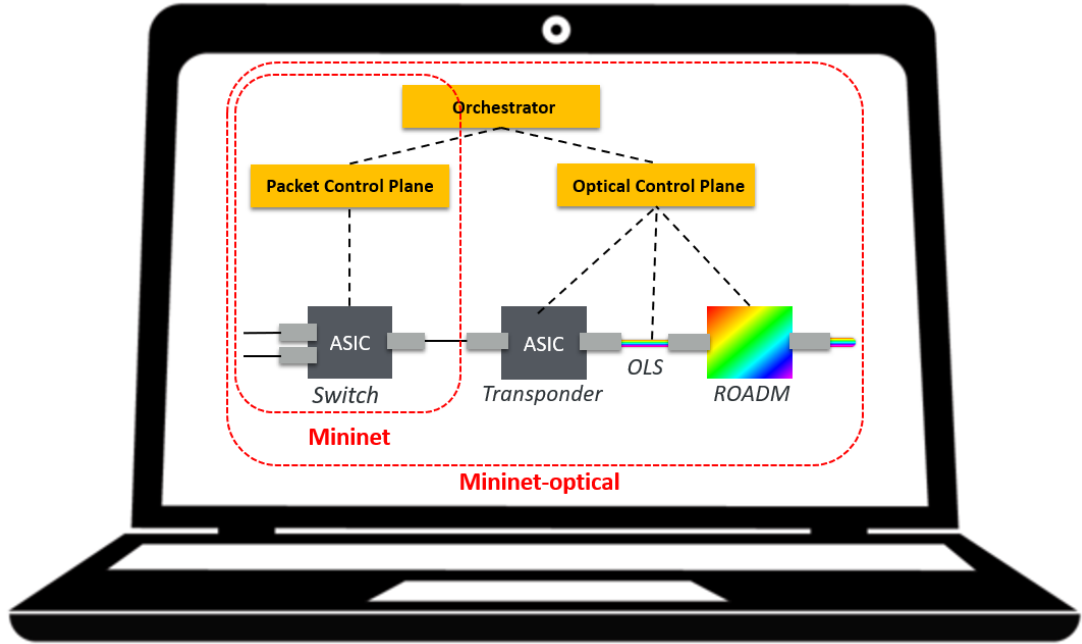


Figure 4.4: Virtual testbed: Mininet and Mininet-optical.

The reason why mininet-optical was created is because a major challenge in packet-optical research is the reliance on expensive, small-scale testbeds, which lack full modelling of physical, data plane, and control plane behaviours. While packet networks can be emulated through virtual network interfaces, optical networks require simulation of light transmission with optical devices functional-

ities that are emulated and accessible via the controller. To address this, Mininet-optical integrates an optical transmission simulator via a built-in API, allowing the instantiation of optical components (e.g., transponders, ROADMs, EDFAs) within a virtual network environment. Optical elements are abstracted as electronic Mininet switch instances, with optical fibre links represented as virtual Ethernet links. Optical parameters are encapsulated in packet headers, enabling optical transmission modelling within the emulated network [122].

For SDN integration, Mininet-optical includes a REST server written in Python, allowing OpenFlow-based SDN controllers or any other controller relying on REST, to manage the emulated packet-optical devices. By enabling optical impairment simulation and SDN-driven network control, Mininet-optical serves as a cost-effective platform for testing packet-optical network architectures and their control algorithms. Besides REST, Mininet-optical can also be extended to support future implementation of standard APIs such as OpenROADM or OpenConfig for plug-and-play compatibility with hardware networks.

Mininet-optical is an open-source project developed through a collaboration between Trinity College Dublin, the Centre for Integrated Access Networks (CIAN) at the University of Arizona, and the Mininet open-source community. Since the start of my PhD, I have been actively involved in the project, contributing to the codebase and conducting use-case experiments to validate its functionality.

In this work, we build upon and extend Mininet-optical as a packet-optical Digital Twin to enable large-scale SDN controller testing. Specifically, we implement a *packetponder* node model, scale packet-optical network topologies, and evaluate a controller capable of demonstrating the critical role of wavelength conversion in modern packet-optical architectures.

4.3.1 Packetponder node within Mininet-optical

Mininet-optical was originally designed to support key optical network elements, including transponders (called Line Terminals in Mininet-optical), ROADMs, and Erbium-Doped Fibre Amplifiers (EDFAs). In this framework, the closest abstraction to our *packetponder* is the Transponder/Line Terminal node, which is abstracted from real OVS-switch instances [123], for which the Switch model from Mininet has been extended to distinguish between electronic and optical Python classes. The Line Terminal node (the optical classe) is equipped with transceivers which are launching the optical signal through the network, and also receiving

the signal at the other end.

Typically, in Mininet-optical, when a signal traverses the network, optical parameters are encapsulated within the packet header, and extracted at each node to be processed by the simulation engine. After computation, the updated optical information is embedded back into the headers, and the packet is forwarded to the next node based on the defined switching rules in both the optical and electronic domains.

In real-world networks, transponders traditionally enforce a static mapping between client-side (grey interface) and line-side (coloured interface). In Mininet-optical, this means that the original conception of the Line Terminal node only allowed an optical transceiver port to be mapped to an Ethernet port and vice versa. Since this static mapping does not align with the *packetponder*'s capabilities, we extended the Line Terminal functionality to support line-side to line-side transmission. This enhancement allows the signal to perform O/E conversion and being forwarded through another line-side transceiver (E/O conversion), effectively enabling in-switch regeneration within the emulated environment, and potential wavelength conversion.

A simple example is illustrated in Fig. 4.5 which uses the new model of the Line Terminal, enabling wavelength conversion. The topology consisted of two hosts (h1 and h2 from Fig. 4.5) and three line terminals (t1, t2, t3 from Fig. 4.5), where the line terminals are enhanced with O/E/O (Optical-Electrical-Optical) conversion between two line-side transceivers, effectively transforming them into *packetponders*. The goal of this experiment was to establish two separate wavelength channels: one from t1 to t3 (red) and another from t3 to t2 (green).

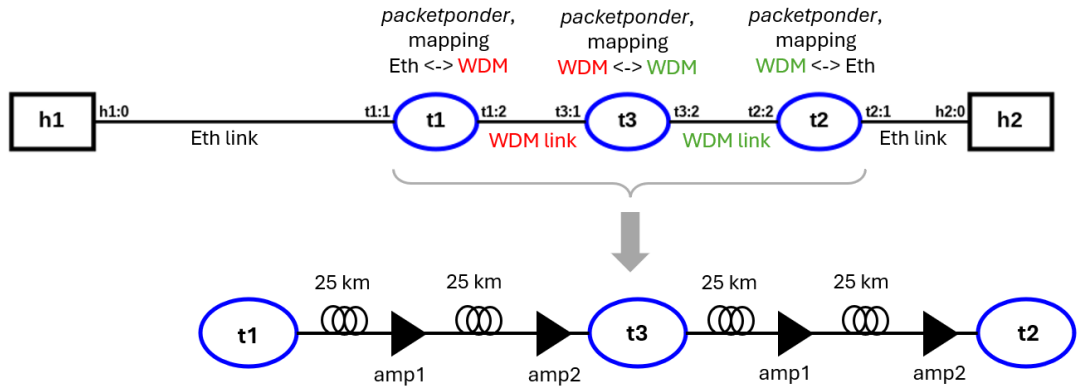


Figure 4.5: 3-nodes topology to test the *packetponder*'s implementation in mininet-optical. "h": host, "t": terminal, or *packetponder*.

The setup included WDM fibre links with two optical amplifiers, each configured with a target gain of 5.5 dB, following the span pattern: [25 km standard single mode fibre, amp1, 25 km standard single mode fibre, amp2] as depicted in Fig. 4.5. Once the topology (data plane) is emulated, the REST controller was used to populate the network with the appropriate wavelength channels.

4.3.2 Software-defined control of the *packetponder* element with Mininet-optical

The current APIs in Mininet-Optical uses REST, so all network elements are configured and controlled via the available REST APIs. To support wavelength conversion, the existing "connect" function, which links two ports of a network element (ROADM or Line Terminal), was extended to allow two WDM ports to be connected. In this setup, t1 and t2 function similarly to the previously supported transponder model, while t3 must forward the signal from one WDM port to another and change the associated wavelength. The corresponding controller is depicted in Fig. 4.6. Here, the curl commands are used to send RESTCONF requests that install forwarding rules on terminals t1 and t2 (a single wavelength is set on the WDM side). On the *packetponder* node t3, two channels (or wavelengths) are configured: one coming from t1 at port 1 (channel 1, shown in red) and one going to t2 at port 2 (channel 2, shown in green). This configuration enables the ports to switch between channels, thereby performing OEO and wavelength conversion.

```
#!/bin/bash -x
set -e
# For now, devices are all at the same IP address and port
mn=localhost:8080; t1=$mn; t2=$mn; t3=$mn
echo "* Configure hairpin.py network"
curl="curl -s"
$curl "$t1/connect?node=t1&ethPort=1&wdmPort=2&channel=1"
$curl "$t2/connect?node=t2&ethPort=1&wdmPort=2&channel=2"
$curl "$t3/connect?node=t3&wdmInPort=1&wdmPort=2&inChannel=1&channel=2"
$curl "$t1/turn_on?node=t1"
$curl "$t2/turn_on?node=t2"
$curl "$t3/turn_on?node=t3"
echo "* Done."
```

Figure 4.6: *packetponder* control using REST in mininet-optical.

To verify the network behaviour, we monitored the signals traversing the topology. Fig. 4.7 presents the results after configuring the network through the controller:

Fig. 4.7(a) shows a successful "ping" command from h1 to h2, confirming the seamless integration of the optical domain with the packet layer and the correct configuration of t1, t2, and the *packetponder* t3. Fig. 4.7(b) displays the Mininet-Optical CLI output, indicating successful bidirectional signal reception between t1-t3 and t3-t2. The OSNR (Optical Signal-to-Noise Ratio) values are calculated at each link's endpoints, with the observed high OSNR attributed to the short fibre spans and multiple amplifiers in the setup. Fig. 4.7(c) monitors the signals entering t1 and t2, clearly showing the frequency shift: 191,350 GHz between t1 and t3, and 191,400 GHz between t3 and t2. This confirms the successful end-to-end connection between the hosts, with t3 performing both wavelength conversion and signal regeneration.

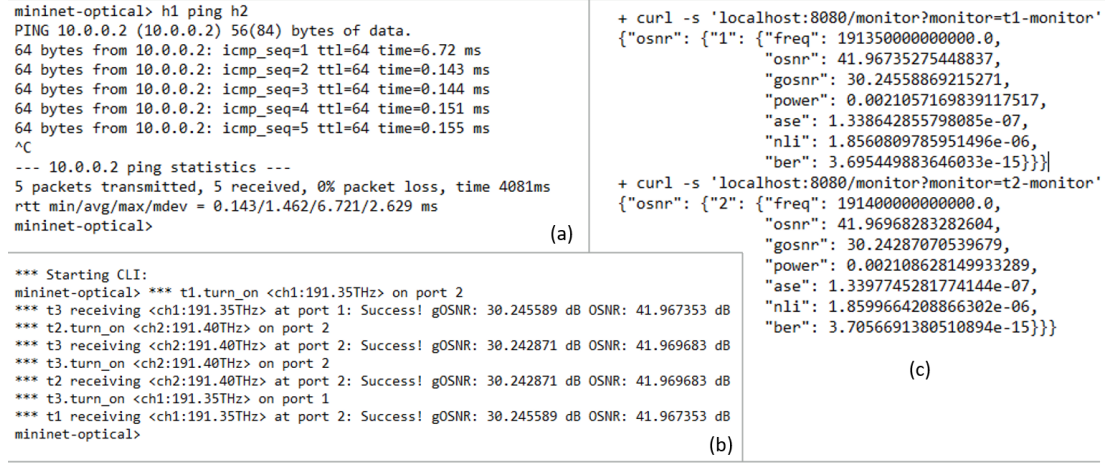


Figure 4.7: (a) successfully end-to-end connectivity between h1 and h2; (b) Mininet-optical CLI showing successful reception of bi-directional signal at t1, t2 and t3; (c) Monitoring function showing associated parameters to signal arriving at t1 (up) and t2 (down).

4.4 Failure recovery scenario with packetponders in a 9-ROADMs mesh topology

Now that we have developed an optical element that emulates the behaviour of a *packetponder* and provided a control API for its configuration, we can design and evaluate a controller that leverages this element to offer greater flexibility and release additional end-to-end routing paths to maximise the collective bandwidth throughput [JRC4].

Indeed, in this section, we show how a single open-source SDN-based network controller can simultaneously co-optimize end-to-end packet and optical data-

paths by exploiting the wavelength conversion feature of a *packetponder*. This feature augments a RWA algorithm within an SDN controller to relax end-to-end constraints, by using multi-hop path routing instead.

4.4.1 Emulated testbed components

We emulated a 9-POP (Point of Presence) network in a ring configuration (as depicted in Fig. 4.8): each POP consisted of a host *h*, an electronic packet switch *s*, an optical terminal (or *packetponder*) *t*, and a ROADM *r*. We assumed *packetponders* were implemented at every POP, and Terminal node *t* models the *packetponder*. The ROADMs were linked by two bidirectional standard single mode optical fibre links of 50km, and each span loss was compensated by ED-FAs with target gains of 11 dB. The choice of topology was arbitrary and tried to simulate a real network scenario as best as possible. The number of degree chosen for each ROADM is also arbitrary and can be adjusted in the dataplane creation of the topology. The *packetponders* are configured to support up to 64 transceivers, with the assumption that all deployed transceivers are tunable.

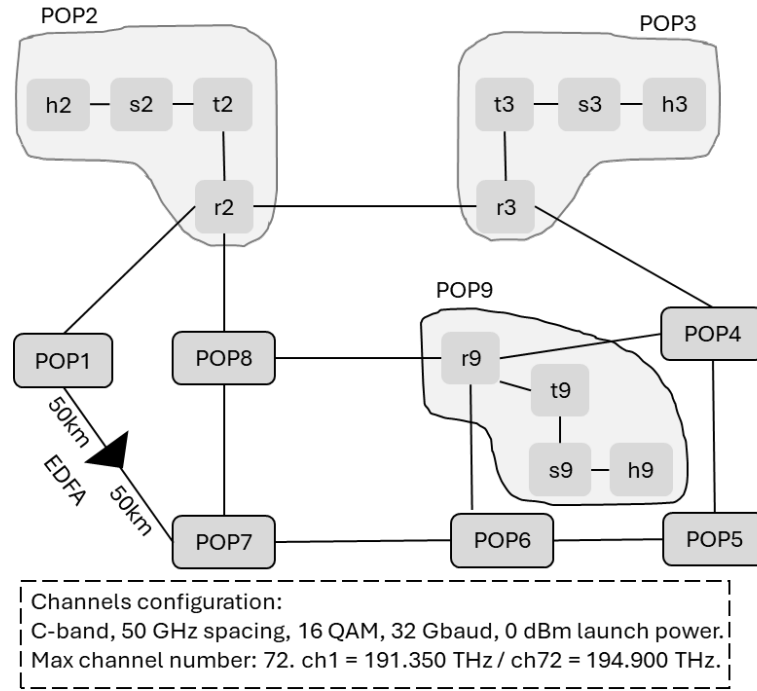


Figure 4.8: 9-POP topology for failure recovery experiment. POP1-POP3 is an example of link configuration and the same applied between every POP connected to each other.

4.4.2 Network configuration and load

Each lightpath (LP), or demand, was established from terminal to terminal using a simple shortest-path (S-P) algorithm to determine the end-to-end path. The first-fit (F-F) algorithm was then applied to assign a wavelength, selecting the lowest available channel index across the entire path (referred to "standard RWA"). Channels were configured as per depicted in Fig. 4.8: 1) channels in the C-band, 2) 50 GHz spacing, 3) 16 QAM modulation format, 4) 32 Gbaud symbol rate, 5) 0 dB launch power. We arbitrarily set the maximum number of channels supported by the network to 72, with channel 1 = 191.350 THz and channel 72 = 194.900 THz. To fully populate the network, we generated 230 lightpaths (LPs) with random source and destination nodes among the 9 available terminals. To ensure a uniform testing environment, we set the demand limit to 230 LPs for all simulation runs. We assume all ROADMs to be colourless and directionless. A colourless ROADM allows any wavelength to be added or dropped on any port, rather than being restricted to specific wavelengths [124]. A directionless ROADM enables any wavelength to be switched to any direction within the node.

In the initial setup, a lightpath is successfully configured if: (i) the transmitter (TX) of the source node and the receiver (RX) of the destination node each have at least one available TX-RX pair of transceivers; (ii) the wavelength is available on all fibre links along the path; (iii) the wavelength is not already added or dropped at the source or destination ROADM (this is a non-contentionless scenario, where two signals on the same wavelength cannot be added/dropped at the same node without interference); and (iv) the maximum number of channels per fibre is 72.

The third constraint arises from the Mininet-optical core system, which currently does not support a single line terminal handling the same wavelength on multiple ports/transceivers. If a lightpath encountered any of these four limitations, it was dropped. Once the network reached capacity, i.e., wavelength unavailability in certain hops, any additional LPs were dropped. The choice of 230 demands was based on preliminary simulations, which revealed that after about 215 configured LPs, blocking became consistent and no more LPs could be established before encountering one of the four network constraints. The objective was to fully utilise network capacity while accounting for the random initialisation of lightpaths. The configuration of ROADMs, terminals, and switches with their switching rules was then sent to the emulated network via a set of python APIs exposed in the northbound interface of mininet-optical data-plane, using REST.

4.4.3 Fault recovery scenario

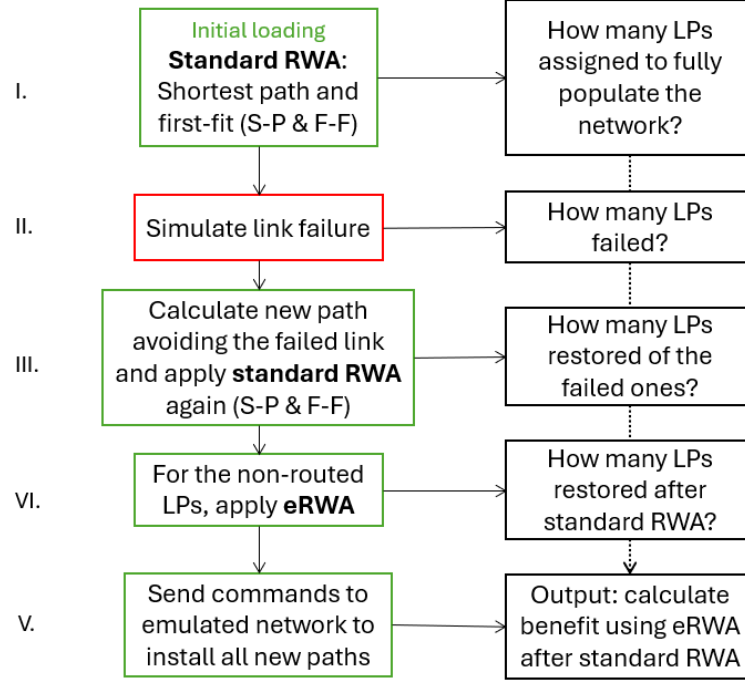


Figure 4.9: Network configuration and failure recovery steps during the experiment.

Once the network is fully loaded (step I in Fig. 4.9), we simulated a link failure between two ROADMs (step II), which gives us the number of failed lightpaths, i.e., the number of LP that were traversing the failed link. In the example shown in Fig. 4.10, the broken link was r8-r9, causing all LPs traversing that link to fail.

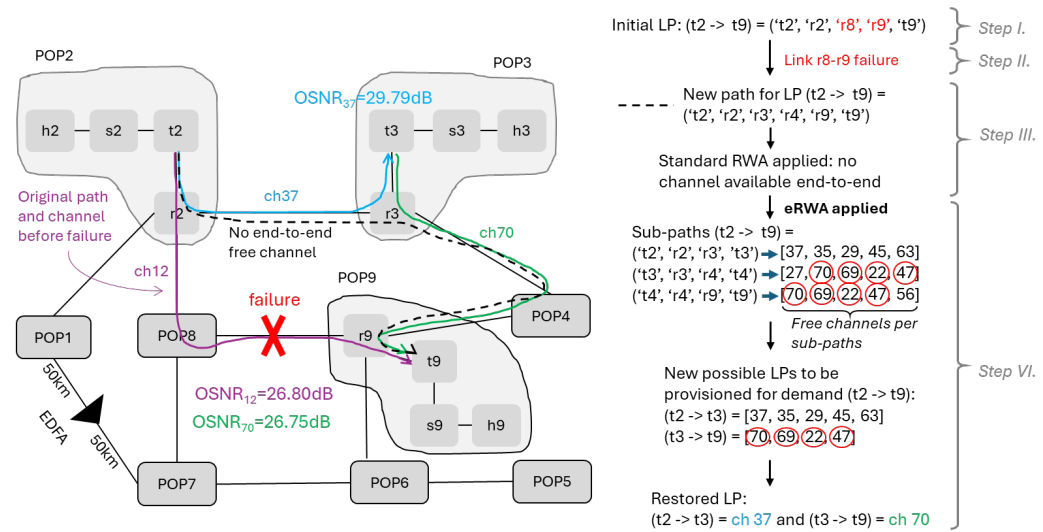


Figure 4.10: Emulated topology in mininet-optical with failure example r8-r9.

A typical approach to evaluate our algorithm would be to perform a statistical

analysis, comparing the results of applying the standard RWA after a link failure against those obtained using our enhanced RWA (eRWA) with wavelength conversion after the failure. This would help determine which algorithm performs better. However, the standard RWA relies on shortest path routing combined with first-fit wavelength assignment, which is a relatively basic approach. As a result, outperforming it would not be particularly challenging. Instead, we chose to assess the added value of our solution by applying it after the standard RWA has already been executed to try to recover LPs after the failure. This allows us to determine whether any "extra" additional lightpaths can be recovered.

In this way, the controller first attempted to reroute these failed LPs using the same standard RWA algorithm (step III), and for the LPs that could not be recovered, a second attempt was made using our enhanced algorithm (eRWA) (step IV). This algorithm leveraged *packetponders*, using their line-side to line-side forwarding with wavelength conversion and 3R regeneration.

Algorithm 1 Enhanced RWA (eRWA) pseudo code

Input: • Non-routed pairs (src $\rightarrow$ dst): failed lightpaths (LPs) that are not re-routed with standard RWA
• Network $\mathcal{N}$ information (neighbours, links, ...)

Output: • New LPs to be installed: set of sub-paths with associated λ

```

1: for pair (src,dst) of LP in non-routed pairs do
2:   Break down the path of pair into sub-paths including packetponder
   elements as 3R regenerator and  $\lambda$  converter.
3:   for each sub-path do
4:     Calculate non-used  $\lambda$  per link and non-used  $\lambda$  per roadms com-
       posing the path.
5:   end for
6:   for each sub-path if there is at least one free  $\lambda$  per path do
7:     if two consecutive sub-paths share at least one free  $\lambda$  then
8:       Merge both sub-paths (i.e., remove the packetponder node of
       the path and use the roadm as pass-through) and associate
       all common free  $\lambda$  to the merged sub-path.
9:     else
10:      Store current sub-path with new  $\lambda(s)$ 
11:    end if
12:  end for
13: end for

```

Figure 4.11: eRWA pseudo code.

The pseudo-code of our eRWA is depicted in Fig. 4.11, which takes all the LPs that failed with the standard RWA, and returns the restored LPs which can be added using wavelength conversion, if any.

In the example in Fig. 4.10 with the purple arrow, LP $t2 \rightarrow t9$ failed as the LP was traversing through r8-r9. The standard RWA re-computed the new shortest

path to be $t2 \rightarrow t9$ via $r3$ and $r4$. However, wavelength continuity cannot be preserved, so the LP is "dropped" at step III of Fig. 4.9. Here, we then apply eRWA to enable wavelength conversion with *packetponders* at $t3$ and $t4$, restoring and re-enabling this demand. The eRWA algorithm in Fig. 4.11, operating under the same constraints (i), (ii), (iii) and (iv) outlined in the previous section, first recalculates the available channels for each sub-path (Line 4). It then identifies consecutive sub-paths that share common available channels (Line 7), allowing the signal to traverse multiple hops without requiring wavelength conversion at every node (Line 8). In such cases, the signal remains optically transparent in these consecutive sub-paths, meaning it is not dropped at any of the ROADMs along the route.

For LP $t2 \rightarrow t9$; ($t3$, $r3$, $r4$, $t4$) and ($t4$, $r4$, $r9$, $t9$) shared free channels: 70, 69, 22, 47 as circled in red in Fig. 4.10, therefore both sub-paths were merged to form ($t3$, $r3$, $r4$, $r9$, $t9$) and the channel 70 was chosen, as shown with the green arrow. In this example, for initial LP $t2 \rightarrow t9$, the new provisioned LP was: $t2 \rightarrow t3$ with channel 37 (blue), and $t3 \rightarrow t9$ with channel 70 (green). Here, $t3$ performed wavelength conversion and regenerated the signal.

4.4.4 SDN controller for emulated network

The new paths were automatically sent from the SDN controller to the emulated network elements of mininet-optical data-plane (step V from Fig. 4.9), sending instructions for both packet switching (to the *packetponders*) and circuit switching (to the ROADMs) from a single controller.

ROADM rules	*** Installing path ($t2$, $r2$, $r3$, $t3$) channels [37]
	connect {'node': $r2$, 'port1': 15, 'port2': 82, 'channels': '37', 'action': 'install'}
	connect {'node': $r3$, 'port1': 81, 'port2': 12, 'channels': '37', 'action': 'install'}
	*** Installing path ($t3$, $r3$, $r4$, $r9$, $t9$) channels [70]
	connect {'node': $r3$, 'port1': 40, 'port2': 82, 'channels': '70', 'action': 'install'}
	connect {'node': $r4$, 'port1': 81, 'port2': 83, 'channels': '70', 'action': 'install'}
	connect {'node': $r9$, 'port1': 83, 'port2': 6, 'channels': '70', 'action': 'install'}
Packetponder rules	*** Configuring packetponders connect {'node': $t3$, 'ethPort': None, 'wdmPort': 120, 'inChannel': 37, 'channel': 70, 'power': 0.0, 'wdmInPort': 92}

Figure 4.12: New rules to be installed for the ROADMs $r2, r3, r4, r9$ and the *packetponder* $t3$.

In our previous example, the enhanced algorithm generates a set of new rules to be implemented in the network, configuring both the ROADMs along the new paths, and the *packetponder* $t3$ (Fig. 4.12). All configuration commands were

translated into OpenFlow rules and sent to the emulator via the REST interface. The advantage of using Mininet-optical is that the same SDN controller and eRWA implementation could potentially be applied to both software-emulated and hardware networks with minimal adjustments. Unlike traditional simulation tools such as the Python-based RWA Simulator over WDM Optical Networks [125], Optical Network Simulator [126], Net2Plan, or SimulNet — Mininet-optical provides a more realistic testing environment by using actual network stacks and protocols. This enables realistic interactions between software and hardware, making it a suitable platform for both theoretical analysis and practical implementation.

Additionally, Mininet-optical supports a NETCONF interface for ROADMs configuration, and a gNMI interface could be integrated to replicate hardware-based methods for programming pluggables within the *packetponder*, further improving its applicability to real-world deployments.

4.4.5 Random ROADM-mesh network results

The improvement in lightpath (LP) availability using eRWA can be quantified by measuring the number of LPs successfully restored. To account for the randomness in initial LP generation, the previously described experiment was repeated 100 times across various link failure scenarios, ensuring that different LPs were blocked in each run.

To perform a statistical analysis of 100 runs efficiently, the main code was executed in simulation mode within Mininet-optical. This means that after the topology is emulated and all nodes/links are created, the code only runs within the simulator and no configuration commands are sent to the emulator, which saves processing time.

4.4.5.1 Analysis of Recovery Attempts for Link Failure r8–r9

Figure 4.13 illustrates the results of 100 recovery attempts following a failure on link r8–r9. The figure distinguishes between: LPs recovered using standard RWA (blue); additional LPs recovered after applying eRWA (red); and LPs that remained unrecoverable (grey). On average, 192 LPs were successfully assigned before the failure, though this number varies across runs due to random LP generation.

This link failure scenario was chosen because r8–r9 occupies a strategic central

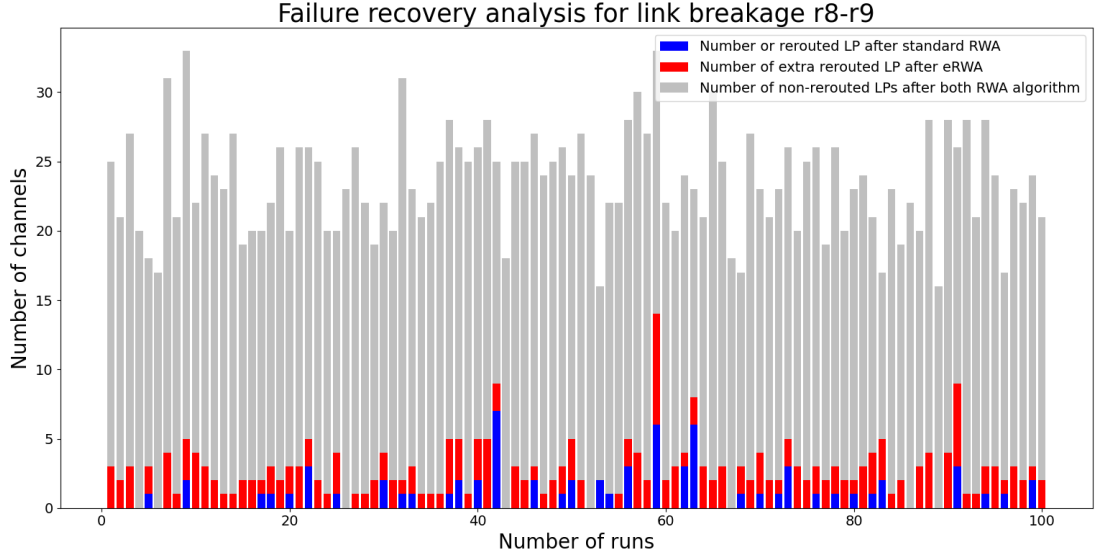


Figure 4.13: Link failure of a less-congested central link in the network, forcing re-routing through highly congested links and limiting standard RWA performance (blue), as opposed to eRWA which improves recovery (red).

position within the network topology. Due to the shortest-path routing algorithm, r8–r9 is less congested compared to r2–r3 and r6–r7, which consistently emerge as the most congested links in nearly every simulation. As a result, when r8–r9 fails, traffic must be rerouted through these already congested links, making recovery extremely difficult. This is evident in Figure 4.13, where only a few blue bars appear, indicating that the standard RWA alone cannot restore most of the failed lightpaths. However, when wavelength conversion is enabled, the network can exploit the few available wavelengths (of different frequencies) on the congested links, as shown by the red bars, demonstrating the clear benefit of eRWA and the use of wavelength conversion in *packetponders*.

This makes r8–r9 a critical failure point, requiring careful monitoring by operators, as signal recovery may be challenging in the event of a failure. To mitigate this risk, additional redundancy mechanisms, network redesign, or improved lightpath allocation strategies should be considered to enhance overall resilience.

The improvement achieved with eRWA can also be calculated using the following formula:

$$\text{Gain (\%)} = 100 \times \frac{\text{Number of extra rerouted LPs using eRWA}}{\text{Number of failed LPs after link failure}} \quad (4.1)$$

For the r8–r9 link failure, the average gain using eRWA is 8.49%, with improvements reaching up to 24%. The variation in results is due to the random distri-

bution of initial LPs, which can lead to congestion on certain links.

Since recovery performance is also highly dependent on which link fails, we extended our analysis by simulating additional failure scenarios within the same topology but affecting different links.

4.4.5.2 Analysis of Recovery Attempts for Link Failure r3–r4

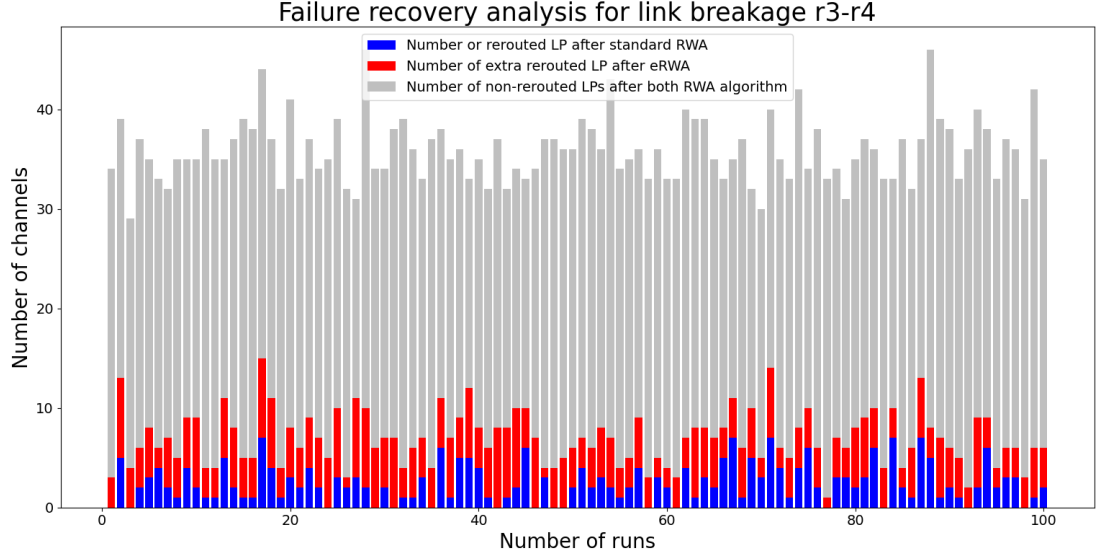


Figure 4.14: Link failure of a moderately congested link, enabling more LPs to be recovered via less congested paths using standard RWA (blue). Additionally, eRWA (red) consistently improves recovery by leveraging wavelength tuning, allowing the network to relax wavelength continuity constraints over longer paths.

For the r3–r4 link failure, the average gain using eRWA is 12.46%, with improvements reaching up to 25%. Figure 4.14 shows that in this scenario, at least one LP is always recovered, and both standard RWA and eRWA demonstrate improved performance, with a higher average number of recovered LPs. This is partly because more LPs fail initially, but the better gain can be explained because r8–r9 starts with lower congestion. As traffic is rerouted through this link, a greater number of LPs can be restored. However, due to this re-routing process, r2–r8 becomes congested, significantly limiting standard RWA, as seen by the few recovered LPs (blue bars). In contrast, eRWA performs well, as the *packetponder* at t8 (POP8) can tune wavelengths, allowing traffic to traverse between the congested r2–r8 link and the less congested r8–r9 path, effectively improving recovery rates.

Overall, in this scenario, eRWA consistently improves recovery, with red bars

appearing in every test run, demonstrating its effectiveness.

4.4.5.3 Analysis of Recovery Attempts for Link Failure r6–r7

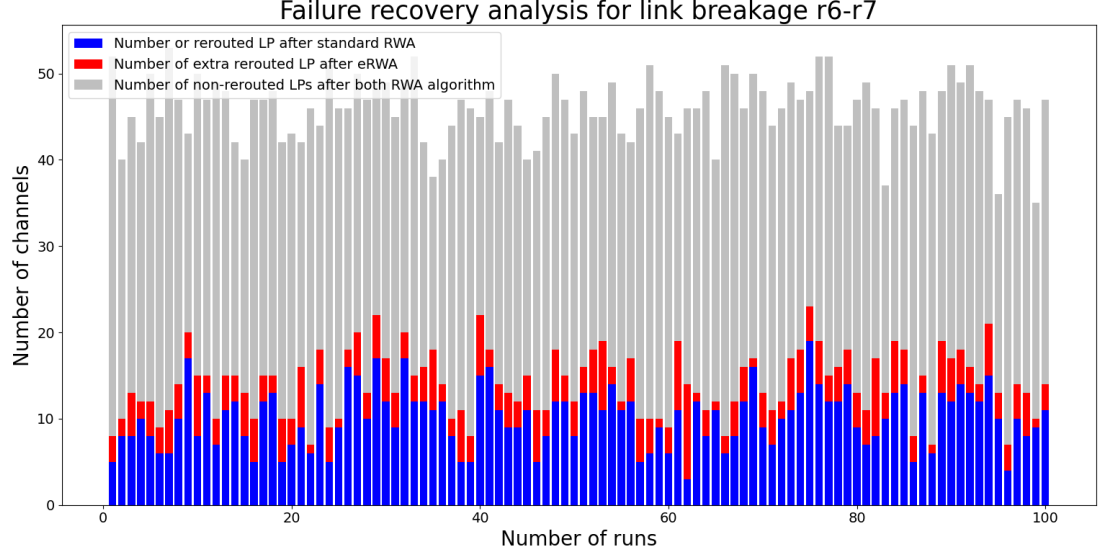


Figure 4.15: Recovery from a highly congested link failure, demonstrating that while standard RWA performs well, eRWA consistently identifies additional paths, enabling more demands to be successfully restored.

For the r6–r7 link failure, the average gain, calculated using Equation (4.1), is 8.36%. This is slightly lower than in the previous scenario because the standard RWA already performs well, recovering a significant number of LPs before eRWA is even applied. The reason for this is similar to the r3–r4 failure case: (i) a high number of initial LP failures creates more re-routing opportunities; (ii) most recovered LPs can be re-routed through r8–r9, benefiting from the under-utilise path due to the shortest algorithm being applied; and (iii) unlike the previous use-case, r6–r7 is surrounded by multiple alternative paths, allowing traffic to be better distributed between r2–r3 and r8–r9, reducing overall blocking. However, even in this case, eRWA remains beneficial, enabling additional LP recovery beyond what standard RWA achieves: Figure 4.15 reveals a 100% success rate in recovering additional LPs after eRWA is applied.

These three examples collectively highlight the advantages of eRWA when using *packetponders*, providing greater flexibility in LP allocation and enhancing network resilience. As shown by the figures, the variations in the results graphs for each link failure highly depend on the location of the failed link. As will be discussed in the conclusion of this chapter, shortest path allocation plays a significant role in determining how many demands can be recovered, whether

using the standard RWA or our eRWA approach.

Enabling wavelength conversion per hop allows for the configuration and reconfiguration of more lightpaths. On a larger scale, this enables localised reconfiguration during failures by adjusting the wavelength per hop, leaving the rest of the configuration unaffected. This is possible as long as the SDN controller has an end-to-end view of both the packet and optical networks, including visibility and knowledge of *packetponder* elements, which can serve not only as endpoints for lightpaths or demands but also as intermediate nodes, providing greater flexibility.

4.4.5.4 QoT and latency measurements

In mininet-optical, the Optical Signal to Noise Ratio (OSNR) can be obtained at every node and notably at the receiver end and interrogated through the control plane interface. We retrieved the OSNR of the channel under test, the purple channel 12 from Fig. 4.10, before and after the failure for comparison. Wavelength conversion doesn't impact the integrity of the signal, as Fig. 4.10 shows a marginal change in the OSNR at the destination t9 of the restored path (26.75 dB) when compared to initial LP (26.80 dB).

Furthermore, to ensure that the new LP is actually effective, a simple Ping Ethernet Device (PING) operation was completed, as depicted in the right-hand side of Fig. 4.16, showing the end-to-end restoration of the new LP t2 → t9, and the successful end-to-end connectivity of the packet transmission between h2 and h9.

The PING test also provides latency measurements. While the values shown in Fig. 4.16 do not accurately reflect physical network conditions and may not be entirely reliable for real-world performance analysis, they are still useful for relative comparisons before and after reconfiguration. As expected, the reconfigured path is longer and traverses more hops, resulting in a slight increase in latency, from an average of 0.162 ms to 0.210 ms. However, this increase is negligible considering the benefit of enabling more lightpaths to be routed through the network. This approach could serve as an effective temporary solution while awaiting physical link repairs.

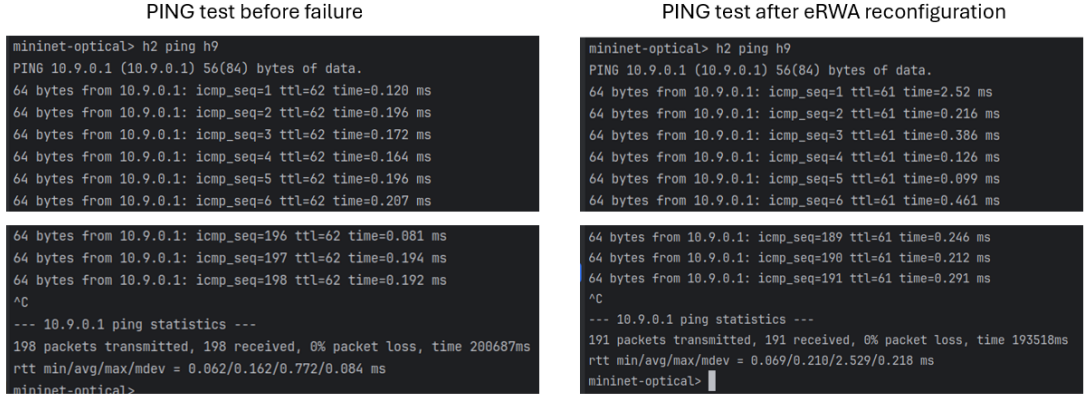


Figure 4.16: PING test from h2 $\rightarrow$ h9 before and after reconfiguration, showing latency measurements and successfully end-to-end connectivity of the new light-path t2 $\rightarrow$ t9 using λ conversion.

4.5 GEANT network topology experiment

The performance and efficiency of eRWA depends heavily on which link fails and its position within the network, as well as the overall network topology. While we are confident that eRWA, when combined with a more intelligent routing algorithm, can easily outperform standard RWA — those which lacks wavelength discontinuity capabilities — we acknowledge that its effectiveness is highly dependent on the topology. To further evaluate its performance, we tested it on a more realistic network topology: the GÉANT network [119]. The GÉANT network is a large data network that connects national research and education networks across Europe. It provides fast and reliable connectivity for universities, research centres, and scientific projects, making it easier for researchers and students in different countries to work together. Supported by the European Union and its members, GÉANT is widely used in fields such as physics, astronomy, and life sciences, where huge amounts of data need to be shared quickly and securely.

As discussed in the previous section, shortest-path routing does not consider physical link distances. Therefore, we modelled the GÉANT network based on its topology shape, rather than replicating its exact link lengths — which is a feature that can be adjusted in Mininet-optical. Our primary focus was to analyse eRWA's behaviour in a topology like GÉANT, rather than simulate precise transmission impairments.

For this experiment, each link between cities (PoPs) was set to 150 km, with two EDFAs positioned at 50 km and 100 km. Figure 4.17 illustrates the simplified

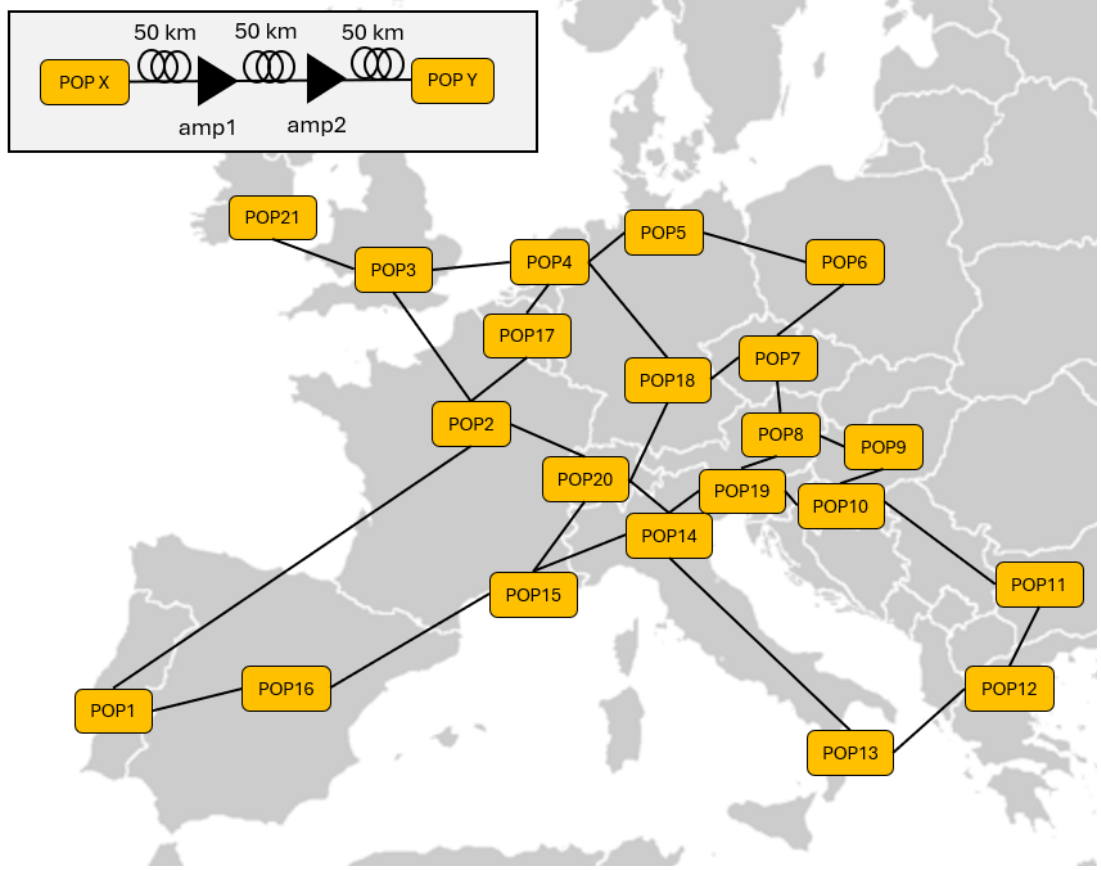


Figure 4.17: Simplified version of the GEANT optical network, linking European cities together.

GEANT topology, which does not include all ROADMs in the actual network. The switches, *packetponders*, and ROADMs were configured identically to previous experiments and deployed at every network hop. To fully load the network, we generated 1,210 random traffic demands per run, with a maximum of 72 wavelength channels per link.

The first link failure test was conducted on r8–r9 and is shown in Fig. 4.18, a link that turned out to be highly congested. Although, at first glance, this link did not appear to be a congestion hotspot, the shortest-path algorithm led to heavy traffic concentration on it, making it a critical point in the network. As observed in the previous topology, standard RWA performs well in this scenario, successfully recovering a high number of LPs. This is primarily due to the multiple alternative less congested routing paths available in the network around that link. However, eRWA once again proves beneficial, consistently enabling the recovery of additional lightpaths.

For the r18–r20 link failure in Fig. 4.19, which is less congested but centrally

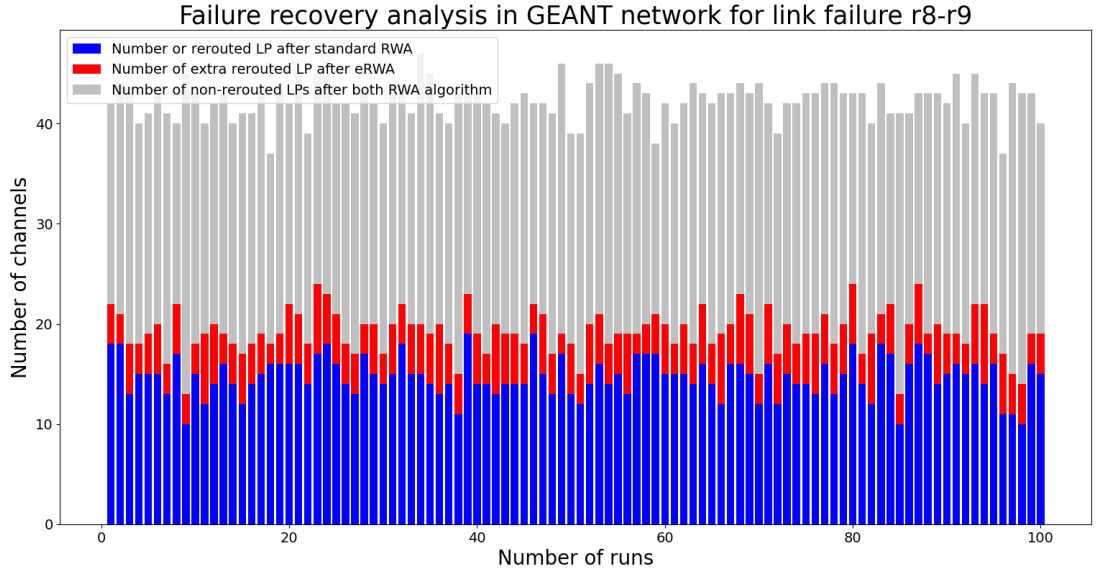


Figure 4.18: Recovery from a highly congested link failure, demonstrating that while standard RWA performs well, eRWA consistently identifies additional paths, enabling more demands to be successfully restored.

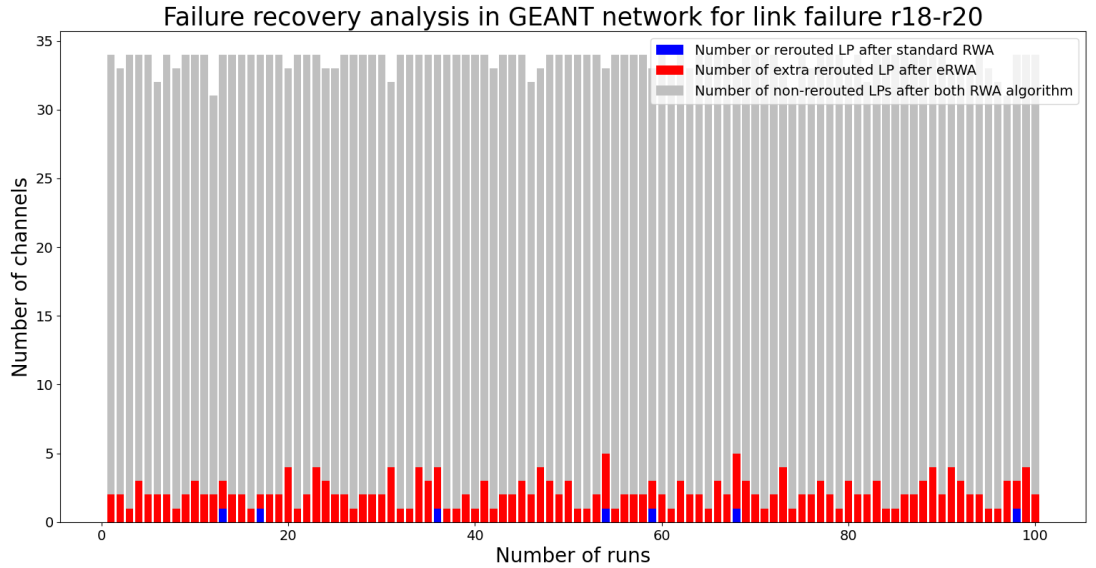


Figure 4.19: Link failure of a less-congested central link in the network, forcing re-routing through highly congested links and highly limiting standard RWA performance (blue), as opposed to eRWA which improves recovery (red).

located within the network, the results are consistent with the previous topology. Out of 100 runs, standard RWA successfully recovered LPs in only 7 instances, whereas eRWA was effective in every case. This can be attributed to the fact that the surrounding links were already congested, making re-routing difficult when maintaining wavelength continuity. By enabling wavelength conversion, eRWA overcomes this limitation, allowing for more successful lightpath recovery.

While the primary objective of this demonstration was to demonstrate the effectiveness of eRWA, it also highlighted the potential of Mininet-optical as both an SDN controller and a network testing platform. First, it enabled real SDN control over an emulated network, using state-of-the-art SDN interfaces, protocols, and standards; and second, different failure scenarios could be assessed by using the simulator, making Mininet-optical a valuable tool for network resilience testing. This approach is similar to the "Chaos Monkey" strategy [127], introduced by Netflix in 2011, which tests IT infrastructure resilience by intentionally disabling components to evaluate failure recovery. Similarly, we showed several link failure at different places in the network topology, revealing that certain links are more vulnerable than others, as recovery methods fail to restore a significant number of lightpaths. Such insight can assist network operators in optimising their network by *a priori* decanting traffic flows and demands away from crucial points of failure.

4.6 Discussion and conclusion

We demonstrated a single open-source SDN controller performing end-to-end optimisation of disaggregated packet-optical networks, exploiting dynamic wavelength (re)allocation following a hop-by-hop approach. This approach substitutes end-to-end wavelength continuity with judicious per-hop O/E/O wavelength conversion, enabling more lightpaths to be configured and reconfigured, particularly for localised failure recovery, without disrupting the entire network. This optimises bandwidth usage during fault scenarios and enhances network availability. The *packetponder* could potentially make wavelength conversion within RWA algorithms economically viable, a concept that was previously considered too costly. This flexibility relies on the SDN controller's end-to-end visibility of both packet and optical networks, recognising *packetponder* elements as both endpoints and intermediate nodes, enhancing overall network adaptability.

While the algorithm used in this paper focused only on wavelength availability, it is important to remember that *packetponders* can also provide 3R regeneration capabilities, which could potentially enhance the algorithms further. As mininet-optical contains a simulator, it should also be possible to integrate OSNR-based QoT-E into the controller alongside wavelength decisions and create a physically aware adaptive RWA. Given the mix of signal rates traversing today's networks, integrating the *packetponder*'s flexibility into open SDN controllers is a useful addition to addressing issues such as fragmentation or underutilised bandwidth,

ultimately enabling more efficient channel allocation.

Our enhanced Routing and Wavelength Assignment (eRWA) algorithm is a simplified version of what could be implemented in more complex scenarios. In this work, we applied eRWA only after a failure, following the standard RWA process. However, an alternative approach could apply eRWA immediately after a failure or even from the start, during network provisioning. While all-optical paths are often preferred, some longer transmission paths could benefit from 3R regeneration and wavelength conversion to maintain signal quality. However, the aim of this study was not to compare different strategies and RWA algorithms but instead to demonstrate the advantages of incorporating wavelength conversion into an SDN-controlled ROADM-based network, particularly for flexible and efficient failure recovery.

One limitation of our specific eRWA implementation is that it requires redundant hardware, including additional transceivers and fibre links between the *packetponders* and the ROADMs at each Point of Presence (POP). But while these resources are reserved, they can also be shared across multiple backup routes for different services, making them useful in various failure scenarios rather than being dedicated to a single recovery path.

Another limitation, as observed in the results section, is the reliance on Dijkstra's shortest path algorithm for routing. This significantly restricts the number of LPs that can be recovered, even when using eRWA. In our current implementation, the shortest path is recalculated after the link failure, and only one path per demand (i.e., between two endpoints) is selected. Ideally, if eRWA fails to reroute a lightpath, the controller should attempt to compute an alternative route, exploring longer paths with or without additional wavelength conversion and regeneration. While this approach may introduce longer transmission distances, potentially impacting signal quality, many real-world networks have under-utilised links. If latency is not a critical factor, selecting an alternative path could improve overall network efficiency and LP recovery rates.

This chapter demonstrates how the optical and packet layers can be seamlessly managed while leveraging the benefits of both. Although this approach may introduce additional management and control complexity, the flexibility it provides is invaluable. While the *packetponder* and its management framework appear to be an ideal solution for bridging the packet and optical layers in metro and core networks, an important question arises: can this flexibility extend across multiple domains, including the access network? If so, the *packetponder* could

evolve into a fully converged packet-optical node at the network edge, bridging multi-layer and multi-domain architectures. This would enable operators to manage a single, unified platform, rather than maintaining multiple separate devices for different services, and this is the subject of the next chapter.

Chapter 5

Packetponder for access, metro and core networks

Chapter 5 introduces the concept of the packetponder at the network Edge: we report a unique and novel architecture for the consolidation of metro and access network services (i.e. point-to-point leased lines, point-to-multipoint Passive Optical Networks (PONs) and 5G fronthaul transport). The novel architecture is based on the use of the packetponder that can accommodate a variety of modular, common-form factor pluggable transceivers. We show experimentally the feasibility of a packetponder at the edge, aggregating mixed signals from the access domain with common form factor pluggables and open software/APIs, enabling transmission of an aggregated 99.56 Gbit/s signal (single wavelength channel) through 80km single-mode fibre. This work is based on published work at ONDM 2024 [JR3] and IEEE Photonics Society Summer Topicals Meeting Series 2024 [JR4].

Fundamental redesign and simplification of the network architecture are needed. By unifying hardware and consolidating functionalities onto fewer, more versatile platforms, operators can streamline management and enable cross-domain interoperability. Beyond operational efficiencies, this approach has the potential to contribute significantly to energy savings by reducing the number of devices, transmission lines, and associated components.

5.1 Edge and services consolidation

Convergence at the edge refers to the integration of multiple network functions and service types into a common, shared infrastructure located close to end-users. Traditionally, the access and metro layers were built and operated as separate domains, each with dedicated equipment and management systems. However, advances in SDN now make it possible to extend centralised control, programmability, and automation from the core and metro domains down into the access layer.

While both research and industry are actively pursuing network simplification and consolidation, most hardware and software solutions remain proprietary and lack the flexibility required for multi-vendor environments. Currently, there is no open whitebox platform, managed by open software, that can support both access clients and metro use cases within the same device. Most existing open solutions are concentrated in metro networks, leveraging advancements in coherent transceivers to enable cost-effective IP-over-DWDM deployments as shown in the last two chapters. This section explores the innovations in the access domain as well as several technologies that are driving edge and access network consolidation, particularly in optical networks, by simplifying infrastructure, reducing hardware needs, and improving efficiency.

5.1.1 Software-Defined Passive Optical Networks (SD-PON)

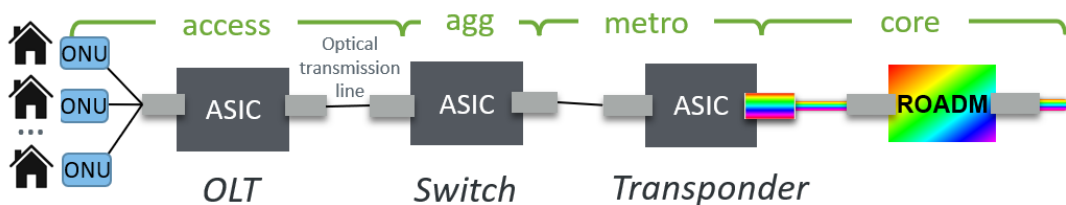


Figure 5.1: Typical PON architecture.

Passive Optical Networks (PONs), as shown in Fig. 5.1 have been a cornerstone of access infrastructure for decades, providing cost-effective, high-capacity connectivity to residential and business users. As the basis for most Fibre-to-the-Home (FTTH) deployments, they now face growing bandwidth demands driven by modern applications.

Traditionally, these systems relied on vendor-specific management and static configurations, which limited flexibility and slowed service innovation. The in-

roduction of SDN into PON environments offers an opportunity to simplify architectures, improve interoperability, and enable programmable solutions that integrate seamlessly with other network domains. Applying SDN to PON in access networks essentially treats Optical Line Terminals (OLTs) as enhanced Layer 2 switches [128], but achieving a fully Software-Defined PON requires overcoming challenges such as interoperability and standardisation issues (PON equipment vendors often implement proprietary management interfaces) or the indirect management of each home-based Optical Network Units (ONUs).

Disaggregated PON solutions represent a transformative approach in broadband access networks, emphasising the separation of hardware and software components to enhance flexibility, scalability, and cost-effectiveness. Central to this paradigm is the decoupling of the OLT functions from proprietary hardware, enabling integration with SDN controllers and facilitating the deployment of pluggable OLT modules within standard network devices.

Currently, fully open, disaggregated platforms that natively support PON applications are still emerging, but there are a few key initiatives and solutions that integrate PON with open networking and optical transport.

5.1.1.1 SDN-controlled OLTs

One of the main challenges in PON networks is the lack of standardisation across different technologies and vendors. There are multiple PON variants — including GPON [129], XG-PON [130], 50G-PON [131], EPON [132], and 10G-EPON [133] — each complying with different standards bodies (e.g., IEEE PON, ITU-T PON) and using vendor-specific software to manage and control the OLT. This fragmentation means that no universal PON device or management interface exists, making interoperability and multi-vendor integration complex.

To address this, researchers and industry groups have been working on software abstraction layers that provide a unified interface for controlling different OLT chipsets in a standardised way. A comprehensive state-of-the-art review of these efforts can be found in [134], which highlights various approaches to abstracting PON functions into SDN controllers. Several works [135]–[137] have chosen OpenFlow as the protocol for managing OLTs, leveraging its existing use in packet-based devices (for switches and routers). The idea is to represent OLTs as virtual switches within SDN controllers, allowing configuration and reconfiguration through standard SDN commands. For instance, the study in [134] used gRPC to connect an OLT abstraction layer with an SDN agent, enabling seamless

integration with ONOS. This approach allows remote management and traffic rule installation, which, in ONOS, is traditionally only possible for switches. The SDN agent translates OpenFlow rules from ONOS into PON-specific commands for the OLT, ensuring a uniform control mechanism.

To further enhance multi-vendor compatibility, the community has been converging towards open-source SDN projects aimed at creating a common software layer (SDN agent) to translate SDN commands into proprietary OLT configuration commands. Notable initiatives include VOLTHA (Virtual OLT Hardware Abstraction) [138]; and Open Broadband - Broadband Access Abstraction (OB-BAA) [139].

VOLTHA:

VOLTHA, a hardware abstraction layer initially developed by ONF, was created to act as an intermediary between the SDN controller, such as ONOS, and the individual OLTs. VOLTHA is an open-source project now under the Linux Foundation's LF Broadband initiative, which abstracts PON hardware, presenting it as a programmable Ethernet switch to SDN controllers. This abstraction allows for multi-vendor interoperability and simplifies the management of PON networks. VOLTHA has been successfully deployed in production environments by major operators such as Deutsche Telekom and Türk Telekom [140]. It provides a north-facing OpenFlow interface, allowing ONOS to manage OLTs thanks to vendor-specific adaptors translating between OpenFlow and each OLT. Moreover, since ONOS needs awareness of the ONUs, which are not directly controllable via OpenFlow or other APIs, the architecture provides a switch abstraction on top of the OLT and its connected ONUs. ONOS then treats this as a logical switch, allowing it to recognise events such as a customer powering up their home ONU, represented as a "port active" event on the switch-like model. ONOS then executes the necessary actions through a suite of control applications, enabling efficient management of the PON.

OB-BAA:

Open Broadband - Broadband Access Abstraction (OB-BAA) is another open-source project facilitating SDN-based management and control of multi-vendor, multi-technology access networks via a standardised abstraction northbound interface. OB-BAA simplifies the integration of various access device types, including legacy implementations, under a unified management and control framework.

These efforts represent significant progress toward disaggregated, open, and programmable PON networks, eliminating vendor lock-in and enabling seamless integration with SDN controllers. And although control and management are becoming more centralised, these solutions still depend on a dedicated OLT chassis which add costs, space, and higher power consumption.

5.1.1.2 MicroPlug OLT and control

At the same time, advances in access networks and the evolution of OLTs facilitated the deployment of MicroPlug OLT transceivers [141]. These pluggable OLTs, characterised by a compact Small Form-Factor Pluggable (SFP) design, can interface with Ethernet switches and routers, effectively transforming them into OLT devices. TiBiT Communications was the first start-up to commercially propose such device (referred as "Tibit" in this Chapter), potentially removing the need of blackbox/whitebox dedicated OLTs. This innovation reduces the physical footprint and power consumption compared to traditional chassis-based OLTs.

Although PON devices are standardised, meaning MicroPlug OLTs could theoretically be integrated into a switch's NOS, in a similar way to coherent pluggables as we demonstrated in Chapter 3, several challenges remain in SDN-based OLT and ONUs configuration. This has made the seamless integration of MicroPlug OLTs into Ethernet switches more complex.

One major challenge is the in-band control required to access and configure the MicroPlug OLT. Unlike traditional OLT devices, modular MicroPlug OLTs do not have a separate management port, meaning that management traffic is overlaid into data traffic rather than being physically separated. This creates additional complexity, as the control plane must be designed to distinguish and handle management flows appropriately. Furthermore, PON functionalities such as ONU authentication, Dynamic Bandwidth Allocation (DBA), and ONU Management Control Interface (OMCI) are fundamentally different from what a standard Ethernet switch supports. To support those functions, the MicroPlug OLT itself contains a chipset that requires distinct management mechanisms, separate from other pluggable transceivers commonly used in packet networks. Additionally, the control and management of OLTs and ONUs remain largely proprietary, as different vendors implement their own management protocols, making interoperability a significant challenge. Just as with standalone OLTs, an SDN controller must still access the OLT chipset to send configuration commands, requiring an abstraction layer to unify control across different vendors and technologies.

To address these challenges, two primary approaches have emerged for integrating MicroPlug OLTs into switches and routers. One method involves extending an existing switch NOS to support MicroPlug OLT functionalities by adding software layers that enable direct control and management. Several research efforts have attempted to address these complexities by developing abstraction layers that automate provisioning and management [142] or by implementing SDN-based traffic management solutions [143]. The second approach is to develop a completely new NOS that natively supports OLT applications while also managing Ethernet switching. Solutions like BISDN's pluggable OLT abstraction (pOLTA) [144] follow this path by offering a software platform that abstracts the complexity of MicroPlug OLTs, providing a standardised interface for service orchestration, as well as supporting switch APIs using BISDN Linux NOS. BISDN follows an open-core model, which means that BISDN Linux (switch APIs) is open-source, although to the best of our knowledge, pOLTA is not yet open-source. Another promising solution, also not open-source, is IP Infusion OcNOS [145], which provides a NOS for disaggregated hardware, including Ethernet switches equipped with Tibit MicroPlug OLTs.

Although work is ongoing to facilitate disaggregation in access networks and devices, the software used within these devices (and for their control) is typically tailored to the specific requirements of access networks, and often lacks the capabilities needed to support metro/core application. Features such as coherent and tunable pluggable integration, as well as advanced programmability using P4 to support a broader range of packet headers, are not typically included. This highlights a broader limitation: while disaggregation is improving access networks, true convergence across access, metro, and core domains remains an ongoing challenge.

5.1.2 **Converged Metro-Access Platforms: Universal Edge Devices**

When considering metro-access convergence, the focus is on aggregating multiple lower-speed optical signals into a higher-speed wavelength channel. Aggregation is a key function at the Edge of the network, making muxponders a natural candidate for access/metro convergence. Unlike transponders, muxponders offer greater flexibility through $n:1$ mapping, allowing multiple lower-rate signals to be consolidated into a higher-rate signal before entering the metro domain. This capability is essential for efficiently handling increasing bandwidth demands.

Recent advancements have led to disaggregated muxponders that can be controlled via open APIs, improving their adaptability within open network architectures. However, muxponders lack packet-level switching and traffic management capabilities (i.e., they do not perform functions such as packet buffering, traffic prioritisation, or dynamic rerouting based on network conditions), limiting their role in more dynamic network environments. Additionally, while they are well-suited for aggregation into the metro domain, they do not support access applications such as PON.

Given these limitations, the most promising option for metro and access network convergence appears to be an Ethernet switch or router equipped with pluggable transceivers. This approach aligns with the long-standing concept of packet over DWDM, which allows packet-optical convergence in metro/core networks as discussed in Chapter 3 and 4. By leveraging pluggable transceivers, operators can create a streamlined architecture that bridges the access, aggregation, and metro domains, supporting a wide range of applications from high-bandwidth FTTH deployments to latency-sensitive services driven by 5G, IoT, and AI.

Several proprietary solutions are available for network edge deployments, offering integrated support for Ethernet, IP, xHaul, and PON services within a single platform. These solutions help operators consolidate multiple networking functions, reducing operational complexity and infrastructure costs. A few examples of those boxes are:

- Ciena's Universal Aggregation Solution [146] integrates 10 Gbit/s PON capabilities into its packet networking platforms (model 3000 and 5000 families), allowing operators to simultaneously support PON, IP, and Ethernet services on a single platform;
- Edgecore Networks' CSR440 [147] is a 400 Gbit/s disaggregated cell site gateway built for 5G and xHaul applications. It combines various interface speeds, from 1 Gbit/s to 400 Gbit/s, and features a high-capacity 800 Gbit/s switching architecture powered by Broadcom's Qumran2a chipset. It also supports advanced transport technologies such as Segment Routing over IPv6 (SRv6) and enhanced Common Public Radio Interface (eCPRI) encapsulation, and the CSR440 can accommodate MicroPlug OLTs, enabling PON services within a disaggregated switching framework;
- DZS's M4000 [148], which is a converged transport switch system optimised for 5G xHaul (fronthaul, midhaul, backhaul) and edge aggregation.

It offers SDN-enabled programmability in a compact design (1 Rack Unit) suitable for network edge environments comprising of 5G and Open RAN applications;

- Ribbon Communications' M6424 [149] is a multi-service access router supporting 4G CPRI, 5G eCPRI and Ethernet Services.

While these platforms provide comprehensive solutions for network edge consolidation, they are mostly proprietary, limiting interoperability with third-party hardware. Operators using these solutions are often required to adopt the vendor-specific software, restricting their flexibility in designing a fully disaggregated network. Among them, Edgecore's CSR440 aligns more closely with open networking principles, offering a disaggregated approach that allows greater flexibility compared to fully proprietary alternatives. However, the NOS used to support the Tibit is ExaDOS from Exaware [150] which we believe to be proprietary.

The lack of such an open unified solution creates a gap, as metro and access networks increasingly converge to support diverse services and traffic types. The "open" aspect is critical here because it ensures interoperability across vendors, allowing operators to mix and match coherent pluggables, Ethernet switches, and MicroPlug OLTs within a unified architecture, reducing costs. An open software stack would enable flexible and dynamic network configurations, empowering operators to meet the demands of evolving applications while fostering innovation and competition in a multi-vendor ecosystem, and currently, there is no open NOS capable of accommodating PON, Ethernet, mobile backhaul, programmable switching for dynamic traffic handling, and DWDM technology within the same platform.

5.1.3 Alternative solutions for aggregation and consolidation

Traditional optical networking has long suffered from inefficiencies due to a mismatch between the way traffic flows and transport technologies. For example, traditional DWDM networks assign fixed wavelength channels for transport, relying on strict point-to-point connections requiring transceivers at both ends to operate at the same speed. However, real-world traffic demand fluctuates, with peak usage varying depending on the time of the day and the geographic location. This results in underutilised wavelengths during low-traffic periods or bottlenecks in high-demand areas, leading to inefficient use of network capacity.

More recently, the XR optic, from Infinera [151], addresses this issue by introduc-

ing a point-to-multipoint (P2MP) architecture using digital subcarriers. A 400G XR pluggable can generate 16 x 25 Gbits/s subcarriers, which can be dynamically allocated to different destinations, eliminating the need for intermediate aggregation devices. This enables direct communication between low-speed and high-speed transceivers, which also helps reducing the need for network-wide upgrades when adding higher-speed endpoints, as more subcarriers can be directed toward that particular end-point. By integrating XR optics into routers, switches, servers, and access devices, operators could potentially achieve better resource utilisation and lower CapEx/OpEx while simplifying metro-access and aggregation networks [152].

CapEx reduction with XR optics is achieved by minimising the number of transceivers required, as eliminating the aggregation box reduces overall hardware costs. Studies have demonstrated the cost-saving benefits of XR pluggable optics, enabling operators to optimise transceiver capacity on a per-site basis [153], [154]. Additionally, other research [155] has explored the broader advantages of P2MP coherent technologies across metro and core networks, highlighting a significant reduction in transceiver count and overall infrastructure costs.

However, the deployment of XR optics in optical networks requires architectural and hardware adaptations. Since XR pluggables communicate using digital subcarriers, they must be paired with another XR transceiver at the receiving end to correctly interpret the transmission. Each 400G XR pluggable consists of 16×25 Gbits/s subcarriers, allowing it to distribute capacity across multiple endpoints (e.g., connecting to 16 different 25 Gbits/s receivers). This means that every receiving node must also be equipped with an XR-compatible pluggable, which may limit backward compatibility with existing non-XR transceivers.

Currently, 100 Gbits/s and 400 Gbits/s XR optics are well-developed, and from a real-world deployment perspective, adopting XR optics requires replacing a significant number of pluggables across the network. Additionally, switches and routers must support XR port cages. Beyond the physical hardware, the management and control integration of XR optics is still being agreed by the Open XR Forum, with ongoing demonstrations and developments [156]. While XR presents a long-term solution for scalable aggregation and flexibility allowing for easy upgrades (e.g., enabling a 25 Gbits/s receiver to support two subcarriers for 50 Gbits/s via a software update), it may not be an immediate priority for operators due to the required infrastructure changes.

Moreover, XR optics primarily serve as an aggregation technology, eliminating

the need for additional switches and routers in leaf-spine topologies. However, it does not natively support access applications like PON. In a PON deployment, XR optics can coexist with existing PON infrastructure [157], but a separate OLT device is still required to interface with ONUs in residential and rural areas.

In contrast, our *packetponder* approach leverages a programmable switch to accommodate various transmission rates, supporting both legacy and next generation pluggables. Unlike XR optics, which fix subcarriers at 25 Gbits/s slots, our solution provides greater flexibility by managing heterogeneous traffic types and network conditions.

5.2 Packetponders for access/metro consolidation

This section suggests the use of the *packetponder* concept to bridge the aforementioned gap.

5.2.1 Hardware consolidation with *packetponders* at the edge

Following our demonstration in Chapter 3 of one approach of metro networks rationalisation and consolidation, which exploits the pipeline of innovations within the lower layers of the OSI protocol stack underpinned by Moore’s Law, we believe that the *packetponder* concept with its open software and APIs is ideally suited to support access services. Its flexibility makes it an excellent candidate for aggregating access signals into metro networks, bridging the gap between these two historically distinct domains while enabling seamless integration and greater network efficiency.

Figure 5.1 (also in Fig. 5.2a)) shows a typical PON network architecture aggregating many ONUs with a singular OLT port within a chassis at an exchange/central office (CO). It is also composed of an aggregation switch, a transponder, and a ROADM. Underlying this infrastructure relies a succession of ASICs, and active ports that are interconnected by copper transmission interconnect on the motherboard, which all consume power. Figure 5.2b) shows how the proposed *packetponder* can usefully substitute the traditional transponder function in a single open hardware element as discussed in Chapter 3. Thus, it replaces the aggregating switch ASIC and transponder ASIC and their respective active ports with one single ASIC and respective active ports. Hence, by leveraging new technologies such as coherent pluggable (100ZR, 400ZR, 400ZR +, etc.), the

transponder can be omitted to reduce energy consumption by minimising the number of: transmission lines, format converters (e.g. pluggables) and switching ASICs along the network path. Additionally, it reduces OpEX, CapEX and space [158]. In our *packetponder*, the SERDES interfaces of the P4-programmable switching ASIC presented externally as common-form-factor QSFP28 frontplate port cages can accommodate a variety of pluggable transceivers.

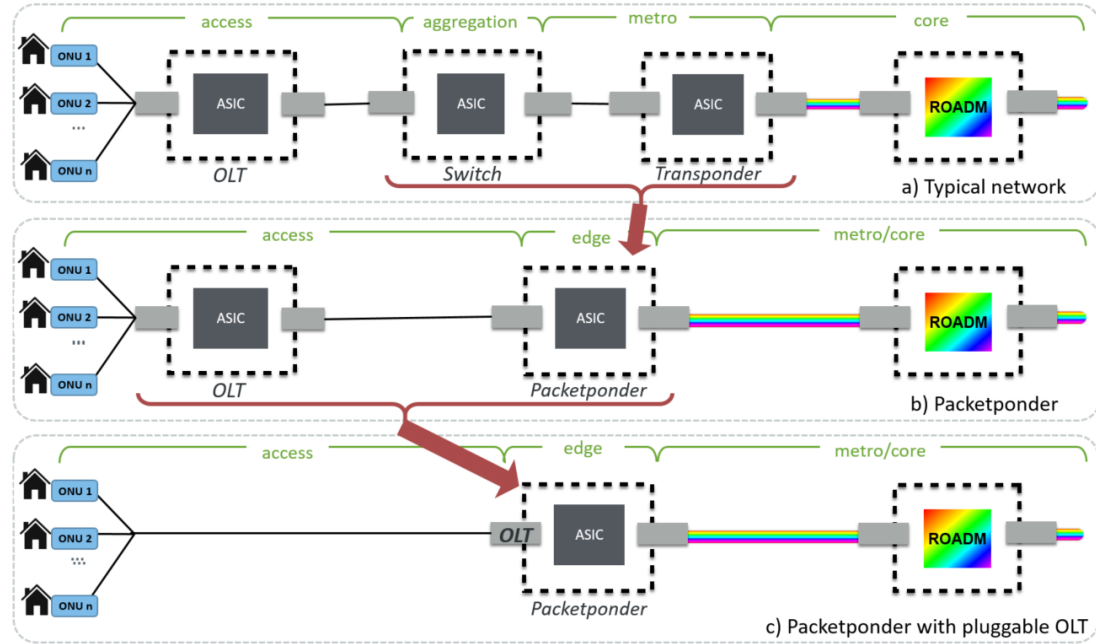


Figure 5.2: Network element consolidation: a) Typical network; b) Packetponder implementation at the metro/core edge; c) Proposed Packetponder implementation for Access/Metro edge with pluggable OLT and DWDM transceivers.

As a result, additional utility of the *packetponder* can be found by combining access, metro and long-haul optical networks. A single network element, the *packetponder*, can not only remove the need for transponders in metro/core network, but consolidate access network services (i.e. point-to-point leased lines, point-to-multipoint PONs and 5G fronthaul transport) that are shown in Figure 5.3, thus also removing surplus ASICs, internal/external transmission lines (e.g. backplane lanes, external optical fibre patchcords or DAC cables) and the associated transceivers from the data plane. This Chapter describes the results of an enhanced implementation of the *packetponder* that serves the access network and more particularly point-to-multipoint PON network architectures shown in Figure 5.2c).

As mentioned in section 2.3.4 of Chapter 2, by integrating coherent optics into switches/routers, networks become more power efficient and cost-effective. On

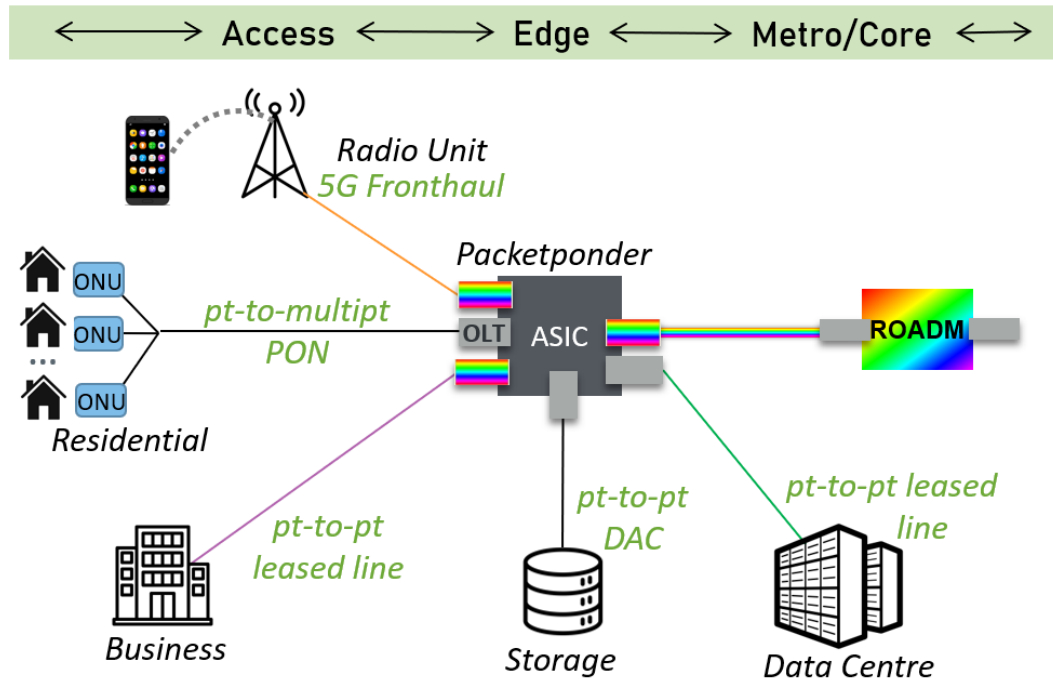


Figure 5.3: Packetponder at the network edge: supporting 5G fronthaul, point-to-multipoint PON and point-to-point leased line Access services; and coherent optical DWDM back-haul and long-haul transmission across the Metro/Core.

the access side, the integration of the Tibit MicroPlug OLT eliminates the need for traditional, bulky OLT chassis, as shown in Fig. 5.4, and reduce the physical footprint by 95% and power consumption by 75% [159]. However, specific simulation studies or real-world measurements detailing the exact energy savings achieved by deploying the Tibit MicroPlug OLT are not available in the public domain, and the reported 75% reduction is based on internal assessments by TiBiT Communications.

While the energy savings may not always follow a strict 1:3 ratio — since a single *packetponder* may not handle the exact same traffic load as a separate OLT, switch, and transponder — the overall efficiency gains remain substantial, and a short review of potential gains will be shown in section 5.4.

5.2.2 Multi-services and aggregation node

One key advantage is the flexibility in pluggable selection, allowing operators to tailor their network based on service demands: a single device can support multiple access types simultaneously, meaning operators no longer need separate hardware for different services.

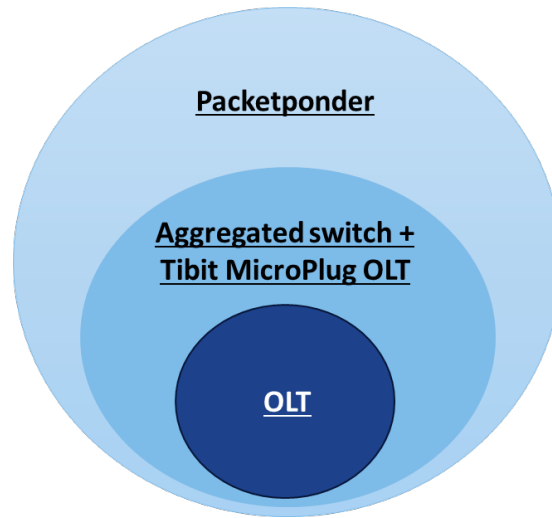


Figure 5.4: *Packetponder* with PON and aggregation functionalities.

In theory, the *packetponder* can aggregate multiple services coming from the access network, as exemplified in Fig. 5.3. It can also supports future anticipated enhancements including multiple wavelength bands [160]. It can support short-reach O-band (1260-1360nm) transmission between datacentres; or FTTx applications [161], while also supporting C-band (1530-1565nm) transmission by using different pluggables; and potentially enabling 3R regeneration and wavelength conversion in between. In addition, the *packetponder* can support multiple rates, and simultaneously support legacy, current, and future bandwidths that share a common form factor.

5.2.3 Interoperability and management framework

To date, the NOS and software control systems for MicroPlug OLT devices and coherent/tunable transceivers have been developed independently, as they were originally designed for deployment in separate hardware platforms. In the case of the MicroPlug OLT, there is no direct Command Line Interface (CLI) management application as there is no direct access to the MicroPlug. Therefore, the OLT requires in-band network management, as opposed to the coherent/tunable transceivers which can be managed using an out-of-band connection. As a result, there is currently no unified open software solution capable of managing both technologies under a single framework. The ultimate goal is to develop a common, fully open software platform capable of controlling all components, ensuring seamless integration and flexibility across diverse network applications.

In the absence of such a unified solution, we experimentally demonstrated the fea-

sibility of a consolidated access and metro network hardware using open-source software where possible. Specifically, we leveraged Stratum, which we have already successfully implemented for managing tunable and coherent transceivers. For the control of the Tibit MicroPlug OLT, we integrated the MicroClimate Management System (MCMS) software [162], which, while not open-source, allowed us to validate the concept of a unified platform. MCMS is a commercial software to control the PON pluggable, which provided a complete management solution for the MicroPlug OLT device as well as the third-party ONUs, compliant with the 10G-EPON [133] and XGS-PON [163] standards. MCMS also provides a NETCONF server enabling SDN applications to retrieve and submit new configurations to the OLT or ONUs using generic YANG data model.

5.3 Multi-services aggregation demonstration with a *packetponder*

The demonstration is a proof-of-concept of the *packetponder* at the edge, showing the feasibility of such a box to consolidate access, metro and core networks within one box. The first part of the experiment shows the integration of the OLT pluggable into the whitebox. The second part shows how different access services, using different pluggable transceivers, can be aggregated into higher capacity pluggable modules toward metro/core networks. The software stack used during this demonstration is similar to the one used in Chapter 3 as is depicted in Figure 5.5.

5.3.1 Hardware testbed

Our proof-of-concept demonstrator is shown in Fig. 5.6. This experimental setup will be discussed throughout this section and comprised of: two ONUs, 1) from Iskratel (serial number: ISKT71e80040), and 2) from Arcadyan (serial number: ARCA20207737) that terminated our XGS-PON; 3) a commercial Lumentum 10 Gbits/s SFP+ tunable transceivers (gray and tunable); 4) a 10 Gbits/s fixed wavelength (850nm-band short reach) SFP+ using a Cisco SFP-10G-SR; 5) a Adtran tunable 25 Gbits/s SFP28 transceiver; 6) a QSFP28 100Gbits/s direct attach copper (DAC) transceiver; 7) a O-band 100 Gbit/s QSFP28 pluggable; a coherent QSFP28 100ZR from Adtran, performing 100 Gbit/s dQPSK @27.952GBd (port 30/0 in Fig. 5.6); and a Tibit MicroPlug XGS-PON SFP+ OLT pluggable inserted into one of the *packetponder* cage using a Mellanox QSFP to

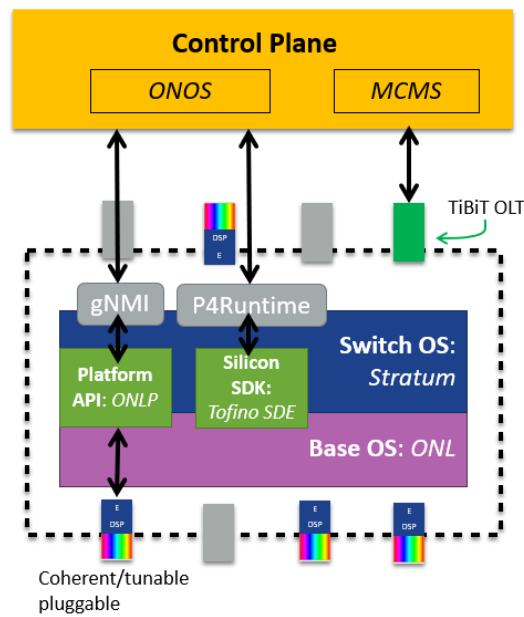


Figure 5.5: Proof-of-concept demonstration: SDN software control of the *packet-ponder* to support access/metro services.

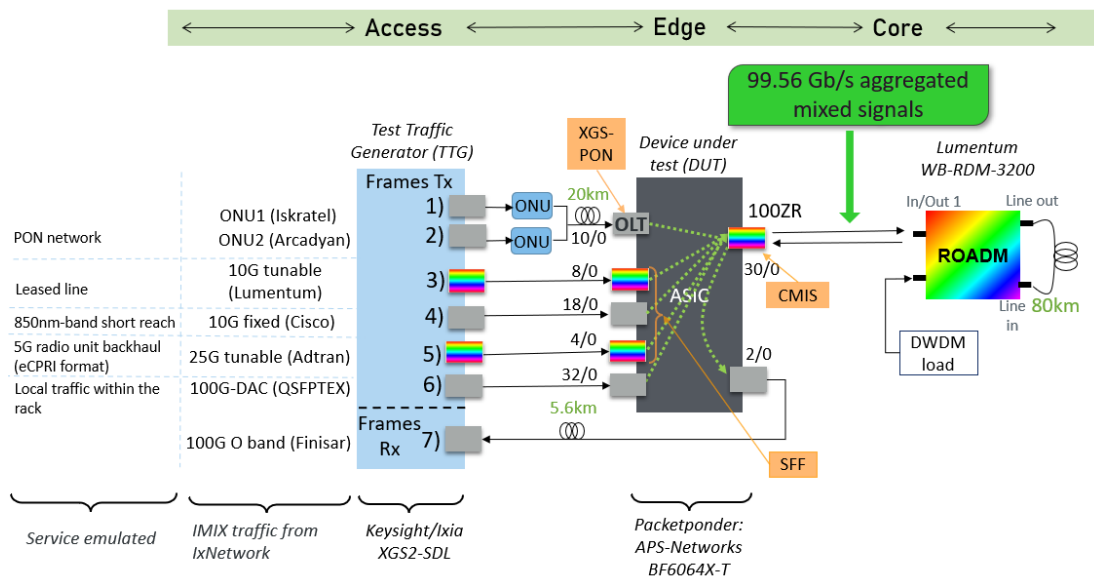


Figure 5.6: Experimental arrangement: Packet flow to/from TTG via DUT & ROADM.

SFP+ Cable Adapter. The MicroPlug OLT is connected to a 1/8 splitter with an extra 10 dB attenuation, before reaching the two ONUs. A 20 kilometres single mode fibre (SMF) spool separates the MicroPlug and the ONUs, emulating a typical access link. Our device under test, the *packetponder*, was the same P4 programmable APS Networks BF6064X-T Ethernet switch described in Chapter 3. We also used the previously described Lumentum whitebox ROADMs. To gen-

erate traffic, we used the Keysight/Ixia XGS2-SDL test traffic generator (TTG). Its chassis contained: I) Novus-S: QSFP-28 port cages supporting 100/50/25/10 GbE (Gbit Ethernet); and II) Novus-NP: SFP+ and 10GBase-T port cages supporting 10GbE/1GbE/100MbE load modules to generate traffic of different line rate coming for the access or metro side. It emitted Ethernet frames of different format, and used the *IxNetwork* software for building test scenarios and viewing statistics.

This demonstration was made possible through the collaboration with Jim Zou from Adtran, who provided the coherent 100ZR; Ian Cooper from BT, for his assistance with the TiBiT; and Scott Hill and John Griffith from Keysight, who supplied the traffic generator and helped with its software.

5.3.2 PON integration and management

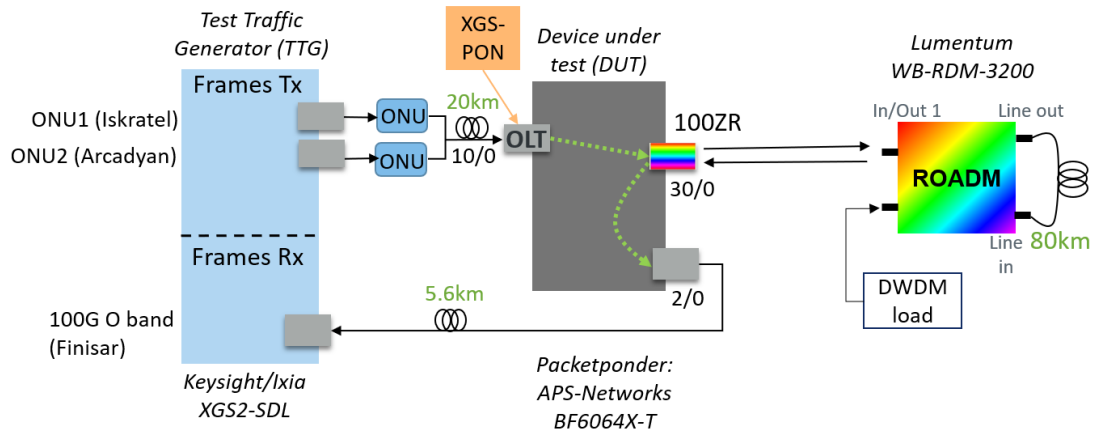


Figure 5.7: PON experimental setup to test MicroPlug Tibit inside the *packetponder*.

The first part of the experimental set up is shown in Fig. 5.7, which is a simplified version of the full experimental setup shown in Fig. 5.6, and is intended to show the Tibit OLT integration into the *packetponder*. Although the purpose of the MicroPlug is to be inserted into a generic switch, in practice its integration and management is not so seamless. As stated previously, the challenge comes from the fact that the Tibit requires a specific management scheme that is not included into switch operating systems and switch/router controller at the moment.

When the Tibit MicroPlug OLT is first installed, it comes pre-configured from the factory to use C-VLAN 4090 for management traffic. This means that all communication between the MicroPlug OLT and the management system (such as an SDN controller, or MCMS) is tagged with VLAN 4090. VLAN 4090 is specifically

used for management traffic to isolate it from normal user traffic. To manage the Tibit pluggable, which requires in-band management, we established a management connection with an Ethernet cable from a remote desktop (hosting the Tibit controller) to one of the switch ports. Since the *packetponder* uses QSFP28 cages, we used an SFP+ to QSFP28 adaptor to ensure compatibility. On the desktop, the management interface is identified as eth2. To enable communication with the Tibit MicroPlug OLT, we needed to create a VLAN sub-interface tagged with VLAN 4090. This was done by configuring eth2.c4090, allowing the desktop to interact with devices that require VLAN 4090-tagged traffic. Any packets sent from eth2.c4090 were automatically tagged with VLAN 4090, ensuring they are correctly recognised by the Tibit OLT. The resulted interface created is shown as follows:

```
root@linuxlab:/home/julie_raulin/Documents/tibit/USB# ifconfig
eth2: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 169.254.221.171 netmask 255.255.0.0 broadcast 169.254.255.255
    ether 00:e0:4c:78:76:f5 txqueuelen 1000 (Ethernet)
    RX packets 34 bytes 11321 (11.3 KB)
    RX errors 0 dropped 24 overruns 0 frame 0
    TX packets 47 bytes 7758 (7.7 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
eth2.c4090: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    ether 00:e0:4c:78:76:f5 txqueuelen 1000 (Ethernet)
    RX packets 1 bytes 47 (47.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 1 bytes 60 (60.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

Our disaggregated whitebox with open NOS, the APS-Networks, was compatible with the Tibit MicroPlug OLT although required extra work to configure management. Since the MicroPlug OLT is designed as an SFP+ form-factor device and hence following the SFF MSA, it was physically compatible with our switch, and inserting it into the *packetponder* did not cause any issues. The main requirement was ensuring that VLAN 4090 traffic could pass through the network, particularly for management communication between the OLT controller (hosted in the Desktop) and the ONUs. To achieve this, we made a minor enhancement to our P4 code, incorporating a switching rule that forwards packets based on VLAN tags. Specifically, any packet tagged with VLAN 4090 was programmed to be forwarded between the Tibit data port and the Tibit management port, ensuring

proper communication and management of the ONUs.

We then installed the management software provided with the Tibit (which is called MicroClimate Management System, version R3.2.0) in the remote Desktop. The MicroClimate Management System (MCMS) is the management solution for Tibit PON networks. It consists of the MCMS PON Manager graphical user interface, MCMS Netconf Server, Mango Database and MCMS PON Controller. Together these components provide a complete network management solution for provisioning and monitoring MicroPlug OLT devices, as well as the subtended third-party ONUs devices. Figure 5.8 shows the software architecture to control the Tibit, using MCMS that is connected to the *packetponder* using an RJ45 cable.

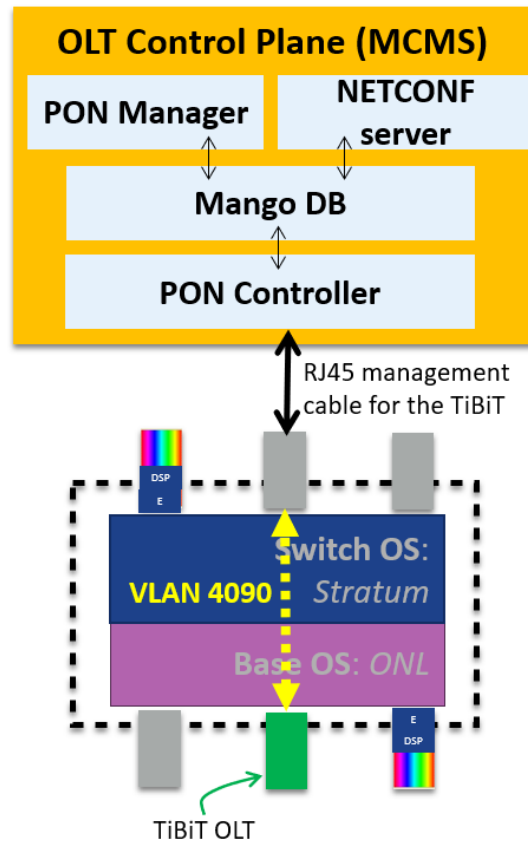


Figure 5.8: Software architecture to control the Tibit using MCMS software.

To test our PON device, we sent traffic through the ONUs and leveraged the Tibit MicroPlug XGS-PON SFP+ OLT pluggable to emulate a PON architecture. The ONUs were separated by a 20km SMF28 fibre spool before reaching the MicroPlug OLT. ONU 1, from Iskratel and ONU 2, from Arcadyan terminated our XGS-PON. The XGS-PON was inserted into our *packetponder* and Fig. 5.9, which is the graphical user interface of the PON Manager as it appears on the MCMS software, shows that the PON controller has successfully been connected to the

MicroPlug OLT in the switch and its two ONUs, using the tag VLAN 4090.

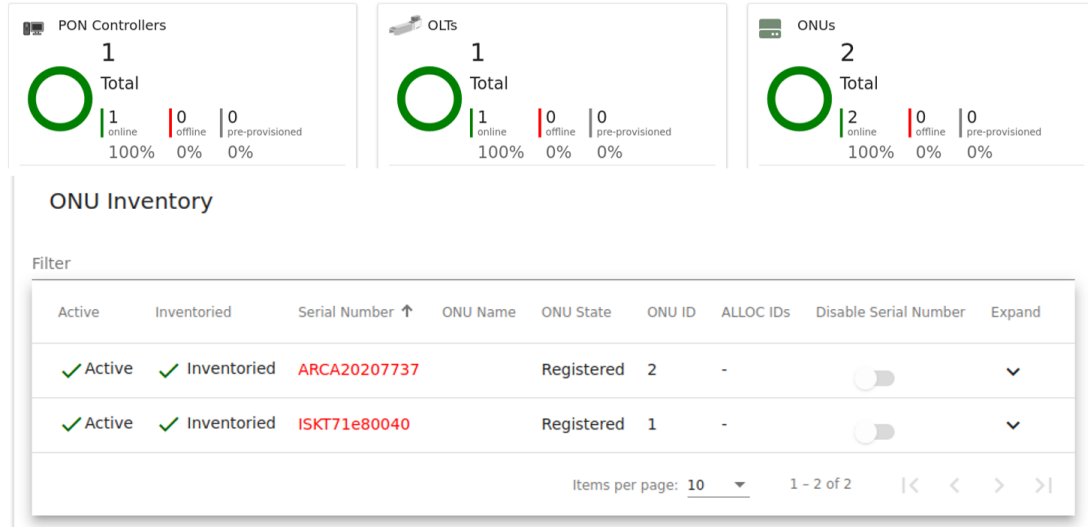


Figure 5.9: MCMS dashboard, showing the integration of the OLT pluggable and the two ONUs.

In the experimental demonstration with the PON, we focused on evaluating upstream traffic. Upstream testing provides valuable insights into the ONU's ability to transmit data efficiently, and to ensure the *packetponder* can aggregate traffic coming from the access network. From a physical layer perspective, upstream transmission is typically the most demanding, as aggregation must accommodate multiple services with varying power levels. Upstream traffic is mainly dominated by acknowledgments, requests, voice/video conferencing, IoT traffic, and network management signals and is generally of smaller frames than the downstream traffic, dominated by videos. Downstream traffic, on the other hand, requires different configuration since the Tibit must correctly distribute data to multiple ONUs. One key challenge is ensuring that each ONU properly recognises its assigned traffic, especially when working with IP-based addressing and service VLANs.

Hence, to validate the core functionality of the PON network integration into the *packetponder*, test traffic was generated from two independent ports of the Keysight traffic generator as shown in Fig. 5.7. The test traffic was launched to the client-side port and transmitted upstream to the headend at a wavelength of around 1270nm. ONU 1 (Iskratel) was a 1 Gbit/s device that sent a flow of random-sized frames (from 66 — the minimum Ethernet frame size when using VLAN tagging — to 1518 bytes) at a rate of 0.86 Gbit/s. ONU 2 (Arcadyan) was a 10 Gbit/s speed device so we sent a flow of Internet Mix (IMIX) packets at a rate of 7.7 Gbit/s. The sum of both traffic rate didn't reach 10 Gbit/s,

which is the capacity of the XGS-PON MicroPlug, because XGS-PON uses error correction that sends additional data on top of the transmitted user's data. This extra data is one form of overhead that consumes a portion of the total available capacity of the PON, and results in a reduction of the usable capacity. In the end, the usable capacity for XGS-PON is approximately 8.5 Gbit/s in downstream and 8.6 Gbit/s in upstream [164]. The data rate from both ONUs have been experimentally chosen to stretch the upstream PON speed at its limit with error-free transmission: ONU 1 has error free transmission with a signal rate of 0.87 Gbit/s with random sized frames while ONU 2 was set to 7.8 Gbit/s with an Internet Mix of traffic with various frame rates as well. The total of both traffic was 8.67 Gbit/s, slightly better than the estimated usable capacity in [164]. For the rest of the demonstration, ONUs rate were set to 0.86 Gbit/s and 7.7 Gbit/s for ONU 1 and 2 respectively, for a total of 8.56 Gbit/s over the Tibit MicroPlug OLT to give a small margin for potential errors.

Frames from both ONUs were then aggregated through the Tibit OLT pluggable which was inserted into port 10/0 of our *packetponder*. We used the IP destination address to distinguish test packet flow from Keysight: all packets from traffic ONU 1 had a source IP address of 10.1.0.2, and all packets from traffic ONU 2 had a source IP address of 10.1.0.3. We compiled a simple P4 program that populated the forwarding table of the Tofino 1 ASIC so that the traffic coming from the ONUs were forwarded to the same egress ports. Indeed, based on their 2-tuple, all packets were forwarded to a single 100 Gbit/s port cage, 30/0, that contained an industry-first 100ZR QSFP28 coherent tunable pluggable transceiver from Adtran. Data was emitted from the Tx port of the 100ZR at 1524.94nm, directed through ingress add port In 1 of an open, optical whitebox ROADM (Lumentum WB-RDM-3200) and transmitted over 80km of single-mode optical fibre to simulate a link to a datacentre, together with a fully loaded DWDM channels (emulating $76 \times 50\text{GHz}$ grid spaced data channels with two 100GHz empty bandwidth slots). The signal was then looped back through port Out 1 of the ROADM to the Rx port of the Adtran 100G ZR pluggable, emulating an edge to metro node. The P4 program then used the 2-tuple `<DUT_ingress_port_label, IPv4_source_address>` to direct test traffic to port 2/0 which contained an O-band 100GbE QSFP28 pluggable (1302.35nm), and transmitted over 5.6km of single-mode fibre. Traffic is finally received to the Keysight traffic generator, which captured packet statistics.

Fig. 5.10 represents frames sent to ONU1 (Traffic item: ONU 1) and ONU2 (Traffic item: ONU 2), and received back to the traffic generator. The table

shows error free transmission and no frames were lost during the experiment. Fig. 5.11 shows the receiving rate on each traffic flow during the run, which corresponds to the sending rate of both traffic.

Traffic Item	Tx Frames	Rx Frames	Frames Delta	Loss %	Store-Forward Avg Latency (ns)
ONU1	5,000,000	5,000,000	0	0.000	987,655
ONU2	85,000,000	85,000,000	0	0.000	700,361

Figure 5.10: Traffic items sent during the experiment, showing no loss of packets, with associated latency measurements.

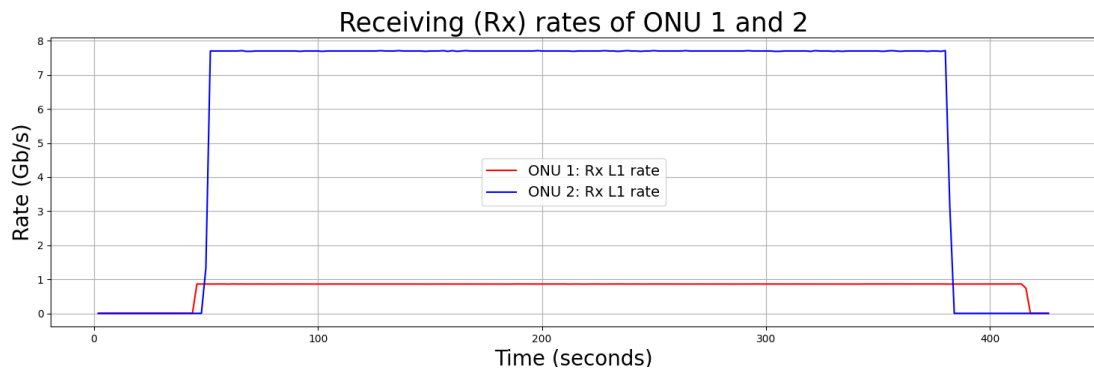


Figure 5.11: Upstream traffic rate (Rx) of ONU1 and ONU2 during the experiment.

The initial part of the demonstration showed how PON functionality can be seamlessly incorporated into a programmable Ethernet switch. While this demonstration required the installation of a third party software to control the PON network, the ideal scenario and hopefully in the future, would involve integrating the MicroPlug OLT's management and control directly into the switch NOS and controller, via open APIs and protocols, treating the MicroPlug OLT as any other pluggable for streamlined operation, and is outside of the scope of this thesis.

5.3.3 Traffic aggregation from different services

In addition to the ONUs traffic, additional traffic flows were sent to the *packetponder* and aggregated into port 30/0 of the programmable switch, to replicate the functions of a Muxponder. The aggregation use-case emulated in this experiment, used the same Keysight Test Traffic Generator (TTG) to emit from port 1) and 2) two unidirectional flows for transmission across a 20km PON and received by a Tibit XGS-PON OLT SFP+ pluggable transceiver in the Device under test

(DUT), as described in the previous section and shown in Fig. 5.7. The Keysight traffic generator provides flexible packet generation with various formats. For simplicity, packets from ONUs 1 and 2 included an Ethernet header, an IP header, and a payload set to "Increment Byte" to introduce variation in content. Except for traffic 5), all generated traffic followed this same packet format. TTG port 3) generated point-to-point traffic using a Lumentum 10GbE tunable optical SFP+ transceiver in the C-Band for leased line emulation. TTG port 4) emitted point-to-point traffic with an 850nm-band short-reach SFP+ transceiver, specifically a Cisco SFP-10G-SR. TTG port 5) transmitted point-to-point traffic in the C-Band using an Adtran 25 Gbits/s tunable SFP28 optical pluggable transceiver, emulating traffic from a 5G radio unit. Communications between remote radio heads (RRHs) and baseband units (BBUs) in modern 5G mobile networks use the enhanced Common Public Radio Interface (eCPRI) packet format, therefore the packets from 5) were created using the TTG with an additional eCPRI header, which typically consists of 4 Bytes. Lastly, TTG port 6) generated point-to-point traffic using a QSFP28 100G direct attach copper (DAC) cable to emulate local traffic within the rack, connecting to an intra-rack storage array for local content delivery network emulation. Fig. 5.12 summarise the pluggable transceivers used in the Test Traffic Generator and their associated line rate, as well as their connection to the device under test, the *packetponder*. In general, the packets were created to emulate a real access/metro aggregation node with mixed services, pluggables, reaches and efficiencies.

The flow of test traffic is represented by arrows in Fig. 5.6 between the Tx ports labelled 1)-6) and the single Rx port labelled 7) of the TTG. Test traffic emitted from the TTG for all ports was configured using *IxNetwork* with a unique IPv4 source address particular to the Tx ports. The P4 program on the DUT used the 2-tuple `<DUT_ingress_port, IPv4_source_address>` for ingress flow classification. The same way it was done for the first experiment, we populate the P4 flow table so that all packets were forwarded based on their 2-tuple to port 30/0, that contained the 100ZR (the *incast* forwarding test traffic flows within the DUT to a common egress port cage is shown by the green dotted line of Fig. 5.6) The table in Fig. 5.13 summarises the number of frames sent for a 10-hours run. It shows successful error-free reception of all 10^{10} frames — that were proportionally divided depending on the line-rate of each Tx port of the TTG — at the conclusion of a ten-hour test run.

In a follow-up experiment, we monitored the throughput of each traffic flow. Traffic flows were added at 10-second intervals, as shown in Fig. 5.14 (coloured

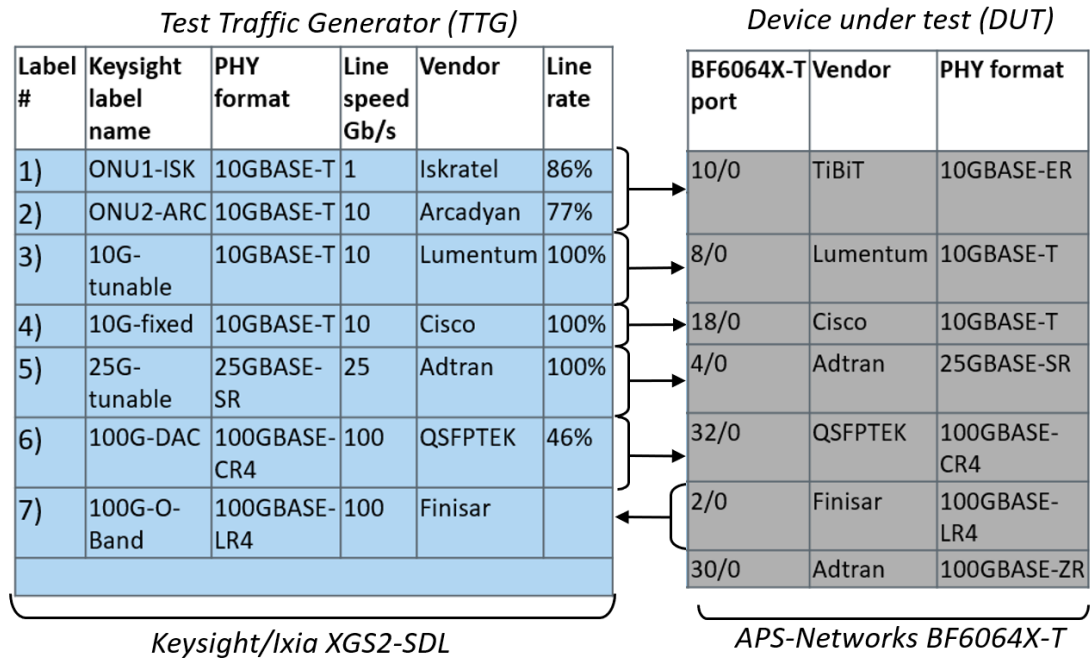


Figure 5.12: Common-form-factor pluggable identification, port-labelling & connectivity mapping scheme between TTG, DUT and ROADM.

	Port Name	Line Speed	Link State	Frames Tx.	Valid Frames Rx.
1)	10G-tunable	10GE LAN	Link Up	10,000,000,000	0
2)	10G-fixed	10GE LAN	Link Up	10,000,000,000	0
3)	ONU1-ISK	1000 Mbps	Link Up	500,000,000	0
4)	ONU2-ARC	10GE LAN	Link Up	8,500,000,000	0
5)	25G-tunable	25GE	Link Up	25,000,000,000	0
6)	100G-DAC	100GE	Link Up	46,000,000,000	0
7)	100G-O-band	100GE	Link Up	0	100,000,000,000

Figure 5.13: Number of frames sent and received by the traffic generator during a 10-hour run.

lines 1) to 6)), and the aggregated throughput was successfully received at the Rx (black line). This demonstrated the successful multiplexing of traffic from different ports with varying header formats, validating the *packerponder*'s aggregation functionality.

The aggregated traffic flowed at a speed of 99.56 Gbit/s as shown in Fig. 5.15, which correspond to the sum of the 6 traffic flows depicted in the table in Fig. 5.12. The speed didn't reach 100 Gbit/s as the MicroPlug OLT and ONUs added overhead data. Fig. 5.15, which is taken from the software *IxNetwork*, shows the

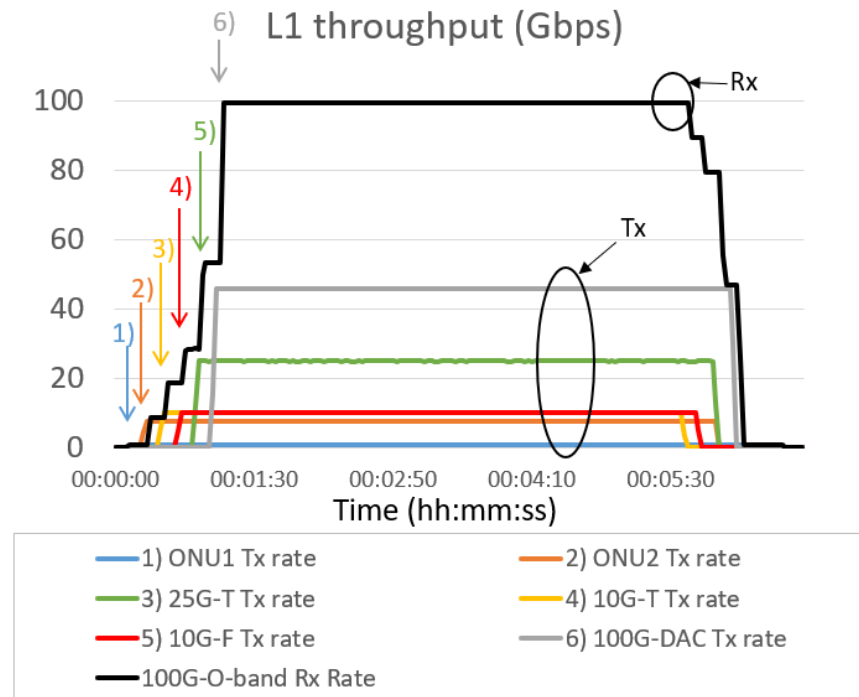


Figure 5.14: Throughput — dynamic, sequential addition of traffic.

Layer 1 throughput of all transmitters combined (Tx — left) and at the receiver port (Rx — right). The screenshot was taken while the system was running, hence the difference between Tx and Rx frames were not zero.

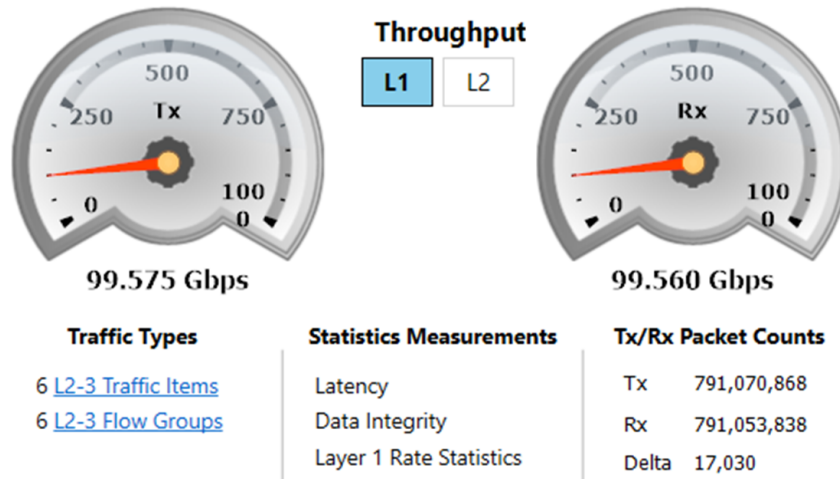


Figure 5.15: Snapshot of the throughput during the experiment with IxNetwork software, showing L1 data rate and the number of packets traversing.

This proof of concept demonstration showed that the *packetponder* can be used as a converged access/metro box to support existing and future services, with at its heart a P4 programmable ASIC that can flexibly support any frame protocol. The demonstration highlighted the muxponder capabilities of the *packetponder*

by aggregating multiple client traffic streams into a DWDM pluggable, while also demonstrating that any front-panel port can be dynamically assigned to different traffic types or services. This enables a true plug-and-play solution where a single box can support multiple services, all managed centrally.

This is the first demonstration, to the best of our knowledge, to integrate a variety of pluggables — each supporting different services and wavelength bands — within the same device, while maintaining control via open APIs. Although MCMS is proprietary, it remains a widely adopted common management scheme, enabling northbound API access via NETCONF for seamless integration into an open, SDN-controlled environment.

While we did not have access to XR pluggables, we do not rule out their potential integration within the *packetponder*. XR optics could further enhance aggregation networks, as the *packetponder*'s flexibility allows it to be deployed at various points in the network, playing different roles depending on the required function.

5.4 Impact on network topologies

5.4.1 BT network use-case

In a real network, the consolidation of hardware could have several benefits as well. To illustrate them, we took as an example the topology of the British Telecom (BT) network, based on information in [13], [165]: the network topology consists of 4375 Multi-Service Access Node (MSAN), which aggregates voice and data from the customer to the backhaul (via copper for telephone cables or fibre optic lines) and contains OLTs for FTTH applications; 925 Tier 1 MSAN nodes which are located in larger cities and contains a mix of Access devices, Ethernet switch/routers, and WDM devices; 63 metro nodes; 12 outer core nodes and 8 inner core nodes which carry the most amount of data in the network. Fig 5.16(a) shows the aforementioned 5383 nodes that are spanning across the entire country. The links between them are not represented but the inner core nodes are fully meshed, while outer core nodes are triple-parented to the inner core. Metro nodes are dual-parented to the nearest core nodes, and lower-level exchanges connect to the two closest exchanges. Remote areas and islands use a minimum number of cables to connect exchanges efficiently before linking to the mainland via the nearest Tier 1 MSAN exchange. The network also includes a cellular infrastructure, which is not represented in Fig 5.16, and comprises

33,062 macro cell nodes. The cellular traffic is connected via a straight-line path to the nearest serving exchange in the Internet exchange network. The large number of nodes in Fig. 5.16(a) highlights the need for topology simplification, as each node represents a location housing multiple devices, all contributing to significant power consumption, physical space requirements, and maintenance effort across a large geographical area.

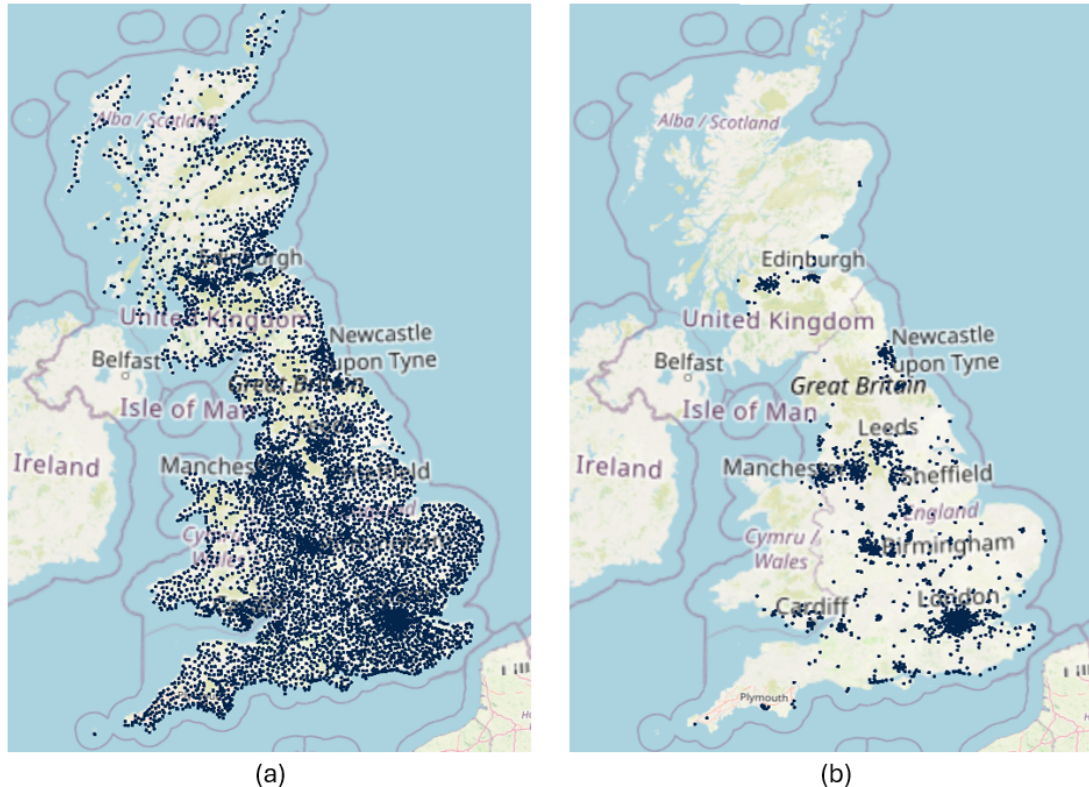


Figure 5.16: BT Topology: (a) Current architecture with Core, Metro, Tier 1 and Multi-Service Access Node (MSAN) nodes; (b) Hypothetical 2040+ network architecture showing a simplified topology without MSAN nodes. While not fully representative of real deployment constraints, it illustrates the potential site reduction achievable through network consolidation, retaining only Core, Metro, and Tier 1 nodes.

Recently, BT and Telefónica have announced that they will move forward with shutting down copper exchanges as part of their transition to an all-fibre network [166], [167]. This does not necessarily mean they will replace every copper exchange with a new "fibre" exchange. Instead, the approach involves consolidating network infrastructure by relocating services to fewer, more centralised fibre-enabled exchanges, such as Tier 1 MSAN or metro/core nodes already in the network. This is possible because fibre can reach further, and requires far fewer intermediate exchanges than copper. As a result, BT will be left with thousands

of redundant exchanges, potentially worth billions of pounds in real estate value for resale.

Fig 5.16(b) presents a conceptual illustration of a simplified BT network architecture in which MSAN nodes are fully decommissioned, highlighting the potential reduction in network hubs and associated operational costs. However, this example is idealised and does not fully reflect practical deployment constraints, as the achievable level of consolidation is limited by cost considerations and the optical reach of commercially viable, low-cost transmission equipment. Indeed, it is important to note that, in practice, Tier 1 MSAN, metro, and core nodes may remain more widely distributed across the UK than shown in Fig 5.16(b) to ensure that remote locations can connect to an exchange within a reasonable distance. Advancements in technology are continuously increasing the reach of fibre networks, making it likely that even the most remote areas will have fibre coverage in the near future. In the worse case scenario, in some rural regions where households are far from the nearest metro node, existing street cabinets could be repurposed and upgraded to house OLTs, replacing outdated DSL equipment and bringing fibre closer to end users.

5.4.2 *Packetponders* for more simplified network, more flexibility and better management

In the BT case, one can assume that OLTs will be relocated into Tier 1 MSAN or metro nodes, which will serve as key aggregation points in the network. These nodes typically house Fibre MSANs (which aggregate broadband services such as PON and Active Ethernet, as well as voice and leased lines), OLT platforms, aggregation routers, Ethernet switches, and optical transport equipment, including transponders and ROADMs. Tier 1 nodes are critical in metropolitan areas, acting as convergence points for multiple access networks before traffic is transported towards the core.

This diversity of hardware makes Tier 1 nodes complex to manage, as they involve multiple software stacks from different vendors. Our proposed *packetponder* significantly reduces this complexity by consolidating multiple network functions into a single platform. Instead of managing three or more separate devices, operators can deploy a single programmable and centralised unit, customising it to support any combination of access services, making it a flexible solution for operators.

This small example illustrates how operators could achieve network simplification by using our concept of the *packetponder* at the Edge, also enhancing network management and control. Additionally, it has the potential to further improve energy efficiency, making the transition even more beneficial.

5.4.3 Preliminary results on potential energy savings by using the *packetponder*

To illustrate the potential energy savings of our network architecture with the *packetponder*, we aimed to compare its total power consumption with the one of a more conventional setup. Fig. 5.17(a) is an example of a simple network model on how to handle traffic from ONUs to the core network in a simplistic way. This scenario serves as a reference point to highlight how our solution could reduce overall energy usage.

It is important to note that this is not a rigorous power consumption analysis, as conducting a comprehensive study would require accounting for numerous parameters such as varying equipment efficiencies, power scaling, cooling requirements, and long-term operational considerations, which is outside the scope of this thesis. Instead, this comparison serves as a high-level indication of how our approach could eliminate some of the extra links, pluggables, and network elements present in a more traditional architecture. The goal is not to claim an exact energy saving figure but rather to demonstrate the potential for optimisation in a typical network deployment.

We structured the traffic as follows: the access side carries 320 Gbit/s of data towards ONUs/customer homes, 1.6 Tbit/s is arbitrarily directed towards the metro network or nearby destinations via point-to-point links (which uses grey optics), and an additional 1.6 Tbit/s is routed towards the core network using DWDM devices with coherent pluggables. This results in a total system capacity of 3.52 Tbit/s.

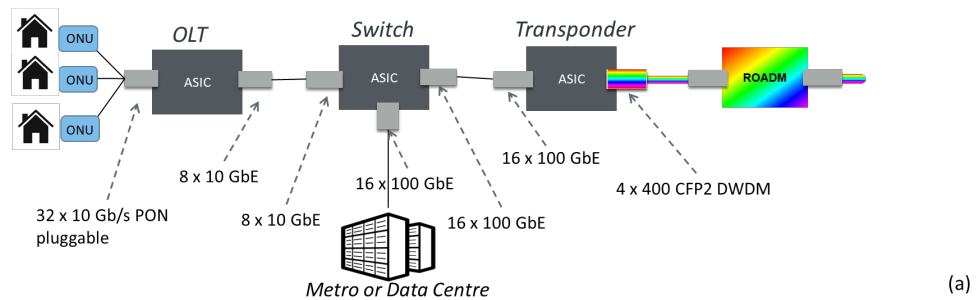
5.4.3.1 Typical architecture

For the access segment, a 320 Gbit/s PON network serves approximately 1,000 customers. This can be achieved using an XGS-PON OLT with 32 ports of 10 Gbit/s capacity and 1:32 splitters (so 32 end-customers per port), such as the Qualfiber XGS-PON OLT 32 Ports [168], which provides 32×10 Gbit/s PON interfaces towards ONUs (downlink) and 8×10 GbE uplinks. Although the OLT

has a total downstream capacity of 320 Gbit/s, the actual traffic at any given time is usually lower so operators design networks with over subscription ratios (e.g., 4:1 or 8:1) assuming users won't fully utilise their bandwidth at the same time. Consequently, an uplink capacity of 80 Gbit/s is typically sufficient.

The OLT connects to an aggregation switch (assumed to be an Ethernet switch, as shown in Figure 5.17(a)) to aggregate the combined PON and metro to the DWDM traffic. Given the 1.6 Tbit/s metro traffic, 1.6 Tbit/s DWDM traffic, and 80 Gbit/s OLT uplinks, the aggregated switch requires at least 40×100 Gbit/s ports ($2 \times 16 \times 100$ Gbit/s plus 8×10 Gbit/s). A suitable switch for this setup could be the Arista 7260CX3-64 [169], which provides sufficient capacity while consuming approximately 640W.

For the traffic directed towards the core, transponders are used as shown in Figure 5.17(a). A suitable reference device is the Edgecore Phoenix [36], which consists of two sleds, each supporting 8×100 GbE client-side interfaces and $2 \times 200/400$ Gbit/s line-side interfaces. For this example, we assumed the use of 400 Gbit/s CFP2 modules, requiring four transceivers (2×400 G per sled).



Device name (vendor – model)	# of items	Power per item (Typical/Max) [W]	Total power (Typical/Max) [W]
OLT (Qualfiber – XGS-PON OLT 32 Ports)	1	320 (Typical)	320 (Typical)
Switch (Arista – 7260CX3-64)	1	314/616	314/616
Transponder (Edge-core – AMX320)	1 (2 sleds)	850 (Max)	850 (Max)
10G OLT pluggables	32	3 (Typical)	96
10 GbE	16	2/4	32/64
100 GbE	48	3.5/5	168/240
400 Gb/s coherent CFP2	4	24/28	96/112

Figure 5.17: (a) Potential topology to meet the 3.52 Tbit/s capacity requirement from access to core; (b) Power consumption of each components.

The example setup is shown in Figure 5.17(a) and the total power consumption, based on the datasheets of the proposed devices, is summarised in the table shown in Figure 5.17(b) with 1708W (typical)/ 2298W (max). Some datasheets don't give the typical power in which case the maximum power is considered for both calculations.

5.4.3.2 Architecture with *packetponders*

The *packetponder* leverages the Tibit MicroPlug OLT, significantly reducing power consumption by eliminating the need for an external OLT chassis. In a traditional OLT chassis, the system consists of a PON MAC ASIC, an Ethernet switch chip (either on the line card or, in newer designs, integrated within the PON MAC chip), and a chassis fabric chip to interconnect line cards. With the Tibit's approach, the PON MAC layer is directly integrated into an aggregated/Ethernet switch, eliminating an entire "switching" layer. Additionally, conventional OLTs require dedicated management cards, often deployed redundantly to ensure reliability. With Tibit's disaggregated approach, the control is separated from the OLT, allowing centralised management of multiple devices further back in the network architecture hierarchy (closer to the metro/core node). Offloading the management burden from the OLT chassis to a scalable, centralised compute node reduces power consumption at each access location, and allow the access equipment to be more distributed, and closer to the end user, while pulling the complexity back to the metro/core.

On the core-facing side, coherent pluggables replace power-hungry transponders, as they can now be inserted, and managed, into an Ethernet switch. The studies described in Chapter 3 have demonstrated that this approach can reduce power consumption by approximately 33% [170]. As a result, the *packetponder* consolidates multiple network functions, reducing energy consumption by optimising the use of pluggables across different applications, as depicted in Figure 5.18(a).

In our simple use-case, the *packetponder* would support 32 Tibit MicroPlug OLTs, 16 grey 100GbE pluggables, and $16 \times 100\text{ZR}$ coherent pluggables within the same box to meet the required capacity toward access, metro and core. Figure 5.18(b) presents an estimated maximum power consumption of 964W (typical)/ 1531W (Max), with a comparative analysis indicating a reduction in energy consumption from 33% to 44% — a substantial improvement.

While real-world deployments may not follow a strict 1:1 replacement model as we just described, this study highlights the potential benefits of network simplification through the adoption of programmable, open, and flexible architectures leveraging advanced pluggable technologies.

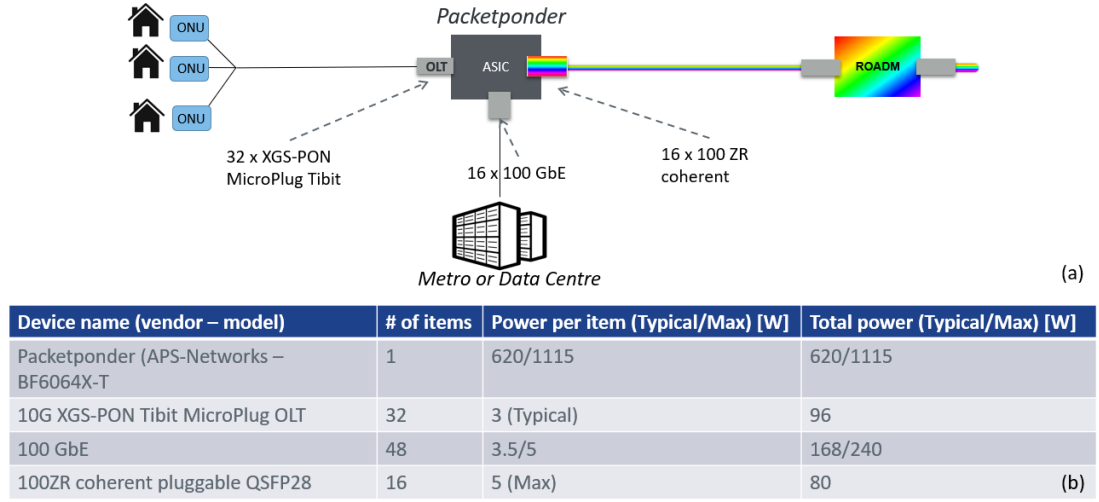


Figure 5.18: (a) Potential topology to meet the 3.52 Tbit/s capacity requirement from access to core using *packetponder* nodes; (b) Power consumption of each components [171].

5.5 Conclusion and Discussion

We demonstrated how a fully open and programmable *packetponder* can be re-defined to serve various purposes, unifying both metro and access networks and simplifying architectures. By integrating the OLTs and aggregation functionalities onto the same platform, we consolidated and simplified PON networks, and more generally, access deployment, in an open, standards-based orchestration to avoid vendor lock-in. It enabled platform and service consolidation by eliminating: surplus ASICs, internal/external transmission lines (e.g. backplane lanes, external optical fibre patchcords or DAC cables) and the associated transceivers from the data plane, which would promote greater energy efficiency.

Thus, we addressed the key challenges of simplifying transport networks, reducing capital, space, and power requirements, enabling dynamic wavelength allocation, and eliminating intermediate aggregation in optical communication networks. Our solution is software-configurable, enabling intelligent provisioning, and unit tunable wavelength backhaul with PON fronthaul mediated by a Tofino ASIC. It can aggregate different types of traffic and services coming from mobile devices, homes and businesses, into DWDM metro/core networks supporting OpenROADM architectures. This provides a straightforward plug-and-play approach for installations and reduced inventory expenses by allowing remote controlled optics with open-source APIs. The *packetponder* offers simplicity, flexibility and resilience in today's networks, with dynamic adaptability and the opportunity of performing 3R regeneration, wavelength conversion, and

multiple wavelength bands conversion within the same box [160]. All of this is achieved without impacting the cost level for operators who will need to reconcile long-term plans with an existing installed base.

In addition to optimising network architecture, various techniques are used to reduce energy consumption in networks. One approach involves dynamically switching interfaces, ASICs, processors, or transmission links on and off based on traffic demand [12]. This allows networks to scale down power usage during periods of low utilisation. Another method is carbon-aware routing [13], which directs traffic through paths with lower environmental impact, such as those powered by renewable energy. Additionally, reducing link capacity during off-peak times can further lower power consumption by adjusting transmission speeds to match demand. These strategies help improve energy efficiency while maintaining network performance.

Chapter 6

Conclusion

Chapter 6 concludes this thesis and presents potential avenues for future research.

Achieving sustainability improvements in our digital infrastructure landscape remains a complex challenge, yet urgent action is needed, as the energy consumption of digital infrastructure is expected to rise significantly over the next five years [172]. Although much of this growth will originate from data centres, the connectivity between computing and storage facilities, as well as between end users and their devices, must evolve similarly with the expansion of renewable energy and the networks that support these infrastructures [173]. TELCOs and data centre operators are already making significant steps in reducing carbon emissions and preparing for more sustainable next-generation networks, but their efforts have primarily focused on areas such as improved cooling systems, more energy-efficient computing hardware, and optimised methodologies to reduce the cost, time, and energy required for the explosive use of AI.

An equally important approach is the consolidation of network devices to eliminate redundant ASICs and transmission components such as SERDES, optical fibres, and pluggable transceivers, by leveraging advanced pluggables that integrate multiple hardware functions into a compact module harnessing Moore's Law. However, as we saw in this thesis, the true enabler of this shift lies in the software control of these multi-purpose devices. Recent standardisation efforts, driven by both academia and industry, are paving the way for open and interoperable management frameworks, making the adoption of such architectures increasingly feasible.

6. CONCLUSION

In the introduction of this thesis, we outlined the main challenges in achieving network consolidation. First, (i) merging packet and optical layers requires the seamless integration of pluggable transceivers into routers and switches, which must be enabled and managed by the Network Operating System. Second, (ii) a remote and centralised control plane is essential, relying on standardised models and open APIs to ensure vendor-neutral interoperability. Third, (iii) the controller must offer intelligent, end-to-end visibility across the network, enabling multi-layer routing decisions that make use of packet-optical nodes not just as termination points but also as in-path, tandem elements. Finally, (iv) consolidating the network edge involves combining metro/core functions, using coherent pluggables, with access services such as PON. These converged edge devices must also be centrally managed through open protocols and software interfaces.

To answer these challenges, this thesis introduced the *packetponder* concept; a fully open and programmable whitebox switch capable of accommodating a wide range of modular, common-form-factor pluggable transceivers. While traditionally designed to support electronic switching functions, its open software stack makes it a strong candidate for broader applications spanning access, metro, and core networks. As a proof of concept, this thesis demonstrated the *packetponder*'s feasibility across different network domains. At the edge, it efficiently aggregates high-radix, lower-speed access terminations with low-radix, high-bandwidth metro/core terminations within a single hardware element. By co-locating packet switching with 3R optical-electrical-optical wavelength conversion, the *packetponder* introduces a new architectural paradigm that has significant implications for the future evolution of TELCO networks.

Challenge (i) was particularly relevant at the start of my PhD in 2021, when coherent pluggables were just entering the market and beginning to be deployed by a few hyperscalers. At that time, the Common Management Interface Specification (CMIS) was still in its early stages, and no unified control mechanism existed for these pluggables. As a result, operators primarily used them for simple point-to-point links, and no open-source NOS supported their integration. Initially, the focus of this thesis was on legacy tunable optics, which themselves were underutilised, lacking deep integration into NOS environments. However, as coherent pluggables rapidly gained traction during the course of my PhD, the research became even more relevant: the techniques, software enhancements, and architectural considerations developed for legacy tunable transceivers were directly transferable to coherent optics, reinforcing the broader applicability and

timeliness of the *packetponder* concept.

Therefore, challenges (i) and (ii) were addressed in Chapter 3, where we demonstrated the successful integration of both coherent and legacy tunable pluggables into a programmable switch using standardised interfaces and common APIs. While other and more recent open software exist for programming pluggables, this thesis details the mechanism and integration steps required to embed these devices seamlessly into the NOS of the host system, and shows how to connect them to a centralised controller to enable dynamic reconfiguration.

Challenge (iii) was addressed in Chapter 4, where we demonstrated how a centralised controller can leverage the capabilities of *packetponders* to optimise wavelength availability in fault recovery scenarios. Specifically, we show that when the Routing and Wavelength Assignment algorithm takes into account the wavelength conversion feature enabled by the *packetponder*, the number of recoverable lightpaths significantly increases. This highlights the potential of such devices to act as intermediate elements in multi-layer recovery strategies, rather than mere endpoints. In the longer term, a more comprehensive controller should incorporate detailed physical layer parameters such as OSNR, non-linear impairments, and other signal quality metrics, to enable intelligent, end-to-end decisions across both packet and optical domains. This chapter also underlined the value of emulation tools like Mininet-optical, which allow researchers to scale network scenarios and validate controller logic before real-world deployment.

Finally, Challenge (iv) is addressed in Chapter 5, where the *packetponder* concept is extended to encompass the consolidation of access, metro, and core networks. The chapter demonstrates the feasibility of this unified architecture at the network edge, highlighting its potential for infrastructure simplification and energy savings. It details the integration of PON pluggable transceivers, particularly the MicroPlug OLT, into a common whitebox switch, using currently available software solutions to manage both the PON and the coherent/tunable pluggables introduced in Chapter 3. While our implementation validates the technical feasibility of this consolidation, a key area for future work lies in natively incorporating PON functionality into the host management system. In this work, we relied on MicroClimate Management System (MCMS), a widely adopted but proprietary software, for managing the TiBiT MicroPlug OLT. MCMS is the industry standard, and many companies build their own software layers on top of it. So a possible avenue for future work is to integrate PON management directly into the NOS of the switch. This requires the development of a vendor-

agnostic PON hardware abstraction layer that interfaces with the pluggable OLT, handles in-band management traffic, and abstracts hardware-specific details. The abstraction layer would expose on the northbound interface a switch-like OpenFlow/P4Runtime interface to the SDN controller (e.g., ONOS), so that OLTs could be managed similarly to packet-optical networks. However, this also introduces challenges, particularly in defining abstraction models and representing PON-specific QoS parameters, which are not trivial to map onto conventional SDN control paradigms [174]. Such control mechanism would enable more effective centralised management and support a truly plug-and-play architecture for flexible, vendor-neutral networks.

With such a hardware solution and a controller with full network visibility, seamless flexibility between network domains and layers becomes possible. In the event of a failure, such as a submarine cable break that we mentioned in the introduction, the network could dynamically reconfigure in real time, selecting the most optimal alternative path. This could involve maximising utilisation of already congested routes (such as other submarine cables), leveraging the packet layer alone, or even switching to alternative transport technologies if they offer lower latency. Additionally, the intelligent control system could temporarily reduce capacity on lower-priority links, while maintaining signal integrity and service quality. Intelligent management can also enable closed-loop control, where the network continuously monitors performance metrics and automatically adjusts configurations, ensuring rapid recovery and optimisation without requiring manual intervention. Although this thesis did not specifically focus on closed-loop control, adopting *packetponders* with open software and standards paves the way for its realisation.

Naturally, achieving this level of flexibility introduces additional software complexity. However, the benefits outweigh the challenges, particularly when intelligent management is paired with standardised and widely adopted APIs, ensuring interoperability with any SDN controller, including those from proprietary vendors/operators.

Further research opportunities

A promising area for future work is the expanded use of P4. In this thesis, P4 was primarily used for basic forwarding functions based on common header fields e.g., VLAN, ingress port, IPv4. However, its potential extends far beyond this,

particularly because it is capable of forwarding based on a wide range of headers and supporting multiple transport technologies. Given that the *packetponder* enables convergence across different network layers and domains, P4 could be leveraged not only for packet-based use cases but also for novel applications, such as optical QoS monitoring [175]. Additionally, exploring P4's role in access networks could further enhance its applicability, providing greater flexibility and programmability in disaggregated access architectures.

Another area for further development is the Mininet-optical Digital Twin controller. Mininet-optical presents a significant opportunity for packet-optical controller innovation and development, as it enables the creation of emulated networks that replicate real-world topologies. A primary goal would be to continue the development of state-of-the-art northbound interfaces, such as T-API or NETCONF, which are widely used in real devices. By ensuring compatibility with these models, Mininet-optical allows SDN controllers to be tested first on an emulated network before being deployed on a real testbed, requiring minimal or no modifications to the control algorithms. As an open-source project, Mininet-optical continues to evolve and supports more interfaces, with efforts aimed at tighter integration with popular SDN controller frameworks like ONOS. This makes it a valuable tool for testing complex algorithms and evaluating their impact in realistic network scenarios. As demonstrated with the incorporation of wavelength conversion into Routing and Wavelength Assignment algorithms, additional capabilities such as OSNR, BER, or other advanced transmission parameters, can be introduced to enhance network modelling. Mininet-optical provides a powerful research platform for developing and refining SDN controllers and algorithms, particularly in the areas of network resilience and performance evaluation.

The final area of exploration is the accurate assessment of the *packetponder*'s impact on network energy consumption using a dedicated sustainability or energy modelling tool. While this thesis has qualitatively demonstrated the potential for power savings through hardware consolidation and reduced transceiver and link usage, a detailed energy analysis is necessary to quantify the exact reduction in power consumption across different network scenarios. A more data-driven approach, incorporating real-world operator network statistics, as well as operator's external constraints, would improve the accuracy of such an analysis. Indeed, regulatory requirements, physical infrastructure limitations (e.g., fibre availability), or specific service-level agreements (SLAs) with enterprise customers, can influence network design choices that may impact the energy efficiency gains

achievable with the *packetponder*. Conducting a broader, real-world assessment with actual network data would provide a more precise estimation of its sustainability benefits.

Final remarks

This thesis has demonstrated that the availability of open and programmable switch has the potential to revolutionise network architectures, enabling them to keep pace with the rapid growth of current and future applications. By consolidating hardware functions and simplifying software management, the proposed *packetponder* solution offers a clear path toward more efficient, flexible, and sustainable networks.

This thesis was prepared using the LaTeX typesetting system, with Overleaf as the editing platform.

Acronyms

3R	Reamplification, Reshaping, and Retiming
AI	Artificial Intelligence
API	Application Programming Interface
ASE	Amplified Spontaneous Emission
ATM	Asynchronous Transfer Mode
BBU	Baseband Unit
BGP	Border Gateway Protocol
CFP	C form-factor pluggable
CIAN	Centre for Integrated Access Networks
CLI	Command Line Interface
CMIS	Common Management Interface Specification
CO	Central Office
DAC	Direct Attach Copper
DBA	Dynamic Bandwidth Allocation
DEMUX	Demultiplexer
DP-QPSK	Dual-Polarisation Quadrature Phase-Shift Keying
DSP	Digital Signal Processor
DUT	Device Under Test
DWDM	Dense Wavelength Division Multiplexing
eCPRI	enhanced Common Public Radio Interface

EDFA	Erbium Doped Fibre Amplifier
eRWA	enhanced RWA
F-F	First Fit
FEC	Forward Error Correction
FTTH	Fiber-to-the-Home
GFP	Generic Framing Procedure
gNMI	Network Management Interface
IA	Implementation Agreement
IETF	Internet Engineering Task Force
IMIX	Internet MIX
IoT	Internet of Things
IPoDWDM	IP-over-DWDM
JSON	JavaScript Object Notation
L2	Layer 2
L3	Layer 3
LP	Lightpath
MANTRA	Mandatory Transceiver Requirements Agreement
MCMS	MicroClimate Management System
MPLS	Multiprotocol Label Switching
MSA	Muti-Source Agreement
MSAN	Multi-Service Access Node
MUX	Multiplexer
NBI	Northbound Interface
NOS	Network Operating System
NRZ	Non-Return-to-Zero
OAM	Operations, Administration and Maintenance
OB-BAA	Open Broadband - Broadband Access Abstraction

OCP	Open Compute Project
ODL	OpenDaylight
ODM	Original Design Manufacturer
ODTN	Open and Disaggregated Transport Network
ODU	Optical Channel Data Unit
OEO	Optical-Electrical-Optical
OLS	Optical Line System
OLT	Optical Line Terminals
ONF	Open Networking Foundation
ONL	Open Network Linux
ONLP	ONL Platform
ONOS	Open Network Operating System
ONU	Optical Network Unit
OOPT	Open Optical & Packet Transport
OS	Operating System
OSFP	Octal Small Form-factor Pluggable
OSNR	Optical Signal-to-Noise Ratio
OTN	Optical Transport Network
P2MP	Point-to-Multipoint
P4	Programming Protocol-independent Packet Processors
PBB-TE	Provider Backbone Bridging Traffic Engineering
PDU	Protocol Data Unit
PHAL	Physical Hardware Access Layer
pOLTA	pluggable OLT abstraction
PON	Passive Optical Network
POP	Point of Presence
Protobuf	Protocol Buffers

QoS	Quality of Service
QoT-E	Quality-of-Transmission Estimation
QSFP-DD	Quad Small Form-factor Pluggable Double Density
ROADM	Reconfigurable Optical Add/Drop Multiplexer
RRH	Remote Radio Head
RU	Rack Unit
RWA	Routing and Wavelength Assignment
RX	Receiver
S-P	Shortest Path
SAI	Switch Abstraction Interface
SBI	Southbound Interface
SDE	Software Development Environment
SDH	Synchronous Digital Hierarchy
SDN	Software-Defined Networking
SERDES	Serializer/Deserializer
SFF	Small Form Factor
SLA	Service-level Agreements
SONET	Synchronous Optical Networking
TAI	Transponder Abstraction Interface
TAPI	Transport API
TDM	Time-Division Multiplexing
TELCO	Telecommunications network operators
TIP	Telecom Infra Project
TTG	Test Traffic Generator
TX	Transmitter
VOLTHA	Virtual OLT Hardware Abstraction
WDM	Wavelength Division Multiplexing

WSS Wavelength-Selective Switch

XC cross-connect

XML Extensible Markup Language

YANG Yet Another Next Generation

Appendix

Transceivers type with corresponding data rate, standards and SERDES speed

Transceiver Type	Typical Data Rate	Standard	Typical SERDES (Lanes × Speed)
SFP	Up to 1 Gbit/s	SFF	1 × 1.25 Gbit/s
SFP+	10 Gbit/s	SFF	1 × 10.3125 Gbit/s
SFP28	25 Gbit/s	SFF	1 × 25.78125 Gbit/s
QSFP	4 × 1G or 4 × 10G	SFF	4 × 10.3125 Gbit/s
QSFP+	40 Gbit/s	SFF	4 × 10.3125 Gbit/s
QSFP28	100 Gbit/s	SFF / CMIS	4 × 25.78125 Gbit/s
QSFP-DD	400 Gbit/s	CMIS / SFF	8 × 50 Gbit/s (PAM4, 25 Gbaud)
OSFP	400G	CMIS	8 × 50 Gbit/s (PAM4, 25 Gbaud)
OSFP	800G	CMIS	8 × 100 Gbit/s (PAM4, 56 Gbaud)
CFP	100 Gbit/s	CFP MSA	10 × 10 Gbit/s or 4 × 25 Gbit/s
CFP2	100–200 Gbit/s	CFP MSA	8 × 25 Gbit/s or 4 × 50 Gbit/s (PAM4)
CFP4	100 Gbit/s	CFP MSA	4 × 25 Gbit/s
QSFP112	400 Gbit/s	CMIS	4 × 100 Gbit/s (PAM4, 56 Gbaud)

References

- [1] *Shadow fleets and subaquatic sabotage: Are europe's undersea internet cables under attack?*, https://www.theguardian.com/world/ng-interactive/2025/mar/05/shadow-fleets-subaquatic-sabotage-europe-undersea-internet-cables-under-attack?CMP=share_btn_url, Accessed: 2025-03-18.
- [2] *Powering intelligence: Analyzing artificial intelligence and data center energy consumption, 2024*, <https://www.epri.com/research/products/3002028905>, Accessed: 2025-03-14.
- [3] *Data centres metered electricity consumption 2020*, <https://www.cso.ie/en/releasesandpublications/ep/p-dcmec/datacentresmeteredelectricityconsumption2020/>, Accessed: 2025-03-14.
- [4] *Data centres metered electricity consumption 2023*, <https://www.cso.ie/en/releasesandpublications/ep/p-dcmec/datacentresmeteredelectricityconsumption2023/keyfindings/>, Accessed: 2025-03-14.
- [5] *Ia : Le mur de l'énergie*, https://www.epsilon.com/tous-les-numeros/n45/ia_le_mur_de_l_energie/, Accessed: 2025-03-14.
- [6] E. Commission, D.-G. for Research, and Innovation, *European Green Deal – Research and innovation call*. Publications Office of the European Union, 2021. DOI: doi/10.2777/33415.
- [7] *Guidance for ict companies setting science based targets - mobile networks operators, fixed network operators and data centre operators*, https://sciencebasedtargets.org/resources/legacy/2020/04/GSMA_IP_SBT-report_WEB-SINGLE.pdf, Science Based Targets initiative (SBTi), 28 February 2020, London. Accessed: 2025-03-14.
- [8] L. A. Barroso, U. Hölzle, and P. Ranganathan, *The Datacenter as a Computer: Designing Warehouse-Scale Machines, Third Edition*. Springer, 2018.
- [9] S. Alouf, "Technical perspective: Can we uncover private backbone infrastructures?" *Commun. ACM*, vol. 66, no. 8, p. 94, Jul. 2023, ISSN: 0001-0782. DOI: 10.1145/3604621.
- [10] N. Zilberman, E. M. Schooler, U. Cummings, *et al.*, "Toward carbon-aware networking," *SIGENERGY Energy Inform. Rev.*, vol. 3, no. 3, pp. 15–20, Oct. 2023. DOI: 10.1145/3630614.3630618.

- [11] J. Frazelle, “Power to the people: Reducing datacenter carbon footprints,” *Queue*, vol. 18, no. 2, pp. 5–18, May 2020, ISSN: 1542-7730. DOI: 10.1145/3400899.3402527.
- [12] *Towards a more energy efficient future telecoms network*, <https://www.bt.com/bt-plc/assets/documents/about-bt/tech-fellowship/towards-a-more-energy-efficient-future-telecoms-network.pdf>, Accessed: 2025-02-23.
- [13] S. El-Zahr, P. Gunning, and N. Zilberman, “Exploring the benefits of carbon-aware routing,” *Proc. ACM Netw.*, vol. 1, no. CoNEXT3, Nov. 2023. DOI: 10.1145/3629165.
- [14] *Open compute project*, <https://www.opencompute.org/>, Accessed: 2025-03-14.
- [15] *Open networking foundation*, <https://opennetworking.org/>, Accessed: 2025-01-27.
- [16] *Telecom infra project (tip)*, <https://telecominfraproject.com/>, Accessed: 2025-01-27.
- [17] *Oif, "implementation agreement 400zr" oif-400zr-02.0*, <https://www.oiforum.com/wp-content/uploads/OIF-400ZR-02.0.pdf>, Accessed: 2025-03-05.
- [18] A. Chappe *et al.*, *Histoire de la télégraphie*. Ch. Richelet, 1840.
- [19] *Open optical and packet transport group*, <https://telecominfraproject.com/oopt/>, Accessed: 2025-01-27.
- [20] L. S. de Sousa and A. C. Drummond, *Metropolitan optical networks: A survey on new architectures and future trends*, 2022. arXiv: 2201.10709 [cs.NI].
- [21] N. Zilberman, P. M. Watts, C. Rotsos, and A. W. Moore, “Reconfigurable network systems and software-defined networking,” *Proceedings of the IEEE*, vol. 103, no. 7, pp. 1102–1124, 2015. DOI: 10.1109/JPROC.2015.2435732.
- [22] M. Casado, M. J. Freedman, J. Pettit, J. Luo, N. McKeown, and S. Shenker, “Ethane: Taking control of the enterprise,” ser. SIGCOMM ’07, Kyoto, Japan: Association for Computing Machinery, 2007, pp. 1–12, ISBN: 9781595937131. DOI: 10.1145/1282380.1282382.
- [23] N. McKeown, T. Anderson, H. Balakrishnan, *et al.*, “Openflow: Enabling innovation in campus networks,” *SIGCOMM Comput. Commun. Rev.*, vol. 38, no. 2, pp. 69–74, Mar. 2008, ISSN: 0146-4833. DOI: 10.1145/1355734.1355746.
- [24] I. R. T. F. (IRTF), *Software-defined networking (sdn): Layers and architecture terminology*, <https://datatracker.ietf.org/doc/html/rfc7426>, Accessed: 2025-03-05.
- [25] R. Wilton and N. Corran, *Device network management - current status, and future direction*, <https://datatracker.ietf.org/doc/slides-nemopsws-paper-device-network-management-current-status-and-future-direction/00/>, Accessed: 2025-03-05.
- [26] *Wedge 100*, <https://engineering.fb.com/2016/10/18/data-center-engineering/wedge-100-more-open-and-versatile-than-ever/>, Accessed: 2025-03-05.

- [27] Intel® tofino products, pcn 827577-00, product discontinuance, tofino end of life, <https://www.intel.com/content/www/us/en/content-details/827577/intel-tofino-products-pcn-827577-00-product-discontinuance-tofino-end-of-life.html>, Accessed: 2025-03-05.
- [28] E. H. Campbell, *P4 is not going anywhere*, <https://erichthewry.github.io/posts/2023-02-27-tofino.html>, Accessed: 2025-03-05.
- [29] Cisco silicon one q211 and q211l processors data sheet, <https://www.cisco.com/c/en/us/solutions/collateral/silicon-one/datasheet-c78-744834.html>, Accessed: 2025-03-05.
- [30] A. L. Stancu, S. Halunga, A. Vulpe, G. Suciu, O. Fratu, and E. C. Popovici, “A comparison between several software defined networking controllers,” in *2015 12th International Conference on Telecommunication in Modern Satellite, Cable and Broadcasting Services (TELSIKS)*, 2015, pp. 223–226. DOI: 10.1109/TELSKS.2015.7357774.
- [31] R. K. Arbettu, R. Khondoker, K. Bayarou, and F. Weber, “Security analysis of opendaylight, onos, rosemary and ryu sdn controllers,” in *2016 17th International Telecommunications Network Strategy and Planning Symposium (Networks)*, 2016, pp. 37–44. DOI: 10.1109/NETWKS.2016.7751150.
- [32] “software-defined networks: A systems approach” chapter 6: Network os, <https://sdn.systemsapproach.org/onos.html>, Accessed: 2025-03-05.
- [33] X. Xue, F. Nakamura, K. Prifti, *et al.*, “Sdn enabled flexible optical data center network with dynamic bandwidth allocation based on photonic integrated wavelength selective switch,” *Opt. Express*, vol. 28, no. 6, pp. 8949–8958, Mar. 2020. DOI: 10.1364/OE.388759.
- [34] H.-J. Schmidtke, I. Lyubomirsky, and B. Taylor, “Packet-optical integration and trend towards white boxes,” in *2017 Optical Fiber Communications Conference and Exhibition (OFC)*, 2017, pp. 1–3.
- [35] *Cassini at a glance - tip*, https://telecominfraproject.com/wp-content/uploads/Cassini_TIP_v0.8_2019_10_28_approved.pdf, Accessed: 2025-03-05.
- [36] *Edge-core’s phoenix amx3200*, <https://www.edge-core.com/wp-content/uploads/2025/01/AMX3200-Datasheet-R01.pdf>, Accessed: 2025-02-23.
- [37] *Czech light roadm*, <https://czechlight.cesnet.cz/en/network-elements/reconfigurable-add-drop-roadm/>, Accessed: 2025-03-05.
- [38] *Lumentum’s roadm graybox*, <https://www.lumentum.com/en/products/dci-roadm-graybox>, Accessed: 2025-03-05.
- [39] L. Liu, D. Zhang, T. Tsuritani, *et al.*, “Field trial of an openflow-based unified control plane for multilayer multigranularity optical switching networks,” *Journal of Lightwave Technology*, vol. 31, no. 4, pp. 506–514, 2013. DOI: 10.1109/JLT.2012.2212179.

- [40] H. Y. Choi, L. Liu, T. Tsuritani, and I. Morita, “Demonstration of ber-adaptive wson employing flexible transmitter/receiver with an extended openflow-based control plane,” *IEEE Photonics Technology Letters*, vol. 25, no. 2, pp. 119–121, 2013. DOI: 10.1109/LPT.2012.2228186.
- [41] N. Sambo, G. Meloni, F. Paolucci, *et al.*, “Programmable transponder, code and differentiated filter configuration in elastic optical networks,” *Journal of Lightwave Technology*, vol. 32, no. 11, pp. 2079–2086, 2014. DOI: 10.1109/JLT.2014.2319859.
- [42] R. Casellas, R. Martínez, R. Vilalta, and R. Muñoz, “Control, management, and orchestration of optical networks: Evolution, trends, and challenges,” *Journal of Lightwave Technology*, vol. 36, no. 7, pp. 1390–1402, 2018. DOI: 10.1109/JLT.2018.2793464.
- [43] J. Kunderát, J. Vojtěch, P. Škoda, R. Vohnout, J. Radil, and O. Havliš, “Yang/netconf roadm: Evolving open dwdm toward sdn applications,” *Journal of Lightwave Technology*, vol. 36, no. 15, pp. 3105–3114, 2018. DOI: 10.1109/JLT.2018.2822268.
- [44] L. Velasco, A. Sgambelluri, R. Casellas, *et al.*, “Building autonomic optical whitebox-based networks,” *Journal of Lightwave Technology*, vol. 36, no. 15, pp. 3097–3104, 2018. DOI: 10.1109/JLT.2018.2829700.
- [45] A. Campanella, H. Okui, A. Mayoral, *et al.*, “Odn: Open disaggregated transport network. discovery and control of a disaggregated optical network through open source software and open apis,” in *2019 Optical Fiber Communications Conference and Exhibition (OFC)*, 2019, pp. 1–3.
- [46] *Open disaggregated transport network*, <https://opennetworking.org/reference-designs/odtn/>, Accessed: 2025-02-05.
- [47] M. N. Ellanti, S. S. Gorshe, L. G. Raman, and W. D. Grover, *Next Generation Transport Networks: Data, Management, and Control Planes*, 1st. Springer Publishing Company, Incorporated, 2010, ISBN: 1441936874.
- [48] S. Bjornstad, “Can otn be replaced by ethernet? a network level comparison of otn and ethernet with a 5g perspective,” in *2018 International Conference on Optical Network Design and Modeling (ONDM)*, 2018, pp. 220–225. DOI: 10.23919/ONDM.2018.8396134.
- [49] J. L. Vizcaíno, Y. Ye, F. Jiménez, *et al.*, “Quality of protection schemes with extended flexibility for improved energy efficiency in transport networks,” in *2013 9th International Conference on the Design of Reliable Communication Networks (DRCN)*, 2013, pp. 28–35.
- [50] E. Hernandez-Valencia, M. Scholten, and Z. Zhu, “The generic framing procedure (gfp): An overview,” *IEEE Communications Magazine*, vol. 40, no. 5, pp. 63–71, 2002. DOI: 10.1109/35.1000215.
- [51] B. Cai, R. Dunsmore, R. Naphan, S. Lisle, D. Patel, and I. Syed, *Odu path protection in a disaggregated otn switching system*, <https://patentimages.storage.googleapis.com>.

- com/00/c8/0f/1c3d5ddedd32ee/US20170310387A1.pdf, US 2017 / 0310387 A1, 2017.
- [52] *Ribbon communications blog: Are we finally entering the ip-over-dwdm era?* <https://ribboncommunications.com/company/media-center/blog/are-we-finally-entering-ip-over-dwdm-era>, Accessed: 2025-03-05.
 - [53] A. Reid, P. Willis, I. Hawkins, and C. Bilton, "Carrier ethernet," *IEEE Communications Magazine*, vol. 46, no. 9, pp. 96–103, 2008. DOI: 10.1109/MCOM.2008.4623713.
 - [54] R. Vaishampayan, A. Gumaste, S. Rana, and N. Ghani, "Application driven comparison of t-mpls/mpls-tp and pbb-te - driver choices for carrier ethernet," in *IEEE INFOCOM Workshops 2009*, 2009, pp. 1–6. DOI: 10.1109/INFCOMW.2009.5072112.
 - [55] S.-A. Li, H. Huang, Z. Pan, *et al.*, "Enabling technology in high-baud-rate coherent optical communication systems," *IEEE Access*, vol. 8, pp. 111 318–111 329, 2020. DOI: 10.1109/ACCESS.2020.3003331.
 - [56] G. G. Shahidi, "Chip power scaling in recent cmos technology nodes," *IEEE Access*, vol. 7, pp. 851–856, 2019. DOI: 10.1109/ACCESS.2018.2885895.
 - [57] *Ethernet interfaces user guide for routing devices*, <https://www.juniper.net/documentation/us/en/software/junos/interfaces-ethernet/topics/concept/400g-zr-zrm-overview.html>, Accessed: 2025-03-05.
 - [58] T. Zami and B. Lavigne, "Optimal deployments of 400 gb/s multihaul cfp2-dco transponders in transparent ipowdm core networks," in *2022 Optical Fiber Communications Conference and Exhibition (OFC)*, 2022, pp. 01–03.
 - [59] N. Camuzcu, Q. Zhang, M. Ibrahimi, R. Wang, L. Zhang, and M. Tornatore, "Minimizing power consumption in ipowdm networks with zr/zr+ using network reconfiguration," in *2024 Asia Communications and Photonics Conference (ACP) and International Conference on Information Photonics and Optical Communications (IPOC)*, 2024, pp. 1–6. DOI: 10.1109/ACP/IPOC63121.2024.10809949.
 - [60] Q. Zhang, A. Morea, P. Layec, M. Ibrahimi, F. Musumeci, and M. Tornatore, "Power-consumption analysis for different ipowdm network architectures with zr/zr+ and long-haul muxponders," *Journal of Optical Communications and Networking*, vol. 16, no. 12, pp. 1189–1203, 2024. DOI: 10.1364/JOCN.531536.
 - [61] R. P. Davey, M. A. Iqbal, and P. D. Wright, "Zr 400 gbit/s and 800 gbit/s use cases, trials, deployments, and future prospects [invited]," *Journal of Optical Communications and Networking*, vol. 16, no. 1, A33–A39, 2024. DOI: 10.1364/JOCN.499365.
 - [62] A. Gumaste, M. Sosa, H. Bock, and P. Kandappan, "Optimized ip-over-wdm core networks using zr+ and flexible muxponders for 400 gb/s and beyond," *Journal of Optical Communications and Networking*, vol. 14, no. 3, pp. 127–139, 2022. DOI: 10.1364/JOCN.444943.

- [63] *Programmability configuration guide for cisco 8000 series routers, ios xr release 7.10.x*, <https://www.cisco.com/c/en/us/td/docs/iosxr/cisco8000/programmability/710x/b-programmability-cg-8000-710x/drive-network-automation-using-programmable-yang-data-models.html>, Accessed: 2025-03-05.
- [64] *Openroadm multi-source agreement*, <http://openroadm.org/>, Accessed: 2025-03-05.
- [65] *Openconfig website*, <https://www.openconfig.net/>, Accessed: 2025-03-05.
- [66] *Common management interface specification (cmis)*, <https://www.oiforum.com/technical-work/hot-topics/management/>, Accessed: 2025-01-22.
- [67] "open zr+ msa technical specification", rev. 1.0, https://openzrplus.org/site/assets/files/1075/openzrplus_1p0.pdf, Accessed: 2025-03-05.
- [68] *Cisco 400g qsfp-dd high-power (bright) optical module data sheet*, <https://www.cisco.com/c/en/us/products/collateral/interfaces-modules/transceiver-modules/400g-qsfp-dd-high-power-optical-module-ds.html>, Accessed: 2025-01-22.
- [69] *The open xr optics forum*, <https://openxropticsforum.org/>, Accessed: 2025-03-05.
- [70] *Oif to take center stage at ofc 2024 with groundbreaking multi-vendor interoperability demonstrations and expert panels*, <https://www.ofcconference.org/en-us/home/news-and-press/exhibitor-press-releases/2024/oif-to-take-center-stage-at-ofc-2024-with-groundbr/>, Accessed: 2025-03-05.
- [71] T. I. Project, *Telecom infra project: Mantra whitepaper - ipowdm convergent sdn architecture - motivation, technical definition and challenges*, https://telecominfraproject.com/wp-content/uploads/TIP_OOPT_MANTRA_IP_over_DWDM_Whitepaper-Final-Version3.pdf, Accessed: 2025-01-27.
- [72] *Github: Tai (transponder abstraction interface)*, <https://github.com/Telecominfraproject/oopt-tai>, Accessed: 2025-01-27.
- [73] *Github: Goldstone nos*, <https://github.com/Telecominfraproject/oopt-goldstone>, Accessed: 2025-01-22.
- [74] *Stratum*, <https://p4.org/onf-product/stratum/>, Accessed: 2025-01-22.
- [75] *Oif*, <https://www.oiforum.com/>, Accessed: 2025-01-22.
- [76] *Sonic*, <https://sonicfoundation.dev/>, Accessed: 2025-01-22.
- [77] D. Scano, A. Giorgetti, A. Sgambelluri, *et al.*, "Hierarchical control of sonic-based packet-optical nodes encompassing coherent pluggable modules," in *2021 European Conference on Optical Communication (ECOC)*, 2021, pp. 1–3. DOI: 10.1109/ECOC52684.2021.9605850.
- [78] M. Bjorklund, *Rfc 6020: Yang - a data modeling language for the network configuration protocol (netconf)*, USA, 2010.
- [79] *P416 portable switch architecture (psa)*, <https://p4.org/p4-spec/docs/PSA.html>, Accessed: 2025-01-22.

- [80] B. Ramamurthy and B. Mukherjee, "Wavelength conversion in wdm networking," *IEEE Journal on Selected Areas in Communications*, vol. 16, no. 7, pp. 1061–1073, 1998. DOI: 10.1109/49.725178.
- [81] *Sff-8472 specification for management interface for sfp+*, <https://members.snia.org/document/dl/25917>, Accessed: 2025-01-22.
- [82] *Common management interface specification (cmis) rev 5.0*, <http://www.qsfp-dd.com/wp-content/uploads/2021/05/CMIS5p0.pdf>, Accessed: 2025-01-22.
- [83] S. Ahearne, Y. Verbishchuk, C. Sreenan, and F. Gunning, "Software defined control of tunable optical transceivers using netconf and yang," in *2018 European Conference on Networks and Communications (EuCNC)*, 2018, pp. 1–86. DOI: 10.1109/EuCNC.2018.8443188.
- [84] *Sff-8690 specification for tunable sfp+ memory map for itu frequencies*, <https://members.snia.org/document/dl/51536>, Accessed: 2025-01-22.
- [85] *Brian o'connor; stratum overview and update - onfconnect19*, <https://opennetworking.org/wp-content/uploads/2019/09/2.00pm-Brian-OConnor-Stratum.pdf>, Accessed: 2025-01-22.
- [86] *Onos*, <https://opennetworking.org/onos/>, Accessed: 2025-01-22.
- [87] A. Giorgetti, A. Sgambelluri, R. Casellas, R. Morro, A. Campanella, and P. Castoldi, "Control of open and disaggregated transport networks using the open network operating system (onos) [invited]," *Journal of Optical Communications and Networking*, vol. 12, no. 2, A171–A181, 2020. DOI: 10.1364/JOCN.12.00A171.
- [88] *Openconfig yang model: Openconfig-platform*, <https://openconfig.net/projects/models/schemadocs/yangdoc/openconfig-interfaces.html>, Accessed: 2025-01-22.
- [89] *Openconfig yang model: Openconfig-interfaces*, <https://openconfig.net/projects/models/schemadocs/yangdoc/openconfig-platform.html>, Accessed: 2025-01-22.
- [90] *Github pyang*, <https://github.com/mbj4668/pyang?tab=readme-ov-file>, Accessed: 2025-01-22.
- [91] *Github: Onos - wavelength-config cli command*, <https://github.com/opennetworkinglab/onos/blob/master/apps/optical-model/src/main/java/org/onosproject/net/optical/cli/PortWaveLengthCommand.java>, Accessed: 2025-01-28.
- [92] A. Sgambelluri, D. Scano, R. Morro, *et al.*, "Failure recovery in the mantra architecture with an ipowdm sonic node and 400zr/zr+ pluggables," *Journal of Optical Communications and Networking*, vol. 16, no. 5, B26–B34, 2024. DOI: 10.1364/JOCN.514179.
- [93] *Tip mantra whitepaper ipodwdm poc 2024 results readout*, https://cdn.mediavalet.com/usva/telecominfraproject/QQhtPYmJVkm88JiJ5xGULA/eg_gHL_g2kGxGDHQisAs3A/Original/TIP_OOPT_MANTRA_IPoDWDWM_PoC_Results_readout_whitepaper_v.1.0.pdf, Accessed: 2025-01-22.

- [94] R. Morro, E. Riccardi, D. Scano, *et al.*, “First demonstration of mantra ipowdm convergent sdn architecture using sonic white box and 400zr/zr+ pluggables,” in *2023 International Conference on Optical Network Design and Modeling (ONDM)*, 2023, pp. 1–3. DOI: 10.23919/ONDM57372.2023.10144848.
- [95] I. Maor, O. Gerstel, V. Lopez, *et al.*, “First demonstration of sdn-controlled multi-layer restoration and its advantage over optical restoration,” in *ECOC 2016; 42nd European Conference on Optical Communication*, 2016, pp. 1–3.
- [96] A. Felix, N. Borges, H. Wu, M. Hanlon, M. Birk, and A. Tschersich, “Multi-layer sdn on a commercial network control platform for packet optical networks,” in *OFC 2014*, 2014, pp. 1–3. DOI: 10.1364/OFC.2014.Th5A.1.
- [97] A. M. Lopez-de-Lerma, V. López, M. L. Bravo, *et al.*, “Multi-layer service provisioning over resilient software-defined partially disaggregated networks,” in *2019 Optical Fiber Communications Conference and Exhibition (OFC)*, 2019, pp. 1–3.
- [98] E. Riccardi and D. Scano, “Sdn control of packet-optical edge network nodes,” in *2022 13th International Symposium on Communication Systems, Networks and Digital Signal Processing (CSNDSP)*, 2022, pp. 817–819. DOI: 10.1109/CSNDSP54353.2022.9907986.
- [99] A. Giorgetti, D. Scano, A. Sgambelluri, *et al.*, “Enabling hierarchical control of coherent pluggable transceivers in sonic packet-optical nodes,” *Journal of Optical Communications and Networking*, vol. 15, no. 3, pp. 163–173, 2023. DOI: 10.1364/JOCN.477732.
- [100] O. Gerstel, G. Galimberti, and B. Foster, “A control hierarchy for integrated packet-optical networks utilizing pluggable transceivers,” in *2022 Optical Fiber Communications Conference and Exhibition (OFC)*, 2022, pp. 1–3.
- [101] J. Ali, G.-m. Lee, B.-h. Roh, D. K. Ryu, and G. Park, “Software-defined networking approaches for link failure recovery: A survey,” *Sustainability*, vol. 12, no. 10, 2020, ISSN: 2071-1050. DOI: 10.3390/su12104255.
- [102] A. A. Díaz-Montiel, B. Lantz, J. Yu, D. Kilper, and M. Ruffini, “Real-time qot estimation through sdn control plane monitoring evaluated in mininet-optical,” *IEEE Photonics Technology Letters*, vol. 33, no. 18, pp. 1050–1053, 2021. DOI: 10.1109/LPT.2021.3075277.
- [103] E. W. Dijkstra, “A note on two problems in connexion with graphs,” *Numer. Math.*, vol. 1, no. 1, pp. 269–271, Dec. 1959, ISSN: 0029-599X. DOI: 10.1007/BF01386390.
- [104] R. Bellman, “On a routing problem,” *Quarterly of applied mathematics*, vol. 16, no. 1, pp. 87–90, 1958.
- [105] G. Shen, S. K. Bose, T. H. Chengand, C. Lu, and T. Y. Chai, “Performance analysis under dynamic loading of wavelength continuous and non-continuous wdm networks with shortest-path routing,” *International Journal of Communication*

- Systems*, vol. 14, no. 4, pp. 407–418, May 2001, ISSN: 1074-5351. DOI: 10.1002/dac.479.
- [106] P.-H. Ho and H. Mouftah, “An approach for enhancing fixed alternate routing in dynamic wavelength-routed wdm networks,” in *Global Telecommunications Conference, 2002. GLOBECOM '02. IEEE*, vol. 3, 2002, 2792–2797 vol.3. DOI: 10.1109/GLOCOM.2002.1189138.
 - [107] L. H. Bonani, J. C. F. Queiroz, M. L. F. Abbade, and F. Callegati, “Load balancing in fixed-routing optical networks with weighted ordering heuristics,” *J. Opt. Commun. Netw.*, vol. 11, no. 3, pp. 26–38, Mar. 2019. DOI: 10.1364/JOCN.11.000026.
 - [108] S. Chen and K. Nahrstedt, “On finding multi-constrained paths,” in *ICC '98. 1998 IEEE International Conference on Communications. Conference Record. Affiliated with SUPERCOMM'98 (Cat. No.98CH36220)*, vol. 2, 1998, 874–879 vol.2. DOI: 10.1109/ICC.1998.685137.
 - [109] T. Korkmaz and M. Krunz, “Multi-constrained optimal path selection,” in *Proceedings IEEE INFOCOM 2001. Conference on Computer Communications. Twentieth Annual Joint Conference of the IEEE Computer and Communications Society (Cat. No.01CH37213)*, vol. 2, 2001, 834–843 vol.2. DOI: 10.1109/INFCOM.2001.916274.
 - [110] H. Zhou, Z. Tian, S. M. Sefat, and S. M. Mostafa, “An optimal sdn-based wavelength allocation and routing method for 5g network,” *Mobile Information Systems*, 2022.
 - [111] I. Martín, S. Troia, J. A. Hernández, *et al.*, “Machine learning-based routing and wavelength assignment in software-defined optical networks,” *IEEE Transactions on Network and Service Management*, vol. 16, no. 3, pp. 871–883, 2019. DOI: 10.1109/TNSM.2019.2927867.
 - [112] K.-C. Lee and V. Li, “Routing and switching in a wavelength convertible optical network,” in *IEEE INFOCOM '93 The Conference on Computer Communications, Proceedings*, 1993, 578–585 vol.2. DOI: 10.1109/INFCOM.1993.253315.
 - [113] R. Ramaswami and K. Sivarajan, “Routing and wavelength assignment in all-optical networks,” *IEEE/ACM Transactions on Networking*, vol. 3, no. 5, pp. 489–500, 1995. DOI: 10.1109/90.469957.
 - [114] E. Riccardi, P. Gunning, Ó. G. de Dios, M. Quagliotti, V. López, and A. Lord, “An operator view on the introduction of white boxes into optical networks,” *Journal of Lightwave Technology*, vol. 36, no. 15, pp. 3062–3072, 2018. DOI: 10.1109/JLT.2018.2815266.
 - [115] G. Davoli, R. Di Tommaso, A. Giorgetti, and C. Raffaelli, “Impact of lightpath selection on end-to-end service orchestration in disaggregated optical networks,” in *2024 International Conference on Optical Network Design and Modeling (ONDM)*, 2024, pp. 1–6. DOI: 10.23919/ONDM61578.2024.10582736.

- [116] A. Sgambelluri, D. Scano, R. Morro, *et al.*, “Failure recovery in the mantra architecture with an ipowdm sonic node and 400zr/zr+ pluggables,” *Journal of Optical Communications and Networking*, vol. 16, no. 5, B26–B34, 2024. DOI: 10.1364/JOCN.514179.
- [117] Y. R. Zhou, K. Smith, M. Gilson, *et al.*, “Field trial demonstration of real-time 400gbe optical transport over both conventional and non-contiguous superchannels using configurable modulation formats,” in *2019 Optical Fiber Communications Conference and Exhibition (OFC)*, 2019, pp. 1–3.
- [118] *Mininet-optical*, <https://mininet-optical.org/>, Accessed: 2025-02-05.
- [119] *Geant network*, <https://network.geant.org/>, Accessed: 2025-02-05.
- [120] *Mininet*, <https://mininet.org/>, Accessed: 2025-02-05.
- [121] B. Lantz, J. Yu, A. Bhardwaj, *et al.*, “Sdn-controlled dynamic front-haul provisioning, emulated on hardware and virtual cosmos optical x-haul testbeds,” in *2021 Optical Fiber Communications Conference and Exhibition (OFC)*, 2021, pp. 1–3.
- [122] *On the disaggregation of optical networks - alan a. diaz montiel, ph. d. dissertation, trinity college dublin, 2021.* <http://www.tara.tcd.ie/handle/2262/96451>, Accessed: 2025-02-05.
- [123] *Open vswitch*, <https://www.openvswitch.org/>, Accessed: 2025-02-05.
- [124] R. Jensen, A. Lord, and N. Parsons, “Colourless, directionless, contentionless roadm architecture using low-loss optical matrix switches,” in *36th European Conference and Exhibition on Optical Communication*, 2010, pp. 1–3. DOI: 10.1109/ECOC.2010.5621248.
- [125] *Github: Rwa simulator over wdm optical networks*, <https://github.com/cassiotbatista/rwa-wdm-sim>, Accessed: 2025-02-05.
- [126] *Optical network simulator (ons)*, <https://ons-simulator.com/>, Accessed: 2025-02-05.
- [127] *Chaos monkey documentation*, <https://netflix.github.io/chaosmonkey/>, Accessed: 2025-02-05.
- [128] *Software-defined networks: A systems approach - chapter 9: Access networks*, <https://sdn.systemsapproach.org/access.html>, Accessed: 2025-02-23.
- [129] *Gigabit-capable passive optical networks (GPON): General characteristics*, Series G: Transmission systems and media, digital systems and networks, International Telecommunication Union, Telecommunication Standardization Sector (ITU-T), 2008.
- [130] *10-Gigabit-capable passive optical networks (XG-PON): General requirements*, Series G: Transmission systems and media, digital systems and networks, International Telecommunication Union, Telecommunication Standardization Sector (ITU-T), 2016.
- [131] *Higher speed passive optical networks (50G-PON): General requirements*, Series G: Transmission systems and media, digital systems and networks, International

- Telecommunication Union, Telecommunication Standardization Sector (ITU-T), 2021.
- [132] *IEEE Standard for Information technology—Telecommunications and information exchange between systems - Local and metropolitan area networks—Specific requirements Part 3: Carrier Sense Multiple Access with Collision Detection (CSMA/CD) Access Method and Physical Layer Specifications—Amendment: Media Access Control Parameters, Physical Layers, and Management Parameters for Subscriber Access Networks*, Institute of Electrical and Electronics Engineers (IEEE), 2004. DOI: 10.1109/IEEESTD.2004.94612.
 - [133] *IEEE Standard for Information technology—Telecommunications and information exchange between systems - Local and metropolitan area networks—Specific requirements Part 3: Carrier Sense Multiple Access with Collision Detection (CSMA/CD) Access Method and Physical Layer Specifications—Amendment: Physical Layer Specifications and Management Parameters for 10 Gb/s Passive Optical Networks*, Institute of Electrical and Electronics Engineers (IEEE), 2009. DOI: 10.1109/IEEESTD.2009.5306327.
 - [134] D. de Pintos, N. Merayo, C. Sangrador, J. C. Aguado, I. de Miguel, and R. J. Duran Barroso, “Software defined networking agent demonstration to enable configuration and management of xgs-pon architectures,” *Journal of Optical Communications and Networking*, vol. 15, no. 9, pp. 620–637, 2023. DOI: 10.1364/JOCN.494694.
 - [135] S. S. W. Lee, K.-Y. Li, and M.-S. Wu, “Design and implementation of a gpon-based virtual openflow-enabled sdn switch,” *Journal of Lightwave Technology*, vol. 34, no. 10, pp. 2552–2561, 2016. DOI: 10.1109/JLT.2016.2540244.
 - [136] R. G. Clegg, J. Spencer, R. Landa, M. Thakur, J. Mitchell, and M. Rio, “Pushing software defined networking to the access,” in *2014 Third European Workshop on Software Defined Networks*, 2014, pp. 31–36. DOI: 10.1109/EWSDN.2014.11.
 - [137] N. Merayo, D. de Pintos, J. C. Aguado, *et al.*, “Experimental validation of an sdn residential network management proposal over a gpon testbed,” *Opt. Switch. Netw.*, vol. 42, no. C, Nov. 2021, ISSN: 1573-4277. DOI: 10.1016/j.osn.2021.100631.
 - [138] *Virtual olt hardware abstraction (voltha)*, <https://opennetworking.org/voltha/>, Accessed: 2025-02-23.
 - [139] *Open broadband – broadband access abstraction (ob-baa)*, <https://www.broadband-forum.org/open-broadband/open-broadband-software/open-broadband-broadband-access-abstraction-ob-baa>, Accessed: 2025-02-23.
 - [140] *The linux foundation press release - lf broadband drives development and adoption of open source broadband transforming disaggregated pon architectures*, <https://www.linuxfoundation.org/press/lf-broadband-drives-development-and-adoption->

- of - open - source - broadband - transforming - disaggregated - pon - architectures, Accessed: 2025-02-23.
- [141] *Tibit microplug olt*, <https://www.ciena.com/interconnects/tibit-technologies>, Accessed: 2025-02-23.
- [142] K. Nishimoto, K. Asaka, J.-i. Kani, and J. Terada, “Automated provisioning method for a modular pon-olt device toward plug and provision,” *Journal of Optical Communications and Networking*, vol. 12, no. 9, pp. D9–D16, 2020. DOI: 10.1364/JOCN.390901.
- [143] M. Wang, G. Simon, L. A. Neto, *et al.*, “Sdn-oriented disaggregated optical access node for converged 5g mobile and residential services,” in *2021 European Conference on Optical Communication (ECOC)*, 2021, pp. 1–4. DOI: 10.1109/ECOC52684.2021.9606171.
- [144] *Pluggable olt abstraction (polta)*, https://www.bisdn.de/wp-content/uploads/BISDN_A4_pOLTA_DataSheet_ok.pdf, Accessed: 2025-02-23.
- [145] *Ip infusion network operating system software ocnos*, <https://www.ipinfusion.com/solutions/>, Accessed: 2025-02-23.
- [146] *Ciena’s universal aggregation solutions*, <https://www.ciena.com/solutions/universal-aggregation>, Accessed: 2025-02-23.
- [147] *Edge-core’s 800g cell site router csr440*, <https://www.edge-core.com/product/csr440/>, Accessed: 2025-02-23.
- [148] *Dzs’s m4000 datasheet*, https://dzsi.com/wp-content/uploads/M4000_Datasheet.pdf, Accessed: 2025-02-23.
- [149] *Ribbon communications’ high availability multi-service aggregation m6424*, <https://ribboncommunications.com/products/service-provider-products/ip-routing/access-routers/m6424>, Accessed: 2025-02-23.
- [150] *Edge-core press release: Edgecore networks partners with exaware and fonex to deliver an open multi-service olt access router using tibit transceivers*, <https://www.edge-core.com/news-inquiry.php?cls=1&id=1003>, Accessed: 2025-02-23.
- [151] *Press release - infinera announces xr optics – game-changing technology for transport network transformation*, <https://shorturl.at/RNxtp>, Accessed: 2025-02-23.
- [152] D. Welch, A. Napoli, J. Bäck, *et al.*, “Point-to-multipoint optical networks using coherent digital subcarriers,” *Journal of Lightwave Technology*, vol. 39, no. 16, pp. 5232–5247, 2021. DOI: 10.1109/JLT.2021.3097163.
- [153] J. Bäck, P. Wright, J. Ambrose, *et al.*, “Capex savings enabled by point-to-multipoint coherent pluggable optics using digital subcarrier multiplexing in metro aggregation networks,” in *2020 European Conference on Optical Communications (ECOC)*, 2020, pp. 1–4. DOI: 10.1109/ECOC48923.2020.9333233.
- [154] A. Napoli, Z. Stevkovski, J. D. M. Jiménez, *et al.*, “Enabling router bypass and saving cost using point-to-multipoint transceivers for traffic aggregation,” in *2022 Optical Fiber Communications Conference and Exhibition (OFC)*, 2022, pp. 1–3.

- [155] A. Gumaste, J. Pedro, and H. Bock, “Exploring point-to-multipoint coherent capabilities across metro and core networks,” in *2022 European Conference on Optical Communication (ECOC)*, 2022, pp. 1–4.
- [156] D. Hillerkuss, S. J. Hand, K. Braet, *et al.*, “Field trial of a management architecture for advanced smart pluggable transceivers in diverse hosts,” in *49th European Conference on Optical Communications (ECOC 2023)*, vol. 2023, 2023, pp. 1282–1285. DOI: 10.1049/icp.2023.2526.
- [157] *Press release - infinera announces xr optics – game-changing technology for transport network transformation*, <https://www.infinera.com/blog/turbocharge-your-single-fiber-access-pon-network/tag/access-and-aggregation/>, Accessed: 2025-02-23.
- [158] *Adtran blog: What is ip-to-dwdm and why is it important?* <https://www.edge-core.com/news-inquiry.php?cls=1&id=1003>, Accessed: 2025-02-23.
- [159] *Lightwave article: Tibit intros pluggable pon olt*, <https://www.lightwaveonline.com/broadband/article/55027007/tibit-intros-pluggable-pon-olt>, Accessed: 2025-02-23.
- [160] C. DeSanti, L. Du, J. Guarin, J. Bone, and C. F. Lam, “Super-pon: An evolution for access networks [invited],” *Journal of Optical Communications and Networking*, vol. 12, no. 10, pp. D66–D77, 2020. DOI: 10.1364/JOCN.391846.
- [161] G. Ronniger, I. Sackey, T. Kernetzky, *et al.*, “Efficient ultra-broadband c-to-o band converter based on multi-mode silicon-on-insulator waveguides,” in *2021 European Conference on Optical Communication (ECOC)*, 2021, pp. 1–4. DOI: 10.1109/ECOC52684.2021.9606033.
- [162] *Microclimate™ management system [mcms] pon manager (webui) user guide*, <https://www.juniper.net/documentation/us/en/software/unified-pon/unified-pon-microclimate-pon-manager-user-guide.pdf>, Accessed: 2025-02-23.
- [163] *10-Gigabit-capable symmetric passive optical network (XGS-PON)*, Series G: Transmission systems and media, digital systems and networks, International Telecommunication Union, Telecommunication Standardization Sector (ITU-T), 2020.
- [164] *Intel white paper: Micro-pop - a converged vbng and olt solution*, <https://www.silicom-usa.com/wp-content/uploads/2021/10/micro-pop-a-converged-vbng-and-olt-solution-1632524288.pdf>, Accessed: 2025-02-23.
- [165] *Resilience study research for nic - systems analysis of interdependent network vulnerabilities*, <https://nic.org.uk/app/uploads/Infrastructure-network-analysis.pdf>, Accessed: 2025-02-23.
- [166] *Bt newsroom: Bt group refines its digital switchover programme for the uk’s full fibre future*, <https://newsroom.bt.com/bt-group-refines-its-digital-switchover-programme-for-the-uks-full-fibre-future/>, Accessed: 2025-02-23.

- [167] *Telefonica communication room: Telefónica closes all its copper central offices*, <https://newsroom.bt.com/bt-group-refines-its-digital-switchover-programme-for-the-uks-full-fibre-future/>, Accessed: 2025-02-23.
- [168] *Qualfiber's xgspon xgpon olt 32 ports*, <https://www.qualfiber.com/xgspon-xgpon-olt-32-ports-product/index.htm>, Accessed: 2025-02-23.
- [169] *Arista's 7260x3 series*, https://www.arista.com/assets/data/pdf/Datasheets/7260X3_Datasheet.pdf, Accessed: 2025-02-23.
- [170] *Datacentre solutions news: Cutting power consumption by 33% and doubling core capacity*, <https://datacentre.solutions/news/69098/cutting-power-consumption-by-33-and-doubling-core-capacity>, Accessed: 2025-02-23.
- [171] *Aps-networks: Stordis advanced programmable switch™ bf6064x-t - hardware design specification*, <https://www.opencompute.org/documents/210216-bf6460x-t-switch-specifications-v2-pdf-1>, Accessed: 2025-02-23.
- [172] *Leveraging digital infrastructure to drive sustainability and revenue*, <https://www.analysysmason.com/research/content/reports/leveraging-sustainable-networks-rma08/>, Accessed: 2025-03-21.
- [173] *How infrastructure is driving our digital world*, <https://www.allianzgi.com/en/insights/outlook-and-commentary/how-infrastructure-is-driving-our-digital-world>, Accessed: 2025-03-21.
- [174] *Beyond 5g - optical network continuum, deliverable d4.1, infrastructure control and service management architecture, initial requirements, and existing frameworks*, https://www.b5g-open.eu/wp-content/uploads/Deliverables/B5G-OPEN-D4_1.pdf, Accessed: 2025-03-21.
- [175] B. Niu, J. Kong, S. Tang, Y. Li, and Z. Zhu, "Visualize your ip-over-optical network in realtime: A p4-based flexible multilayer in-band network telemetry (ml-int) system," *IEEE Access*, vol. 7, pp. 82 413–82 423, 2019. DOI: 10.1109/ACCESS.2019.2924332.