



SEASON

Self-Managed Sustainable High-Capacity Optical Networks

This project is supported by the SNS Joint Undertaken through the European Union's Horizon RIA research and innovation programme under grant agreement No. 101096120

Deliverable D5.3

Final report of demo 2

Editor Stefano Tennina, Andrea Marotta (WEST)

Contributors CNIT, CTTC, ADTRAN, TID, HHI, TIM, SSSA, UPC, INF-G, WEST, ACC, WIN, ERI

Version 2.0

Date February 26, 2026

Distribution PU

DISCLAIMER

This document contains information which is proprietary to the SEASON consortium members that is subject to the rights and obligations and to the terms and conditions applicable to the Grant Agreement number 101096120. The action of the SEASON consortium members is funded by the European Commission.

Neither this document nor the information contained herein shall be used, copied, duplicated, reproduced, modified, or communicated by any means to any third party, in whole or in parts, except with prior written consent of the SEASON consortium members. In such case, an acknowledgement of the authors of the document and all applicable portions of the copyright notice must be clearly referenced. In the event of infringement, the consortium members reserve the right to take any legal action it deems appropriate.

This document reflects only the authors' view and does not necessarily reflect the view of the European Commission. Neither the SEASON consortium members as a whole, nor a certain SEASON consortium member warrant that the information contained in this document is suitable for use, nor that the use of the information is accurate or free from risk, and accepts no liability for loss or damage suffered by any person using this information.

The information in this document is provided as is and no guarantee or warranty is given that the information is fit for any particular purpose. The user thereof uses the information at its sole risk and liability.

REVISION HISTORY

Revision	Date	Responsible	Comment
1.0	October, 06, 2025	A. Marotta	Initial ToC version.
1.1	October, 31, 2025	A. Marotta	First set of contributions.
1.2	November, 30, 2025	A. Marotta	Final contributions.
1.3	December, 04, 2025	A. Marotta	Start quality review.
1.4	January, 22, 2025	C. Pinho	Quality check.
2.0	February, 26, 2025	A. Marotta	Final version for submission

LIST OF AUTHORS

Partner	Name Surname
Accelleran	Stephen Parker
Adtran (ADVA)	Vignesh Karunakaran, Nikhil Dsilva, Lutz Rapp, Dominic Schneider, Achim Autenrieth
CNIT	Andrea Sgambelluri, Filippo Cugini
CTTC	Ramon Casellas, Ricardo Martinez, Laia Nadal, Luca Vettori, J. Vilchez, Carlos Hernández-Chulde.
FIB	Anna Chiado' Piat, Roberto Mercinelli, Roberto Morro, Massimo Nilo, Annachiara Pagano, Emilio Riccardi, Maurizio Valvo
HHI	Abdelrahmane Moawad
TID	Pablo Armingol, Oscar Gonzalez
UPC	Luis Velasco, Marc Ruiz, Jaume Comellas, Salvatore Spadaro, Fernando Agraz, Joan Gené, Albert Pagès
WEST	Stefano Tennina, Andrea Marotta, Carlo Centofanti
WIN	Vasilis Tsekenis, Sokratis Barmponakis, Panagiotis Demestichas
SSSA	Nicola Sambo

EXECUTIVE SUMMARY

Deliverable D5.3 presents the final validation results of the SEASON project through two major contributions: (i) a consolidated assessment of all project-wide Key Performance Indicators (KPIs); and (ii) the complete evaluation of Demo 2 (L'Aquila), focused on end-to-end orchestration across access, metro, and radio domains.

The KPI assessment provides a project-level view of SEASON's technical and operational achievements, drawing on experimental measurements, simulation results, and demonstrations spanning all work packages. It evaluates progress in areas such as capacity scaling, energy efficiency, reliability, self-management, service creation time, and AI/ML-driven automation.

Demo 2 validates SEASON's real-time orchestration capabilities using an immersive eXtended Reality (XR) workload and a fully integrated platform combining SDM-PON access, programmable packet-optical midhaul transport, and O-RAN mobile connectivity over a multi-core fiber deployment. The system demonstrates dynamic resource activation, including optical lane provisioning, midhaul path configuration, and DU/RU scaling, while maintaining service continuity.

Three additional demo extensions complement the core field trial, i.e., Extension 1: cloud-native deployment of the XR backend on a Kubernetes environment; Extension 2: unified orchestration of legacy PON and coherent P2MP technologies; and Extension 3: intent-based provisioning of coherent P2MP services.

Furthermore, an appendix is provided to document the supplementary activities enabled by reallocated project funds, including 5G infrastructure deployment, AI/ML dataset development, 6G scheduling research, and extended techno-economic studies.

Collectively, the KPI assessment and the demonstration outcomes confirm SEASON's operational readiness and its capacity to deliver a sustainable, high-capacity optical-mobile infrastructure consistent with the project's strategic objectives.

TABLE OF CONTENTS

1	Introduction.....	6
2	SEASON KPIs	8
3	Demo 2: Field Trial of Spatial PON and P2MP Metro Convergence with O-RAN for AR/VR Services with Edge Offload	25
3.1	Use Case Description.....	25
3.2	Demo components.....	26
3.2.1	PON Controller.....	26
3.2.2	Telemetry Controller / Infrastructure Agent.....	27
3.2.3	Metro Controller.....	27
3.2.4	Network Service Orchestrator	29
3.2.5	XR Service	30
3.3	Demo Workflow	31
3.4	Results.....	33
4	Demo 2 extension 1: Validation of Season XR Service Backend Deployment	36
4.1	Use Case Description.....	36
4.2	Components.....	37
4.3	Workflow.....	38
4.4	Results.....	39
5	Demo 2 Extension 2: Dynamic Control of Multi-Technology Optical Access Networks with PON and Coherent P2MP	41
5.1	Use Case Description.....	41
5.2	Components.....	42
5.3	Workflow.....	44
5.4	Results.....	45
6	Demo 2 Extension 3: Dynamic Service Provisioning in IPoDWM scenarios with P2MP pluggables.....	46
6.1	Use Case Description.....	46
6.2	Components.....	47
6.3	Workflow.....	48
6.4	Results.....	51
7	Conclusion	52
8	Appendix 1.....	54
8.1	Dedicated Open 5G SNPN for SEASON demo at HHI.....	54
8.2	Measurement Framework (New data set creation) and Operational Economics of AI/ML-enabled Orchestration	56

8.2.1	Methodology overview	57
8.2.2	KPI identification	57
8.2.3	Additional techno-economic assessment of AI/ML-enabled Service and Application-driven Orchestration.....	59
8.3	Scheduling of mixed 6G traffic in TDM-based horse-shoe optical metro-access with burst-mode transceivers	63
8.3.1	Background on 6G radio and scheduling approach	64
8.3.2	Proposed ILP & Heuristic Scheduling Algorithms	70
8.4	Contribution to the techno-economic assessment Errore. Il segnalibro non è definito.	
8.5	Additional Techno-Economic Analysis for Advanced Mobile Transport Architectures	84
8.5.1	Cost Model Parametrization for Transport Solutions	85
8.5.2	Impact of Future 400G Fronthaul Requirements on Cost Analysis	88
8.5.3	Assessment of Single-Fiber Ring Topology.....	91
8.5.4	Study and Modeling of the New SEASON Railway Geotype	93
8.6	On the exploitation of segmented Mach-Zehnder modulators in the front/mid-haul and access	95
	Glossary	101
	References.....	104

1 INTRODUCTION

This deliverable consolidates two key outcomes of the SEASON project: the comprehensive assessment of project-wide Key Performance Indicators (KPIs) and the final validation results of Demo 2. This deliverable complements D5.2, which focuses exclusively on Demo 1, together providing comprehensive experimental validation of the SEASON architecture. It includes a dedicated KPI section that evaluates progress across all technical and operational targets defined in the project proposal. Section 2 therefore serves as a primary reference for assessing SEASON's performance against objectives related to network capacity scaling, energy efficiency, reliability, self-management, service provisioning time, and AI/ML-driven automation. The assessment integrates experimental measurements, simulation results, and demonstration outputs generated across WP2, WP3, WP4, and WP5, providing a consolidated view of project's overall accomplishments.

Demo 2 complements Demo 1 by shifting the validation focus from backbone-level automation and failure recovery to user-facing service delivery and real-time resource adaptation. While Demo 1 addresses network resilience and MBoSDM switching at the operator infrastructure layer, Demo 2 targets the end-to-end orchestration chain from access to edge, using XR traffic as a demanding reference workload. The demonstration integrates SDM-enabled PON access, programmable packet-optical metro transport, and an O-RAN radio access network deployed over a real multi-core fiber ring. Together, the two demonstrations provide comprehensive coverage of the SEASON architecture across all network segments.

The XR-based workflow used in the field trial serves as a realistic stress case for evaluating end-to-end responsiveness. Through continuous telemetry and distributed decision logic, SEASON dynamically activates optical lanes, provisions midhaul connectivity, and scales DU/RU resources in response to application load. This provides concrete evidence of the system's capability to execute coordinated adaptations across optical, packet, and radio domains—directly supporting the KPI validation reported in Section 2.

The deliverable further includes three demo extensions that complement the main experiment and broaden the scope of the validation. Extension 1 validates Kubernetes-based deployment of the XR backend, confirming SEASON's cloud-native orchestration capabilities. Extension 2 demonstrates coexistence between legacy PON systems and coherent P2MP channels under unified control, addressing realistic brownfield scenarios. Extension 3 validates automated, intent-driven provisioning of coherent point-to-multipoint services, illustrating the practical benefits of SEASON's hierarchical SDN architecture for service lifecycle management.

The remainder of this document is organised as follows. Section 2 presents the SEASON KPI assessment. Section 3 describes the core Demo 2 field trial, including the use case, demo components, workflow, and results. Sections 4, 5, and 6 report on Demo Extensions 1, 2, and 3, respectively, each following a consistent structure covering use case description, components,

workflow, and KPI validation. Section 7 presents the conclusions of the deliverable. A glossary and list of references complete the document.

Finally, Appendix 1 (Section 8) documents additional activities carried out through the reallocation of available project funds, extending the project's impact beyond its original scope.

2 SEASON KPIs

Obj. 1 (O1): Design a sustainable network architecture able to scale up network capacity and cope with challenging user needs through improved power efficiency, reliability, and self-management capabilities

Description: Design a comprehensive end-to-end network solution that scales network capacity thanks to truly self-managed operations applied to innovative MBoSDM technology and specifically designed subsystems and solutions (e.g., multi granular node, power-efficient transmission, and DSP). The MBoSDM technology will be applied from access and front/mid-haul through metro-aggregation up to the cloud, minimizing O/E/O conversions thanks to power-efficient functional split implementations as well as relying on new integrated computing-networking nodes equipped with smart coherent pluggable modules. A pervasive monitoring infrastructure will guarantee network security against data leakage as well as the maximization of resource usage. At the control level, truly self-management capabilities will be achieved thanks to a control service orchestration system for the SEASON MBoSDM technology, able to manage the lifecycle of pluggables, network elements, and integrated computing/packet/optical systems, and services. The control system will support AI/ML-based closed-loop operation, and an innovative NetDevOps paradigm for continuous development/integration.

KPI 1: This overall objective is addressed through all subsequent specific project objectives and KPIs, and it is validated in two comprehensive demonstrations focusing on both infrastructures and user needs.

Obj.1 (O1) Project results

All subsequent specific project objectives and KPIs have been successfully achieved. In this document, objectives and KPIs related to experimental validation are reported. For objectives and KPIs related to techno-economic validations, please refer to D2.3 [D2.3-25].

Obj. 2 (O2): Design and validate a scalable, ultra-high capacity, and power efficient Multi-Band over Space Division Multiplexing (MBoSDM) network infrastructure from access to cloud

Description: Design an innovative MBoSDM architecture able to provide x120 capacity increase compared to current C-band solutions. MBoSDM maximizes the cost-effective usage of available and new (including multi-core) fiber resources. The MBoSDM will be investigated in the context of both metro/aggregation and front/mid-haul, aiming at supporting densification of cells and cell-free solutions from a transport/capacity perspective.

KPI 2.1: increase the available bandwidth of the fiber from actual C-band (~35 nm) to O, E, S, L, U bands (~415 nm) that, together with the usage of SDM, e.g., with >10 fibers / cores, will make the available bandwidth grow by a factor x120. [SRIA, mid-term evo ~2028]. **KPI 2.2:** 50% CAPEX reduction by (1) designing an architecture that jointly leverages on parallel fibers (where fiber resources are abundant), multiple bands (where fiber resources are scarce), and multi-core fibers (where fibers are not present, e.g., for cell densification); (2) limiting intermediate aggregation in routers thanks to the ultra-high capacity of MBoSDM and by exploiting smart coherent pluggable

to remove aggregation layers and unnecessary O/E/O conversions. **KPI 2.3:** Network connectivity service with creation time < 3 min combining control and data planes. In [Sha21], 3 mins were needed for network connectivity in the metro segment only, mainly due to laser configuration. In SEASON, connectivity will be extended to cover end-to-end, including front-haul, PON and metro/core.

Obj.2 (O2) Project results

All KPIs of this Objective have been successfully achieved.

KPI 2.1

Achieved. Reported on D2.3 [D2.3-25].

KPI 2.2

Achieved. Reported on D2.3 [D2.3-25].

KPI 2.3

Achieved. The KPI is justified by the results demonstrated at HHI Demo. During the demonstration, the end-to-end network connectivity was provisioned with a setup time consistently below 3 minutes. This includes all control plane signaling and data plane configuration, such as optical laser tuning and multi-domain device orchestration.

The measurement data from the demo shows average provisioning times around 1 minute and 30 seconds, reflecting significant reduction compared to previous state-of-the-art times limited to metro segments. The results confirm that the integrated automation and orchestration tools used can reliably and repeatedly meet the KPI target of sub-3-minute network connectivity setup across multiple network domains. This validates the KPI as a realistic and achievable metric based on concrete empirical evidence. Details reported on D5.2 [D5.2-25].

Obj. 3 (O3): Design and development of novel optical systems and subsystems for multiband over SDM

Description: Design and development of different switching and transmission systems and subsystems which will be validated in the integrated SEASON demonstrators. Regarding switching, it entails the study of architectural options for MBoSDM multi-granular switching nodes with the following design goals: i) extension of the operating spectral window to include additional spectral bands beyond C+L; ii) enable fiber/core switching for SDM (i.e., switching the entire spectrum range among fibers/cores); iii) enable band switching (i.e., switching one entire band among fibers); iv) energy efficiency; v) cost efficiency. In terms of transmission, two different paths will be followed: i) upgrade a scalable and modular S-BVT (e.g., employing multicarrier modulation) capable to establish high-capacity point-to-point (P2P) and P2MP connections in a MBoSDM environment; ii) the optimization of the DSP algorithms to enable up to 4x data-rate increase, while keeping power consumption low enough and including novel functionalities focused on power management (e.g., dynamic power scaling, sleeping mode/spectrum).

KPI 3.1: Design and implement flexible and modular MBoSDM node prototypes able to switch/add/drop channels in at least 3 different bands (e.g., S, C, L) in an SDM/MCF fiber infrastructure featuring up to 10 fibers/cores, able to cope with switching capacities scalable up to between 2.4-3.6 Pb/s (considering a 4-degree node with 50% local add/drop and depending on the number of used bands and SDM cores/fibers) [SRIA, mid-term evo ~2028], by approaching (fractional) space-wavelength flexible architectures [Mar15]. **KPI 3.2:** MBoSDM transceivers able to increase the capacity of SoA transceivers [Nad22] up to 2× - 4× by exploiting enhanced wavelength/space dimensions while enabling appropriate slice/band/core/fiber selection according to the network path. **KPI 3.3:** Optimized DSP for metro/core coherent applications, able to increase 4× data-rate, reaching up to 1.6Tb/s per port with power consumption suitable for future pluggable modules.

All KPIs of this Objective have been successfully achieved.

KPI 3.1

Achieved. Details reported in D3.3 [D3.3-25]. Related techno-economic studies in D2.3 [D2.3-25].

In deliverable D3.2 [D3.2-24], an initial discussion was included regarding the achievement of this KPI. In deliverable D3.3 [D3.3-25], more details can be found related to the validation of the KPI. Specifically, different node architectures were presented and evaluated for high-capacity MBoSDM optical networks. These advanced node architectures, based on spatial optical cross-connects (S-OXCs) or multi-band wavelength-selective switches (MB-WSS), provide enhanced switching capabilities and allow trade-offs between flexibility/performance and cost/complexity. While these architectures are not yet commercially available, they can be implemented using mature components such as band-pass filters (BPF), fan-in/fan-out devices, OXCs, and WSS operating in single or dual bands (C+L). Each degree of these switches can handle several hundred Tb/s, and nodes with degrees greater than four can easily achieve overall switching capacities in the Pb/s range. In deliverable [D2.3], node traffic capacities were analyzed considering the reference backbone network topology proposed in section 5.2.2 of SEASON deliverable D2.1 [D2.1-24].

To support the development of a scalable infrastructure meeting these high switching capacities, SEASON has developed and implemented two prototypes (C3.1 and C3.2, within [D3.2] and [D3.3]), enabling both spatial (core, fiber) and spectral (wavelength/band) switching. These prototypes are designed to switch, add, and drop channels across at least three different bands (S+C+L) within an SDM infrastructure, closely aligned with the KPI requirement. The first node prototype C3.1 offers wavelength granularity within the C+L-bands and supports band and spatial switching. This makes it particularly suitable for the metro-aggregation segment, where finer granularities and channel flexibility are required. The second prototype exploits whole-band and core switching, adopting a more coarse-grained approach to accommodate higher traffic demands, and is therefore better suited for exploration within the core segment.

Successful transmission of three slices of the MB(oSDM) S-BVT prototype at 41.3 Gb/s (C-band), 45.4 Gb/s (L-band), and 34 Gb/s (S-band) over 25 km of MCF, has been demonstrated in section 4.1 of D3.3. This multiband flow is switched using the three-degree SEASON node prototype (C3.1). Leveraging the full S+C+L spectral range over a 19-core MCF can significantly boost total capacity. Under this setup, 350 channels at 25 GHz (S-band), 175 channels at 25 GHz (C-band), and 150 channels at 50 GHz (L-band) envision a total capacity of **28.2 Tb/s**. When extending to the spatial dimension of the 19-core MCF, **0.5 Pb/s** can be envisioned. For higher-degree node configurations, target KPI switching capacities between **2.4 Pb/s and 3.6 Pb/s** would be enabled.

KPI 3.2

Achieved. Details reported in D3.3 [D3.3-25].

Evaluation results presented in deliverable D3.3 [D3.3-25], together with previous findings in deliverable D3.2 [D3.2-24], in section 4.1, confirm that operating across the full S+C+L band on a single core of a 19-core MCF can envision, through scalability analysis, an aggregated capacity of 28.2 Tb/s. This assumes full population of 350 channels in the S-band (SSB), 175 channels in the C-band (SSB), and 150 channels in the L-band (DSB).

For comparison, a state-of-the-art (SoA) C-band S-BVT populated with 175 C-band (SSB) channels would achieve approximately 7 Tb/s. Therefore, adopting a multiband (S+C+L) approach enables a theoretical capacity increase by a factor of up to **4x**, which can be further scaled by exploiting the spatial dimension of MCFs.

However, expanding transmission across multiple cores and multiple bands introduces new challenges, such as stimulated Raman scattering (SRS) and crosstalk (XT), which may impact the actual achievable capacity and scalability factor, as discussed in D3.3. These considerations highlight the need for advanced mitigation strategies to fully realize the benefits of multiband and spatial multiplexing in future high-capacity optical networks.

KPI 3.3

Achieved. We optimized DSP techniques for metro and core coherent systems, efficiently supporting 240 GBd per port, thus meeting the requirements of upcoming 1600ZR pluggable standards. We specifically focused on realizing low-power DSP architectures by adopting frequency-domain (FD) filtering for group-velocity dispersion (GVD) compensation—one of the most computationally demanding DSP functions. We implemented an FD-based GVD compensator using FFT/IFFT processing, enabling efficient dispersion equalization in the frequency domain.

As symbol rates scale to 120 GBd and beyond, DSP complexity and power demand grow rapidly. To address this, we conducted a detailed complexity analysis—published in an accepted PTL paper—showing that our optimized architecture achieves a 12% reduction in non-trivial complex multiplications at 120 GBd and up to 18% at 240 GBd. We translated this computational saving directly into DSP power efficiency: because complex multiplications dominate the energy consumption of FFT-based architectures, the reduction yields meaningful power-budget

improvements. Overall, we demonstrated, quantified, and validated that our FD filtering approach provides tangible DSP power savings, establishing a key KPI for next-generation coherent pluggable designs and supporting the feasibility of compact 1600ZR-class optical modules. Details are reported in D3.2 [D3.2-24]. and D3.3 [D3.3-25].

Obj. 4 (O4): Develop an innovative access and front/mid-haul transport solution supporting power-efficient functional split implementations as well as cost-effective small/free cells solutions

Description: Study and develop power-efficient solutions for high capacity and high density 5G+ access. This includes assessment and comparisons among: (1) SDM-PON, (2) low-latency TDM transmission, (3) coherent P2P and P2MP, e.g., with digital subcarrier multiplexing. The SEASON solution will then study i) SDM-PONs and related control mechanisms for optical access networks providing high energy efficiency via activation and deactivation of spatial channels, ONUs and OLTs based on traffic conditions; ii) integration between optical transport and O-RANs to coordinate resource reservation at the optical access and management of the radio access network, e.g., user-cell association; iv) power efficient implementation of functional splits closer to the cell sites; v) fronthaul transport and associated time & frequency synchronization research advances to support user-centric cell-free.

KPI 4.1: <1ms mobile user latency via coordinated resource allocation at optical access and mobile network for SDM-PON mid-hauled RU/DU as a result of SEASON's target integration against >5ms delay in non-integrated approach [Li18]. **KPI 4.2:** >50% contribution in energy saving via dynamic spatial channels aggregation and deactivation of unused transceivers at the OLT side basing on traffic conditions over total 70% energy saving targeted by [SRIA]. **KPI 4.3:** 400Gb/s RAN fronthaul ports capacity. **KPI 4.4:** High accuracy profile for IEEE 1588-2019 or better, evolving $\pm 1.5 \mu\text{s}$ of LTE and Release-15 Standalone, aiming to ns with target IEEE P802.1CM A+ networks, demanding an accuracy better than 12.5 ns.

All KPIs of this Objective have been successfully achieved.

KPI 4.1

Achieved. The experimental activity carried out in SEASON demonstrates that the integrated optical–mobile control plane can significantly reduce the end-to-end user latency when the optical access segment supports software-driven spatial resource allocation. Measurements on the Spatial PON infrastructure confirm that the *expedited-forwarding traffic class consistently achieves sub-millisecond latency* across the PON segment. In particular, the latency remains well below 1 ms across all measurement points, with values in the order of 0.15 ms depending on the offered load. This behaviour is enabled by the combination of two SEASON innovations: the software control of spatial lane activation in the SDM-PON and the mapping of latency-critical flows onto the expedited-forwarding profile. Together, these mechanisms minimise queueing effects in the optical access and ensure deterministic forwarding even under varying traffic conditions.

When compared to the best-effort profile, which exhibits latency values approaching 1 ms under increasing load, the benefit of the SEASON integrated approach becomes clear. The ability to dynamically allocate additional spatial lanes and reserve optical forwarding resources for critical mobile traffic avoids congestion build-up, allowing the PON segment to sustain latency levels that are significantly lower than those reported in non-integrated architectures. It is important to note that, despite the optical access segment meeting the <1 ms requirement, current commercial and pre-commercial O-RAN systems do not achieve end-to-end mobile user latency below 1 ms, due to inherent processing delays in RU, DU and scheduling functions. As such, the KPI must be interpreted within the context of SEASON's optical-centric scope: the project demonstrates that the optical portion of the mobile access chain—when enhanced with spatial multiplexing and real-time SDN control—no longer represents a bottleneck for ultra-low-latency communication. This is a fundamental prerequisite for future mobile systems, where radio and edge processing technologies will progressively reduce their internal delays.

KPI 4.2

Achieved. Reported on D2.3 [D2.3-25].

KPI 4.3

Achieved. The proposed fronthaul architecture interconnects vDUs and RRUs via packet switches and an optical line system (OLS) equipped with 400G QSFP-DD coherent transceivers. It employs a bidirectional fiber ring with a logical mesh topology to ensure high throughput and flexible wavelength management. The OLS adopts a filterless design using optical add-drop multiplexers (OADMs) with simple splitters and pluggable erbium-doped fiber amplifiers (EDFAs). Two configurations are considered: one with a single 2:2 coupler for add/drop and another with separate splitters, the latter reducing attenuation and improving signal quality. Amplification strategies include automatic gain and power control, with single- and dual-amplifier modules delivering different output powers. High-power transceivers significantly enhance optical signal-to-noise ratio (OSNR) and link budgets. The drop-and-continue ring topology requires $N(N-1)/2$ wavelengths for N nodes, supporting up to six nodes at 400G and about ten nodes at 200G. Stronger amplifiers and high-power transceivers further extend reach and performance. Future upgrades under evaluation include variable gain amplifiers and gain-flattening filters for dynamic control and improved stability. Overall, the design offers a flexible, scalable solution for next-generation fronthaul transport with robust signal quality. Details reported in D2.3 [D2.3-25].

KPI 4.4

Achieved. To fulfil the KPI requirements, Accelleran integrated a Telecom Grandmaster (T-GM) clock within the CU server, equipped with an Intel E810 NIC. The T-GM is synchronised using a GNSS satellite timing reference, and the resulting timing information is distributed over the IP network. This architecture enables the O-DUs and O-RUs to achieve accurate time and phase synchronisation with the T-GM via the PTPv2 (IEEE 1588-2019) protocol, through which timing

signals are disseminated from the T-GM to the Telecom Time Slave Clocks (T-TSCs) implemented in the O-DU and O-RU.

Across the SEASON demonstration sites, this solution achieved timing accuracy between -10.1012 ns and 11.1012 ns with a confidence level of 99.999%, thereby exceeding the KPI requirement of 12.5 ns.

Obj. 5 (O5): Develop a pervasive monitoring infrastructure for secure and truly self-managed networking

Description: Study and develop a comprehensive monitoring infrastructure of the whole end-to-end packet-optical transport network for continuous optical performance monitoring relying on streaming telemetry. In addition to characterizing the end-to-end performance characteristics of a lightpath, the approach is also designed to identify parts of the network causing strong degradation of signal quality. The monitoring infrastructure will leverage on several key innovations for: i) estimating fiber link characteristics from powerful signal processing algorithms performed by DSPs in the receivers; ii) signal identification and characterization by modulation format independent monitoring devices located in the network, iii) adaptive methods for data sampling, compressing, reporting and further processing by AI techniques; iv) optimized interrogators for characterizing individual fiber sections and v) improved amplifier control and link monitoring by embedded OSAs.

KPI 5.1: Achieve sub-km (<500 m) and sub-dB (<0.5) resolution in the estimation of longitudinal fiber attenuation points and optical amplifier gain, respectively, using DSP-based monitoring scheme.

KPI 5.2: Performance improvement achievable with an OSA embedded in the amplifier setup and control identified for different link designs and applications. **KPI 5.3:** OTDR Interrogator for latency / position measurement with 4 ns / < 1 meter accuracy respectively. **KPI 5.4:** Applicability of modulation format insensitive OSNR measurement techniques in different scenarios determined, sources of inaccuracy identified, impact of signal distortions worked out.

All KPIs of this Objective have been successfully achieved.

KPI 5.1

Achieved. The SEASON project focuses on advancing optical network monitoring by significantly improving the resolution and accuracy of longitudinal power-profile estimation using DSP-based methods. In this part, we applied a Linear Least Squares (LLS)–based approach that provides enhanced precision in detecting and localizing anomalies along the fiber link. The technique, originally developed in [C3.5], delivers **sub-kilometer (<1 km)** spatial resolution and **sub-dB (~0.46 dB)** accuracy in identifying attenuation events. A comprehensive description of this development has been published in a JOCN journal article and is summarized in Section 6.4 of this deliverable, highlighting its relevance for real-time diagnostics and improved visibility in modern optical networks.

KPI 5.2

Achieved. Dynamic gain equalization in EDFAs via adaptive gain-flattening and per-amplifier spectral monitoring can significantly enhance link performance [Rap03], but high component

costs made it uneconomical. A reconfiguration-based approach delivered similar noise figure at lower cost. With recent price reductions in wavelength-configurable devices and advances in photonic integration, spectral reconfigurability has regained relevance—especially for multi-band systems using TDFAs in the S-band. In total, NF improvements of around 0.5 dB at lower input powers and around 1.0 dB at high input powers turned out to be achievable. Furthermore, the premium is expected to amount to a few hundreds of dollars per amplifier for large volume production. A simplified TDFA simulation model was developed to assess this potential. A full rate-equation-based model was beyond project scope but will be validated later using results from an ongoing project. More details are reported in deliverable D3.3 [D3.3-25].

KPI 5.3

Achieved. The Digital Optical Time Domain Reflectometer (D-OTDR) enables precise latency and position measurements in live fiber networks, meeting KPI 5.3 requirements. It achieves 4-ns latency resolution and sub-meter spatial accuracy using Golay-coded sequences and correlation-based processing on reserved wavelengths. The architecture integrates FPGA and SFP components for continuous, in-service diagnostics without traffic disruption. Details in Appendix 1 (Section 8.4) of the document.

KPI 5.4

Achieved. The applicability of modulation-format-insensitive optical signal-to-noise ratio (OSNR) measurement techniques has been assessed across various scenarios with different impact of nonlinear fiber effects. Sources of inaccuracy and distortion impacts have been identified. Spectral correlation proved promising for signal quality monitoring in agile, disaggregated networks [Rap21], capturing major fiber impairments. However, sensitivity varied by impairment. Therefore, the work done in SEASON focused on better aligning the technique for providing a meaningful and universally applicable generalized signal-to-noise ratio (GSNR) value. This has been partly achieved by halving the impact of nonlinear fiber effects on the measurements results. More details are reported in deliverable D3.3.

Obj. 6 (O6): Provide and validate smart edge nodes for packet/optical integration with computing resources

Description: Design and develop a smart edge node based on latest generation Data Processing Units (DPU) enhanced with coherent intelligent pluggable modules for: i) supporting integrated computing and optical networking solutions in a single element, minimizing the number of active electronic nodes; ii) supporting selected ultra-low latency services at the edge with direct S-BVT and P2MP optical communication to 5G+ access; iii) providing fully transparent power-efficient edge to edge/cloud communication with >50% reduction of intermediate O/E/O; iv) supporting selected 5G functions closer to cell-sites for >50% reduced O/E/O.

KPI 6.1: 40% CAPEX reduction by collapsing computing, IP networking, and usage of high-speed coherent optical transmission in a single element (i.e., DPU) not designed for the Telecom market but for much wider computing markets and verticals (e.g., automotive). **KPI 6.2:** >40% reduction of O/E/O conversions in edge-edge and edge-cloud communications by developing smart edges with

high-speed coherent intelligent pluggables and by moving 5G functions closer to the cell sites. [SRIA, short/mid-term evo]. **KPI 6.3:** Supporting traffic adaptation/monitoring at μ s granularity on innovative HW-accelerated networking smartNICs/DPUs (including computing, networking and optical resources) for selected low-latency services.

KPI 6.1

Achieved. Reported in D2.3 [D2.3-25].

KPI 6.2

Achieved. Reported in D2.3 [D2.3-25].

KPI 6.3

Achieved. Reported in D4.2 [D4.2-24] and D4.3 [D4.3-25] SEASON developed a fully integrated hardware-accelerated monitoring and adaptation architecture built on next-generation smartNIC/DPU (Data Processing Unit) that combine embedded computing, high-speed packet processing, and coherent optical interfaces within a single data-plane element. The implementation uses NVIDIA BlueField-2 DPUs with an embedded GPU paired with 400ZR+ coherent pluggable transceivers, allowing the DPU to ingest and process optical telemetry while simultaneously executing line-rate packet inspection. In this architecture, DOCA Flow, hardware-offloaded DPI, stateful flow tracking and a RegEx accelerator enable per-packet feature extraction at wire speed, ensuring that optical metrics such as BER and received power, as well as traffic and security descriptors, are parsed within the microsecond domain before software or host CPU involvement. Optical telemetry from the coherent modules is consumed directly by the DPU using REST interfaces and then injected into the data-plane processing pipeline, enabling a unified, tightly coupled monitoring framework across optical and packet layers. SEASON further developed a hierarchical inference pipeline in which DPU hardware offload performs microsecond-scale packet parsing and telemetry extraction, while the embedded GPU executes lightweight Convolutional Neural Network (CNN) models for optical degradation prediction and DDoS classification. The transfer of DPI-extracted features to the GPU is optimized through zero-copy or minimal-overhead memory operations, producing end-to-end inference times well below a millisecond. Experimental results demonstrate that optical quality of transmission prediction achieves an average inference latency of approximately 0.5 ms when DPI is enabled—substantially faster than the 2.5 ms measured without hardware offload—while DDoS analysis completes in approximately 0.74–0.84 ms versus >2.3 ms without DPI support. Because packet parsing and threshold checks remain confined to the hardware pipeline, the monitoring loop operates with deterministic timing properties, allowing the system to react to emerging anomalies at microsecond and sub-millisecond timescales. SEASON also implemented real-time adaptive mechanisms tightly integrated with this fast monitoring layer. For optical transport, algorithms developed within the project use real-time BER and power measurements, extracted through DPI, to perform early anomaly detection and remaining-useful-life estimation on the GPU. When degradations approach critical thresholds, the DPU

initiates proactive rerouting actions toward alternative optical paths, achieving control-plane actuation before service-impacting link failure occurs. On the packet side, SEASON integrated DPU-resident DDoS mitigation logic that combines hardware DPI, traffic-rate counters, RegEx signature matching and AI-driven classification. Upon detecting malicious or abnormal traffic patterns, the DPU enforces mitigation actions (packet drops, rate limiting or steering) directly in hardware, ensuring reaction latencies on the order of microseconds and preventing harmful flows from affecting delay-sensitive services.

SEASON validated this architecture in a realistic multi-span optical testbed composed of two GPU-augmented DPUs interconnected through three 80 km fiber spans with coherent 400ZR+ transceivers. The system consistently achieved high prediction accuracy, with optical-fault CNNs reaching 99.45 % accuracy and DDoS classifiers reaching 99.41 % accuracy, while CPU utilization on the DPU remained around 26 %, demonstrating the capacity to scale multiple concurrent monitoring tasks. These results confirm that the integrated DPU+GPU+optics platform developed in SEASON supports continuous, fine-grained telemetry extraction, AI-assisted decision making and immediate data-plane adaptation with timing characteristics compatible with the stringent requirements of advanced low-latency services.

This work has been published in JOCN Journal [Cast24].

Obj. 7 (O7): Control plane, Monitoring and streaming telemetry

Description: Design, development and validation of a generalized telemetry, infrastructure and control service orchestration system for the SEASON MBoSDM infrastructure, able to deploy and manage the lifecycle of pluggables, network elements, integrated packet/optical systems, and services. To fully exploit monitoring and network telemetry while reducing bandwidth utilization, SEASON will develop intelligent methods to aggregate and compress data before distributing them. Development and validation of the DevOps paradigm augmenting the centralized SDN control plane with tools and processes from continuous development/continuous integration for zero-touch infrastructure configuration and network automation.

KPI 7.1: Intelligent data aggregation to provide data compression ratio >90% without significant information loss. **KPI 7.2:** Reduction on the average setup time of converged connectivity service by 30% compared to serialized provisioning, exploiting approaches relying on parallelism and concurrency. Network Connectivity Service (point to point across different segments (PON/backhaul) considering control plane only < 1 second (not considering hardware configuration latencies). [SRIA, mid-term evo ~2028]. **KPI 7.3:** Configuration of the MBoSDM node prototype agents for simple operations from the SDN control plane of O(100ms). **KPI 7.4:** Demonstration of CI/CD approach with < 10 minutes pipeline execution time after committing or detection of network change (new node/pluggable, LOS, ...), including device configuration modification, testing, validating, and deploying of infrastructure configuration compared to ~ 30 minutes of manual configuration via CLI or SDN intent creation, execution and testing using REST-APIs.

KPI 7.1

Achieved. Reported on D2.3 [D2.3-25].

KPI 7.2

Achieved. This report measures the time elapsed from when the network orchestrator receives an intent until it receives confirmation (OK) from all generated services. It focuses exclusively on the interaction time between the orchestrator and domain controllers. Equipment configuration time is excluded as a constant and non-optimizable factor.

1) Data Overview

Two datasets were analyzed, each with 238 samples. Setup times are measured in seconds.

- **Parallel provisioning:** Setup times where service creation and deletion processes run in parallel.
- **Serialized provisioning:** Setup times where service processes run one after another.

2) Statistical Results

Table 2-1: Network Service Time Statistical Results.

Provisioning Type	Process	Mean Time (seconds)	95% Confidence Interval (seconds)
Parallel	Creation	3,768	3.696 – 3.839
Parallel	Deletion	3,058	3.055 – 3.060
Serialized	Creation	8,027	8.011 – 8.042
Serialized	Deletion	6,798	6.783 – 6.813

3) Formulas Used for Performance Gain Calculation

$$\text{Creation time saving percentage: } saving_{creation} = 1 - \frac{\text{Mean Creation Time (parallel)}}{\text{Mean Creation Time (Serialized)}}$$

$$\text{Deletion time saving percentage: } saving_{creation} = 1 - \frac{\text{Mean Creation Time (parallel)}}{\text{Mean Creation Time (Serialized)}}$$

$$\text{Average total saving percentage: } saving_{creation} = 1 - \frac{\text{Mean Creation Time (parallel)}}{\text{Mean Creation Time (Serialized)}}$$

4) Performance Gains Achieved

Table 2-2: Network Service Time Performance Gains.

Metric	Value (%)
Creation Saving	53.05
Deletion Saving	55.02
Average Saving	54.04

These savings significantly exceed the 30% KPI target.

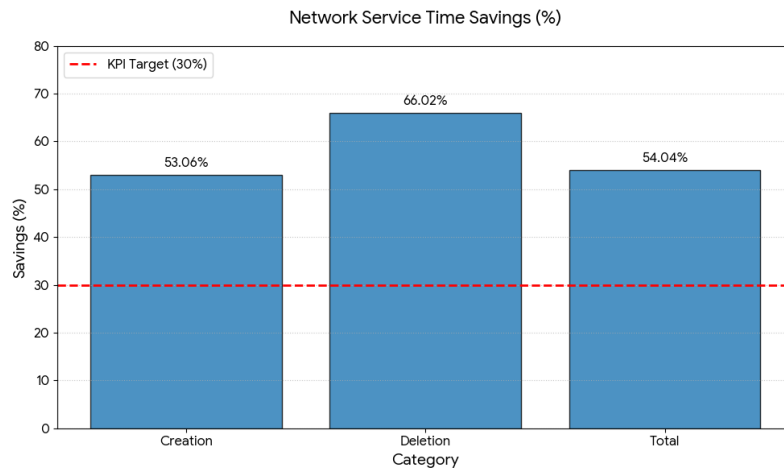


Figure 2-1: Network Service Time Savings.

KPI 7.3

Achieved. The control plane delay is defined by the time it takes to the SDN controller to send a configuration command using the NETCONF protocol and the defined Yang data models for the MBoSDM node prototype. Even in this case, it is important to define whether the NETCONF/SSH session is persistent (it is not closed after each command) or is opened and closed for each configuration. Clearly, the latency is also defined by the propagation (distance) delay between the SDN Controller and the agent as well as the number of intermediate switches/routers in the control path. Assuming a LAN-like control plane, only SDM cross-connections and multiband (e.g., L+S) cross-connections were achieved under the constraint on $O(100\text{ms})$. In particular, a latency of 75.19ms has been measured, as shown in Figure 2-2. This is also reported in D5.2 [D5.2-25]. Let us note that this KPI is referred to control plane latency only, the actual configuration latency is defined by the hardware constraints and is not considered (in some operations, due to technical limitations, the configuration of the WSS may exceed the threshold).

```

| /home/rcasellas/src/spce/src/pce/netconf/connection.hpp:458
[13:23:19.668933] 915358 [T] [NETCONF][CONNECTION]: attribute xmlns=urn:ietf:params:xml:ns:netconf:base:1.0 | /home/rcasellas/src/spce/src/pce/netconf/connection.hpp:531
[13:23:19.668941] 915358 [T] [NETCONF][CONNECTION]: attribute message-id=1 | /home/rcasellas/src/spce/src/pce/netconf/connection.hpp:531
[13:23:19.668950] 915358 [T] [NETCONF][CONNECTION]: rpc-reply OK | /home/rcasellas/src/spce/src/pce/netconf/connection.hpp:535
[13:23:19.668956] 915358 [D] [MBOSDM][PROVISIONER] completed edit_config/merge | /home/rcasellas/src/spce/src/server/plugins/season_mbosdm/provisioner.cpp:128
[13:23:19.668967] 915358 [D] [MBOSDM][PROVISIONER] closing netconf connection | /home/rcasellas/src/spce/src/server/plugins/season_mbosdm/provisioner.cpp:130
[13:23:19.668975] 915358 [D] [TX] 10.1.1.111:830 sending close-session message | /home/rcasellas/src/spce/src/pce/netconf/connection.hpp:157
[13:23:19.668987] 915358 [T] [NETCONF][CONNECTION]: sending 2 | /home/rcasellas/src/spce/src/pce/netconf/connection.hpp:190
[13:23:19.661723] 915358 [D] [SSH][CLIENT]: SSH_EOF while reading non-blocking | /home/rcasellas/src/spce/src/pce/netconf/ssh/ssh.hpp:266
[13:23:19.661744] 915358 [D] [NETCONF][CONNECTION]: 10.1.1.111:830 connection::on_message with mid [2] | /home/rcasellas/src/spce/src/pce/netconf/connection.hpp:437
[13:23:19.661799] 915358 [T] [NETCONF][CONNECTION]: message #2 removed from queue, left 0 | /home/rcasellas/src/spce/src/pce/netconf/connection.hpp:449
[13:23:19.661815] 915358 [T] NETCONF Exchange - on_message
<xml version="1.0" encoding="utf-8" standalone="yes"?>
  <rpc xmlns="urn:ietf:params:xml:ns:netconf:base:1.0" xmlns:nc="urn:ietf:params:xml:ns:netconf:base:1.0" message-id="2">
    <close-session />
  </rpc>
</xml>
<xml version="1.0"?>
  <rpc-reply xmlns="urn:ietf:params:xml:ns:netconf:base:1.0" message-id="2">
    <ok />
  </rpc-reply>
</xml>
| /home/rcasellas/src/spce/src/pce/netconf/connection.hpp:458
[13:23:19.661838] 915358 [T] [NETCONF][CONNECTION]: attribute xmlns=urn:ietf:params:xml:ns:netconf:base:1.0 | /home/rcasellas/src/spce/src/pce/netconf/connection.hpp:531
[13:23:19.661845] 915358 [T] [NETCONF][CONNECTION]: attribute message-id=2 | /home/rcasellas/src/spce/src/pce/netconf/connection.hpp:531
[13:23:19.661850] 915358 [T] [NETCONF][CONNECTION]: rpc-reply OK | /home/rcasellas/src/spce/src/pce/netconf/connection.hpp:535

```

a)

```

[2025-12-10 14:51:41.239] INFO in occ_connection.py: NETCONF RPC start
[2025-12-10 14:51:41.245] INFO in transport.py: Connected (version 2.0, client libssh_0.10.6)
[2025-12-10 14:51:41.307] INFO in transport.py: Authentication (keyboard-interactive) successful!
[2025-12-10 14:51:41.310] INFO in occ_connection.py: Configuration to be processed:
<config xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <node xmlns="urn:season:yang:mbosdm-node">
    <connections>
      <connection nc:operation='create'>
        <name=f34757da-abb1-42d3-9029-1b48180d2b9b</name>
        <input_port>2100</input_port>
        <output_port>21</output_port>
        <input_core>0</input_core>
        <output_core>0</output_core>
        <n>48</n>
        <m>8</m>
      </connection>
    </connections>
  </node>
</config>
[2025-12-10 14:51:41.314] INFO in occ_connection.py: Configuration applied successfully.
[2025-12-10 14:51:41.314] INFO in session.py: Stopping recv thread due to exception
[2025-12-10 14:51:41.314] INFO in occ_connection.py: NETCONF Request latency: 75.19 ms

```

b)

Figure 2-2: a) SDN Controller logs. Control plane latency, b) Calculation for control plane latency.

KPI 7.4

Achieved. The implemented NetDevOps pipeline successfully fulfils KPI 7.4 by executing a complete CI/CD workflow in under 10 minutes, measured from the moment the FastAPI-based proxy receives a service creation request from the optical controller. Upon receiving the request, the proxy commits the service configuration to a Git repository, triggering the pipeline immediately. The system then identifies all impacted devices and applies the necessary configuration changes. Using Ansible’s parallel execution configured through the forks parameter the pipeline efficiently deploys configurations across transponders, ROADMs, and other infrastructure components in parallel, accelerating execution time.

In parallel, the pipeline validates the current state of devices and services, capturing both infrastructure and service-level snapshots, which are stored persistently for traceability. Compared to traditional CLI-based or SDN-driven approaches that typically take around 30 minutes for configuration, execution, and validation, this NetDevOps implementation reduces that time by more than two-thirds. It also supports automated rollback: each pipeline run is versioned via Git tags, the rollback operation restores the entire network—both device configurations and service states—to a previously known good state.

Dissemination Level	PU
---------------------	----

In practice, the full-service creation process—including sending the REST request to the Adtran OLS controller, triggering the pipeline, and completing provisioning on the direct path—takes approximately 30 to 40 seconds. If an anomaly is detected on the direct path, the rerouting to an alternate indirect path is completed within approximately 90 seconds. This includes deletion of the affected service and re-provisioning along the new path. Once the original fault is resolved, the rollback to the direct path—including restoration of both device configurations and service states—is completed in approximately 50 to 60 seconds as shown in Figure 2-3.

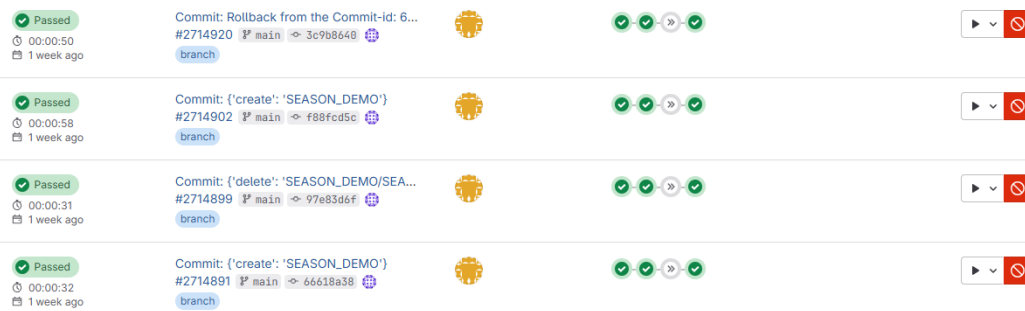


Figure 2-3: NetDevOps Pipelines for creation, deletion, and rollback of a service.

<p>Obj. 8 (O8): AI/ML Service Orchestration and Self-Management and Secure AI</p>
<p>Description: This objective is split into four related subobjectives: i) Develop AI/ML-mechanisms for Service and Infrastructure management, given the services to be offered, the infrastructure (communication, compute) capabilities, including the functional splitting options, find the best allocation of service functionality, functional splitting, and resource allocation; ii) Develop a distributed self-management control infrastructure based on multi-agent systems able to autonomously coordinate resources near real-time for end-to-end service assurance. Each node in the data plane will include specific agents for the different technologies supported in that node, e.g., radio and packet, and the services supported by that node. The different agents coordinate the local resources and exchange data, models and control with other agents; iii) Develop an AI/ML empowered optical layer Digital Twin that will include models for time and frequency domains for optical connections in the SEASON MBoSDM infrastructure; the digital twin will be used for multiple purposes, including soft-failure detection, localization, identification, and detection of anomalies; and iv) Work on attack methodologies against ML-based systems, including the digital twin, as well as on correcting their vulnerabilities and on detection/defence measures.</p>
<p>KPI 8.1: Autonomous operation based on multi-agent systems to reduce >25% OpEx w.r.t. manual/static operation. KPI 8.2: Near-real time local (inside a node) control loops, including data collection, analysis, and decision making in <10 ms, as compared to seconds for control loops involving the SDN controller. KPI 8.3: Optical layer digital twin for gradual soft-failure detection and localization with at least 1min before major impact on the service. >90% accuracy in soft-failure identification. KPI 8.4: Service creation < 90 min (such as ETSI Network Service or Micro-services application deployed with Kubernetes) [Naki20].</p>

KPI 8.1

Achieved. Reported on D2.3 [D2.3-25].

KPI 8.2

Achieved. The distributed architecture proposed in D4.3 fulfills the requirements of KPI 8.2 by achieving a near-real-time control loop duration of less than 10 ms. The solution proposes and evaluates a distributed control architecture based on Multi-Agent Systems (MAS) that assists the SDN controller in managing network services with near-real-time capabilities. Figure 2-4 illustrates the testbed used to evaluate this solution.

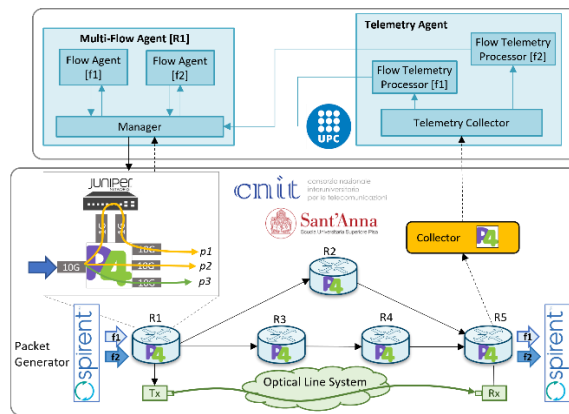


Figure 2-4: KPI Evaluation Testbed.

The control loop starts upon the arrival of a telemetry measurement at the Telemetry Agent, which runs near the egress router. In particular, the telemetry measurement contains QoS statistics for each flow path. The measurement is processed by the Telemetry Collector and conveyed to the corresponding Flow Telemetry Processor. The Flow Telemetry Processor processes, stores, and summarizes the data before sending it to the Multi-Flow Agent. The Multi-Flow Agent, which runs near the ingress router, periodically receives metrics both from the ingress router and from the Telemetry Agent. Telemetry measurements from the ingress router consist of traffic statistics, while those received from the Telemetry Agent are related to the QoS of the flow. These metrics are processed by the Manager and sent to its corresponding Flow Agent for analysis. Each Flow Agent runs a DRL algorithm to make flow routing decisions. In the presence of an anomaly, a new routing policy is generated and sent to the SDN controller to be applied to the network. The logging messages of the control loop execution and their timestamps are shown in Figure 2-5 and Figure 2-6.

```
11:06:29.682 Processing INT measurement: {'flow_id': '10', 'paths': {'p1': {'delay': {'max': 5.3, 'min': 4.0, 'avg': 4.8}, 'jitter': {'max': 1.1, 'min': 0.2, 'avg': 0.8}}, 'p2': {'delay': {'max': 10.2, 'min': 2.0, 'avg': 7.5}, 'jitter': {'max': 2.1, 'min': 0.3, 'avg': 1.1}}}}
11:06:29.683 Elapsed time for processing INT measurement: 0.016450881958007812 milliseconds
11:06:29.683 Sending processed measurement to Multi-Flow Agent
```

Figure 2-5: Telemetry Agent Times Logging.

```
11:06:29.689 Received QoS measurement: {'flow_id': '10', 'paths': {'p1': {'delay': {'max': 5.3, 'min': 4.0, 'avg': 4.8}, 'jitter': {'max': 1.1, 'min': 0.2, 'avg': 0.8}}, 'p2': {'delay': {'max': 10.2, 'min': 2.0, 'avg': 7.5}, 'jitter': {'max': 2.1, 'min': 0.3, 'avg': 1.1}}}}
11:06:29.689 Elapsed time for processing QoS measurement: 0.4429817199707031 milliseconds
11:06:29.691 Elapsed time for DRL routing policy calculation: 1.615762710571289 milliseconds
11:06:29.691 Anomaly detected, changing policy for flow 10
11:06:29.691 Setting new policy: {'flow_id': '10', 'policy': {'p1': 0, 'p2': 0, 'p3': 100}}
```

Figure 2-6: Multi-Flow Agent times logging.

The Table 2-3 summarizes the time duration each step of the control loop and the total time.

Table 2-3: Time duration of each step of the control loop and the total time.

Metric	Description	Value
Telemetry Agent INT process time	Measured time for the process, store and summarization of the INT measurements on the Telemetry Agent.	< 1 ms
MAS Agent communication delay	Added delay for the communication between the Telemetry and the Multi-Flow agent	≈ 5 ms
Multi-Flow Agent QoS process time	Measured time for the process of the QoS measurements on the Multi-Flow agent.	< 1 ms
DRL new policy calculation	Measured time for the calculation of a routing policy by the DRL algorithm on the Flow Agent	1 to 2 ms
TOTAL time for the control loop	Measured time for the execution of the control loop	≈ 9 ms

KPI 8.3

Achieved. Reported on D2.3 [D2.3-25].

KPI 8.4

Achieved. The objective of establishing service creation times under 90 minutes was validated during the HHI Final Demo (Demo 1), where the Service Orchestrator, developed by WINGS, managed the lifecycle of media-processing micro-services on a Kubernetes infrastructure.

The Service Orchestrator is implemented as a Python module coupled with a Flask and Flask-SocketIO web application. It automates the instantiation, scaling, and supervision of video streaming workloads (deploying ue-receiver as a StatefulSet with host networking). The orchestration logic ensures dependent resources are consistent and gates streamer start-up until receivers are ready, significantly streamlining the deployment process compared to manual configuration.

Experimental measurements conducted during the demo execution (D5.2 [D5.2-25], Section 2.3) demonstrated that the initial end-to-end service provisioning time was 1 minute and 30 seconds. This duration encompasses the entire workflow:

1. The instantiation of Docker containers for video streaming by the Service Orchestrator.
2. The request generation to the Network Orchestrator.

3. The subsequent provisioning of the optical and IP connectivity across the red path (between ROADM2 and ROADM3) to support the service.

This result is orders of magnitude faster than the KPI target of 90 minutes. It confirms that the SEASON orchestration framework, by leveraging Kubernetes-native scaling and automated intent translation between the Service and Network layers, enables near-real-time service creation for complex, multi-domain applications.

Obj. 9 (O9): Reinforce European industry leadership and influence major vendors and service providers to adopt SEASON principles by means of standardization, exploitation, and dissemination, activities using Open Science practices

Description: Influence the telecommunications industry on the adoption of SEASON concepts, design methodology, algorithms, and system/node/architectures by means of dissemination, standardization, and exploitation activities and maximize SEASON external reach and impact. This includes: *i)* scientific dissemination in peer-reviewed international conferences, journals, and magazines; *ii)* contributions to SDOs and Open-Source projects; *iii)* actively contributing to and shaping white papers and co-organization of technical events, workshops, and panels; and *iv)* communicate project achievements using different channels (social media, newsletters, participation to public events, and other relevant actions). Open Science practices will be followed, when disseminating and communicating the achieved objectives, including open data sets.

KPI and Means of Validation: Communication, Dissemination and Standardization activities.

Achieved. Details reported on D6.3 [D6.3-25].

3 DEMO 2: FIELD TRIAL OF SPATIAL PON AND P2MP METRO CONVERGENCE WITH O-RAN FOR AR/VR SERVICES WITH EDGE OFFLOAD

This section describes in detail the field trial led by WEST within WP5, aimed at validating the joint operation of the Spatial PON system, the metro IPoWDM transport layer, and the O-RAN radio access infrastructure under the coordinated control of the SEASON orchestration framework. The trial focuses on an XR service delivered at the network edge and accessed by multiple users, which represents a demanding use case due to its sensitivity to latency, bandwidth fluctuations, and temporary peaks of interaction load. The experiment evaluates the behavior of the integrated system when traffic-driven scaling actions are performed across the access, transport, and radio domains, assessing their impact on service continuity and on the overall energy consumption of the infrastructure.

3.1 USE CASE DESCRIPTION

The demonstration targets a multi-user XR service where immersive content is rendered at an edge server and consumed by users through Meta Quest 3 headsets connected via local Wi-Fi and the O-RAN 5G access network. The XR application follows a master–visitor model and supports the manipulation of 3D assets, real-time synchronization of object transformations, and voice interaction. These features produce a continuous stream of positional updates, texture requests, and audio samples, resulting in a traffic pattern characterized by sustained upstream and downstream flows with little tolerance for jitter or delay.

To support such traffic, the trial exploits the Spatial PON infrastructure deployed on a four-core multicore fiber ring within the city of L’Aquila. A single user typically operates within the limits of one spatial lane, while the presence of additional XR users can create temporary load conditions that exceed the available capacity. The key objective is therefore to verify whether the system can recognize these situations via telemetry and automatically activate an additional spatial lane together with a second DU/RU pair in the radio domain, and whether these changes can be executed without disrupting the ongoing XR session. The orchestrator performs the inverse sequence when the additional user leaves the session, restoring the single-lane, single-DU configuration and reducing the associated energy footprint.

The scenario reflects realistic conditions where XR services may experience rapid shifts in demand and where non-intrusive, traffic-driven adaptation is essential to maintain quality of experience without over-provisioning resources during periods of reduced utilization.

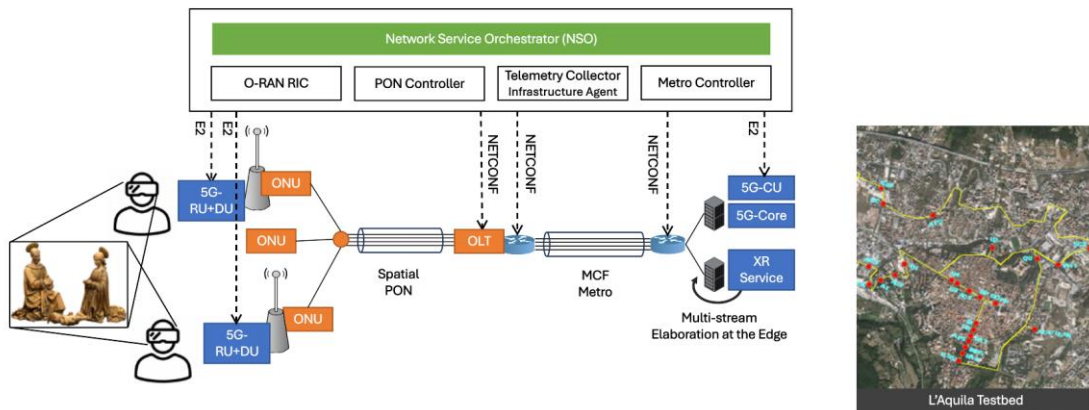


Figure 3-1: Demo Architecture.

3.2 DEMO COMPONENTS

The demonstration relies on a converged architecture combining a Spatial PON access system, an O-RAN deployment composed of two radio sectors, and a metro transport network based on IPoWDM switches equipped with coherent ZR+ pluggables. These domains are jointly controlled through the SEASON orchestration framework, which integrates the Network Service Orchestrator, the metro and PON controllers, the RAN Intelligent Controller, and the Telemetry Controller/Infrastructure Agent.

3.2.1 PON Controller

The PON Controller represents the central element for the management of the Spatial PON access network. It provides the mechanisms required to activate and deactivate spatial lanes on demand, and to propagate configuration changes across the OLT, the optical switch, and the passive splitting structure. Communication with the underlying PON elements is achieved through NETCONF, allowing the controller to adjust wavelengths, lane mappings, and splitter interconnections without service disruption.

Its northbound REST interface allows the Network Service Orchestrator (NSO) to request specific access configurations, including the addition of a new spatial lane when increased traffic is detected. The controller maintains a consistent view of the operational state of all optical components and ensures that transitions between single-lane and multi-lane configurations occur smoothly. Special attention is given to the timing required for the activation of new PON ports and for the optical stabilization of the multicore links, which constitute non-negligible intervals when coordinating multi-domain actions.

Overall, the PON Controller enables the elasticity of the access domain and ensures that additional optical capacity is made available to the RAN elements as soon as required by the XR load.

3.2.2 Telemetry Controller / Infrastructure Agent

The Telemetry Controller collects telemetry from both the transport network infrastructure and the near-Real Time Radio Intelligence Controller (near-RT RIC). This continuous stream of data helps monitor how resources are being used throughout the network. With this data, decisions can be made to optimize resources dynamically and improve energy efficiency. Working alongside the Telemetry Controller is the Infrastructure Agent, which obtains insights from the collected data and takes decisions based on the state of the network. When the telemetry data suggests that some resources are being underused, the Infrastructure Agent issues a petition to the Network Service Orchestrator to disconnect those resources in the RAN and PON. This approach helps save energy when network demand is low. On the other side, if the telemetry data shows increased demand, the Infrastructure Agent triggers the reconnection of resources to handle the traffic spikes, ensuring service quality remains stable. The system relies on Apache Kafka as the message bus to efficiently send performance metrics from the RIC. Within the RIC, a specialized xApp takes charge of handling the performance metrics. This xApp identifies which metrics are the most relevant for managing resources and optimizing the network and conveys it to the Telemetry Controller.

3.2.3 Metro Controller

The Metro controller is an-hoc extended instance of ETSI TeraFlow (TFS) SDN Controller, capable of operating on IPoWDM nodes. The contribution developed within SEASON has been integrated in the development release of the ETSI SDG. In particular, this component is responsible for the comprehensive support of packet-optical devices. Considering the optical domain, the Metro Controller is capable of configuring the key optical transmission parameters of QSFP-DD ZR/ZR+ coherent pluggable modules—such as frequency, TX power level, and application code. In the packet domain, different operations have been realized, including IP address assignment, VLAN setup, activation of BGP adjacencies, and activation of routing policies. The controller communicates with IPoWDM devices via NETCONF using OpenConfig models, while a NETCONF agent on each device enables the translation to physical configurations through CMIS/Sonic REST APIs. At the bootstrap, the NETCONF protocol is adopted to retrieve from each IPoWDM device relevant components, including interfaces and optical channels (as shown in Figure 3-2)-2

In the demonstration, the novel capability of establishing L2 VPN services over disaggregated IPoWDM infrastructures has been exploited. In particular, the TFS controller has been extended to support L2 VPN activation routines on Edgecore SonicOS-based devices. After the activation of the L2 connectivity among ZR/ZR+ pluggable transceivers (tuning frequency, TX power and

application code) the following steps are executed: (i) IP address is set in each device to activate their neighbouring; (ii) the real L2VPN configuration performed, activating a VXLAN tunnel (acting as Network Virtualization Overlay), supporting the Ethernet Virtual Private Network (EVPN); (iii) the BGP instance (AS number 65001) for the exchange of data related to the L2VPN capability is activated, relying on Sonic FRR routing instance; (iv) a VLAN instance is then created (i.e., VLAN 10) on the client interfaces (in the experiment the 10Gbps interfaces of the Edgework switches Ethernet256 and Ethernet257 have been considered; (v) the local VLAN is mapped to a VXLAN Network Identifier (VNI), being able to perform later the route target for the data exchange among the nodes (in the experiment VLAN 10 is mapped to the VNI 1000 in both the nodes). Figure 3-3 shows the Edgework L2VPN configuration in the devices.-3

Device IPoWDM1.1 (012c4e67-0024-5f30-b527-55ec6daf2639)

[Back to device list](#)
[Update device](#)
[Delete device](#)

UUID: 012c4e67-0024-5f30-b527-55ec6daf2639
Name: IPoWDM1.1
Type: optical-transponder
Controller:
Status: ENABLED
Drivers:

- OC

Endpoint UUID	Name	Type	Location
4079dce7-3fe6-50a1-b4b4-6574d2db02cb	0	port-0	
67891767-4d60-53a9-82a4-c2477daca4f7	16	port-16	
6f6cb9af-3905-5fe0-827c-a942ce1b6fb4	32	port-32	
c4b8eb66-2fcf-5f66-802f-6cd9590fe70b	48	port-48	

Configurations:

Key	Value		
_connect/address	• 172.17.254.52		
_connect/port	• 2022		
_connect/settings	<ul style="list-style-type: none"> allow_agent: False commit_per_rule: False device_params: {'name': 'default'} endpoints: [] force_running: False hostkey_verify: False look_for_keys: False manager_params: {'timeout': 120} password: admin type: optical-transponder username: admin 		

Figure 3-2: Discovery of the IPoWDM device.

```

sonic# show bgp l2vpn evpn
BGP table version is 36, local router ID is 1.1.1.1
  Network          Next Hop          Metric LocPrf Weight Path
Route Distinguisher: 1.1.1.1:2
*> [2]:[0]:[48]:[42:73:f2:d8:d8:8b]
    1.1.1.1
    ET:8 RT:65001:1000
    32768 i
*> [2]:[0]:[48]:[42:73:f2:d8:d8:8b]:[128]:[fe80::4073:f2ff:fed8:d88b]
    1.1.1.1
    ET:8 RT:65001:1000 ND:Router Flag
    32768 i
*> [2]:[0]:[48]:[56:9b:e0:f2:a2:b4]
    1.1.1.1
    ET:8 RT:65001:1000
    32768 i
*> [2]:[0]:[48]:[56:9b:e0:f2:a2:b4]:[128]:[fe80::549b:e0ff:fef2:a2b4]
    1.1.1.1
    ET:8 RT:65001:1000
    32768 i
*> [3]:[0]:[32]:[1.1.1.1]
    1.1.1.1
    ET:8 RT:65001:1000
    32768 i
Route Distinguisher: 2.2.2.2:2
*>i[2]:[0]:[48]:[50:7c:6f:57:6c:b0]
    2.2.2.2
    RT:65001:1000 ET:8
    100 0 i
*>i[2]:[0]:[48]:[50:7c:6f:57:6c:b0]:[128]:[fe80::527c:6fff:fe57:6cb0]
    2.2.2.2
    RT:65001:1000 ET:8
    100 0 i
*>i[2]:[0]:[48]:[50:7c:6f:57:6c:b3]
    2.2.2.2
    RT:65001:1000 ET:8
    100 0 i
*>i[2]:[0]:[48]:[50:7c:6f:57:6c:b3]:[128]:[fe80::527c:6fff:fe57:6cb3]
    2.2.2.2
    RT:65001:1000 ET:8 ND:Router Flag
    100 0 i
*>i[3]:[0]:[32]:[2.2.2.2]
    2.2.2.2
    RT:65001:1000 ET:8
    100 0 i

```

Figure 3-3: L2 VPN configuration in the Edgecore devices.

3.2.4 Network Service Orchestrator

The Network Service Orchestrator as the central intelligence for end-to-end service orchestration, dynamically managing resources across optical, IP, and RAN domains to fulfill service-level objectives (SLOs) such as guaranteed bandwidth and latency. It enables autonomous resource optimization, self-healing, and energy efficiency in next-generation optical networks.

The Network Service Orchestrator coordinates spatial PON, IPoWDM metro, and RAN domains for AR/VR services. Dynamically scales DU resources via RIC and activates P2MP transceivers.

It consists of a combination of two components (Figure 3-4): IETF Network Slice Controller (NSC)[Con25], which translates the intents, performs the path computation necessary to determine the equipment and parameters involved in the services to be created, and generates the necessary services for each domain; and TeraFlow SDN (TFS), which communicates with the domain controllers, receives the services generated by the NSC, and sends the corresponding services to each controller in the required format.

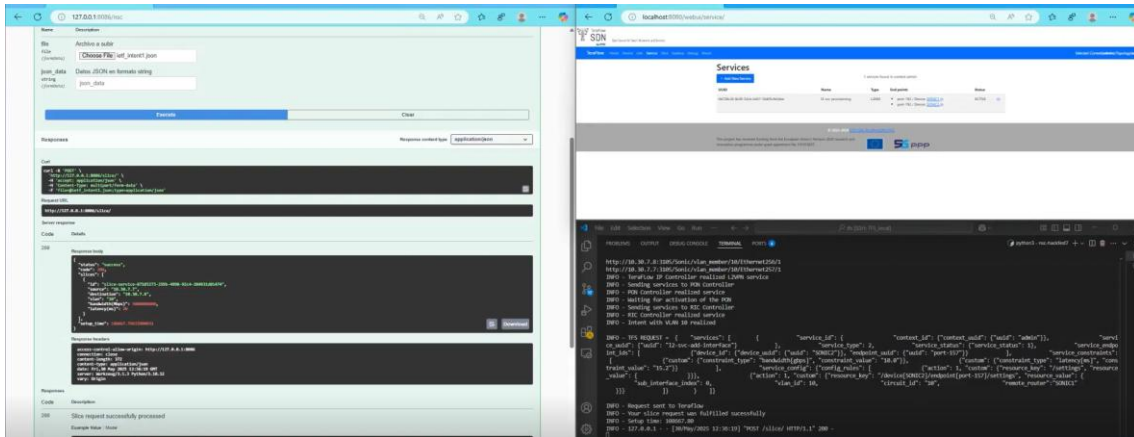


Figure 3-4: Network Orchestrator Components: Left) NSC API, Top Right) TeraFlow, Bottom Right) NSC Logs.

3.2.5 XR Service

The XR service developed by WEST is used as the application workload to validate the end-to-end behavior of the SEASON architecture. The system supports guided virtual visits in a museum-like environment and is designed around a client-server model, as shown in Figure 3-5. The server, implemented in Python, maintains the global state of the virtual environment, manages asset distribution through an HTTP REST API, and provides dedicated TCP channels for real-time interaction updates and voice streaming. Asset data are stored centrally and delivered on demand through JSON-based metadata and OBJ/MTL/texture files retrieved via HTTP.

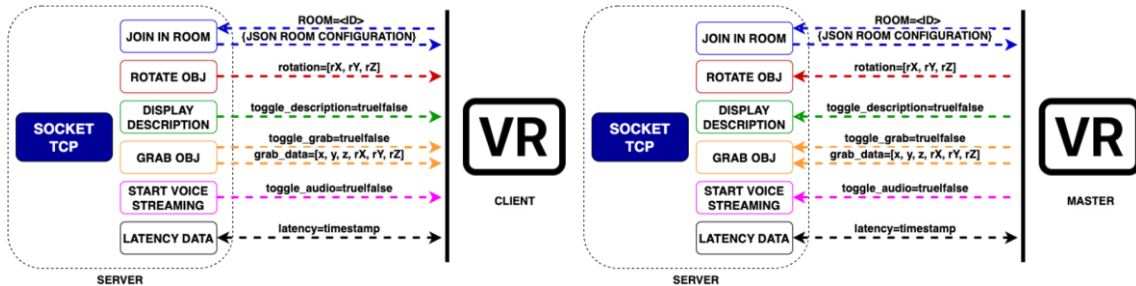


Figure 3-5: XR system architecture. Server (HTTP + TCP) and Unity/OpenXR clients.

The VR client application is implemented in Unity using the OpenXR framework and supports two operation modes: *master* and *visitor*. The master user controls the flow of the virtual experience, manipulates the displayed 3D artefact, and can initiate contextual actions such as the display of textual information or the activation of voice communication. Visitor clients maintain a consistent view of the scene and receive all updates generated by the master. Once an asset is selected, the client establishes a persistent TCP connection that carries all scene updates during the session, including event-based actions and continuous manipulation streams.

Dissemination Level	PU
---------------------	----

The interaction model between master and visitors is illustrated in Figure 3-6. Continuous object manipulation generates high-frequency updates—up to 72 FPS—containing position and orientation vectors, while lower-rate commands are transmitted for discrete actions such as rotation or text display. A separate voice socket is created on demand and scoped to the virtual room, allowing the master to communicate with connected visitors. This mix of steady downstream flows, bursty upstream messages, and continuous high-rate updates makes the XR service particularly sensitive to delay variations. Any increase in latency or jitter immediately affects the alignment between motion, rendering, and perceived interaction.

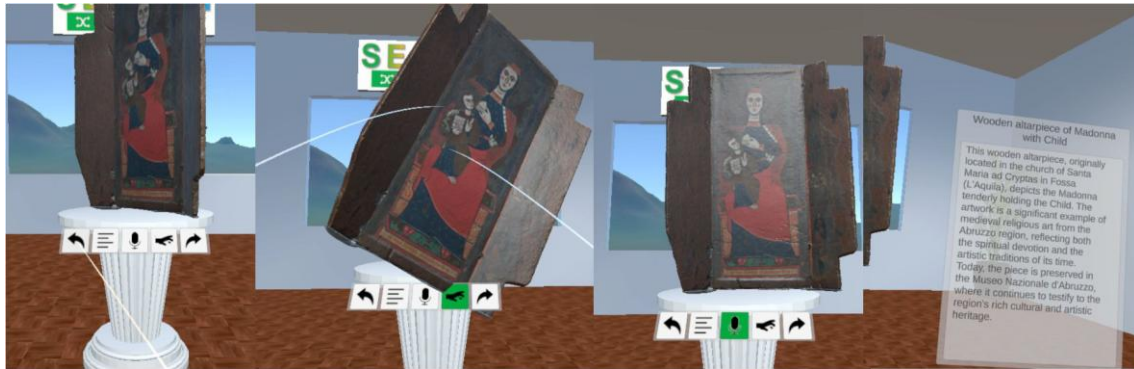


Figure 3-6: Real-time update flow between master and visitor clients, including 72 FPS object manipulation.

This behavior results in a traffic profile that is well suited for evaluating SEASON’s dynamic orchestration mechanisms. The XR service produces reproducible bursts during object manipulation and strict timing constraints for synchronization across clients, creating a realistic load for the Spatial PON, the metro IPoWDM transport, and the O-RAN domain. Within this environment, SEASON validates that spatial lane activation, expedited-forwarding behavior in the PON, and real-time DU scaling can be coordinated without degrading the quality of the XR experience. As such, the XR application acts both as a functional demonstration of edge-hosted immersive services and as a controlled stress source for assessing the responsiveness of the integrated optical-mobile control plane.

3.3 DEMO WORKFLOW

The demonstration begins with a single XR user (UE1) connected through the O-RAN system. In this initial phase, the infrastructure operates in a minimally provisioned configuration, with RU1 and DU1 active and serving all XR traffic, while RU2 and DU2 remain idle. Only one spatial lane of the SDM-PON is enabled, carrying the upstream and downstream flows generated by the XR application, and a single L2VPN midhaul path connects DU1 to the centralized CU functions in the metro domain. As shown in Figure 3-7, this initial state is maintained as long as the

Telemetry Controller (TC/IA) reports that the observed radio utilization and PON throughput remain within the limits that can be handled by the active DU and the single optical lane.

When a second user (UE2) joins the XR session, the increased traffic load is reflected simultaneously in the radio and optical segments. The TC/IA identifies this change by monitoring PRB utilization, PON traffic variations, and midhaul performance. Once the increase persists beyond the predefined threshold, a trigger is sent to the Network Service Orchestrator (NSO), indicating that the system is operating close to resource saturation and requires coordinated scaling.

Upon receiving the notification, the NSO initiates the multi-domain scaling workflow. The first step involves requesting the activation of DU2 via the RAN Intelligent Controller (RIC), which transitions DU2 from an idle to an operational state and prepares RU2 to serve mobile traffic. In parallel, the NSO instructs the PON Controller to activate a second spatial lane on the SDM-PON, thereby increasing the available optical capacity and ensuring that XR traffic from both users can be transported without congestion. At the same time, the Metro Controller provisions a new L2VPN instance to provide midhaul connectivity for DU2, configuring the IPoWDM switches and coherent pluggables in accordance with OpenConfig models.

Once the new DU/RU pair, the second spatial lane, and the additional midhaul path are all active, the O-RAN network automatically redistributes users across DU1 and DU2 according to its standard association procedures. Throughout the scaling sequence, the XR application remains uninterrupted: all real-time position, rotation, and voice-streaming updates continue to be transmitted without noticeable artefacts, confirming the ability of the integrated orchestration layer to perform multi-domain reconfiguration under load.

When UE2 leaves the session, telemetry values revert to their baseline, prompting the NSO to execute the down-scaling procedure. In this phase, DU2 is deactivated, the second spatial lane is released, and the additional L2VPN midhaul is torn down. The system then returns to the single-lane, single-DU configuration used at the beginning of the workflow.

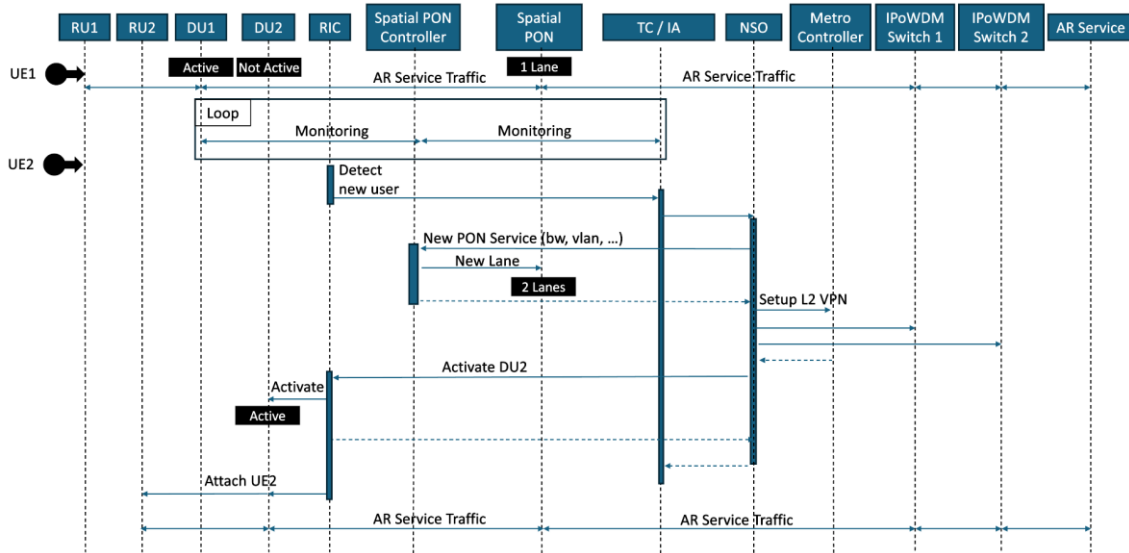


Figure 3-7: Coordinated activation of DU2, second PON lane, and additional midhaul L2VPN following detection of increased XR load.

3.4 RESULTS

The field trial carried out in L’Aquila provides a comprehensive validation of the SEASON integrated architecture across optical access, metro transport, and O-RAN layers. The experiment confirms that the coordinated control framework can react to variations in XR traffic demand by activating additional resources across all domains while maintaining uninterrupted service continuity. The results presented in this section summaries the behavior of the system under scaling conditions and highlight the contribution of the Spatial PON and the SEASON control plane to latency stability, energy efficiency, and optical performance.

At the radio layer, the transition from a single-user to a multi-user XR session triggers the activation of a second RU/DU pair through the RIC. The correct activation and association of users can be observed in Figure 3-8, which reports the network topology after scaling. Here, the two radio units are shown serving independent UEs, confirming that the O-RAN infrastructure successfully transitions to a dual-cell configuration. This transition is crucial for providing sufficient uplink and downlink capacity for the XR workload, particularly during periods of intense object manipulation when fine-grained updates must be delivered with minimal delay. Throughout the scaling process, the XR application preserved stable rendering, synchronized motion updates, and reliable voice communication, demonstrating that the orchestration actions did not introduce artefacts in the user experience.

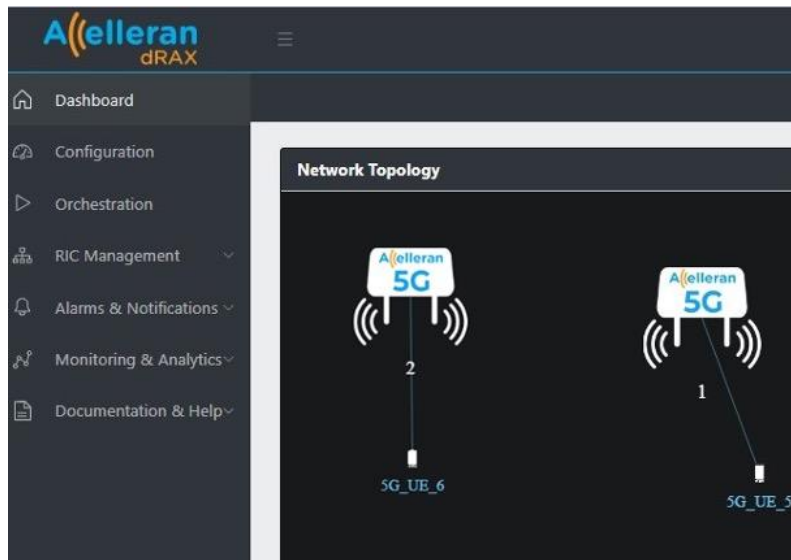


Figure 3-8: O-RAN topology after DU/RU scaling.

The behavior of the Spatial PON during the scaling procedure is illustrated in Figure 3-9, which reports the measured optical spectra on the two active cores of the multicore fiber. In the initial phase, only core #1 carries the upstream and downstream XGS-PON wavelengths, while core #2 remains idle. Once the NSO triggers the activation of a second spatial lane, the PON Controller enables the additional core and configures the corresponding optical switch paths. As shown in the measurements, both cores then carry full bidirectional PON traffic: the upstream signals at ~1270 nm and the downstream signals at ~1577 nm appear clearly on the spectrum for both cores. The absence of measurable inter-core crosstalk confirms the effectiveness of the spatial separation provided by the multicore fiber and the suitability of the SDM-PON platform for multi-lane operation under dynamic control. The stability of the optical power levels following the activation procedure further indicates that the system converges rapidly to a steady state without introducing penalties in the physical layer.

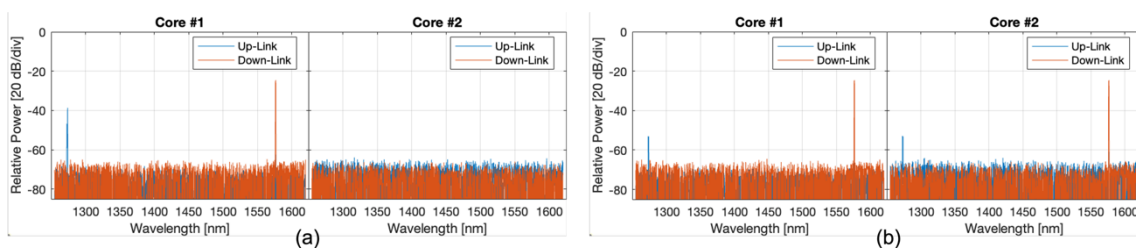


Figure 3-9: Optical spectrum for single-lane (left) and dual-lane (right) Spatial PON operation.

In addition to validating functional behavior, the trial also highlights the impact of multi-domain scaling on energy consumption. As shown in Figure 3-10, three power traces were collected during the experiment. The baseline curve represents a hypothetical scenario where both RUs and PON ports remain continuously active, resulting in a constant consumption of approximately 185 W. The measured curve corresponds to the SEASON integrated operation: starting from

Dissemination Level	PU
---------------------	----

~170 W in single-lane, single-DU mode, the system increases to ~185 W when both RUs and PON ports are activated and returns to ~170 W once the second user disconnects and the NSO triggers the release of resources. The achievable curve represents the power savings that would be possible if full RU shutdown were supported by the commercial equipment used in the trial. In this case, the energy savings would reach approximately 35 W (19%), compared to the 15 W (8%) realised with the available hardware. These measurements confirm that the SEASON orchestration framework can effectively reduce energy consumption by dynamically adapting the access and radio domains to the actual load profile generated by the XR service.

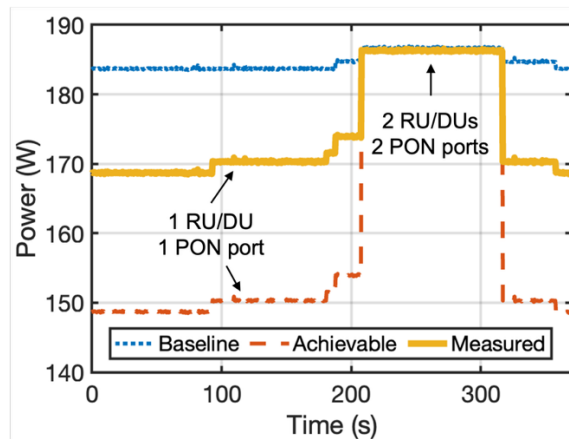


Figure 3-10: Power consumption dynamics during single-lane and dual-lane operation.

Finally, the provisioning of the additional midhaul L2VPN was successfully executed through the TeraFlow SDN controller. The main source of delay in the scaling process originates from the reconfiguration time required by commercial IPoWDM hardware when updating coherent pluggable parameters and transport configurations. Nevertheless, this delay did not affect the continuity of the XR session, as the orchestration logic ensures that the DU activation and user offloading only take place once the midhaul connectivity has been fully established. The system therefore demonstrates the ability to perform coordinated scaling actions across multiple domains while preserving service stability, confirming the suitability of the SEASON architecture for supporting latency-sensitive and bandwidth-variable immersive applications.

4 DEMO 2 EXTENSION 1: VALIDATION OF SEASON XR SERVICE BACKEND DEPLOYMENT

This section details the first extension of Demo 2, led by WINGS, that focuses on the validation of the application layer within the SEASON ecosystem. Specifically, it addresses the deployment and configuration of the SeasonVR backend server on the project's Kubernetes (k3s) infrastructure. Acting as a preparatory phase for the end-to-end field trial, this activity verifies the stability, network reachability, and resource efficiency of the containerized service using a "headless" validation approach. This ensures that the edge computing environment is fully operational prior to the introduction of the optical-mobile transport and client-side hardware.

4.1 USE CASE DESCRIPTION

The primary objective of this extension was to deploy, configure, and validate the SeasonVR backend server service within the project's existing Kubernetes (k3s) infrastructure. This task serves as a critical preparatory phase for "Demo 2," ensuring that the server-side components are fully operational and stable before the introduction of client-side hardware (VR headsets).

Scope and methodology

The scope of this use case is defined by the principle of "headless validation." This methodology decouples the backend deployment from the frontend client availability, allowing the infrastructure team to certify the service's readiness independently. By simulating network connections and verifying port responses without the physical VR interface, we effectively de-risk the final integration phase.

The deployment was measured against four key success criteria:

1. **Service stability:** The SeasonVR container must achieve a stable Running state on the designated worker node without restart loops or crash events.
2. **Network reachability:** The application's critical communication ports, specifically the HTTP API (5001) and the Data Exchange Socket (17000), must be accessible within the network.
3. **Resource efficiency:** The deployed service must demonstrate a minimal resource footprint (CPU and Memory) to ensure it can coexist with other project workloads.
4. **Integration readiness:** The system must present a finalized IP address and configuration set that can be immediately utilized by the VR client application once hardware is available.

Successfully executing this use case ensures that the on-site demonstration can proceed with a "plug-and-play" experience. By resolving architectural dependencies and network

configurations in advance, the project eliminates potential bottlenecks during the live presentation.

4.2 COMPONENTS

The solution architecture integrates the specific requirements of the SeasonVR application with the standardized WINGS Kubernetes platform. The following components were utilized to build the system.

Infrastructure layer

The deployment targets the k3s cluster. The cluster topology used for this extension consists of a high-availability control plane and dedicated worker nodes, see Table 4-1.

Table 4-1: Cluster topology and node inventory for the SeasonVR deployment.

Node Name	Role	IP Address	OS / Kernel	Architecture
season-05	Control Plane (Master)	172.24.36.58	Ubuntu 22.04 LTS	amd64
season-08	Worker Node	172.24.36.61	Ubuntu 22.04 LTS	amd64

Software and application layer

The core service is encapsulated in a Docker container, managed by Kubernetes declarative manifests.

- Container Image: seasonunivaq/season-vr-server:latest
 - Source: Docker Hub Private Registry
 - Authentication: Managed via Kubernetes Secret (type: docker-registry).
- Orchestration Object: Kubernetes Deployment
 - Replica Count: 1 (Singleton pattern required for state management).
 - Namespace: seasonvr-demo (Isolated environment created specifically for this extension).

Networking strategy

To minimize latency and avoid the complexity of NAT or real-time VR data streams, the deployment utilizes the Host Network strategy (hostNetwork: true). This binds the application directly to the physical network interface of the worker node. The port configuration is presented in Table 4-2.

Table 4-2: Port Configuration.

Port	Protocol	Description	Status at Idle
5001	TCP	HTTP API: Handles RESTful requests for room management and client synchronization.	Listening
17000	TCP	Data Socket: The primary channel for real-time object manipulation and state exchange.	Listening
26001-26005	TCP	Voice Streams: Dedicated channels for voice communication (one per room).	Closed (On-demand)*

4.3 WORKFLOW

The implementation followed a structured, multi-phase workflow designed to ensure verifiability and repeatability. All operations were initiated from the master node (season-05).

Phase 1: Environment preparation

The first phase involved preparing the Kubernetes namespace to host the application securely.

1. **Namespace creation:** A dedicated namespace `seasonvr-demo` was initialized to provide logical isolation from other project components.
2. **Credential configuration:** A Kubernetes Secret (`regcred`) was generated containing the Docker Hub account credentials (`seasonunivaq@...`), enabling the cluster to authenticate and pull the private container image.

Phase 2: Node and architecture Enablement

To ensure that the application container which may target specific CPU architectures runs seamlessly on infrastructure, the host node was prepared with compatibility layers.

1. **Node selection:** Worker `season-08` (172.24.36.61) was designated as the target node.
2. **Compatibility layer:** The `qemu-user-static` and `binfmt-support` packages were installed on the worker node. This proactive step ensures that the container runtime can transparently execute binary formats from different architectures (e.g., ARM64) on AMD64 hardware, preventing `ExecFormatError` issue and ensuring stability.

Phase 3: Deployment execution

The application was deployed using a targeted manifest strategy.

1. **Scheduling:** The deployment manifest was pinned to `season-08` node using a `nodeSelector`. This guarantees that the application always receives the specific IP address (172.24.36.61) configured in the VR clients.
2. **Rollout:** The Kubernetes pulled the application image, initialized the container, and mounted the local host path `/var/lib/seasonvr/client_logs` to the container to ensure persistent log storage.

Phase 4: Verification and validation

A headless validation protocol was executed to confirm success:

1. **Status check:** Verified the Pod phase changed to Running state.
2. **HTTP probing:** Used curl to ping the API port (5001).
3. **TCP handshake:** Used netcat (nc) to open a raw TCP connection to port 17000, simulating a client connection.
4. **Metric analysis:** Queried the Kubernetes Metrics Server to baseline resource usage.

4.4 RESULTS

The deployment was completed successfully. The SeasonVR backend service became active, stable, and reachable at 172.24.36.61. The following evidence confirms the system's status.

Service status: The application Pod is running successfully on the designated worker node (season-08):

```
$ kubectl -n seasonvr-demo get deploy,rs,po -o wide
```

NAME	READY	UP-TO-DATE	AVAILABLE	AGE	CONTAINERS
IMAGES					
deployment.apps/seasonvr-server	1/1	1	1	15m	server
seasonunivaq/season-vr-server:latest					
NAME	READY	STATUS	RESTARTS	AGE	IP
NODE	NOMINATED NODE				
pod/seasonvr-server-9d649b564-bz7b4	1/1	Running	0	3m	172.24.36.61
season-08	<none>				

The Orchestrator confirms 1/1 replicas are ready and the pod is assigned to the correct IP.

Network connectivity: The required network endpoints are responsive.

```
$ curl -s -o /dev/null -w '%{http_code}\n' http://172.24.36.61:5001/
404

$ nc -zv 172.24.36.61 17000
Connection to 172.24.36.61 17000 port [tcp/*] succeeded!
```

The HTTP 404 code confirms that the web server is active and handling requests (the root path / has no mapped handler, which is expected behavior). The TCP success on port 17000 confirms the socket server is accepting connections.

Resource consumption: The application is highly efficient and imposes negligible load on the cluster.

```
$ ./metrics-basic.sh
---- kubect1 top nodes ----
NAME          CPU (cores)  CPU (%)  MEMORY (bytes)  MEMORY (%)
season-05     50m          1%       2859Mi          73%
season-07     15m          0%       3003Mi          37%
season-08     17m          0%       2825Mi          35%
---- kubect1 top pods ----
NAME          CPU (cores)  MEMORY (bytes)
seasonvr-server-9d649b564-bz7b4  1m           67Mi
```

5 DEMO 2 EXTENSION 2: DYNAMIC CONTROL OF MULTI-TECHNOLOGY OPTICAL ACCESS NETWORKS WITH PON AND COHERENT P2MP

Future access networks will have to support multiple operators and services within the same physical infrastructure. Operators therefore will require a flexible architecture capable of integrating legacy PON systems with emerging high-capacity optical solutions such as coherent point-to-multipoint (P2MP) XR Optics and introducing the possibility of sharing the optical distribution network infrastructure among different actors to implement resource dynamic allocation and process automation.

The goal of this extension is therefore to demonstrate the feasibility of a multi-technology integrated metro-access scenario, comprising the coexistence over the same single-fiber access network infrastructure of legacy XG-PON services and high-end metro services, based on digital subcarrier multiplexing (DSCM) point to multipoint XR technology, carried transparently from the metro segment network, where both type of services can be dynamically provisioned under a unified control and management framework.

5.1 USE CASE DESCRIPTION

The considered use case, illustrated in Figure 5-1, is that of a business customer having headquarter in the metro-aggregation segment, several km away from a local central office (CO), and branch offices that can be reached by the passive optical infrastructure in the same local CO domain, and already supporting several legacy PON services.

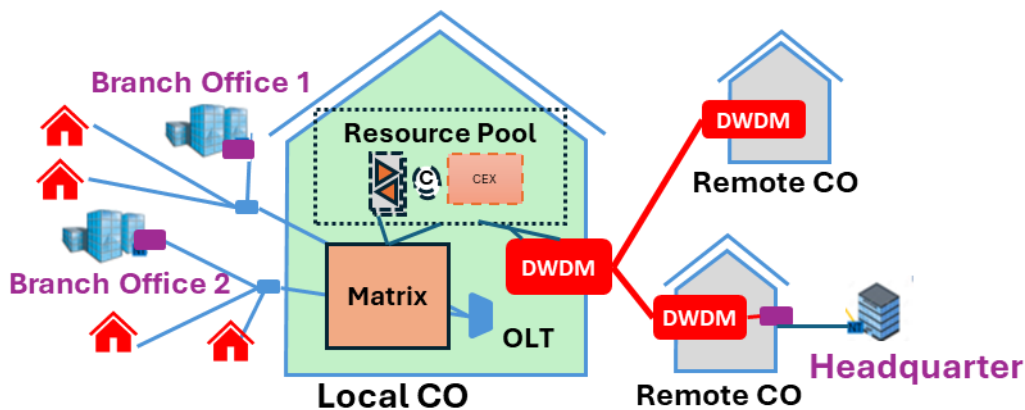


Figure 5-1: Use case logical view.

To provide connectivity between the headquarter and the branch offices of the new customer, coherent point-to-multipoint XR Optics technology can be used. The new services can be

provisioned automatically, without physical intervention in the local CO. This is possible thanks to:

- a flexibility element (a spatial optical matrix) already placed in the local CO between the Optical Line Terminals (OLT) and the Optical Distribution Networks (ODN) that carry the PON services.
- a pull of resources that ensure the adaptation of XR signals transported transparently from the metro to the access segment (EDFA optical amplifiers to cope with the transmission through the link in the metro segment, optical circulators to switch from double fiber to single fiber domains and coexistence elements to avoid disturbance between PON services and XR services).
- an integrated control plane framework that enables control and management under the same orchestrator of packet and optical, metro and access devices.

5.2 COMPONENTS

The experimental testbed shown in Figure 5-2, maps the logical view of Figure 5-1.

Regarding the data-plane, the Local CO (green box) is the demarcation point between two functional domains, the access domain, that comprises the two ODNs in the left side of the picture, and the metro domain, that interconnects the Remote COs (grey boxes in the right side of the picture) through a dual-fiber infrastructure.

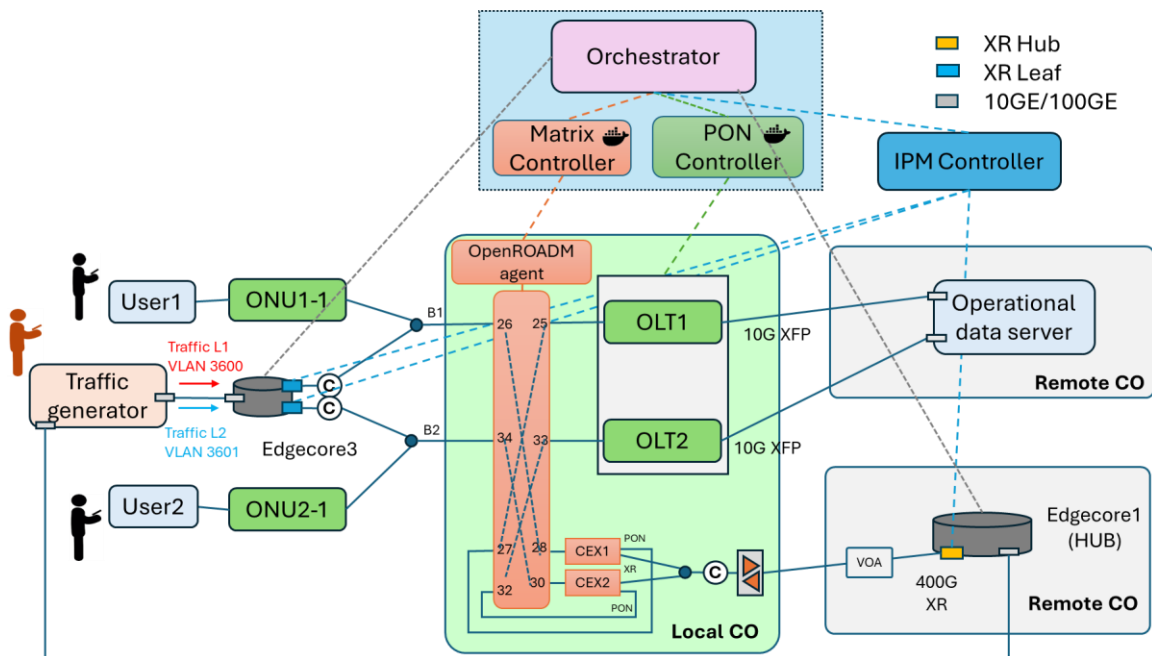


Figure 5-2: Demo setup - final cross connections in the matrix are shown.

In the access domain, two optical line terminals (OLTs) are connected through the matrix to single-fiber distribution branches towards two optical network units (ONUs) to provide standard XG-PON access services.

The Remote CO in the bottom right side of the picture represents the headquarter of the new business customer and is equipped with an IPoWDM box (Edgecore 1) hosting a 400G XR transceiver, configured as a HUB, and interconnected to the local CO through an optical attenuator that emulates a 50km link. Two branch offices of the new business customer, located in the access domain, are emulated by another IPoWDM box (Edgecore 3) equipped with two 100G XR Optic transceivers configured as leaves, exploiting P2MP technique. Optical circulators (C) are adopted at both sides in the P2MP system to convert the dual-fiber transmit/receive XR interfaces into a single-fiber format. Two optical adaptation units, CEX1 and CEX2, are used to aggregate the XR transceivers with the single-fiber bidirectional infrastructure typical of PON systems.

The optical matrix acts as a fiber-programmable interconnection layer among all optical endpoints, including OLTs, ONUs, amplification, circulators and CEX units. By dynamically configuring optical cross-connections, the matrix enables the coexistence of heterogeneous technologies and the on-demand reconfiguration of services within the same physical infrastructure. The picture shows the matrix cross-connections in the case of all the four services (the two XG-PON and the two XR) configured.

Two different applications are adopted for testing of the end-to-end connectivity: end-user servers equipped with 10G interfaces are used for the legacy PON service connection towards the operational server in the metro domain, while a traffic generator is adopted to test the performance of the high-bitrate P2MP-based access services, relying on two traffic streams with different VLAN tags exploiting XR Leaf1 and XR Leaf2 and XR Hub in the aggregation domain.

Regarding the control-plane, a hierarchical control and orchestration framework manages this architecture, exploiting three domain controllers. The PON Controller, is a custom-built framework, handling the PON access provisioning. It exposes a general purpose REST API, exploiting the NETCONF protocol to activate the ONUs and the related connectivity services. The Matrix Controller configures optical cross-connection: by exposing TAPI interface it communicates with an OpenROADM agent that controls the optical matrix. The IPM Controller is a commercial framework, responsible for the management of the XR transceivers, exposing a REST API for the configuration and monitoring of physical layer and connectivity services. On the top, the custom-built centralized Orchestrator supervises all the domain controllers, coordinating resource allocation and service activation across domains, exposing a REST-based API for the services activation. Together, these components enable a software-defined, multi-technology access and metro network where PON and XR Optics services coexist and interoperate dynamically under unified control.

5.3 WORKFLOW

The workflow of the demo is structured in four steps, consisting respectively in the activation of PON-1 service, PON-2 service, XR-1 service and XR-2 service.

step 1 – Activation of the PON branch1 with ONU1-1: the Orchestrator initiates the configuration of the optical matrix to establish the connection between OLT1 and ONU1-1 by cross-connecting port 25 and port 26 of the matrix in Figure 5-2. The Matrix Controller defines the cross-connection parameters, aligning optical channels to ensure transparent transmission. Once the matrix path is established, the PON Controller activates OLT1 and ONU1-1 service, assigning the corresponding VLAN and service identifiers.

step 2 – Activation of the PON branch2 with ONU2-1: similar procedure is performed at this step, interconnecting OLT2 with ONU2-1 (cross-connection of port 33 with port 34 of the matrix in Figure 5-2). The Orchestrator coordinates the process, instructing the Matrix Controller to provision the appropriate optical path and update its configuration database. The PON Controller then enables a second service, associating a distinct VLAN for user traffic, and completing the service setup. Both PON branches now operate simultaneously, each using a dedicated optical path within the matrix, providing independent access services and validating bidirectional transmission through the switching fabric.

step 3 – Activation of the XR service on branch1: with both PON branches operational, the first XR Optics connection is created to provide high-capacity access. The Orchestrator instructs the Matrix Controller to rearrange previous connection among OLT1 and ONU1-1, establishing an optical cross-connection between CEX1 and the first PON branch (removing cross-connection from port 25 to port 26 and substituting it with the two cross-connections of port 25 with port 27 and port 28 with port 26 of the matrix in Figure 5-2). Then the IPM controller is invoked to perform the configuration of the 400G XR Hub and the 100G XR Leaf1 transceivers. This requires both the physical layer and the service activation. Then, the IPoWDM configuration is performed enabling the packet layer configuration. In this case, a service VLAN is configured to convey the traffic generation stream from the ingress port (in the access domain) to the egress port (in the aggregation domain). This enables the first XR transport channel for inter-branch communication while maintaining compatibility with the single-fiber physical topology.

step 4 – Activation of the XR service on branch2: A new XR connection is provisioned to extend the point-to multipoint high-bitrate connectivity service towards the second branch. The Orchestrator triggers the reconfiguration of the optical matrix to enable the communication towards branch2 inserting CEX2 and creating a parallel logical channel with the previous one (removing cross-connection from port 33 to port 34 and substituting it with the two cross-connections of port 33 with port 32 and port 30 with port 34 of the matrix in Figure 5-2). The second XR connectivity service is enabled, via IPM controller, by extending the physical layer to include the new leaf and creating a second connectivity service towards XR Leaf2. Then, as in the previous step, the packet layer configuration is performed enabling the second stream of

traffic. This confirms that multiple XR services can coexist and be dynamically managed alongside PON traffic on the same optical infrastructure.

5.4 RESULTS

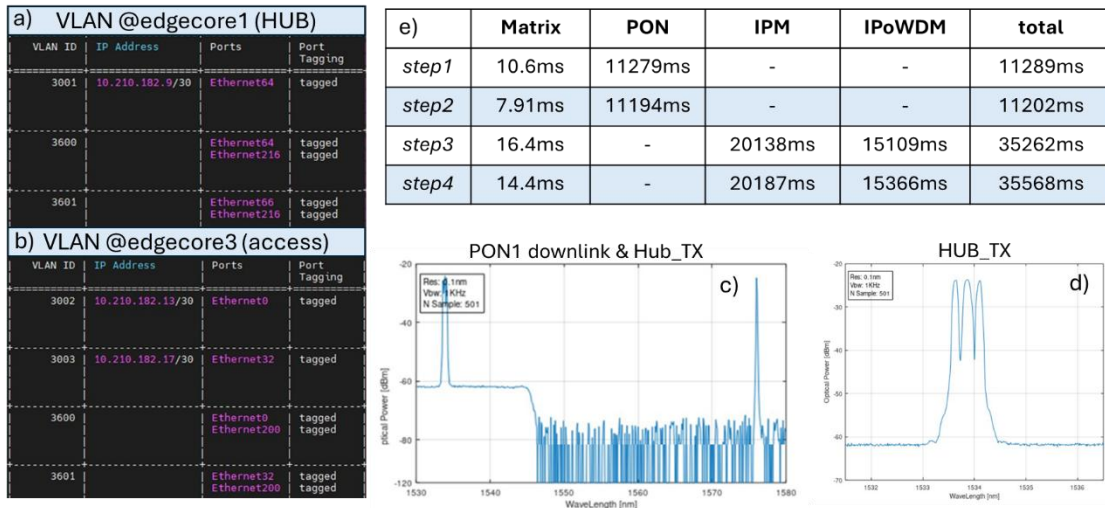


Figure 5-3: Experimental results.

Experimental data-plane and control-plane validation has been performed. Figure 5-3 summarizes the key results. Figure 5-3(a) and Figure 5-3(b) show the status of the VLAN configuration at the end of the procedure (when PON and XR P2MP are up and running). Two sets of VLAN are present, the ones for XR pluggable in-network control and the service VLANs. Considering the branch1, Figure 5-3(c) and Figure 5-3(d) the spectrum view is shown, highlighting the coexistence of PON and XR working at different frequencies. Finally, the control-plane execution intervals of the four phases are reported in Figure 5-3(e), highlighting the time contribution of each of the involved domain controllers.

The matrix control is fast (around 10ms), while longer contributions are due to the configuration of IPM and IPoWDM hardware boxes.

6 DEMO 2 EXTENSION 3: DYNAMIC SERVICE PROVISIONING IN IPOWDM SCENARIOS WITH P2MP PLUGGABLES

To enable fine-grained bandwidth allocation in access, aggregation, and metro segments, the solution builds upon Point-to-Multipoint (P2MP) coherent optics based on digital subcarrier multiplexing (DSCM), which divides a 400G optical carrier into multiple subcarriers, each of which is independently managed in terms of modulation format, power, and routing. By using XR P2MP pluggables in place of traditional point-to-point transceivers.

From a control-plane perspective, the architecture uses a hierarchical SDN model: an XR Intelligent Pluggable Manager (IPM) and an optical line-system controller are coordinated by a FlexOpt network orchestrator, which abstracts the underlying IPoWDM infrastructure for higher-layer systems and exposes a TAPI-based northbound interface for service requests. While maintaining IP and optical domains conceptually distinct but closely coordinated, the orchestrator oversees topology and resource discovery, route calculation, frequency assignment, and life-cycle management of both point-to-point and P2MP services

6.1 USE CASE DESCRIPTION

The particular use case connects a Cisco router and a UfiSpace router via 400G QSFP-DD XR P2MP modules and an optical distribution system made up of 1:4 and 1:8 splitters to create a metro-style IPoWDM testbed in Telefónica's Future Network Laboratory. To simulate a realistic multi-site operating environment, the optical SDN controller is hosted remotely at CTTC and connected via VPN, while an extra whitebox router offers management connection.

In this setup, the demo focuses on leveraging a single XR 400G hub-and-leaf constellation to dynamically supply a 100 Gb/s service between breakout port 1 of the Cisco router and breakout port 1 of the UfiSpace router. The process consists of the following steps: (i) using the IPM to find routers and XR modules and map them to TAPI nodes and DSR service interface points; (ii) activating breakout mode on 400G QSFP-DD ports to create four 100G-capable subcarriers; (iii) choosing a subcarrier and configuring its central frequency and modulation (e.g., 16QAM) based on assignments from the optical controller; and (iv) creating and activating the logical Ethernet service between the chosen breakout interfaces, after which link status and performance indicators are verified.

On the planning side, an AI Agent operates as a network planning assistant that dynamically assigns XR digital subcarriers from a 400G hub to multiple leaf nodes in a P2MP coherent optics scenario, where a single 400G carrier is divided into 16 independent 25 Gb/s subcarriers using DSCM. The agent receives as input the current and historical traffic demands of multiple leaf nodes and, following a system prompt that instructs it to meet all demands with the minimum number of channels, invokes a planning function tool that translates each demand into a

contiguous block of 25 Gb/s subcarriers, capped at four subcarriers (100 Gb/s) per leaf. This tool computes the first and last channel index for every leaf by taking the maximum of current and previous-day traffic (expressed in Gb/s), converting it to a number of 25 Gb/s subcarriers, and packing the users sequentially within the 16-subcarrier budget of the XR hub. By incorporating historic traffic in the decision process, the agent anticipates short-term demand surges and avoids under-dimensioning, while still respecting the hub capacity and the requirement that each leaf receives at least one consecutive subcarrier.

Within the broader SDN-controlled IPoWDM architecture of Telefónica's metro-style XR testbed, the agent behaves as an intent-fulfilment microservice between the high-level service orders and the low-level XR pluggable configuration on Cisco and UfiSpace routers. Once the FlexOpt-like orchestrator has validated that all endpoints support XR pluggables, discovered the topology via the optical controller and IPM, and computed a feasible P2MP tree that satisfies spectrum and QoS constraints, it can call the agent to obtain a concrete subcarrier map—first and last channel per leaf—compatible with the 400G breakout structure. These assignments are then pushed through the TAPI-based northbound interface toward the optical controller, which configures ROADMs paths, activates the chosen subcarriers with appropriate modulation formats (for example, 16QAM), and establishes the logical Ethernet services toward each leaf. In this way, the AI-driven subcarrier allocation complements hierarchical SDN orchestration, enabling fine-grained, automation-driven bandwidth scaling across hub-and-leaf constellations, reducing manual configuration time from hours to minutes and supporting multi-vendor P2MP services for 5G xHaul, enterprise access, or edge-cloud connectivity.

6.2 COMPONENTS

The solution framework builds several integral components working in concert, as shown in Figure 6-1. XR coherent pluggable optics are the linchpin technology, offering standardized, multi-rate interfaces capable of seamlessly integrating with existing network elements while providing the advanced reach and modulation required for P2MP scenarios.

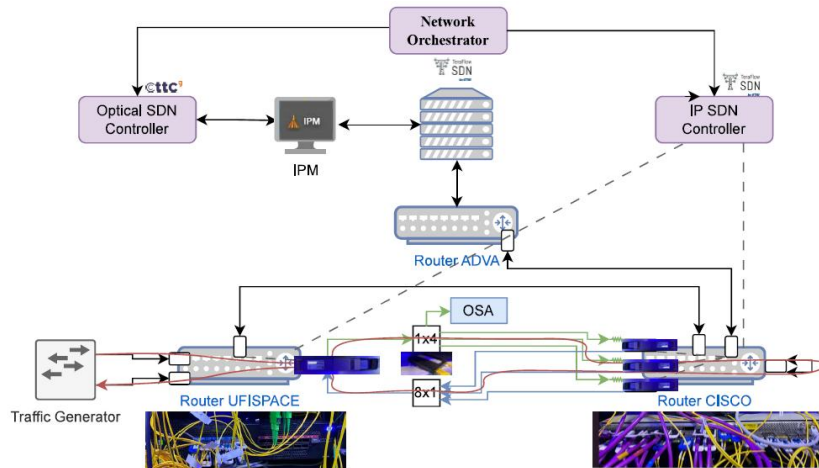


Figure 6-1: Demo PTMP Testbed.

The SDN orchestrator, a complex software entity that converts high-level service intentions into precise provisioning operations, is crucial to the control plane. This orchestrator maintains a full network perspective by interfacing with dispersed domain controllers that manage the optical and IP network domains. It isolates device specificities and provides uniform open APIs, allowing for easy programmability and interaction across vendor equipment.

On the physical layer, network elements such as ROADMs, routers, and transponders equipped with XR pluggables form the tangible infrastructure, enabling high-capacity, low-latency connections. These devices provide telemetry and operational feedback that is continuously fed into the control framework for real-time monitoring and corrective actions.

Furthermore, the architecture is tightly coordinated with OSS and BSS systems providing the essential service management functions such as order handling, inventory tracking, billing, and assurance, ensuring that service provisioning aligns with business processes and customer demands.

6.3 WORKFLOW

Figure 6-2 describes the workflow of the use case.

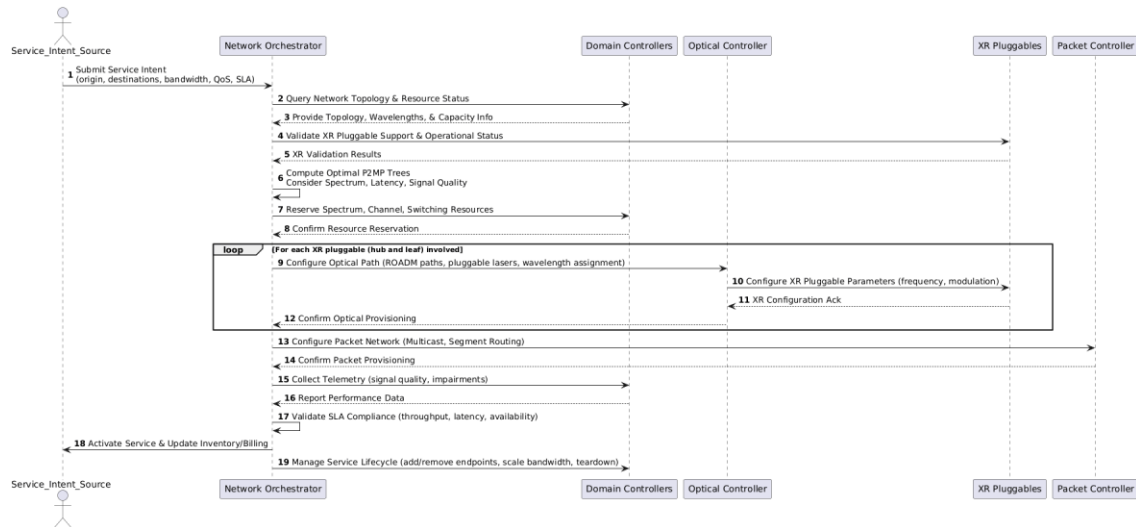


Figure 6-2: PTMP Workflow.

The workflow begins when the orchestrator receives a clear service of intent from operational or business portals. This intent specifies the origin point as well as the multiple destinations to be reached, alongside bandwidth, quality of service, and service-level objectives. Once received, the orchestrator embarks on discovering the current network topology and resource status by querying the associated domain controllers. These controllers abstract the detailed capabilities and states of their managed devices, offering a global yet granular picture of available paths, compatible wavelengths, and reserved capacities. The Optical Controller periodically obtains the IPM OpenID token and proceeds with the discovery of hosts (routers) and pluggable devices (modules) using a REST interface over HTTPs. Discovered network elements are mapped to TAPI network nodes (Figure 6-3).

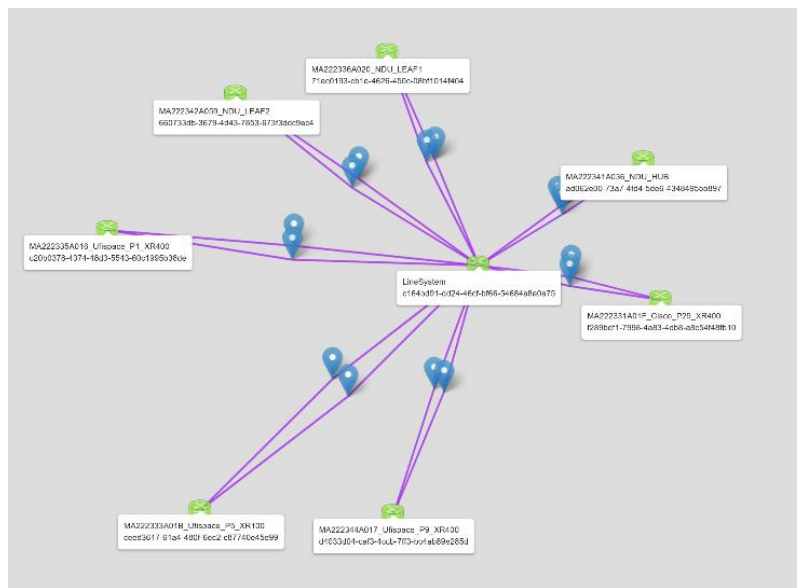


Figure 6-3: Optical Controller: Topology GUI.

Dissemination Level	PU
---------------------	----

An essential early step is validating whether all involved endpoints support XR pluggable modules and verifying their operational status. This validation is critical to ensure service feasibility and to prevent configuration attempts on incompatible hardware. Subsequently, the orchestrator computes optimal P2MP trees taking into consideration network constraints such as spectrum fragmentation, signal degradation risks, and latency requirements. The computation balances resource efficiency with robustness, positioning the network to efficiently handle dynamic scaling demand.

With a path in hand, the orchestrator proceeds with resource reservation, locking down spectrum, channel, and switching resources across the network in a manner that prevents conflict and contention. The actual provisioning is executed through precise commands delivered to domain controllers, which then configure the optical transport network — setting ROADMs paths, activating pluggable lasers, and assigning wavelengths — as well as the IP/MPLS layer for multicast and segment routing where applicable. Figure 6-4 shows the optical services.

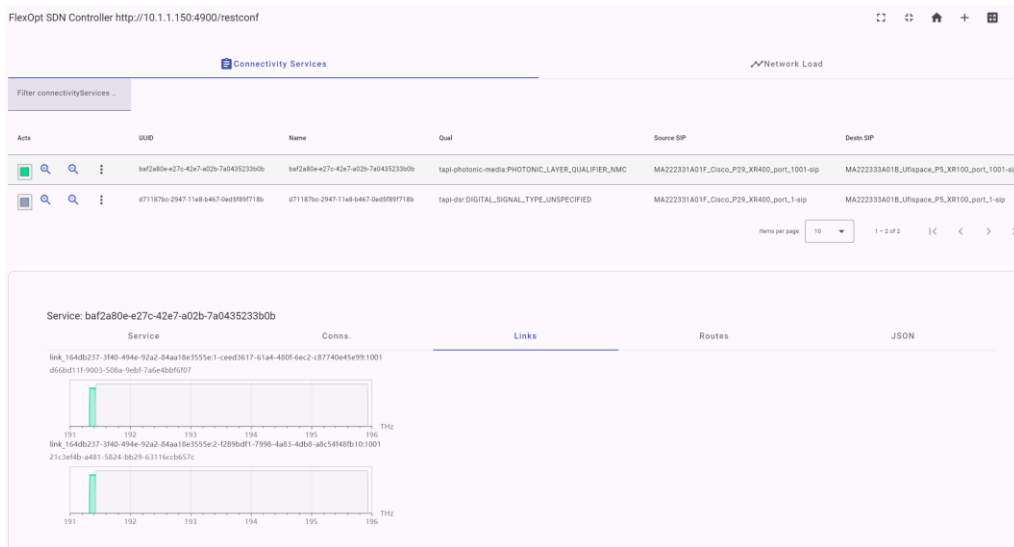


Figure 6-4: Optical Controller: enabling services.

Following provisioning, continual monitoring of performance parameters such as signal quality and optical defects maintains service integrity. This telemetry also allows for verification of stated SLA metrics like throughput, latency, and availability guarantees.

Only once all validation checks pass, the orchestrator activates the service officially, triggering updates to the inventory and billing systems to reflect the service establishment. Through the lifecycle, the system supports operational agility - allowing the addition or removal of endpoints, scaling bandwidth, or fully decommissioning the service based on real-time business needs.

6.4 RESULTS

This use case has demonstrated experimentally the use of the dual management of programmable pluggable optics in commercial routers and whiteboxes. Dynamic configuration capabilities are demonstrated in a hierarchical SDN architecture.

Key Technical Results from the Document:

- Demonstrates real-world dynamic P2MP service provisioning with XR pluggables using a multi-vendor lab setup. Figure 6-5 shows the spectrum before and after the service configuration. It demonstrates that the optical service is configured.
- Automation reduced manual provisioning time from several hours to a few minutes per service.
- Workflow enables bulk setup/teardown for hundreds of endpoints, supporting scenarios like 5G cell site addition or remote cloud site connection.

Lessons learned:

- Improve inventory/telemetry synchronization between SDN orchestrator and domain controllers, particularly across multi-vendor borders.
- Ensure consistent firmware features (e.g., alarm support and statistics) across all XR pluggable devices

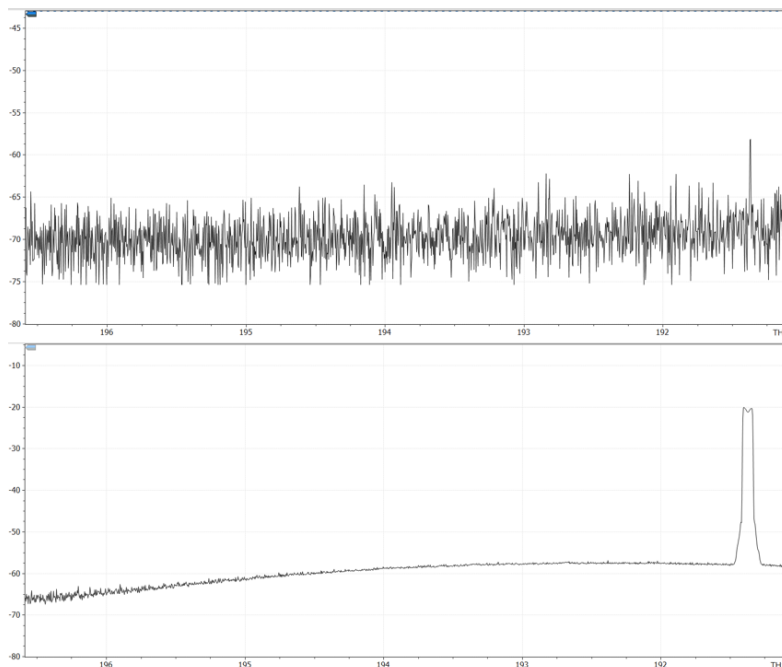


Figure 6-5: Demo PTMP: Spectrum before and after the service creation.

7 CONCLUSION

This deliverable has fulfilled its dual role within the SEASON project by providing the comprehensive assessment of project-wide KPIs and by reporting the final validation results of Demo 2.

The KPI evaluation presented in Section 2 confirms that SEASON has achieved or substantially progressed towards all major performance targets. The project has demonstrated network capacity scaling through MBoSDM and SDM-PON technologies, energy savings ranging from 18% to 38% in access-metro segments, and service creation times well below the 90-minute threshold. AI/ML-driven automation, near-real-time control loops operating under 10 ms, and predictive maintenance capabilities with over 90% accuracy in soft-failure identification have been validated across multiple testbeds. These results collectively demonstrate that the SEASON architecture delivers a sustainable, high-capacity optical network infrastructure aligned with the strategic objectives defined at project inception.

The Demo 2 field trial has validated SEASON's orchestration-driven optical–mobile convergence under realistic operational conditions. The core experiment confirmed that the system can coordinate PON lane activation, midhaul provisioning, and O-RAN DU scaling in real time while maintaining stable XR service delivery. The dynamic resource activation, deterministic optical operation, energy-aware scaling, and service continuity during reconfiguration observed throughout the trial provide concrete evidence that SEASON's control plane delivers the multi-domain coordination required by next-generation converged networks.

The three demo extensions have broadened the validation scope and confirmed SEASON's readiness for diverse deployment scenarios. Extension 1 validated cloud-native backend deployment on Kubernetes, demonstrating the platform's compatibility with modern service orchestration paradigms. Extension 2 proved that legacy PON and coherent P2MP technologies can coexist under unified SDN control, addressing practical brownfield migration requirements. Extension 3 showed that intent-based provisioning can reduce manual configuration efforts from hours to minutes, enabling scalable service lifecycle management across multi-vendor infrastructures.

Together with the results of Demo 1 reported in Deliverable D5.2 [D5.2-25], these outcomes provide comprehensive experimental validation of the SEASON architecture from backbone to access, covering both operator-centric resilience scenarios and user-centric service delivery use cases. The successful integration of data plane innovations from WP3 with the control plane components developed in WP4 confirms the maturity of the end-to-end solution and its potential for adoption in production network environments.

Appendix 1 further documents supplementary activities enabled through fund reallocation, including 5G infrastructure deployment, new AI/ML datasets, 6G scheduling algorithms, and

extended techno-economic analyses—demonstrating the consortium's commitment to maximizing SEASON's scientific and practical impact.

In conclusion, Deliverable D5.3 demonstrates that SEASON has met its core objectives: delivering a self-managed, sustainable, high-capacity optical network architecture capable of supporting advanced services—including bandwidth-intensive immersive applications—through intelligent, automated, and energy-efficient operations across converged optical and mobile domains.

8 APPENDIX 1

This section reports the additional activities carried out through the reallocation of available project funds.

8.1 DEDICATED OPEN 5G SNPN FOR SEASON DEMO AT HHI

Due to budget constraints, the initial project plan allowed for validating the 5G solution at only one of the two planned demonstration sites. Although the original intention was to perform this activity at the Berlin site (HHI), the consortium determined during project execution that the L'Aquila site provided a more suitable environment for demonstrating SEASON innovations from the end-user perspective. Consequently, the validation was carried out in L'Aquila, where the SEASON platform's ability to deliver advanced 5G services was thoroughly assessed. This work confirmed the successful integration of the transport network with disaggregated O-RAN functions and verified seamless interoperability between the transport SDN controllers, the O-RAN RIC, and the deployed xApps.

With the release of additional funds, the project was able to provide 5G equipment and to support integration and validation activities to the Berlin site, as originally planned. For this purpose, the required equipment was provided by Accelleran and deployed at HHI, including:

- 1 x Super micro server AS-1015A-MT
- 1 x E894772X5304649 Benetel 650 RU
- 2 x 6502100341 Omnidirectional Antennas
- 2 x SFP+ SM Fiber transceiver
- 1 x SM Fiber cable 3M
- 1 x SFP+ Ethernet 1/10G
- 1 x SMB to SMA connector
- 1 x DC Power cable 5m
- 1 x Power Supply Meanwell 10A
- 1 x OnePlus Nord CE 5G UE

The Network was deployed with the following the topology shown in Figure 81-1 below.

Fraunhofer HHI – 5G Network

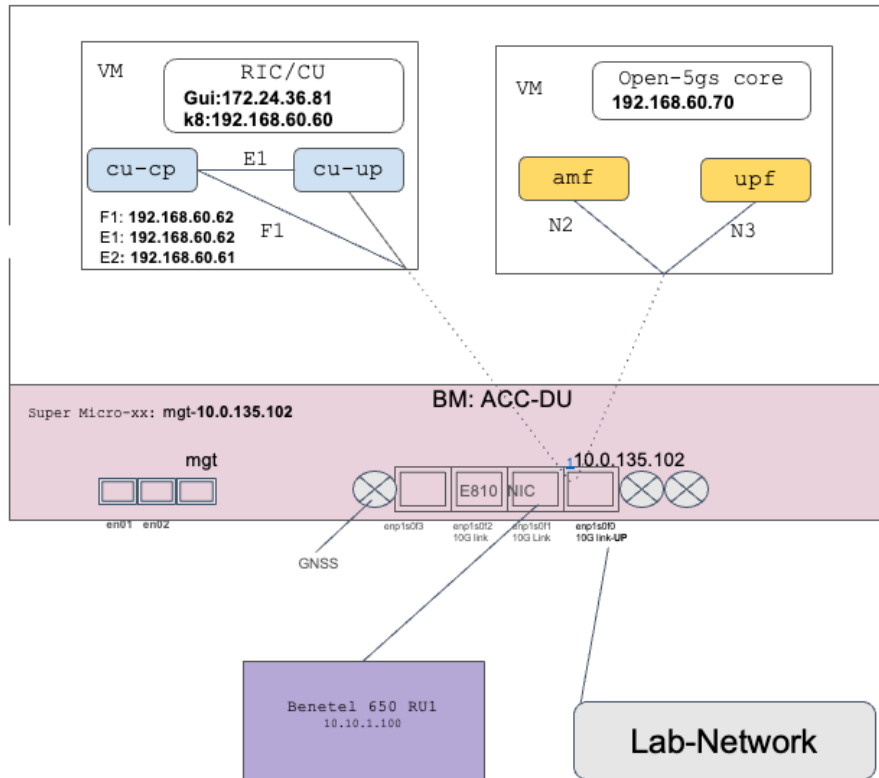


Figure 81-1: 5G Network Topology

The network was deployed using TDD configuration 5 (DSUDDDDDDD pattern). The performance results achieved are in line with expectations for that configuration i.e., ~280 Mbps in DL and 24 Mbps in UL. See Figures 81-2 and 81-3, below.

In addition to the equipment deployment, Accelleran provided dedicated technical support to HHI, assisting in the integration of the 5G network components into the existing HHI test setup. This support included guidance on configuring the O-RAN-compliant elements, aligning the SEASON platform with HHI's infrastructure, and ensuring smooth interworking with the 5G core network.

This expanded setup allowed the Berlin demonstration to focus on the operator perspective, particularly on integrating the SEASON solution with the 5G core network and evaluating how the platform supports operational workflows and network management functions.

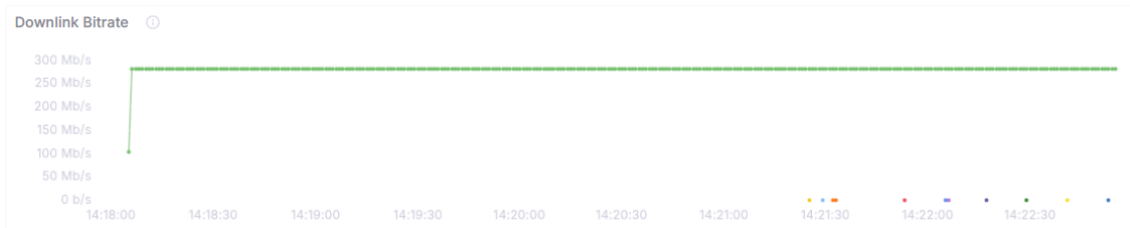


Figure 82-2: 5G Network Downlink Bitrate

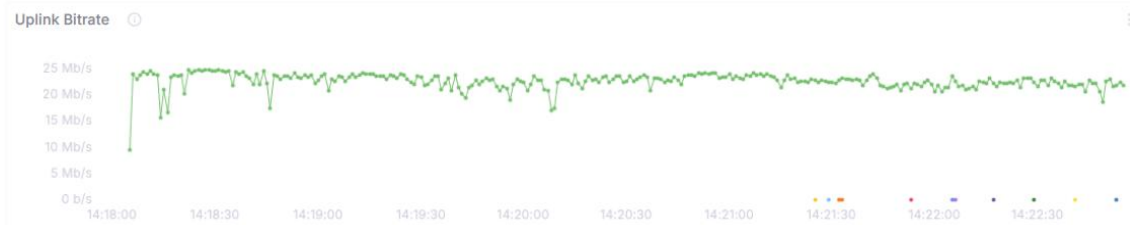


Figure 83-3: 5G Network Uplink Bitrate

Together, the L’Aquila and Berlin demonstrations delivered a comprehensive, end-to-end validation of the SEASON architecture, covering both user-centric and operator-centric viewpoints and providing a robust platform to assess performance, interoperability, and applicability across different deployment contexts.

8.2 MEASUREMENT FRAMEWORK (NEW DATA SET CREATION) AND OPERATIONAL ECONOMICS OF AI/ML-ENABLED ORCHESTRATION

The SEASON Demo 1 “Self-Healing in IPoWDM Network with Dual Protection Using MBoSDM and NetDevOps Solutions” [D5.2-25] targets a core operational problem in multi-domain service delivery: maintaining an application-level service objective while the underlying network and compute resources evolve due to failures, congestion, or planned changes. The value proposition is not a single measurement or a one-off lab run, but a repeatable method for turning an orchestration workflow into measurable evidence. Concretely, the approach couples orchestration actions with observable outcomes (what telemetry and application KPIs report), so that resilience mechanisms such as protection switching and rerouting can be validated, compared, and reasoned about in a way that is useful for engineering decisions.

In the remainder of this section, the results are presented in a layered manner: control-plane state changes provide the causal timeline, network QoS KPIs (*throughput_mbps* and *rtt_avg_ms*) provide the primary service-transport evidence, application QoE and energy metrics provide end-user and cost-related context. The accompanying dataset files and figures are referenced explicitly so that each claim can be traced back to a time-series column and, where applicable, to a phase of the orchestration workflow. This section summarizes a

streaming-service experiment aligned to the SEASON demo architecture (Service Orchestrator, Network Orchestrator, optical/IP controllers, and telemetry). The trace is designed to reflect the self-healing workflow described in D5.2: Failure-1 on the primary (red) path, rerouting via an alternate red path, Failure-2, switching to a green protection path, and rollback to the original path.

The architecture-to-metrics mapping is as follows. Service Orchestrator metrics represent per-receiver and per-stream throughput aggregation and reporting. Control-plane state represents the active end-to-end service path and its expected performance impact. End-to-end QoE proxies respond to network impairments and throughput drops. Finally, compute and energy metrics enable energy-per-bit and cost estimation; cumulative traffic counters provide delivered traffic volume for technoeconomic indicators. Note that the dataset is synthetically generated (model-based). It is intended for architecture-aligned analysis and documentation and should not be interpreted as a direct replacement for physical measurements from the full lab deployment.

8.2.1 Methodology overview

The dataset and analysis are produced using a repeatable, framework-based process intended to generalize beyond a single lab run. Scenarios are defined as structured combinations of a service topology, workload profile, and a sequence of control-plane events (for example, failure injection, reroute, protection switch, and rollback). Each scenario is described by explicit parameters such as target traffic per streamer, collection interval, warmup/cooldown, and a workflow timeline that captures the intended orchestration actions and their timing. This turns the demo narrative into an executable experiment specification and allows consistent comparison across scenarios by changing one factor at a time (for example, number of streamers, impairment level, or path choice) while keeping the rest fixed.

8.2.2 KPI identification

KPIs follow a layered approach that mirrors the SEASON architecture. At the service layer, KPIs capture user-perceived quality and delivered service capacity (for example, *throughput_mbps*). At the network layer, KPIs capture QoS and transport health (for example, RTT, jitter). At the control layer, KPIs capture orchestration state and decision outcomes (for example, active path ID and path multipliers that encode expected BW/RTT impact). At the infrastructure layer, KPIs capture resource and energy drivers (CPU, power). This layered structure makes it clear which subsystem a KPI belongs to, how it should be interpreted, and how it supports the demo claims (self-healing behavior and cost/energy relevance).

A short excerpt from metrics.csv is provided below for traceability and to help navigating the dataset:

timestamp_unix	application.active_streams	application.stream_bitrate_mbps	compute.receiver_cpu_percent	compute.streamer_cpu_percent	compute.total_power_watts	kubernetes.receiver_pod_count	kubernetes.streamer_pod_count	network.jitter_ms	network.receiver_throughput_mbps	network.rtt_avg_ms	telemetry.tellemetry_rx_power_dbm
...											
1765805616	30	1716.21	39.87	46.97	119.4	6	30	1.154	1911	2.058	2.943
1765805617	30	1804.2	38.6	47.78	120.5	6	30	0.495	1799.52	2.148	2.91
1765805618	30	1887.8	39.68	48.52	123	6	30	0.929	1691.16	2.139	3.066
1765805619	30	1830.6	38.74	47.11	120.3	6	30	0.731	1889.83	2.11	3.072
1765805620	30	2172.58	40.65	45.37	118.7	6	30	0.855	1893.35	2.081	2.874
1765805621	30	1875.94	41.15	45.39	122.8	6	30	0.813	1858.72	2.262	3.019
1765805622	30	1841.16	40.05	45.57	116.9	6	30	0.688	1903.74	2.358	3.076
1765805623	30	1948.91	37.4	45.99	118.9	6	30	0.291	1982.53	2.391	3.025
1765805624	30	1789.66	35.96	46.81	120.1	6	30	0.997	1919.4	2.261	2.944
1765805625	30	1864.24	36.33	46.72	121.1	6	30	0.462	2007.83	2.185	2.958
1765805626	30	1768.23	36.28	50.79	124.8	6	30	0.831	2012.83	2.157	2.892
1765805627	30	2109.5	37.81	46.91	122.5	6	30	0.811	1908.77	2.1	3.022
1765805628	30	1689.37	39.74	48.28	124.7	6	30	0.924	1933.96	2.157	3.077
1765805629	30	1897.39	38.37	48.02	124.9	6	30	0.753	1894.48	2.061	3.063
1765805630	30	1721.28	38.53	47.12	122.7	6	30	1.059	1944.91	2.012	3.036
1765805631	30	1716.54	38.15	48.7	125.9	6	30	1.014	1822.07	1.993	3.009
1765805632	30	1930.62	40.26	50.27	126.4	6	30	0.256	1906.77	2.089	3.018
1765805633	30	1852.42	39.88	50.16	116.6	6	30	1.069	1862.03	2.245	3.217
1765805634	30	2101.1	40.66	48.34	126.2	6	30	0.761	1819.2	2.125	2.866
...											

Probes and metric sources are represented via a collector's abstraction to keep the dataset schema consistent across synthetic and real deployments. In a real deployment, the kubernetes collector corresponds to Kubernetes API and cluster state queries (pods, HPA, rollout/pending pods), the compute collector corresponds to node or container resource probes (CPU, memory, power estimates), the network collector corresponds to host or interface counters and active probing (bytes, packets, retransmits), the network collector corresponds to active measurements (ping and iperf), the service orchestrator collector corresponds to application-facing KPIs emitted by the Service Orchestrator (receiver_stats and bw_update/net_update events), and the telemetry collector corresponds to optical-layer telemetry (OSNR, Rx power, BER). In the present dataset, some of these sources are model-based proxies generated by the synthetic model but they preserve the same metric names and relationships, enabling a later swap-in of real probes without changing downstream analysis scripts and deliverable figures.

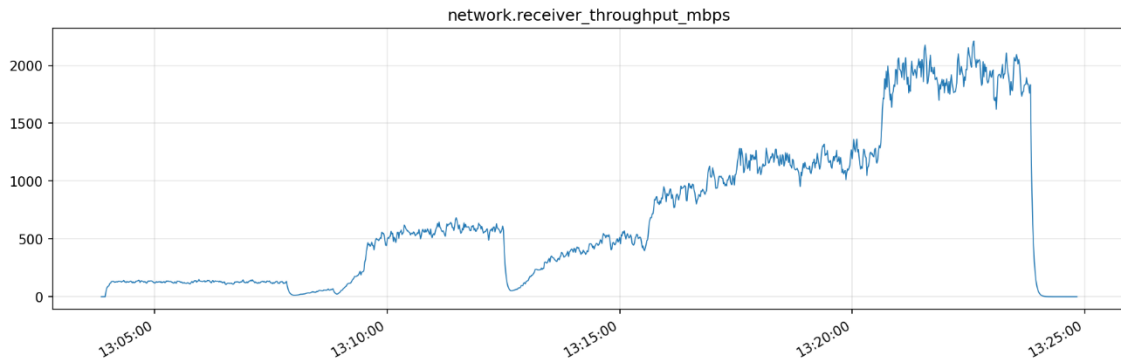


Figure 8-4: Receiver throughput (Mbps) during self-healing (failure → reroute/protection switch → recovery).

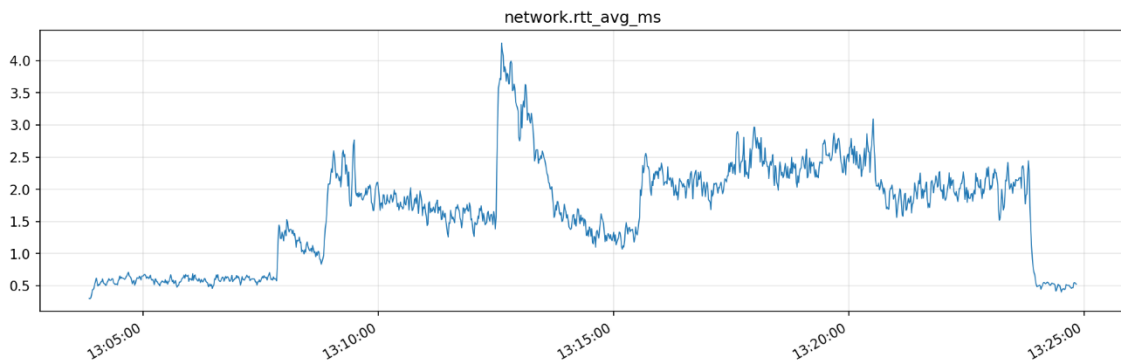


Figure 8-5: End-to-End RTT (ms) during self-healing (failure → reroute/protection switch → recovery).

Overall results across the whole run show mean throughput of 713.8 Mbps and mean RTT of 1.672 ms as depicted in Figure 8-4 and Figure 8-5, and mean total power of 87.73 W. In the optical_network profile, each streamer is streaming at approximately 64 Mbps, so 30 streamers produce approximately 1920 Mbps when the path multipliers and impairments are favorable. Degradation events and reroutes reduce throughput and can impact QoE. In practical terms, the self-healing behavior can be supported by demonstrating that the control-plane state transitions align with the documented orchestration steps, that network QoS and optical telemetry temporarily deteriorate during failure intervals, and that throughput and QoE recover after reroute or protection switching.

The generated dataset is available via the project's SharePoint repository and will also be archived on Zenodo for longer-term access and citation.

8.2.3 Additional techno-economic assessment of AI/ML-enabled Service and Application-driven Orchestration

Following the measurement and dataset creation framework presented earlier, this section presents an additional techno-economic assessment focused on the operational and lifecycle economics of AI/ML-enabled service and application-component orchestration [ETSI19]. In

contrast to infrastructure-centric techno-economic analyses, which primarily address deployment cost and capacity dimensioning, the assessment here targets the runtime cost dynamics introduced once network and compute infrastructures are already in place. Such dynamics are increasingly recognized as a dominant cost factor in softwarised and cloud-native network environments, where operational complexity and service agility directly influence total cost of ownership and sustainability. The objective is to quantify how increasing levels of orchestration intelligence affect operational effort, resource utilization, energy consumption, and service agility over time. Rather than deriving absolute CAPEX or OPEX values, the analysis adopts a comparative and mechanism-driven approach, consistent with established techno-economic practices for network automation and management systems. This approach highlights the structural economic effects that emerge when static or semi-automated orchestration practices are replaced by telemetry-aware, automated workflows, as advocated in recent frameworks for autonomous and zero-touch network operation.

The assessment considers two orchestration regimes. The first corresponds to conventional practices, where service lifecycle management relies on centralized control, manual coordination across layers, and conservative provisioning assumptions. The second reflects an AI/ML-enabled orchestration regime, where service and application components are instantiated, scaled, adapted, and terminated through automated workflows driven by real-time telemetry and policy-based decision logic. In both cases, the underlying infrastructure and service demand are assumed to be comparable; the differentiating factor is the degree of orchestration intelligence applied at runtime.

One of the most immediate techno-economic effects appears at the level of operational effort. In conventional environments, service lifecycle events typically involve a sequence of manual or semi-manual actions across application, compute, and network layers. These actions include configuration, validation, monitoring, and corrective intervention, often distributed across different operational tools and teams. As service complexity and service churn increase, the cumulative operational effort grows rapidly, and the likelihood of configuration inconsistencies rises accordingly. AI/ML-enabled orchestration changes this dynamic by consolidating these actions into executable workflows that can be triggered, supervised, and adapted automatically. To capture this effect, the analysis models operational effort as a function of the number of service lifecycle events and the average number of manual actions required per event. Automation reduces both the action count and the frequency of anomaly-driven interventions, leading to a substantial reduction in recurring operational effort and improved repeatability of service management processes.

A second, closely related dimension concerns resource utilization and elasticity. Static orchestration practices commonly treat peak demand as the dominant design point, resulting in persistent over-provisioning of compute and network resources even during extended low-load periods. Elastic orchestration enables resources to follow demand more closely by activating application components and supporting infrastructure only when required and releasing them when demand subsides. While this does not remove the need to provision for peak demand, it

significantly improves average utilization over time and delays infrastructure expansion decisions. From a techno-economic perspective, this translates into a lower effective cost per delivered service unit and a higher return on already deployed assets.

Energy consumption emerges as a further impact dimension that is tightly coupled to orchestration behavior. When service components and network resources are dynamically activated and deactivated based on real-time conditions, unnecessary energy expenditure during low-load periods is avoided. Importantly, these gains do not rely on aggressive or risk-prone power management strategies, but arise naturally from tighter coupling between service demand and resource activation. Although the absolute magnitude of energy savings depends on traffic patterns and service mix, even moderate reductions at the service level accumulate into meaningful energy-related OPEX reductions when considered across large infrastructures and long operational horizons.

Service agility completes the techno-economic picture. Faster service instantiation and adaptation shorten the delay between demand and availability, improving time-to-revenue and reducing opportunity costs associated with slow provisioning processes. While agility is often discussed primarily as a technical performance indicator, its economic relevance becomes evident in environments where services are short-lived, highly dynamic, or demand-driven. In such cases, the ability to launch, modify, or withdraw services within minutes rather than hours or days has a direct impact on revenue capture and operational flexibility.

To make these effects more concrete, Table 8-1 summarizes representative techno-economic indicators comparing conventional orchestration practices with AI/ML-enabled orchestration. The values are indicative and reflect conservative, order-of-magnitude estimates derived from typical operational environments and observed orchestration behavior. Rather than providing absolute cost figures, the table highlights the relative magnitude and consistency of the improvements across different impact dimensions.

Table 8-1: Representative techno-economic indications comparing conventional with AI/ML-enabled orchestration.

Dimension	Conventional orchestration	AI/ML-enabled orchestration	Indicative impact
Service instantiation time	30–90 minutes	<3 minutes	Strong reduction (≈95%)
Manual actions per service lifecycle	8–15	1–2	Significant OPEX decrease
Average compute utilization	30–40%	60–75%	~2× improvement
Resource over-provisioning margin	30–50%	10–20%	Noticeably reduced
Service-related energy consumption	Baseline (100%)	70–85%	15–30% reduction
Time-to-revenue for new services	Hours or days	Minutes	Strongly improved

To further illustrate the structural effect of orchestration intelligence, Figure 8-6 conceptually depicts the relationship between service dynamics and operational effort. While conventional orchestration exhibits near-linear growth in operational effort as service dynamics increase, AI/ML-enabled orchestration significantly flattens this curve by automating lifecycle management and reducing manual intervention. The figure emphasizes relative scaling behavior rather than absolute cost values.

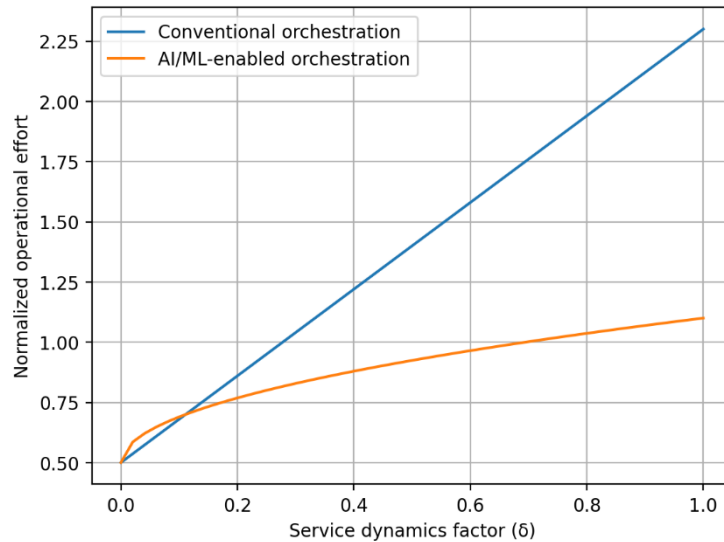


Figure 8-6: Conceptual illustration of operational effort as a function of service dynamics for conventional and AI/ML-enabled orchestration. Intelligent orchestration flattens the operational effort growth by automating service lifecycle management and reducing manual intervention, particularly under highly dynamic service conditions.

Beyond individual metrics, the most significant economic effect emerges at the level of operational scalability. As service portfolios become more diverse and infrastructure more heterogeneous, conventional orchestration approaches exhibit near-linear growth in operational effort. AI/ML-enabled orchestration breaks this coupling by relying on automation, telemetry, and closed-loop control, allowing service growth without proportional increases in operational resources.

Overall, this assessment shows that AI/ML-enabled service and application-component orchestration introduces a structural shift in network economics—from manual, labor-intensive operation toward software-driven efficiency and elasticity. These effects complement infrastructure-focused techno-economic analyses by addressing a different, but increasingly dominant, cost component of future networks: the economics of operation, adaptation, and service agility.

8.3 SCHEDULING OF MIXED 6G TRAFFIC IN TDM-BASED HORSE-SHOE OPTICAL METRO-ACCESS WITH BURST-MODE TRANSCEIVERS

To reduce cost and improve bandwidth efficiency, the radio access network (RAN) has been evolving toward more flexible and disaggregated architectures. This evolution is mainly enabled by the functional split options defined by the 3rd Generation Partnership Project (3GPP), which allow a 5G base station to be decomposed into the Radio Unit (RU), Distributed Unit (DU), and Central Unit (CU). The protocol stack can be partitioned among these entities to meet different requirements and constraints of operators and applications. In this architecture, the interface between the RU and DU is referred to as the mobile fronthaul (FH). For services with stringent latency requirements, such as ultra-reliable and low-latency communication (URLLC) services, adopting lower-layer functional splits (e.g., 3GPP Option 7-2x) offers a balance between FH bandwidth consumption and processing complexity. Lower-layer splits transfer more physical-layer processing to the DU, which reduces the processing load at the RU and enables tighter control of FH transmission delay.

Passive optical network (PON) has become a promising solution for supporting the mobile FH of 6G RAN due to its point-to-multipoint (P2MP) topology. However, traditional dynamic bandwidth allocation (DBA) mechanism in PON introduced non-negligible delay, as the optical line terminal (OLT) must wait for buffer reports from optical network units (ONU) before assigning upstream bandwidth. To address this, a cooperative dynamic bandwidth allocation (Co-DBA) method has been proposed for FH over PON. In Co-DBA, the DU informs the OLT of the uplink scheduling results before the corresponding FH packets reach the ONU. With this advance information, the OLT can pre-allocate upstream transmission windows and thus avoid the grant-report delay. To enable such coordination, ITU-T and 3GPP have standardized the cooperative transport interface (CTI), which provides a protocol for exchanging scheduling information between the RAN and the PON systems. On this basis, several enhanced Co-DBA algorithms have been developed that work for 5G but not for 6G FH, as the latency requirement should be as low as 100 μ s, which is much lower than that in 5G, i.e., 1ms. In addition, traffic with different characteristics, e.g., traffic from applications based on Artificial Intelligence (AI), which require large capacity during some defined times, need to be supported on the same infrastructure.

Therefore, the objective of the work is to study metro-access network architectures and technologies able to serve such a mix of traffic with different characteristics. To that end, we have considered horseshoe topologies together with point-to-multipoint communication (P2MP) based on digital subcarrier multiplexing (DSCM) transceivers operated using time division multiplexing (TDM). Specifically, in this exploratory work, we have focused on scheduling traffic between an access location and the metro location (see Figure 8-7).

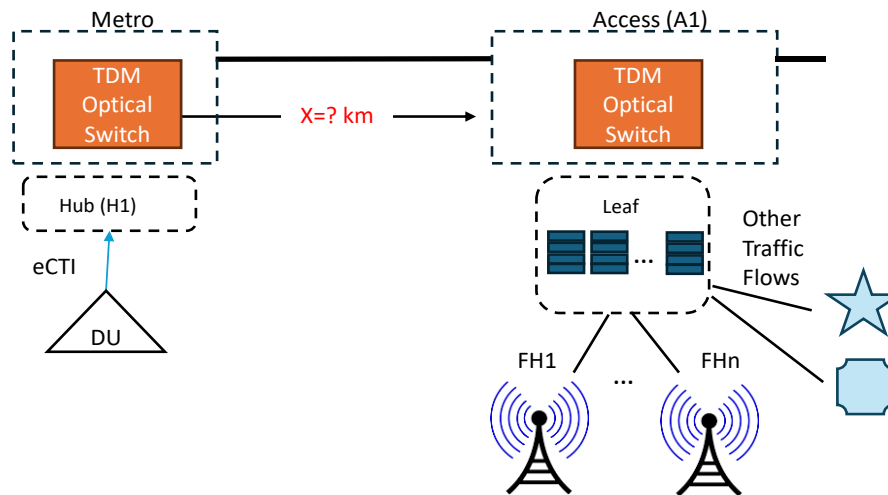


Figure 8-7: Considered metro-access horseshoe architecture.

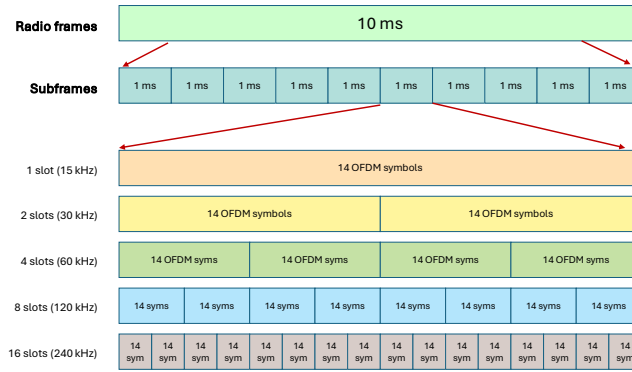
We assume that the DU can predict the FH traffic and inform the scheduler running at the hub (H1) about the arrival of bursts for different FHs via the extended CTI (*eCTI*) interface. At the access node (A1), a dedicated queue is allocated for every incoming flow, so packets are stored until they are transmitted towards the hub, based on a granted timeslot in the scheduling plan, which is managed by the scheduler running at the hub. The scheduling plan is distributed periodically to all the access nodes for them to transmit packets of the specified flows during every time slot.

In this work, we are interested in investigating how the latency and the number of FH flows relates, as well as the maximum distance between the access and the metro that can be supported (denoted as X in Figure 8-7).

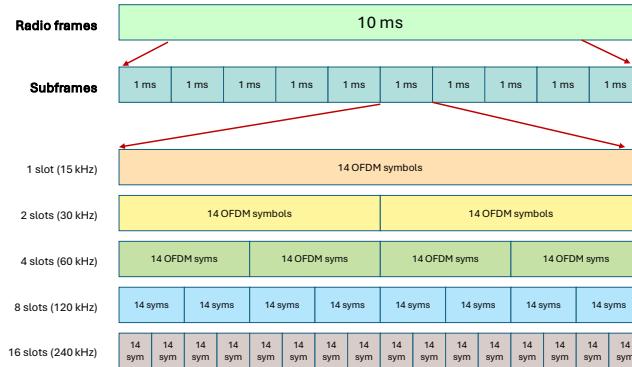
8.3.1 Background on 6G radio and scheduling approach

6G frame, subframe, slot, and orthogonal frequency division multiplexing (OFDM) symbol duration is shown in Figure 8-8 for different subcarrier spacing (SCS) scenarios, 15 kHz, 30 kHz, 60 kHz, and so on. Different numerologies are also shown. The structure and the PRBs are shown in Figure 8-9. PRB is defined as 12 consecutive frequency subcarriers in one OFDM symbol. This is the minimum block that can be allocated to one UE. When the DU allocates PRBs, the RU determine the IQ stream in the PRBs and transmit them to the DU. So, as soon as DU transmit the aggregated number of Bytes to the H1 scheduler for all the PRBs in one OFDM symbol, the scheduler can allocate resources to that flow stream. The allocation to a UE can be spread over 2, 4, 7, and 14 (one slot) symbols, starting from whatever OFDM symbol index. So, the minimum allocation is 2 symbols, which is called *mini-slot*.

NR Radio Frame, Subframe, and Slot Structure



NR Radio Frame, Subframe, and Slot Structure



NR Radio Frame, Subframe, and Slot Structure

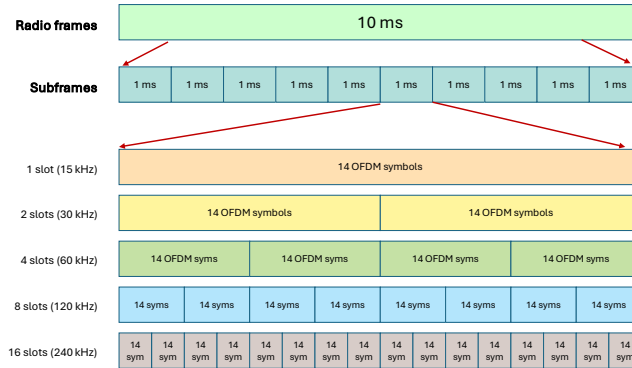


Figure 8-8: 6G frame, subframe, slot, and OFDM symbol duration.

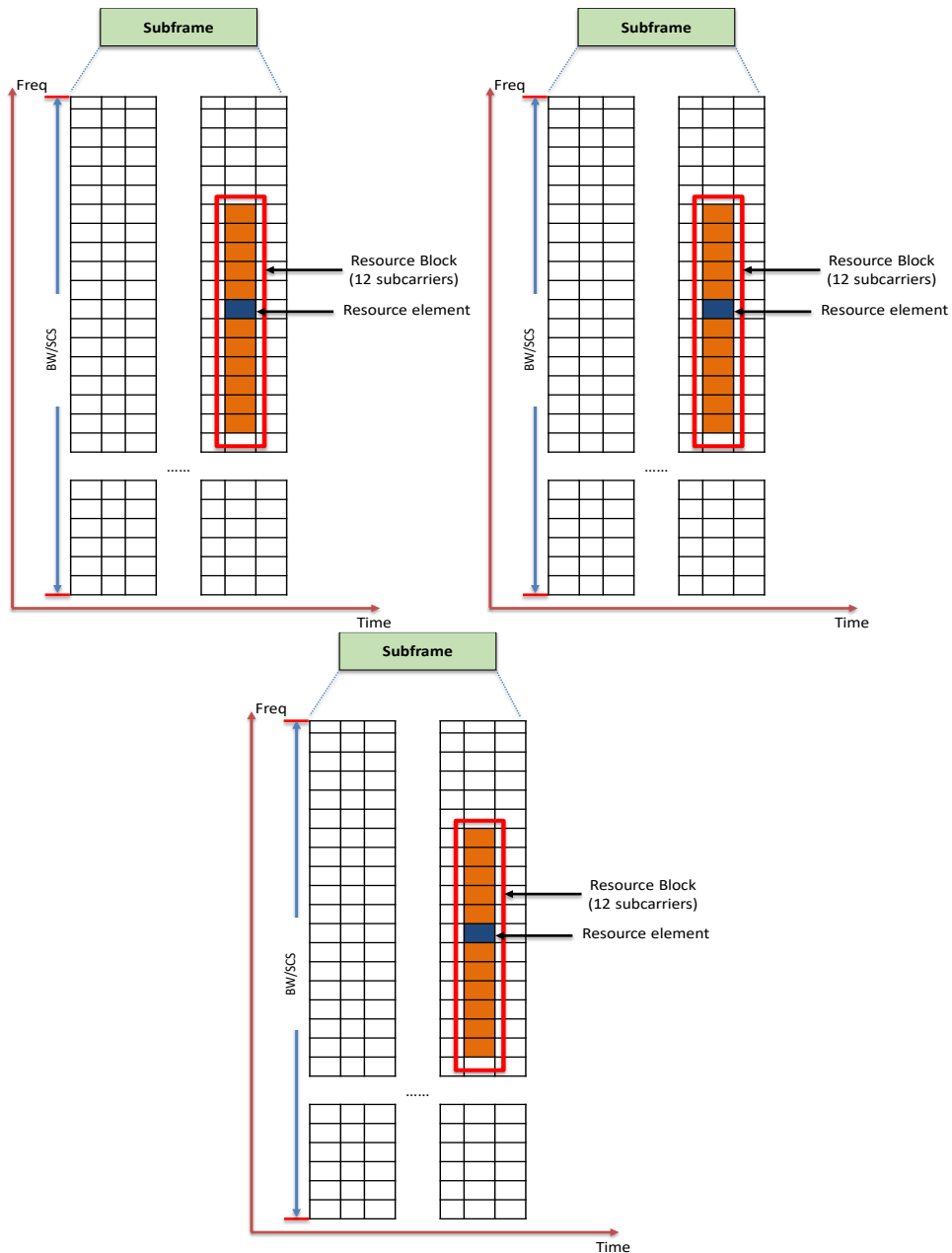


Figure 8-9: Frame structure and PRBs in a subframe.

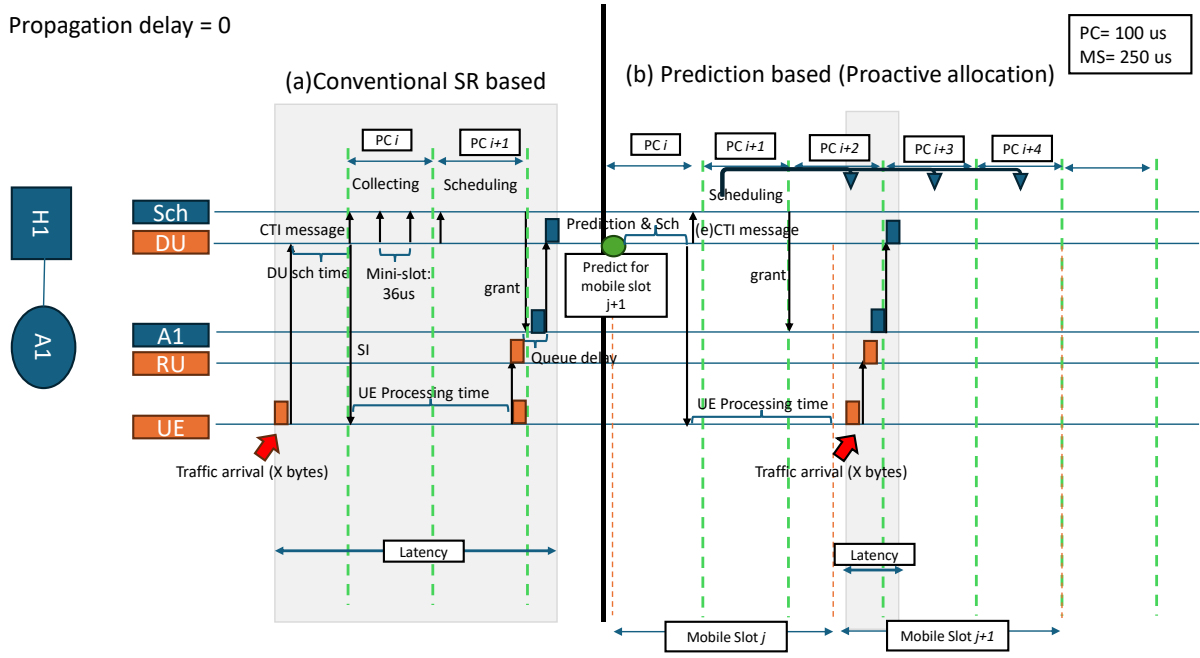
To reduce the latency of the FH flows, we need some kind of prediction to estimate the arrival and size of the packet bursts. To this end, we assume that the DU performs such prediction and informs the scheduler at H1 about the aggregated amount of traffic with estimated arrival time to properly allocate resources in the access-metro network before their arrival.

Figure 8-10 shows the effectiveness of proactive scheduling in terms of reducing latency. We assumed that the mobile slot that DU uses for scheduling, is $MS=250\mu s$ and the polling cycle of the scheduler at H1 is $PC=100\mu s$. The prediction which is performed at DU is done for the next MS and DU will inform UE about the granted physical resource blocks (PRBs) for that UE, then UE requires some time to understand the message from DU. Scheduler at H1, collects all the

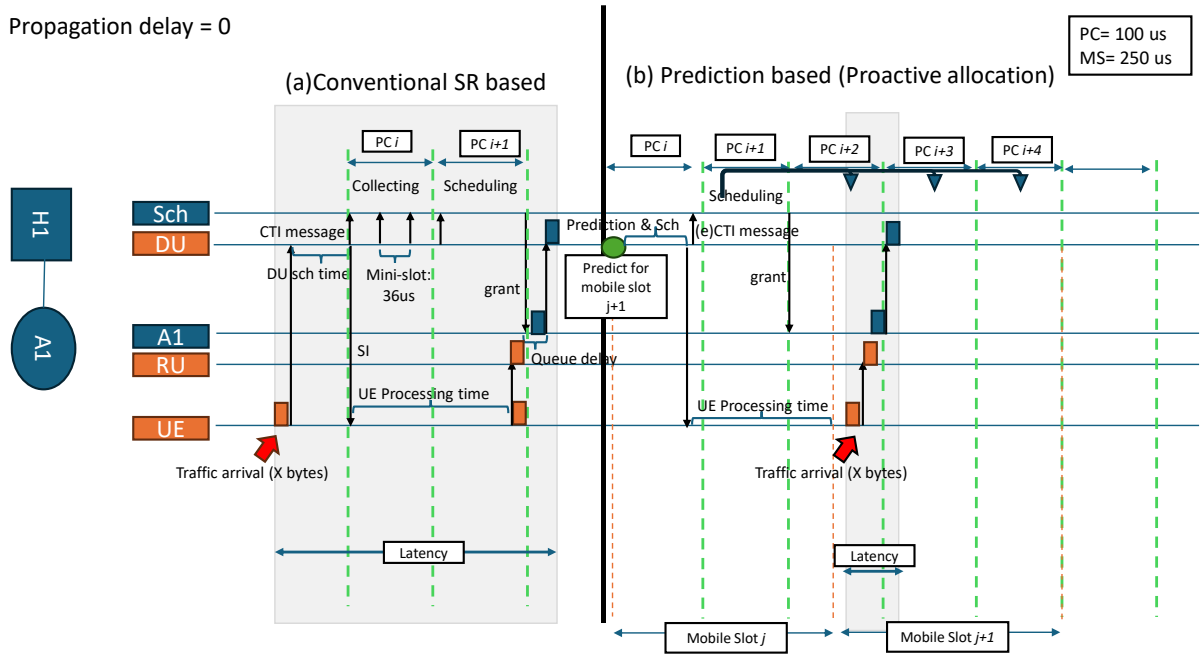
scheduling request from DU which are based on the prediction and performs scheduling during PC $i+1$ and informs A1 about the scheduling plan.

As can be inferred from Figure 8-10, when we use conventional scheduling request (SR)-based resource allocation at DU and then informing scheduler at H1 about the decision, the delay that the arrived packet experiences, is too high. Once we use prediction-based allocation (proactive allocation), the latency is reduced notably.

Propagation delay = 0



Propagation delay = 0



Propagation delay = 0

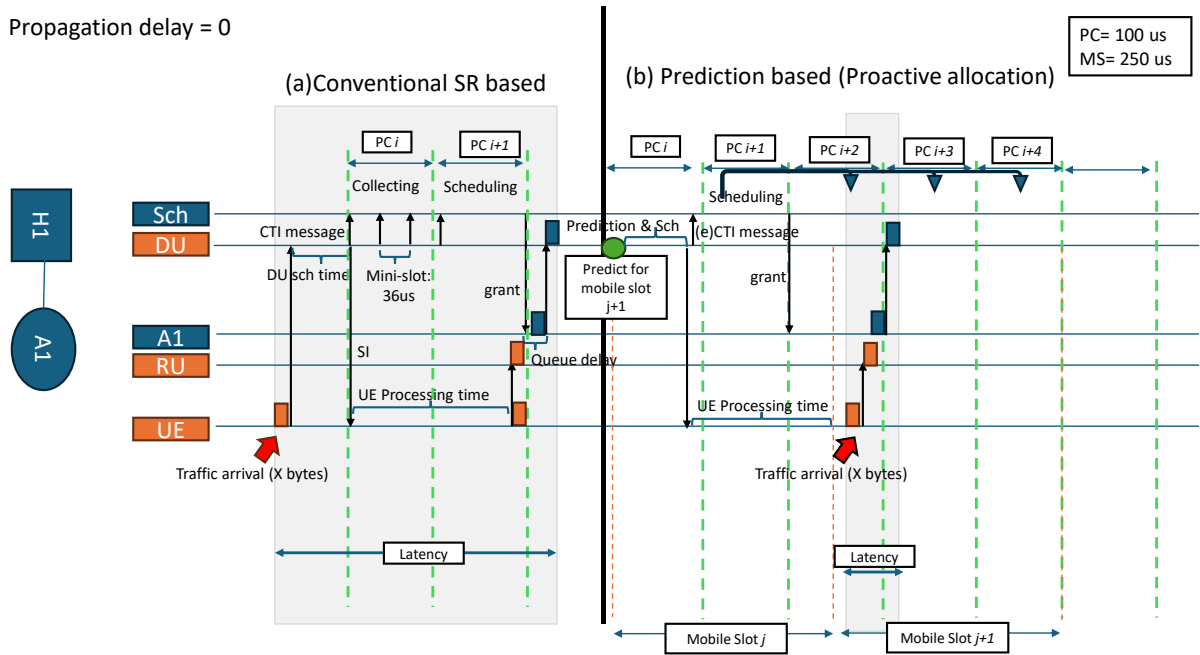


Figure 8-10: Comparing SR-based scheduling (a) and proactive allocation (b).

Error in the synchronization between the RU/DU and the transport may introduce additional delay and mismatch between allocated resources and requested one if we do not deal with it. The effect of synchronization error is shown in Figure 8-11. Let us assume some synchronization mismatch. In this case, since the RU is not able to deliver the packets on time, it may happen that the granted window for the burst can serve only some of the packets in the burst. For example, the last packet(s) will be served in the next granted window which causes huge delay. We observe the effect of synchronization error on the latency of the packets.

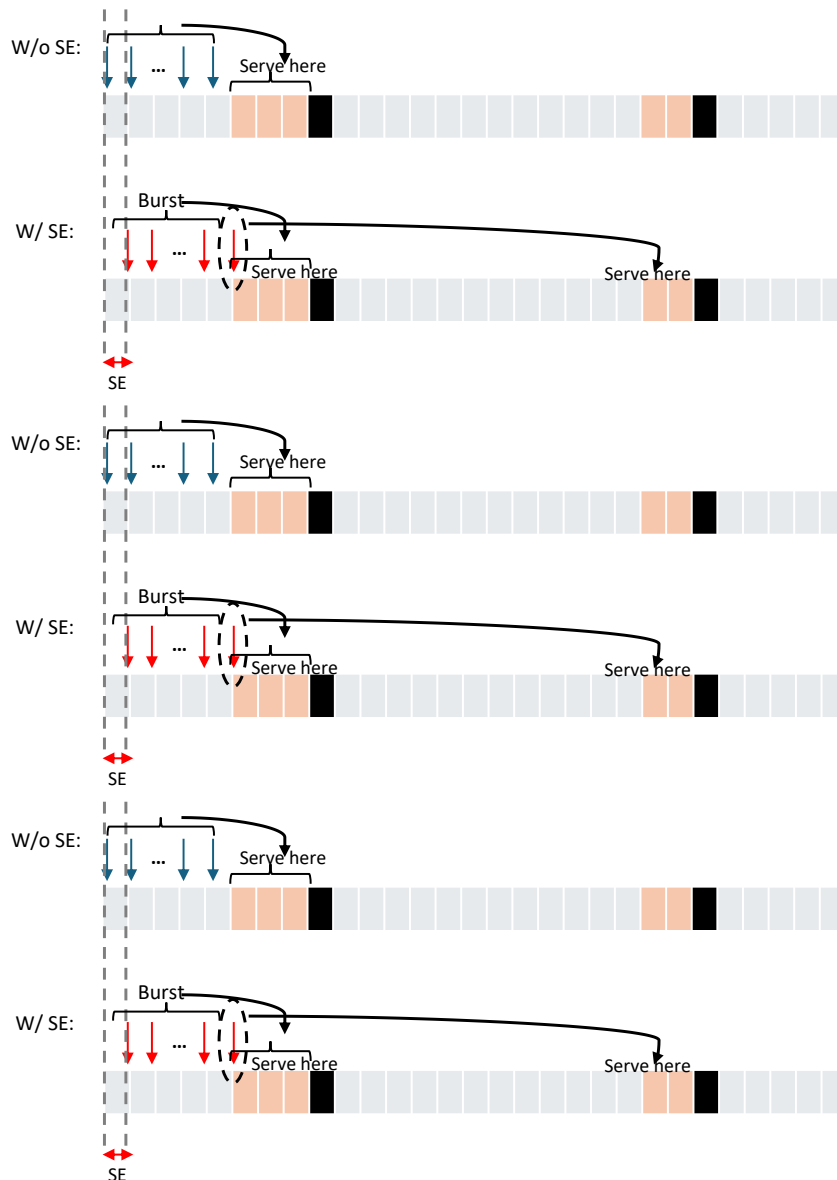


Figure 8-11: Effect of Synchronization mismatch.

8.3.2 Proposed ILP & Heuristic Scheduling Algorithms

In this section, we are going to first, state the problem, formulate the Integer Linear Programming (ILP) model to solve the problem and then, we introduce the proposed scheduling algorithms for solving the problem.

Problem statement

Given:

- A directed optical bus connecting several nodes. The nodes use burst-mode optical transceivers of the same speed B .

- The optical transmission is cyclic with period T_{PC} and divided in time slots of granularity τ .
- A continuous set of time slots can be allocated for a single flow. A guard band of duration G is required between time allocations to avoid collisions.
- Set of flows F : each flow f defined by the source and destination nodes, length of the packets, maximum delay and jitter, priority, i.e., $f = \langle src_f, dest_f, l_f, \delta_f, u_f, r_f \rangle$.

Constraints: each time slot can be allocated to only one flow, and the number of allocated time slots for every flow should be enough to transport the (predicted) traffic during each time interval and has to include the guard band.

Objective: is to allocate resources to the flows considering the delay and jitter requirements for the flow.

ILP formulation

The proposed scheduling ILP model is formulated as follows. It should be noted that since the scheduling ILP has to be solved for every polling cycle, we need some information from previous polling cycles.

Parameters:

- B_{ck} : list of all buckets to be scheduled. Each bucket is defined as $b = (t_{ini}, t_{end}, reqTS)$ where .. is the index of timeslot of the first packet of the bucket, ... is the index of the last packet of the bucket, and the required number of timeslots (see Eq. (13)).
- F : set of flows.
- T : set of timeslots ($t \in [0, 1, \dots, (2 * T_{PC} / \tau) - 1]$) in the polling cycle with the granularity τ .
- h : this is the plan that comes from previous scheduling; since in the scheduling plan in each cycle, we are considering double duration than T_{PC} and pass the second half of the plan to the next scheduling cycle. The size of h is T_{PC} / τ and each $h[t] = 1$ if timeslot t is allocated to one bucket in previous cycle.
- $D_f^{min,prev}$: minimum delay for flow f so far; it is very large value if we do not have history for the flow.
- $D_f^{max,prev}$: maximum delay for flow f so far; it is zero if we do not have history for the flow.
- a_b^f : binary parameter; is one when bucket b is related to flow f ; otherwise, zero.
- d_{prop}^f : the propagation delay for flow f in terms of number of timeslots.
- α : weight for objective function.
- β : weight for objective function.

Decision variables:

- x_t^b : binary; is 1 if timeslot " t " is allocated to bucket " b ".
- z_t^b : binary; is 1 if timeslot " t " is allocated to bucket " b " as first slot.

- d_f^b : integer; the delay for bucket “ b ” from flow “ f ” in terms of number of timeslots.
- d_f^{max} : integer; this is the maximum value among delay for different buckets of flow “ f ” and also $D_f^{max,prev}$.
- d_f^{min} : integer; this is the minimum value among delay for different buckets of flow “ f ” and also $D_f^{min,prev}$.
- J_f : integer; jitter of flow “ f ”.
- D_{cost}^f : real value (i.e., $\in \mathbb{R}$); exceeded delay in this polling cycle for flow f .
- J_{cost}^f : real value (i.e., $\in \mathbb{R}$); exceeded jitter in this polling cycle for flow f .

Objective: The objective is to minimize the exceeded delay and exceeded jitter for the flows in F . It can be also defined as minimizing the number of flows that exceeded jitter or delay, with small modification.

$$\text{minimize } \alpha \sum_{f \in F} D_{cost}^f + \beta \sum_{f \in F} J_{cost}^f \quad (1)$$

Constraints:

Constraints (2)-(6) satisfy the allocation of timeslots to each bucket. Constraint (2) ensures that the allocated timeslots to each bucket is exactly equals to the required number of timeslots for that bucket. Constraint (3) ensures that for each bucket we allocate only one timeslot as the first allocated timeslot while constraint (4) ensures that the allocated timeslot to each bucket, are contiguous. Constraint (5) states that the first allocated timeslot should be after receiving all the packets in the bucket. Constraint (6) ensures that one timeslot can be allocated to at most one bucket.

$$\sum_{t \in T} x_t^b = b.reqTS \quad \forall b \in B_{ck} \quad (2)$$

$$\sum_{t \in T} z_t^b = 1 \quad \forall b \in B_{ck} \quad (3)$$

$$\sum_{t \in [t', t' + b.reqTS)} x_t^b \geq z_{t'}^b \times b.reqTS \quad \forall t' \in T, \forall b \in B_{ck} \quad (4)$$

$$b.t_{end} < \sum_{t \in T} t \times z_t^b \quad \forall b \in B_{ck} \quad (5)$$

$$\sum_{b \in B_{ck}} x_t^b + h \leq 1 \quad \forall t \in T \quad (6)$$

Constraints (7)-(9) compute the maximum delay that one flow experiences. Constraint (7) computes the delay for each bucket from one flow by comparing the time of the first allocated timeslot and the initial timeslot of the bucket (i.e., where the first packet of the bucket has arrived) and also considers the propagation delay. Constraint (8) and (9) compute the maximum and the minimum delay a flow experiences, respectively, considering the history of the flow.

$$d_f^b = \left(\sum_{t \in T} t \times z_t^b - b.t_{ini} + d_{prop}^f \right) \times a_f^b \quad \forall b \in B_{ck}, f \in F \quad (7)$$

$$d_f^{max} \geq \max(d_f^b, D_f^{max,prev}) \quad \forall f \in F, \forall b \in B_{ck} \quad (8)$$

$$a_f^b \times d_f^{min} \leq \min(d_f^b, D_f^{min,prev}) \quad \forall f \in F, \forall b \in B_{ck} \quad (9)$$

Constraints (10) computes the jitter as the difference between the maximum delay and the minimum delay that a flow experiences.

$$J_f = d_f^{max} - d_f^{min} \quad (10)$$

Constraints (11) and (12) compute the exceeded delay and exceeded jitter, respectively. The objective function is defined based on these two variables.

$$D_{cost}^f \geq \max(\tau \times d_f^b - f.\delta, 0) \quad \forall f \in F \quad (11)$$

$$J_{cost}^f \geq \max(\tau \times J_f - f.v, 0) \quad \forall f \in F \quad (12)$$

Proposed algorithm

In this section we dive into the proposed method and algorithms. Algorithm 8-1, defines a scheduling plan as the allocation of flows to time slots. The duration of scheduling plan equals to the double of the polling cycle (T_{PC}) and every polling cycle and it receives the current plan and extends that plan to be double than T_{PC} and starts finding available contiguous time slots to be allocated to each flow. First, it sorts the flows by their priorities and then for each flow, it gets the prediction in format of buckets where a bucket is defined as initial timestamp, finish timestamp and number of bytes (i.e., bucket size). The algorithm first tries to find available time slots to serve requested traffic for each bucket within half of the delay budget δ_f to keep the rest for the propagation delay; if the algorithm can find the requested number time slots, then it continues to the next bucket, but it fails to find, it tries to make space for the flow by moving previously allocated buckets if it is possible. Finally, if the algorithm fails again to find requested resources, it extends the allocation window to consider also the other half of the delay budget. The number of required time slots can be computes as (13).

$$reqTS = \left\lceil \frac{8 \times bucket_size}{B \times \tau} + \frac{G}{\tau} \right\rceil. \quad (13)$$

Algorithm 8-1: Scheduling algorithm.

INPUT: $F, B, T, \tau, current_Plan$ **OUTPUT:** Plan

- 1: $n_TS \leftarrow T / \tau$
- 2: $plan \leftarrow [current_Plan \ll n_TS, [0, 0, \dots, 0]_{1 \times n_TS}]$
- 3: Sort F by $f.r$ in DESC order
- 4: **for each** f in F **do**
- 5: $Buckets \leftarrow f.getPrediction(f.\delta, \tau, f.P, f.g.BufferSize)$
- 6: **for** $bucket$ in $Buckets$ **do**
- 7: $reqTS \leftarrow computeReqTS(f)$ (Eq.(13))
- 8: $allocationFound \leftarrow False$
- 9: $j \leftarrow 0$
- 10: **while NOT** $allocationFound$:
- 11: $allocationWindow \leftarrow \langle bucket.ts_end + j * f.\delta/2, bucket.ts_end + (j+1) * f.\delta/2 \rangle$
- 12: **for** t in $allocationWindow$ **do**
- 13: **if NOT** $plan(t).isAllocated()$ **AND** $freeTS(plan, t, reqTS)$ **then**
- 14: $plan.allocate(t, reqTS, f)$
- 15: $allocationFound \leftarrow True$
- 16: **break**
- 17: **if NOT** $allocationFound$ **then**
- 18: $C \leftarrow Reallocate(plan, f, allocationWindow, reqTS)$
- 19: $C_{best} \leftarrow FindCandidate(C, reqTS)$
- 20: **if** C_{best} **then**
- 21: $C_{best}.f'.move(plan, C_{best}.s')$
- 22: $plan.allocate(t, reqTS, f)$
- 23: $allocationFound \leftarrow True$
- 24: $j \leftarrow j+1$
- 25: **return** Plan

Following, you can find auxiliary algorithms that are used in the scheduling algorithm. Algorithm 8-2 returns the candidates for reallocation. It swipes the time slot in the allocation window and gets the first flow that is allocated in the considered time slots then tries to see if that flow can move (Algorithm 8-3) to any other time slot set considering that this movement makes the required space for allocating to the flow under allocation.

Algorithm 8-2: Reallocation algorithm.

INPUT: $plan, f, allocationWindow, reqTS$ **OUTPUT:** C

- 1: $C \leftarrow \{\}$
- 2: **for** t in $allocationWindow$ **do**
- 3: **if** $t + reqTS$ **NOT in** $allocationWindow$ **then break**
- 4: $s \leftarrow \langle t, t+reqTS \rangle$
- 5: $f' \leftarrow getFirstFlow(plan, s)$
- 6: $move_cost, overlap, s' \leftarrow f'.canMove(plan, f, s)$
- 7: **if** $move_cost < \infty$ **then** $C \leftarrow C \cup \{ \langle f', move_cost, s, s', freeTSSize(s, overlap) \rangle \}$
- 8: **return** C

Algorithm 8-3: CanMove.

```

INPUT: plan, f, f', s      OUTPUT: move_cost, overlap, s'
1:  allocationWindow ← plan.findAllocationWindow(f', ts)
2:  overlap ← ∞
3:  move_cost ← ∞
4:  s' ← None
5:  for t in allocationWindow do
6:    if NOT plan(t).isAllocated() AND freeTS(plan, t, f'.reqTS) then
7:      local_s' ← <t, t+f'.reqTS >
8:      local_overlap ← computeOverlap(local_s', s)
9:      cost ← getCost(local_s', <f'.getFirstTS(), f'.getLastTS()>)
10:     if local_overlap < overlap AND (NOT f'.exceedMaxDelay(local_s') OR f'.r > f'.r) then
11:       overlap ← local_overlap
12:       move_cost ← cost
13:       s' ← local_s'
14:       if overlap = 0 then
15:         return move_cost, overlap, s'
16:   return move_cost, overlap, s'

```

Simulation Results

Let us discuss the simulation scenarios and the results obtained. Table 8-2 shows the simulation parameters and their values.

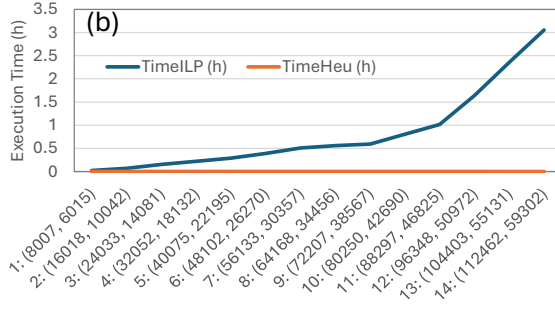
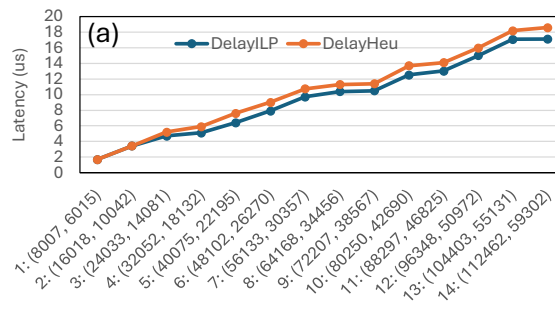
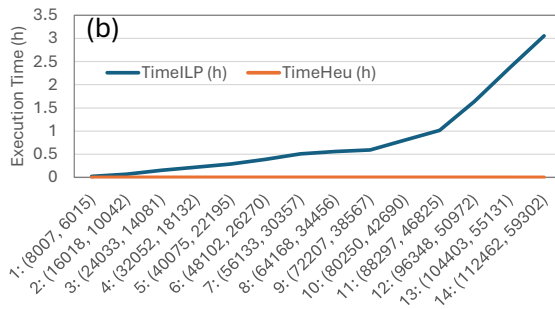
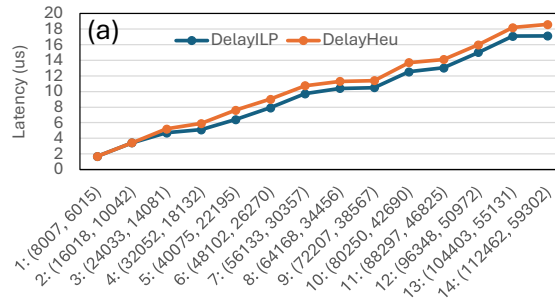
Table 8-2 Parameters and values.

Parameter	Definition	Value
T_{PC}	Polling cycle duration	100 us
SCS	Subcarrier spacing	60 kHz
Mini-slot	Duration of 2 OFDM symbols	36 us
B	Interface speed	100 Gbps
τ	Time slot granularity for scheduling	{10, 20, 50, 80, 100} ns
Burst _{max}	Maximum burst duration	2 us
FH delay	Delay requirements for FH flows	100 us
Burst iat	Burst inter arrival time	36 us
D_{prop}	Propagation delay in fiber	5 us/km

ILP vs Heuristic performance

To compare ILP and the proposed heuristic algorithms, we simulated the simulator for 20 polling cycles and generated the same traffic pattern for ILP and heuristic scenarios. We kept the number of simulating polling cycles low to be able to solve the ILP program in scalable time. Figure 8-12(a) shows the maximum measured delay among different flows in the network for the ILP and Heuristic approach. It can be inferred from the figure that the heuristic approach could find a quit good solution compared to the ILP. Figure 8-12(b) shows the comparison between heuristic and ILP in terms of execution time. The horizontal axis shows the size of the

problem. Each tick in the axis, is represented as $n_f:(n_{var}, n_{const})$, where n_f is the number of flows, n_{var} is number of variables in the ILP and n_{const} is the number of constraints in the ILP problem. As is shown, the execution time for ILP grows very fast and is much more than that of the heuristic approach.



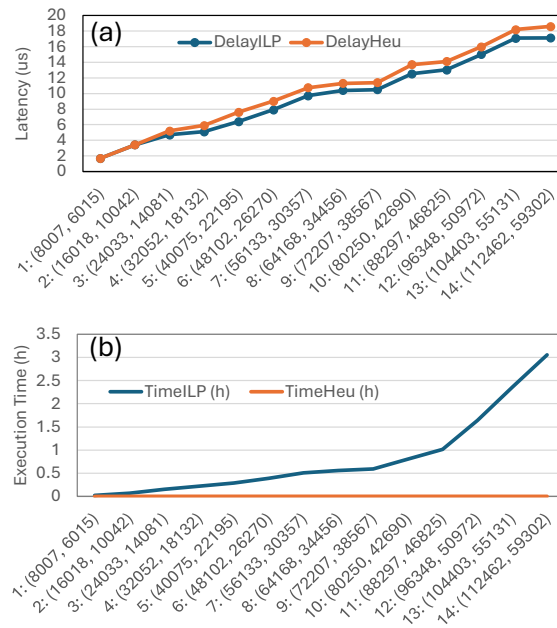


Figure 8-12: ILP and Heuristic comparison; (a) measured delay (b) execution time.

Heuristic Simulation Results

Figure 8-13 shows the impact of having different timeslot granularities on the (a) delay, (b) jitter, and (c) the supported link length. It can be inferred that they have almost the same performance. The considered maximum bucket (burst) size is 2us. We can also see the effect of an increasing number of FH flows in the network. Jitter is the standard deviation of the delays that different packets experienced. It should be noted that the guard band (GB) that we are considering is one time slot.

Figure 8-14 shows the total allocated capacity for two FH flows and the remained capacity for serving other types of traffic in one minute simulation. In this simulation, to consider the GB for other types of traffic, we considered two GBs for each FH flow that we allocated. One, at the beginning of granted window and one at the end that.

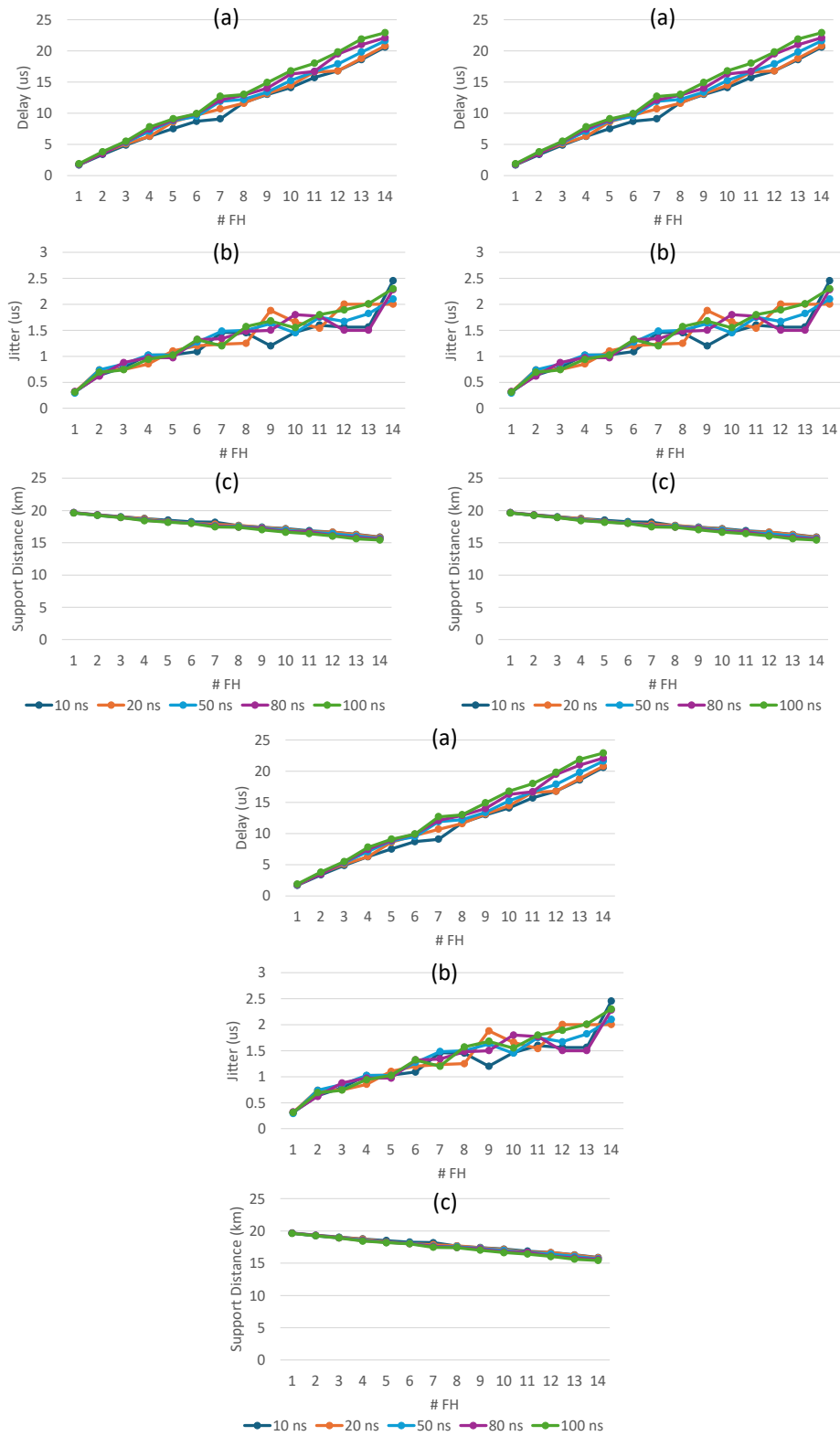


Figure 8-13: Impact of different timeslot granularities on the (a) delay, (b) jitter, (c) and support distance.

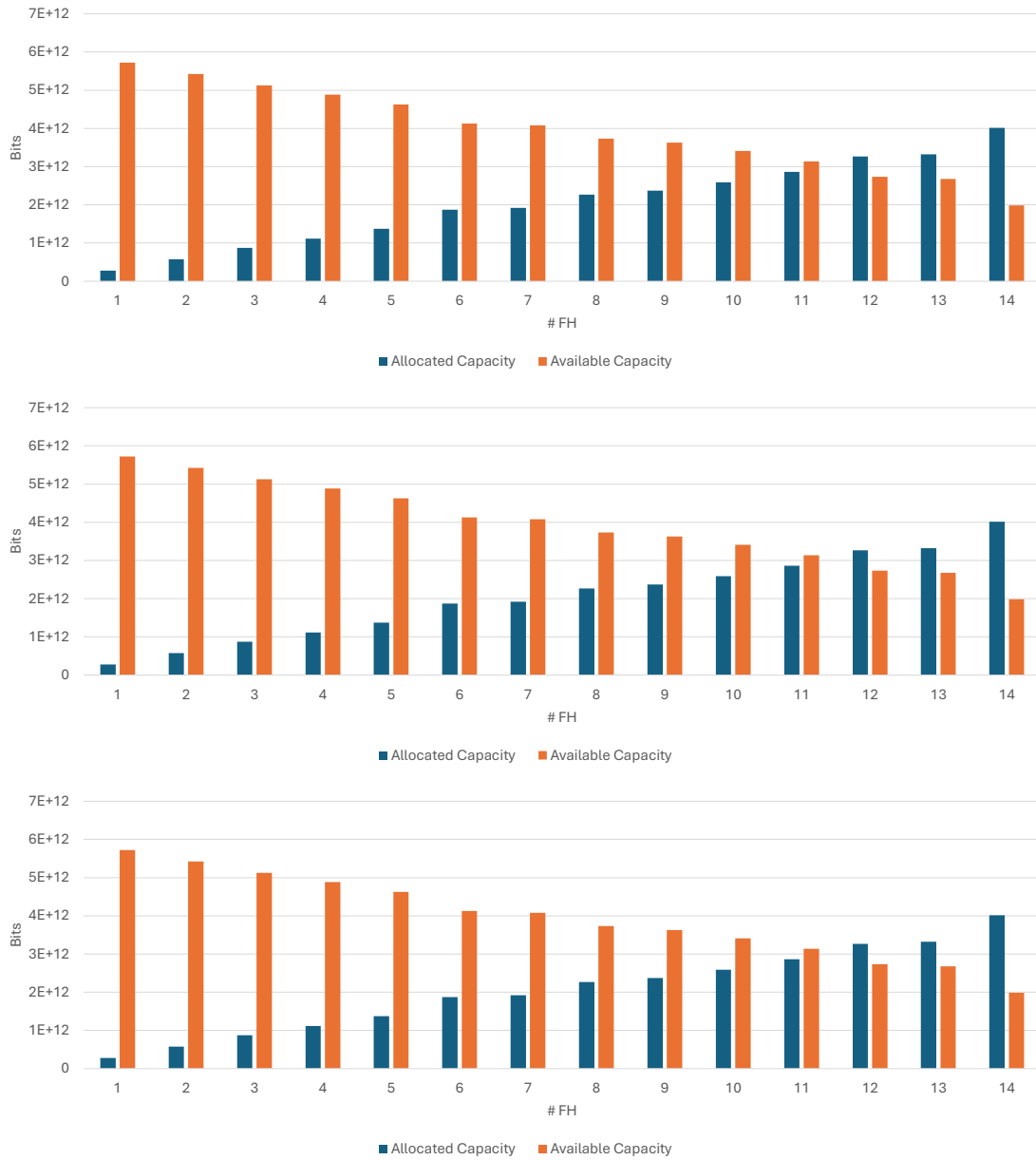


Figure 8-14: Allocated capacity to FH traffic vs Available Capacity to serve other types of traffic in 1 minute duration.

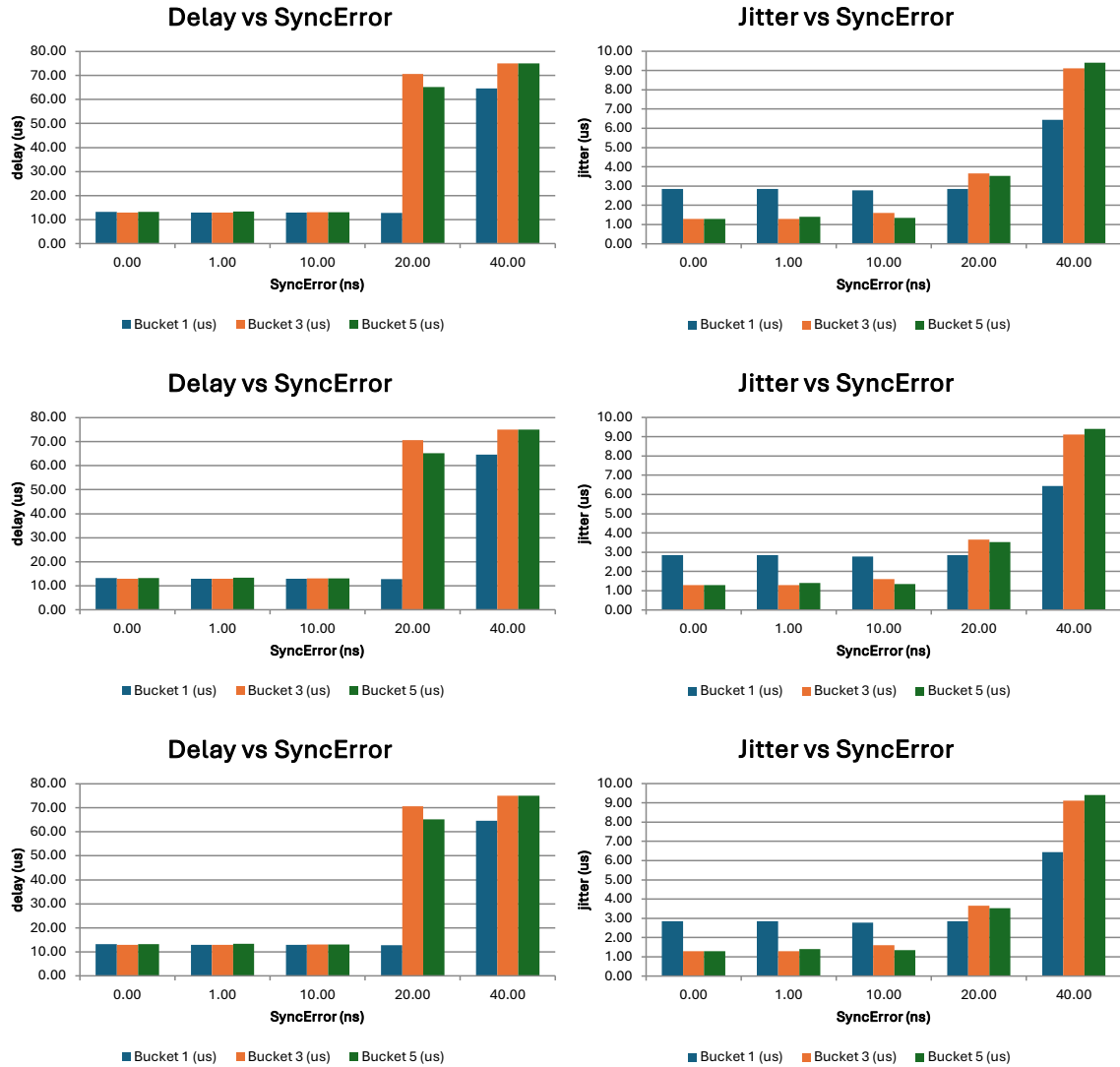


Figure 8-15: Effect of synchronization error on delay and jitter of packets considering 8 FH flows.

Finally, Figure 8-15 shows the effect of synchronization error on the delay and jitter of packets considering 8 FH flows. As you can see if we have 40 ns error in synchronization, we will have a huge delay in all cases. So, we must consider this error in the scheduler to be able to compensate for this error.

8.4 TECHNO-ECONOMIC EVALUATION OF DIGITAL OTDR FOR HIGH-ACCURACY, LOW-LATENCY OPTICAL NETWORK DIAGNOSTICS

The Digital Optical Time Domain Reflectometer (D-OTDR) is engineered and deployed to satisfy KPI 5.3, enabling precise latency and position measurements in fiber-optic transport networks. The target capabilities include achieving 4-ns latency resolution and position accuracy of less than 1 meter, while providing unidirectional latency measurement and continuous, in-service diagnostics. Unlike traditional OTDR systems that require traffic interruption and high-power analog pulses, the D-OTDR adopts a fully digital methodology, allowing supervision without affecting live traffic or altering network operating conditions.

The measurement chain relies on complementary Golay coded sequences transmitted at low optical power (approximately 2 dBm) on reserved out-of-band wavelengths (1610–1630 nm). Correlation-based digital signal processing reconstructs event timing with a temporal resolution of 2 ns, which—combined with the known group index of the fiber—maps to sub-meter spatial accuracy across typical transport fibers. This time-of-flight centric approach is the basis for achieving the required 4-ns latency granularity and position accuracy below 1 meter, while maintaining Class-1 laser safety and compatibility with existing services.

Figure below illustrates the compact architecture used to meet KPI 5.3: an FPGA generates Golay sequences, timestamps samples, and computes the reflectometric trace, while the SFP transceiver integrates the laser diode, linear AGC receiver, and optical circulator/splitter operating at the reserved wavelengths. This one-ended activation avoids any remote active equipment, simplifying deployment and enabling scalable latency/position supervision over distributed nodes.

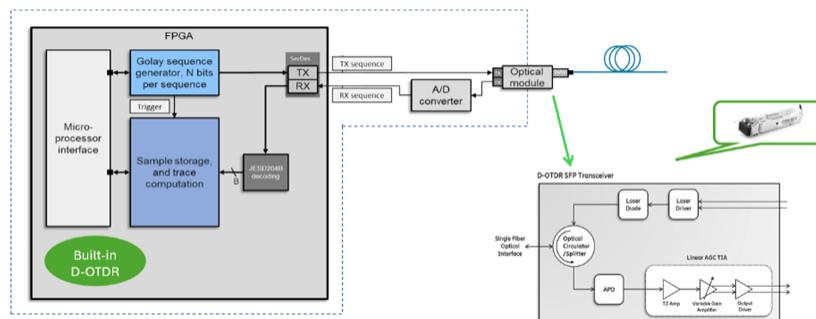


Figure 8-16: D-OTDR architecture integrated in SFP (KPI 5.3-oriented).

Operating from a single fiber end, the D-OTDR continuously acquires event timing along the link and converts it into distance/latency pairs. The intrinsic AC-coupling yields event-based information: discontinuities, open connectors, micro-bends, and Fresnel reflections are identified with high timing fidelity, whereas the continuous attenuation profile and absolute Return Loss are not reconstructed. For KPI 5.3, the emphasis is on rapid and precise timing of reflectometric events to localize positions and quantify segment latency without service disruption.

Figure 8-17 shows a reflectometric trace with an inset zoom on the first meters, demonstrating the absence of dead zones and the capability to resolve closely spaced connectors. The 2-ns timing resolution translates to approximately 20-cm spatial resolution, ensuring that early-stage connectors and short patch segments can be positioned accurately, supporting sub-meter accuracy required by KPI 5.3.

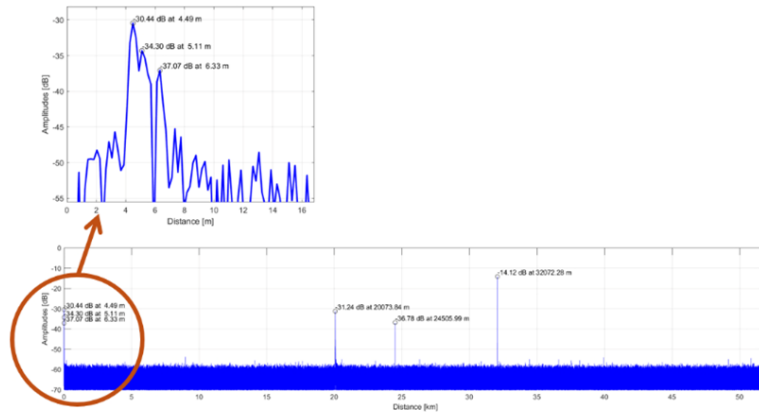


Figure 8-17: Reflectometric trace with zoom on first meters (timing-to-position accuracy).

Precise timing also distinguishes distributed phenomena from hard faults. Figure 8-18 contrasts micro-bending (distributed, frayed timing signature) against a severe open-connector event (narrow, high-amplitude peak). The D-OTDR’s timing correlation allows operators to assign accurate positions and compute segment latencies associated with each event, reducing Mean Time To Detect (MTTD) and Mean Time To Localize (MTTL) while directly contributing to KPI 5.3.

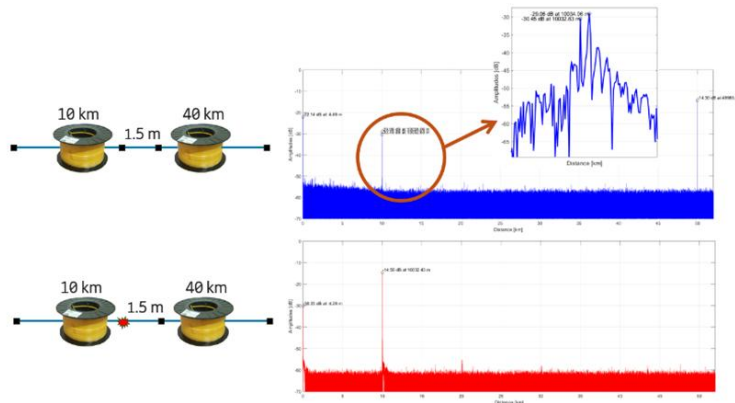


Figure 8-18: Comparison between micro-bending and severe fault (timing signature).

To enable unidirectional latency measurement without impacting traffic, Figure 8-19 presents the integration scheme using out-of-band optical filters. The D-OTDR sequences propagate on reserved wavelengths and their reflections are captured and time-correlated independently of service channels. Demarcation elements (e.g., open connectors or dedicated reflectors) provide

stable timing anchors, enabling accurate one-way latency estimation for monitored segments alongside precise position localization.

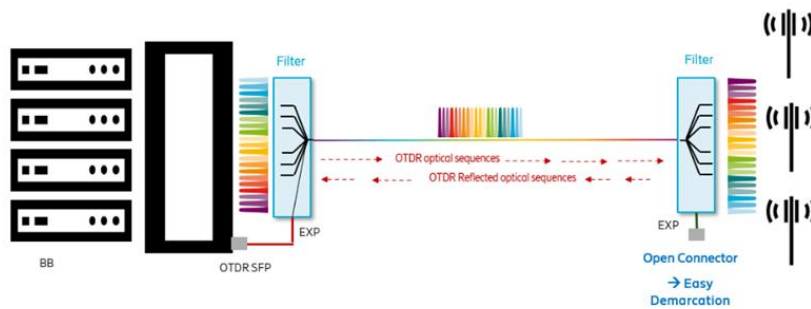


Figure 8-19: Network integration with out-of-band filters (unidirectional latency anchors).

Operational characteristics supporting KPI 5.3 include a detection range up to 52 km, approximately 1.0 W power consumption, and the complete absence of saturation and dead-zone effects, removing the need for launch fibers. The long coded sequences and robust correlation processing deliver high signal-to-noise ratio for timing extraction, enabling 4-ns latency resolution and sub-meter position accuracy across the monitored infrastructure.

In summary, the D-OTDR enhances optical network diagnostics, ensuring rapid detection and localization of issues within fiber-optic infrastructure while providing unidirectional latency measurement. The architecture, timing resolution, and out-of-band integration collectively address KPI 5.3 by enabling precise latency and position measurements—achieving 4-ns latency granularity and position accuracy of less than 1 meter—within live, high-capacity transport networks.

8.5 ADDITIONAL TECHNO-ECONOMIC ANALYSIS FOR ADVANCED MOBILE TRANSPORT ARCHITECTURES

This section presents the additional techno-economic analysis activities conducted by WEST Aquila in the context of the extended project activities. These contributions aim at strengthening the comparative assessment framework developed throughout the SEASON project, with particular focus on optical transport solutions for mobile xHaul applications across diverse deployment scenarios.

The work encompasses four main areas of investigation. The first contribution addresses the parametrization of cost models for transport solutions, with particular emphasis on the role of site sharing factors and infrastructure reuse assumptions in determining overall network deployment costs. A comprehensive simulation framework has been developed to enable systematic comparison across Point-to-Point (P2P), Wavelength Division Multiplexing (WDM),

and Point-to-Multipoint (P2MP) transport architectures, incorporating the SEASON-defined geotypes (Dense Urban, Urban, Suburban, and Rural) as reference deployment scenarios. The second area of investigation assesses the impact of future 400G fronthaul requirements on cost analysis, evaluating how the transition from current fronthaul requirements to higher fronthaul with advanced MIMO configurations affects infrastructure dimensioning and capital expenditure across different optical transport solutions. The third contribution focuses on the assessment of single-fiber ring topology as an alternative to dual-fiber ring deployments, quantifying the trade-offs between infrastructure complexity, network efficiency, and ring utilization when employing XR optics technology with digital subcarrier multiplexing. Finally, the fourth area introduces the study and modeling of a new SEASON geotype specifically tailored for linear infrastructure along railway and motorway corridors, addressing the unique requirements of continuous radio coverage and the corresponding transport network architectures.

The following subsections detail the methodology, results, and key findings for each of these investigation areas.

8.5.1 Cost Model Parametrization for Transport Solutions

Economic comparison of optical transport solutions for mobile xHaul applications. The simulator models the deployment of Point-to-Point (P2P), Wavelength Division Multiplexing (WDM), and Point-to-Multipoint (P2MP) transport architectures across the four SEASON reference geotypes: Dense Urban, Urban, Suburban, and Rural. The cost model adopted in this analysis follows the methodology presented in Deliverable D2.3, where equipment costs are normalized to a defined cost unit (c.u.) to ensure consistency and comparability across different network components and deployment scenarios.

The simulation framework evaluates five distinct transport architecture variants: (i) P2P, employing dedicated grey transceivers for each fronthaul connection; (ii) WDM, utilizing wavelength multiplexing with passive optical components; (iii) WDM-WP (WDM with Pre-aggregation), incorporating traffic aggregation at intermediate nodes; (iv) P2MP, leveraging XR optics technology with digital subcarrier multiplexing; and (v) P2MP-WP (P2MP with Pre-aggregation), combining XR optics with pre-aggregation strategies. For each scenario, the simulator computes the total CAPEX by aggregating equipment costs across the network topology, considering both transmission and switching components for medium-term and long-term deployment horizons.

While previous analyses in the SEASON project employed best-case and worst-case assumptions for XR optics pricing, this study introduces a parametric approach through the definition of a cost scaling factor α (alpha). The XR equipment cost is modeled as a function of the corresponding grey long-reach (LR) transceiver cost according to the relationship presented in (14).

$$C_{XR} = \alpha \times C_{Grey-LR} \tag{14}$$

where C_{XR} represents the normalized cost of the XR transceiver module, $C_{Grey-LR}$ is the normalized cost of the equivalent grey long-reach transceiver at the same data rate, and α is the cost scaling factor. This parametric formulation enables systematic sensitivity analysis across a range of potential market conditions, from optimistic scenarios where XR technology achieves cost parity with grey transceivers ($\alpha \approx 1$) to more conservative projections where coherent technology commands a significant premium ($\alpha > 2$).

The sensitivity analysis was conducted by varying α from 0.5 to 3.0, covering scenarios ranging from aggressive cost reduction through economies of scale to premium pricing for emerging technology. Figure 8-20 and Figure 8-21 present the total cost evolution as a function of α for all five transport solutions across the four deployment scenarios, for medium-term and long-term horizons respectively.

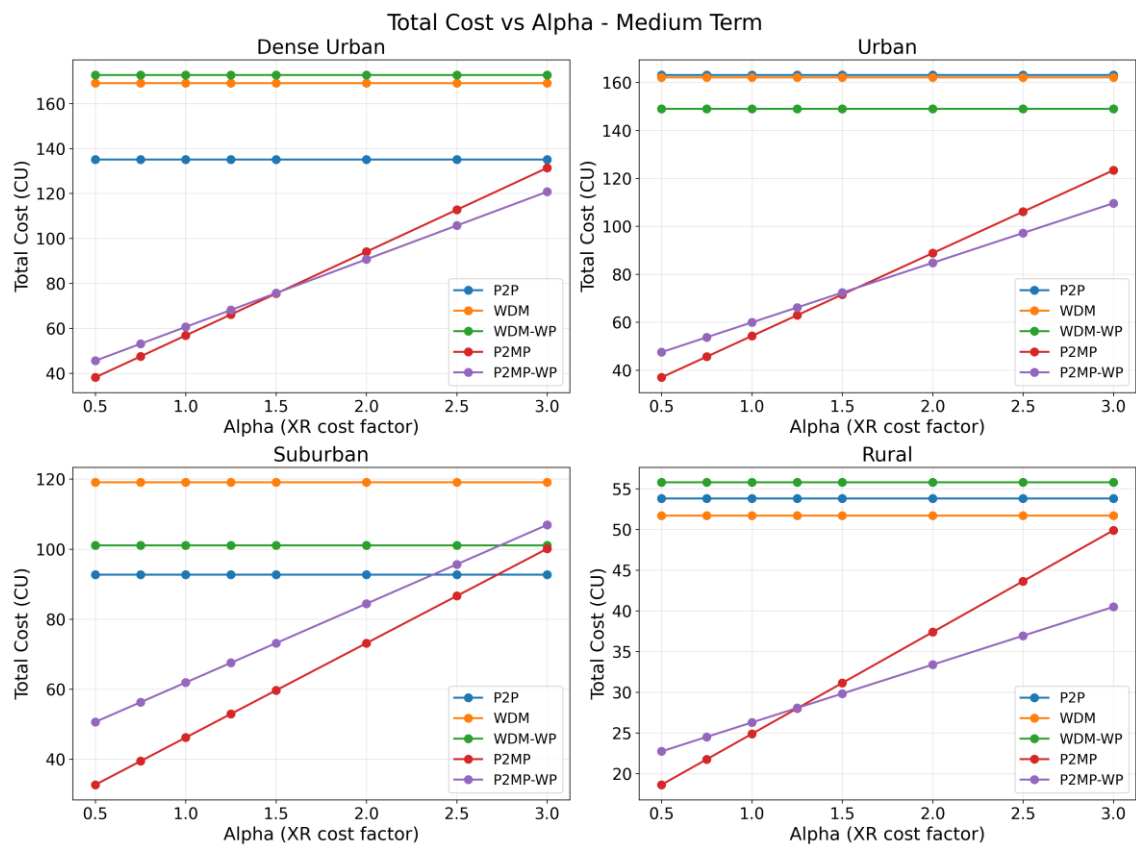


Figure 8-20: Total cost versus XR cost factor (α) for medium-term deployment across all geotypes.

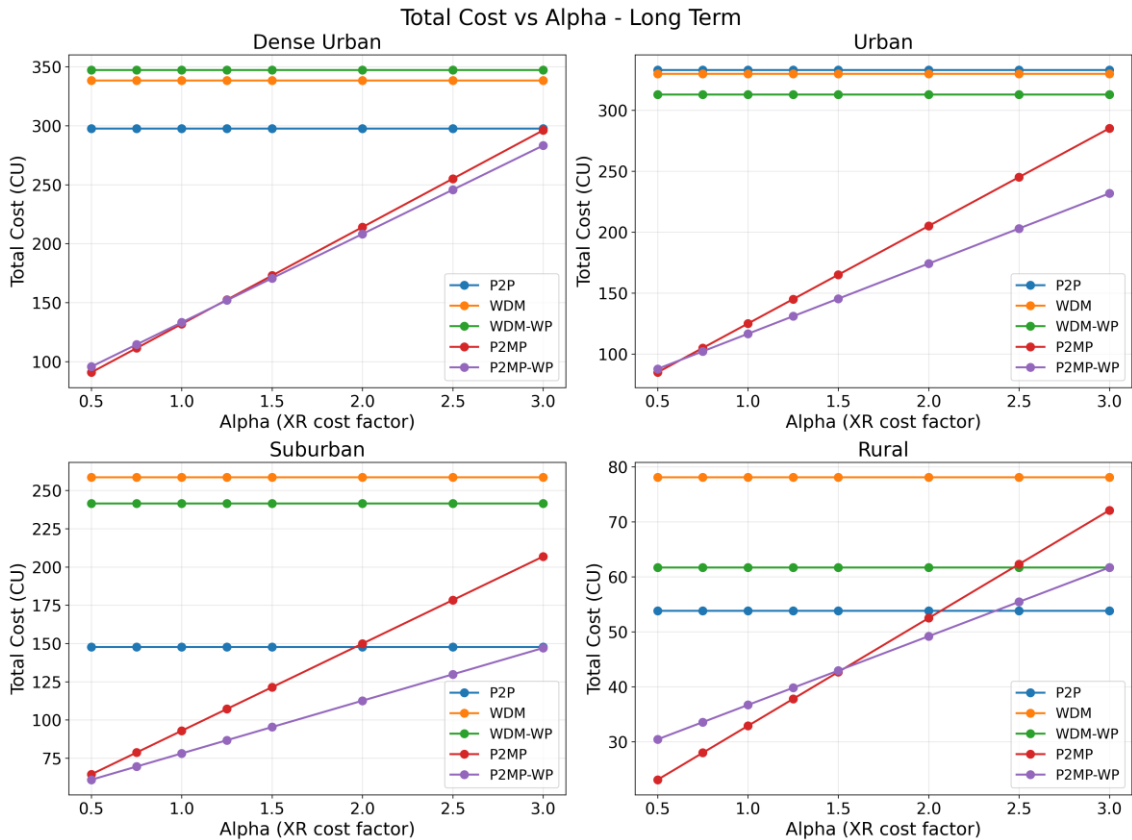


Figure 8-21: Total cost versus XR cost factor (α) for long-term deployment across all geotypes.

The analysis reveals several important insights regarding the cost-competitiveness of P2MP solutions based on XR optics technology. First, the P2P, WDM, and WDM-WP solutions exhibit constant cost profiles with respect to α , as these architectures do not employ XR transceivers. In contrast, P2MP and P2MP-WP costs increase linearly with α , reflecting the direct relationship defined in Equation (1). The crossover points where P2MP solutions become more expensive than traditional alternatives vary significantly across geotypes and deployment horizons.

In Dense Urban and Urban scenarios, characterized by high site density and substantial traffic volumes, P2MP solutions demonstrate cost advantages only for relatively low values of α . For medium-term deployments, P2MP becomes cost-equivalent to P2P at approximately $\alpha = 2.5$ in Dense Urban and $\alpha = 2.0$ in Urban scenarios. The P2MP-WP variant, which incorporates pre-aggregation, extends the competitive range by reducing the number of required XR modules through traffic consolidation at intermediate aggregation points.

Suburban and Rural deployments present the most favorable conditions for P2MP adoption. In these scenarios, the statistical multiplexing gains enabled by XR optics technology translate into significant infrastructure cost reductions. Even at $\alpha = 3.0$, P2MP solutions remain competitive with or superior to WDM alternatives in Rural scenarios. The P2MP-WP architecture demonstrates particular effectiveness in Suburban deployments, maintaining cost advantages

across the entire α range evaluated. These results suggest that operators targeting suburban and rural coverage expansion would benefit most from XR optics technology investments.

Comparing medium-term and long-term deployment horizons, the analysis indicates that the relative cost advantages of P2MP solutions are preserved across both timeframes, with absolute cost values scaling proportionally with increased network capacity requirements. The long-term scenario, which assumes higher traffic demands and correspondingly greater equipment deployment, amplifies both the potential benefits and risks associated with XR technology adoption, underscoring the importance of accurate α estimation for strategic planning purposes.

This parametric cost model provides network planners with a quantitative framework for evaluating P2MP adoption strategies under varying market conditions. By identifying the α thresholds at which different solutions become cost-optimal, operators can make informed decisions regarding technology selection based on their specific deployment scenarios and expectations for XR optics price evolution.

8.5.2 Impact of Future 400G Fronthaul Requirements on Cost Analysis

The evolution of mobile networks toward 6G and advanced 5G deployments is expected to significantly increase fronthaul capacity requirements. A key driver of this increase is the adoption of higher-order MIMO antenna configurations, which enable enhanced spectral efficiency and network capacity but impose proportionally higher demands on the transport infrastructure. This section quantifies the impact of transitioning from MIMO 4 to MIMO 16 configurations on both aggregate fronthaul data rates and the resulting transport network CAPEX across the four SEASON reference geotypes.

The fronthaul capacity requirement for a given radio unit is determined by the functional split option, carrier bandwidth, and MIMO configuration. Following the eCPRI-based functional split 7.2x, which represents a commonly adopted trade-off between fronthaul bandwidth and processing distribution, the fronthaul data rate scales linearly with the number of MIMO layers. This relationship implies that transitioning from MIMO 4 to MIMO 16 results in a four-fold increase in per-carrier fronthaul requirements, assuming equivalent bandwidth allocations. When aggregated across all carriers and sites within a geotype, this multiplication factor propagates through the entire transport network dimensioning.

To assess the infrastructure implications of this transition, a comparative analysis was conducted using the simulation framework described in Section 8.5.1. The analysis considers the long-term deployment scenario, where the radio layer configuration includes multiple carriers spanning sub-GHz, 1-3 GHz, and 3-7 GHz frequency bands. For the MIMO 16 scenario, all carriers that currently employ MIMO 4 or MIMO 8 configurations are upgraded to MIMO 16, representing a forward-looking assessment of future network requirements.

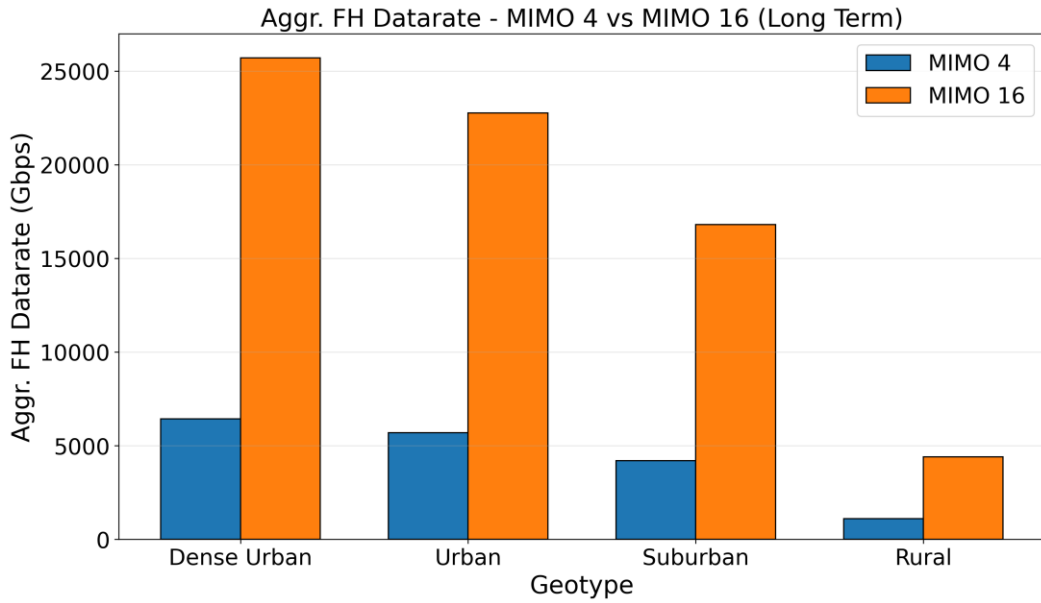


Figure 8-22: Aggregate fronthaul data rate comparison between MIMO 4 and MIMO 16 configurations for long-term deployment across all geotypes.

Figure 8-22 presents the aggregate fronthaul data rate requirements for each geotype under both MIMO configurations. The Dense Urban scenario exhibits the highest absolute values, with aggregate fronthaul demand increasing from approximately 6.5 Tbps under MIMO 4 to nearly 26 Tbps under MIMO 16. Urban and Suburban geotypes follow similar patterns, with MIMO 16 requirements reaching approximately 23 Tbps and 17 Tbps respectively. Even the Rural scenario, characterized by lower site density, shows a substantial increase from approximately 1.2 Tbps to 4.5 Tbps. These results confirm that the transition to MIMO 16 consistently produces a multiplication factor between 3.8x and 4.0x across all deployment scenarios, aligning with the theoretical expectation of a four-fold increase.

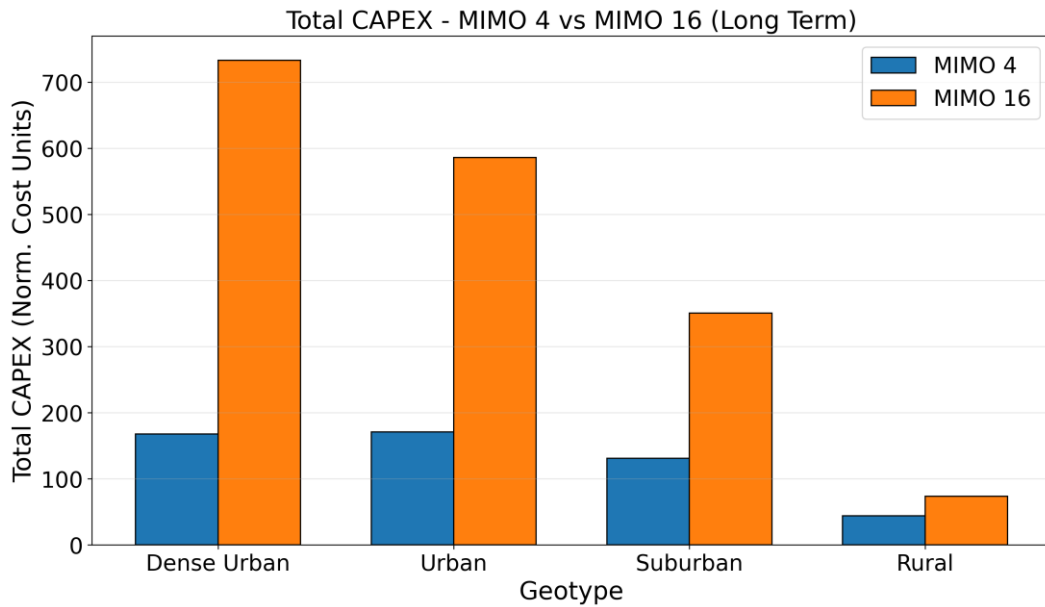


Figure 8-23: Total transport network CAPEX comparison between MIMO 4 and MIMO 16 configurations for long-term deployment across all geotypes.

Figure 8-23 illustrates the corresponding impact on transport network CAPEX. The cost increase associated with MIMO 16 adoption varies significantly across geotypes, reflecting the interplay between traffic growth and the cost structure of different transport solutions. Table 8-3 summarizes the CAPEX multiplication factors observed for each scenario.

Table 8-3: CAPEX multiplication factors for MIMO 16 vs MIMO 4 transition.

Geotype	MIMO 4 (c.u.)	MIMO 16 (c.u.)	Factor
Dense Urban	168	733	4.37×
Urban	172	590	3.43×
Suburban	130	352	2.71×
Rural	45	76	1.69×

The analysis reveals that CAPEX multiplication factors range from 1.69× in Rural deployments to 4.37× in Dense Urban scenarios. Notably, the CAPEX increase does not scale linearly with the traffic increase (which is consistently ~4×) due to several factors. First, switching equipment costs exhibit economies of scale, where higher-capacity switches do not cost proportionally more than lower-capacity alternatives. Second, the statistical multiplexing gains achievable with P2MP solutions become more pronounced at higher traffic loads, partially offsetting the capacity increase. Third, Rural deployments benefit from lower baseline equipment utilization, allowing absorption of additional traffic within existing infrastructure margins.

These findings have significant implications for network planning and technology selection. In Dense Urban environments, where the CAPEX impact is most severe, the choice of transport architecture becomes critical. P2MP solutions with XR optics may offer particular advantages in

these scenarios, as their inherent capacity flexibility enables more efficient accommodation of traffic growth without proportional infrastructure scaling. Conversely, in Rural deployments where the absolute traffic volumes remain manageable even under MIMO 16 assumptions, simpler P2P architectures may remain cost-effective.

The transition to 400G-class fronthaul, driven by advanced MIMO configurations, represents a substantial infrastructure investment challenge. Operators planning network evolution strategies must account for these capacity requirements and evaluate transport solutions that can accommodate the anticipated traffic growth while maintaining acceptable cost profiles. The parametric cost model presented in Section 8.5.1, combined with the MIMO impact analysis of this section, provides a comprehensive framework for such strategic assessments.

8.5.3 Assessment of Single-Fiber Ring Topology

Ring topologies represent a widely adopted architecture for metro and access network deployments, offering inherent protection capabilities and efficient fiber utilization. In the context of mobile fronthaul transport, ring architectures based on XR optics technology with digital subcarrier multiplexing (DSCM) provide an attractive alternative to traditional point-to-point deployments. This section presents a comparative assessment of dual-fiber and single-fiber ring configurations, quantifying the trade-offs between infrastructure complexity, capacity utilization, and network efficiency.

The analysis considers XR optics transceivers operating with 16 digital subcarriers, each providing 25 Gbps capacity. In the dual-fiber ring configuration, two separate fibers are employed for opposite transmission directions (clockwise and counter-clockwise), enabling 400 Gbps capacity per direction (16 subcarriers \times 25 Gbps). In the single-fiber ring configuration, a single fiber carries bidirectional traffic by allocating half of the subcarriers to each direction, resulting in 200 Gbps capacity per direction (8 subcarriers \times 25 Gbps). The single-fiber approach reduces fiber infrastructure requirements but constrains the available capacity per ring.

A ring topology simulator was developed to assess the infrastructure requirements for both configurations across the SEASON reference geotypes. The simulator employs a bin-packing algorithm to allocate individual radio equipment flows to available subcarriers within each ring. This flow-level allocation approach accurately captures the capacity constraints imposed by the 25 Gbps subcarrier granularity, where small traffic flows must occupy entire subcarriers regardless of their actual bandwidth requirements. The algorithm iteratively assigns flows to rings using a first-fit strategy, creating new rings when existing ones cannot accommodate additional flows.

Two key performance metrics are evaluated in this analysis. Network Efficiency measures the ratio between actual traffic and the capacity allocated to transport it, reflecting the overhead introduced by subcarrier granularity. Ring Utilization represents the ratio between actual traffic and the total installed ring capacity, indicating how effectively the deployed infrastructure is

used. Both metrics are constrained by the discrete nature of subcarrier allocation, where flows smaller than 25 Gbps still require a full subcarrier.

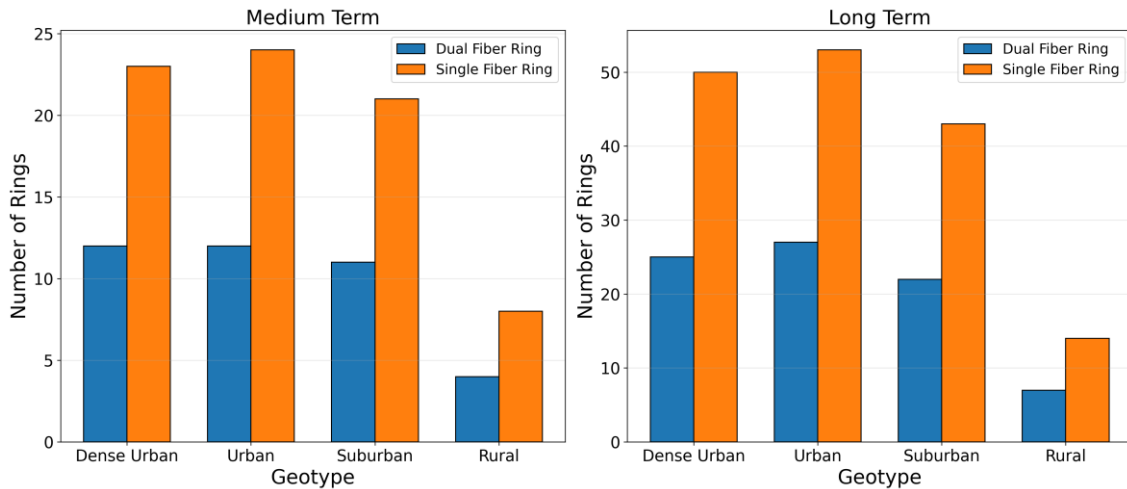


Figure 8-24: Number of rings required for dual-fiber and single-fiber configurations across geotypes and deployment horizons.

Figure 8-24 presents the number of rings required to accommodate the fronthaul traffic for each geotype under medium-term and long-term deployment scenarios. The results demonstrate that single-fiber rings consistently require approximately twice the number of rings compared to dual-fiber configurations. This ratio directly reflects the halved per-direction capacity of the single-fiber approach. For the long-term Dense Urban scenario, dual-fiber deployment requires 25 rings while single-fiber deployment necessitates 50 rings. Urban and Suburban geotypes exhibit similar scaling patterns, while Rural deployments show more modest absolute requirements due to lower traffic volumes.

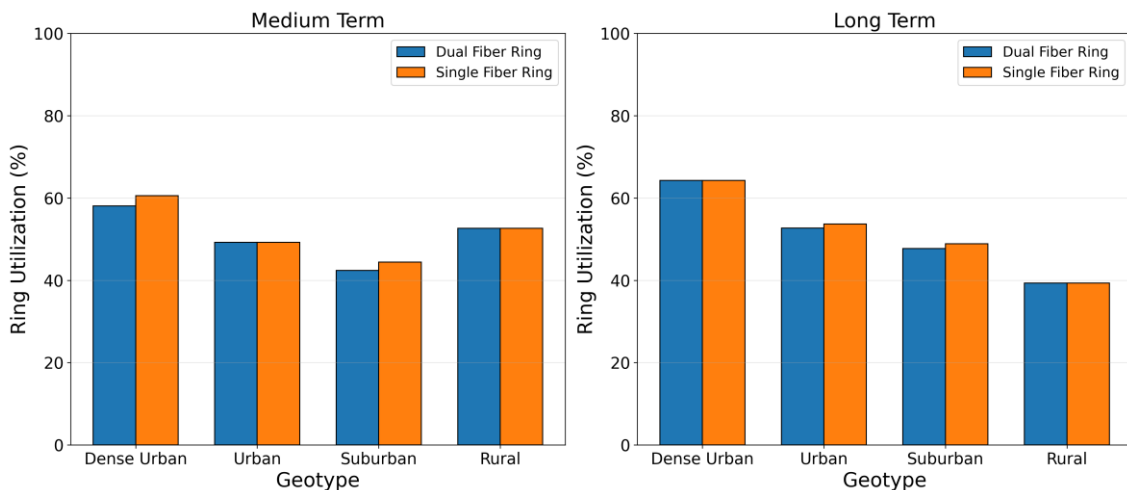


Figure 8-25: Ring utilization comparison between dual-fiber and single-fiber configurations across geotypes and deployment horizons.

Figure 8-25 illustrates the ring utilization achieved by both configurations. Utilization values range from approximately 39% to 64% across all scenarios, with dual-fiber and single-fiber configurations exhibiting comparable utilization levels within each geotype. The Dense Urban scenario achieves the highest utilization (approximately 58-64%), benefiting from larger traffic volumes that enable more efficient packing of flows into subcarriers. Suburban and Rural deployments show lower utilization (42-53%), reflecting the impact of subcarrier granularity on scenarios with smaller individual flows. The similar utilization levels between dual and single-fiber configurations indicate that the capacity overhead introduced by discrete subcarrier allocation is consistent regardless of the ring capacity.

The Network Efficiency metric, which excludes unused ring capacity and focuses solely on allocated subcarrier utilization, ranges between 42% and 65% across all scenarios. This efficiency ceiling is fundamentally constrained by the 25 Gbps subcarrier granularity: fronthaul flows from individual radio equipment typically range from 1 to 22 Gbps depending on carrier configuration and MIMO order, meaning that most flows utilize only a fraction of their allocated subcarrier capacity. Improving this efficiency would require either smaller subcarrier granularity (increasing system complexity) or traffic aggregation strategies that combine multiple small flows before subcarrier allocation.

The choice between dual-fiber and single-fiber ring configurations involves a trade-off between fiber infrastructure costs and equipment costs. Single-fiber rings halve the fiber requirements but double the number of rings needed, implying a proportional increase in XR transceiver equipment at both hub and leaf nodes. In deployment scenarios where fiber availability is constrained or fiber leasing costs are dominant, single-fiber configurations may offer economic advantages despite higher equipment counts. Conversely, in greenfield deployments or scenarios with abundant fiber infrastructure, dual-fiber rings provide more efficient capacity utilization per ring and simpler network planning.

This assessment provides network planners with quantitative guidance for ring topology selection in XR optics-based mobile transport deployments. The consistent 2:1 ratio in ring requirements between single and dual-fiber configurations, combined with the relatively stable utilization levels across both approaches, enables straightforward infrastructure dimensioning based on known traffic projections. Future work may explore hybrid configurations that combine single and dual-fiber rings within the same network to optimize the trade-off between fiber utilization and equipment costs on a per-segment basis.

8.5.4 Study and Modeling of the New SEASON Railway Geotype

The SEASON project initially defined four reference geotypes (Dense Urban, Urban, Suburban, and Rural) to characterize different deployment scenarios for mobile transport networks. However, linear infrastructure corridors such as railway lines and motorways present unique characteristics that are not adequately captured by area-based geotype definitions. These scenarios require continuous radio coverage along narrow but extended corridors, with specific

constraints on site placement and transport network topology. To address this gap, a fifth geotype—the Railway geotype—has been defined and modeled within the extended SEASON techno-economic framework.

The Railway geotype models a chain of mobile sites providing coverage along a railway line or motorway segment. The reference scenario considers a 50 km segment covered by 10 sites placed uniformly, resulting in a 5 km inter-site distance. Each site hosts two Radio Units (RUs) with directional antennas oriented in opposite directions along the corridor to ensure continuous coverage. The radio configuration assumes operation at 3.7 GHz carrier frequency with 100 MHz bandwidth, representing a realistic 5G deployment targeting high-speed train connectivity with speeds up to 300 km/h.

A Monte Carlo radio network simulator was developed to characterize the achievable throughput in the Railway geotype scenario. The simulator implements a downlink physical layer model compliant with 3GPP specifications, including the RMa (Rural Macro) path loss model from 3GPP TR 38.901 with line-of-sight propagation and log-normal shadow fading (6 dB standard deviation). The Modulation and Coding Scheme (MCS) selection follows 3GPP TS 38.214 Table 5.1.3.1-2 for PDSCH, with a target Block Error Rate (BLER) of 10^{-3} . The simulator evaluates system performance across a comprehensive parameter space, including multiple channel bandwidths (20, 50, and 100 MHz), MIMO configurations (SISO, 2×2, 4×4, and 8×8), and inter-site distances ranging from 0.5 to 50 km.

Table 8-4: Downlink throughput [Mbps] as a function of inter-site distance and MIMO configuration for the Railway geotype (100 MHz bandwidth).

RU Distance	SISO	MIMO 2×2	MIMO 4×4	MIMO 8×8
5 km	120	350	500	620
10 km	110	280	400	520
25 km	80	140	200	280
50 km	50	80	100	140

Table 8-4 presents representative simulation results showing achievable downlink throughput as a function of the RU inter-site distance for the 100 MHz bandwidth configuration. The results demonstrate the expected dependency on MIMO configuration. For the reference scenario with 5 km inter-site distance and MIMO 4×4 configuration, the simulator predicts approximately 500 Mbps throughput, degrading to approximately 200 Mbps at 25 km and to 100 Mbps at 50 km inter-site distance. These values are consistent with theoretical expectations based on the selected MCS tables and represent realistic performance bounds for the railway coverage scenario.

The simulation results directly inform the transport network dimensioning for the Railway geotype. Based on the achievable throughput values, the midhaul/backhaul traffic per cell does not exceed approximately 500 Mbps even under full load conditions. Considering the reference

scenario with 10 sites and 2 RUs per site, the aggregate traffic collected at the tail-end site (connected to the Central Office) remains below 10 Gbps. This finding supports the use of 10G transceivers for the chain transport topology, significantly simplifying the transport network design compared to the high-density urban scenarios where fronthaul requirements can reach tens of Gbps per site.

The Railway geotype simulator has been integrated with the SEASON techno-economic framework to enable comparative analysis of different RAN architectures. Specifically, three architectural options have been evaluated: (i) a baseline architecture employing Cell Site Routers (CSR) for hop-by-hop traffic aggregation; (ii) an enhanced baseline using BBUs with integrated networking capabilities; and (iii) an O-RAN compliant architecture with virtualized DU (vDU) functions running on servers equipped with Data Processing Units (DPU) for x-haul traffic handling. The detailed techno-economic analysis of these architectures, including CAPEX comparison and transceiver count assessment, is presented in Deliverable D2.3 Section 4.1.

The key findings from the Railway geotype analysis indicate that the chain topology inherent to linear infrastructure deployments enables significant simplification of transport network equipment compared to traditional cell-site architectures. The O-RAN approach with DPU-enabled servers demonstrates potential for reducing both the number of optical transceivers (33% reduction compared to baseline) and the number of optical-to-electrical-to-optical (O/E/O) conversions (50% reduction). These efficiency gains, combined with the moderate transport capacity requirements derived from the radio simulations, suggest that linear infrastructure scenarios represent favorable deployment contexts for emerging O-RAN and virtualized RAN technologies.

The Railway geotype extension enhances the SEASON techno-economic framework by providing quantitative modeling capabilities for an increasingly important deployment scenario. As railway operators and infrastructure managers seek to deploy 5G and future 6G services along transportation corridors, the simulation tools and architectural analysis developed in this work provide valuable guidance for technology selection and network planning decisions.

8.6 ON THE EXPLOITATION OF SEGMENTED MACH-ZEHNDER MODULATORS IN THE FRONT/MID-HAUL AND ACCESS

Works carried out within the framework of the SEASON project highlighted the role of DSCM in enhancing network and resource efficiency and flexibility. DSCM is currently achieved through DSP. Despite its important role in network systems, currently available DSCM methods are either expensive (due to the electronic domain processing within DSP) or inefficient. Therefore, new opportunities must be explored to achieve low-cost and efficient sub-carrier multiplexing (SCM) implementations, possibly without the use of DSP (thus, migrating from DSCM to a more generic SCM).

To understand the SEASON proposal for new SCM techniques, a brief introduction is given for the generation of DSCM. The process begins with N tributary digital signals, which are processed through DSP to generate the corresponding DSCM signal. The resulting samples are then converted into an analog SCM signal via digital-to-analog converter (DAC). Finally, after amplification by an electrical amplifier, the signal is fed into a Mach-Zehnder Modulator (MZM) for optical transmission. Then, the DSCM is transmitted through an optical carrier to generate a subcarrier-modulated optical signal. Its implementation requires expensive and power-intensive DSP as well as high-speed DACs.

We propose a Segmented Mach-Zehnder Modulator (SEMZM) as a novel concept of frequency multiplexer to generate optical SCM. We use a SEMZM with the concept of a frequency multiplexer to generate a subcarrier-modulated optical signal. Instead of generating digital subcarrier signals using a power-hungry DSP, a SEMZM can generate an optical signal carrying multiple subcarriers by implementing a carefully designed segmentation and using a multi-tone RF drive signal for driving different segments with multiple RF signals at distinct frequencies.

A SMZM is an integrated electro-optic modulator where the phase-shifting region of one or both arms is divided into a series of electrically independent segments that can be driven with different electrical delays. Each segment is associated to a specific sub-carrier. Sub-carriers are thus generated without the need of DSP, while only relying on the electro-optic modulator. An example of a 4-segments MZM is depicted in Figure 8-26.

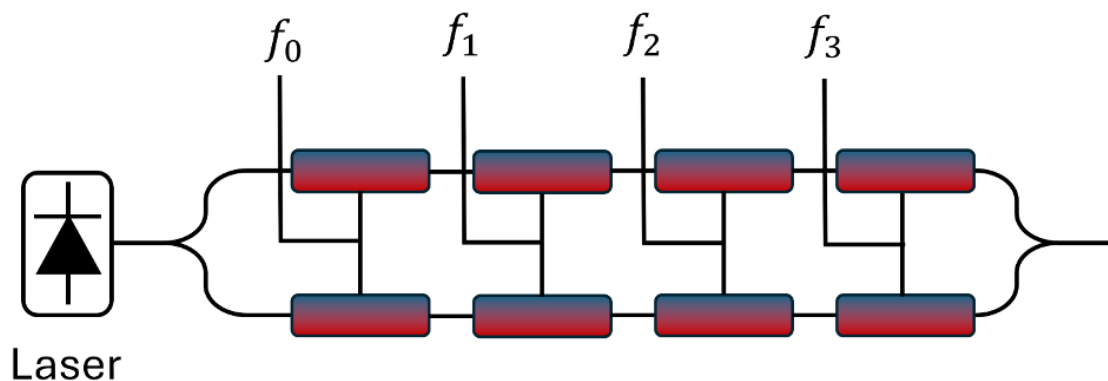


Figure 8-26: Schematic of the segmented modulator and associated carriers.

Using a single segmented modulator rather than multiple single-segment modulators has multiple advantages: (i) It enables a more compact design, reducing the area occupied by chips and making them smaller and cheaper to fabricate; (ii) By segmenting the MZM and driving each segment with an RF signal delayed to match the optical transit time through the modulator, the electrical and optical signals remain in phase throughout the device. This results in higher bandwidth. (iii) The driving voltage per segment is decreased by a factor equal to the number of segments, enabling direct CMOS driving and avoiding the need for bulky, power-hungry and expensive RF amplifiers.

However, the usable dynamic range of each signal is reduced, which decreases the electro-optical conversion efficiency of the input RF signals. Therefore, there is a clear trade-off between the number of channels per segmented MZM and the quality of the outgoing signals. This trade-off has been studied for up to four segments using the photonic-electronic system simulation environment, Lumerical INTERCONNECT. The simulation scheme is shown in Figure 8-27.

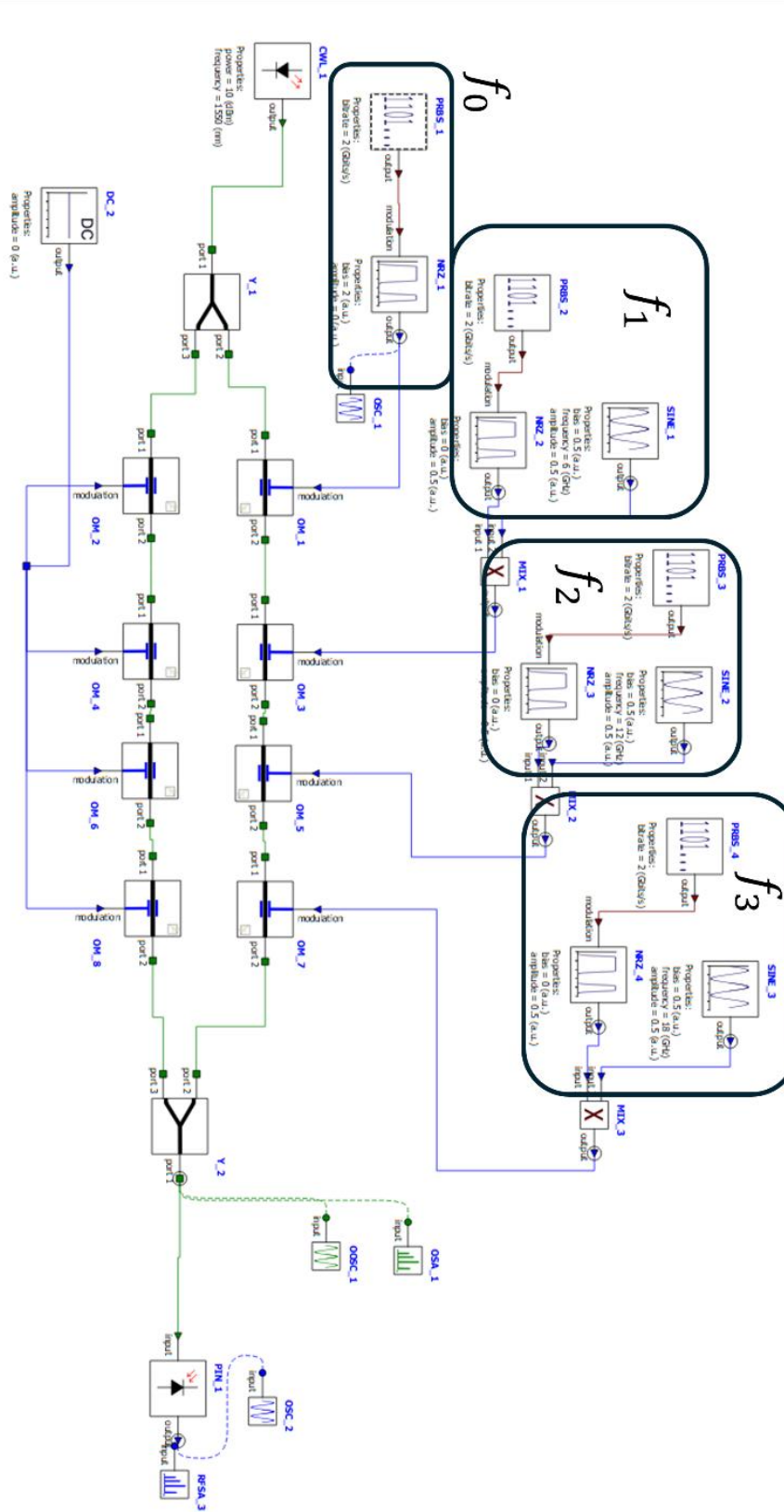


Figure 8-27: Lumerical INTERCONNECT simulation environment for the segmented transmitter.

The structure has been simulated considering the non-idealities of physical devices. The following features have been considered for the simulation:

- Laser power: 10 dBm
- Laser wavelength: 1550 nm
- Laser RIN: -155 dBm/Hz
- Segment half-wave voltage: 2V
- Sub-carrier separation: 6 GHz

All incoming bitstreams were generated as pseudo-random bit sequences and encoded with an OOK-NRZ technique. The outgoing compound optical signal is depicted in Figure 8-28.

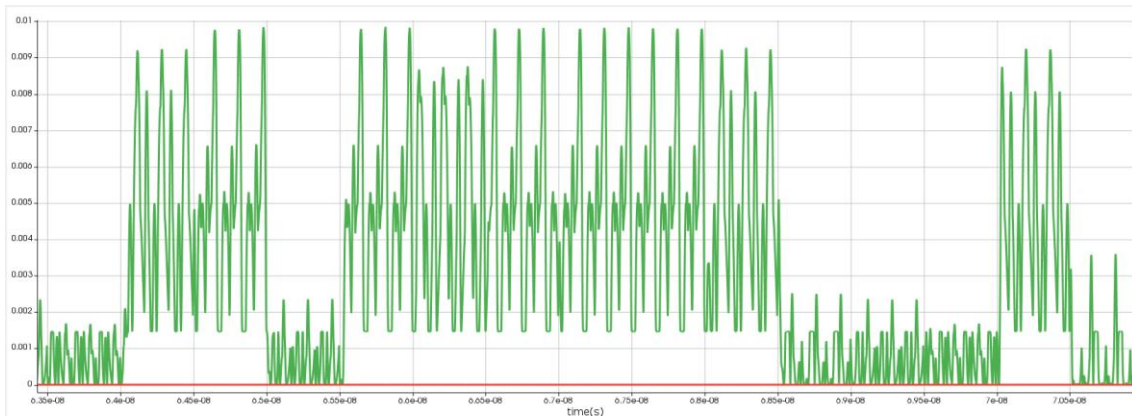


Figure 8-28: Example of output signal in time domain.

The signal has been directly detected by a photodiode to show the signal separation in the electrical spectrum, depicted in Figure 8-29.

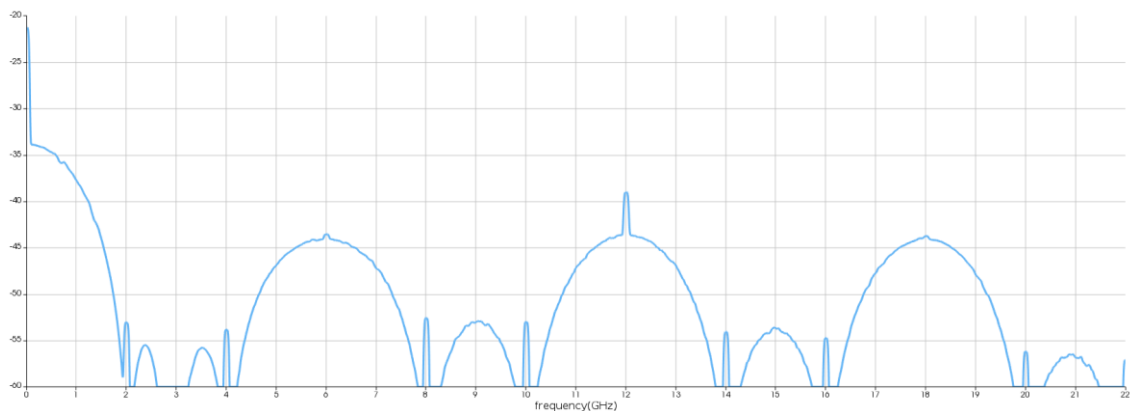


Figure 8-29: Output spectrum with 4 carriers spaced by 6 GHz.

The proposed solution permits to generate sub-carriers without the use of energy-hungry DSP. Possible uses:

- Fine-granularity transceivers: the selective activation of sub-carriers permits to support a fine granularity of bit-rate values to suitably match traffic requests. As an example, considering 2.5Gb/s sub-carriers and four carriers: 2.5Gb/s, 5Gb/s, 7.5Gb/s, and 10Gb/s bit-rate values can be achieved.
- All-optical traffic aggregation: assuming non-overlapped sub-carriers generated by different segment modulators at different transceivers, such sub-carriers can be simply aggregated by relying on passive optical couplers, which would substitute electronic interfaces typically utilized to aggregate traffic. Passive all-optical traffic aggregation may contribute to build sustainable networks, given that the energy consumption of layer-2 or -3 devices, typically, linearly increases with traffic.

SCM achieved through SMZM permits to save 50% at the transmitter side considering that DSP constitutes around 50% of energy consumption of a transmitter supporting DSCM [Hui11].

In conclusion, we explored the possibilities to achieve frequency division multiplexing – that is the basement for digital sub-carrier multiplexing – through segmented Mach-Zehnder modulators. Simulations have shown that segmented Mach-Zehnder modulators may be an effective alternative to DSP for sub-carrier multiplexing (SCM) generation. Such an activity will be soon exploited to design and manufacture a segmented modulator for SCM.

GLOSSARY

Acronym	Description
2G	Second Generation
3GPP	Third-Generation Partnership Project
4G	Fourth Generation
AI/ML	Artificial Intelligence / Machine Learning
API	Application Programming Interface
AR/VR	Augmented Reality / Virtual Reality
AWG	Arrayed Waveguide Grating
BER	Bit Error Rate
BVT	Band Variable Transceiver
CMIS	Common Management Interface Specification
CU	Centralized Unit
DD	Direct Detection
DD-LMS	Decision-Directed Least Mean Square
DNN	Deep Neural Network
DPU	Data Processing Unit
DRB	Data Radio Bearer
DRL	Deep Reinforcement Learning
DU	Distributed Unit
ENC	Ensemble Controller
FC	Fault Condition
FEC	Forward Error Correction
GVD	Group Velocity Dispersion
GW	Gateway
IPM	Intelligent Pluggable Manager
IPoWDM	IP over Wavelength Division Multiplexing
K8s	Kubernetes
KPI	Key Performance Indicator
LLS	Linear Least Squares
MAS	Multi-Agent Systems
MBoSDM	Multi-Band over Spatial Division Multiplexing
MCF	Multi-Core Fiber
MLOps	Machine Learning Operations
NATS	NATS Messaging Protocol
NEP	Node Edge Point

NETCONF	Network Configuration Protocol
OFDM	Orthogonal Frequency Division Multiplexing
OLC	Optical Line Controller
OLL	Optical Line Layer
OLT	Optical Line Terminal
OLT	Optical Line Terminal
ONAP	Open Network Automation Platform
ONOS	Open Network Operating System
OPENROADM	OpenROADM Multi-Source Agreement
O-RAN	Open Radio Access Network
OSC	Optical Supervisory Channel
OSNR	Optical Signal-to-Noise Ratio
OTDR	Optical Time Domain Reflectometer
P2MP	Point-to-Multipoint
P2P	Point-to-Point
PdM	Predictive Maintenance
PON	Passive Optical Network
PRB	Physical Resource Block
QoT	Quality of Transmission
RAN	Radio Access Network
rApps	Applications for Non-Real-Time RIC
REST	Representational State Transfer
RESTful API	Representational State Transfer Application Programming Interface
RIC	RAN Intelligent Controller
RMBSA	Routing, Modulation, Band, and Spectrum Assignment
ROADM	Reconfigurable Optical Add-Drop Multiplexer
RU	Radio Unit
RUL	Remaining Useful Life
SAP	Service Attachment Point
SBI	South Bound Interface
S-BVT	Sliceable Bandwidth Variable Transceiver
SDM	Spatial Division Multiplexing
SDN	Software-Defined Networking
SLO	Service Level Objective
SNMP	Simple Network Management Protocol
SONiC	Software for Open Networking in the Cloud
TAPI	Transport API
TFS	TeraFlow SDN Controller
TWDM	Time-Wavelength Division Multiplexed

UC	Unified Communications (Not explicitly defined; assumed)
UL	Uplink
VDSL	Very-high-bit-rate Digital Subscriber Line
VNF	Virtual Network Function
WDM	Wavelength Division Multiplexing
WSS	Wavelength Selective Switch
xApps	Applications for Near-Real-Time RIC

REFERENCES

- [Cas23] R. Casellas, R. Vilalta, R. Martínez, and R. Muñoz, "SDN control of disaggregated optical networks with openconfig and openroadm," in 23th International IFIP Conference on Optical Network Design and Modeling (ONDM), (2023), pp. 452–464.
- [Cast24] Castoldi, P., Bakar, R. A., Sgambelluri, A., Olmos, J. J. V., Paolucci, F., & Cugini, F. (2025). Programmable packet-optical network security and monitoring using DPUs with embedded GPUs. *Journal of Optical Communications and Networking*, 17(2), A178-A195.
- [Che23] S. Chenumolu, "Open RAN deployments," Open RAN: The Defin. Guide pp. 145–171 (2023).
- [Con25] IETF. (2025). draft-ietf-teas-ns-controller-models-06: Network Slice Controller Models. Internet Engineering Task Force. <https://datatracker.ietf.org/doc/draft-ietf-teas-ns-controller-models/>.
- [D2.1-25] EU Project SEASON deliverable D2.1, (2024). "Definition of Use Cases, Requirements and Reference Network Architecture".
- [D2.3-25] EU Project SEASON deliverable D2.3, (2025). "Techno-economic Analysis".
- [D3.2-24] EU Project SEASON deliverable D3.2, (2024). "Optical systems enabling ultrahigh-capacity access/metro networks".
- [D3.3-25] EU Project SEASON deliverable D3.3, (2025). "The SEASON solution for data plane infrastructure".
- [D4.2-24] EU Project SEASON deliverable D4.2, (2024). "Second year design of control plane infrastructure".
- [D4.3-25] EU Project SEASON deliverable D4.3, (2025). "Final SEASON control plane Infrastructure"
- [D5.2-25] EU Project SEASON deliverable D5.2, (2025). "Final report of demo 1".
- [D6.3-25] EU Project SEASON deliverable D6.3, (2025). "Final project report on standardization, communication, and dissemination activities".
- [ETSI19] ETSI GS ZSM 002 V1.1.1 (2019-08), "Zero-touch network & Service Management (ZSM); Reference Architecture".
- [Her23] C. Hernández-Chulde, R. Casellas, R. Martínez, R. Vilalta, and R. Muñoz, "Experimental evaluation of a latency-aware routing and spectrum assignment mechanism based on deep reinforcement learning," *J. Opt. Commun. Netw.* 15, 925–937 (2023).
- [Hui11] Ron Hui, Andrea Fumagalli, "A digital subcarrier optical network utilizing digital subcarrier cross-connects with increased energy efficiency", WO2012083311A1, 2011.
- [Mar24] R. Martinez, C. Hernandez-Chulde, R. Casellas, R. Vilalta, R. Muñoz, O. Gonzalez de Dios, and J. P. Fernandez-Palacios, "Enhancing network performance and reducing power consumption in elastic optical networks with deep reinforcement learning," in 24th International IFIP Conference on Optical Network Design and Modeling (ONDM), (2024).

- [Mon22] S. Mondal and M. Ruffini, "Optical front/mid-haul with open access-edge server deployment framework for sliced O-RAN," *IEEE Trans. on Netw. Serv. Manag.* 19, 3202–3219 (2022).
- [Nad24] Nadal, L., Martínez, R., Ali, M., Vílchez, F. J., Fàbrega, J. M., Svaluto Moreolo, M. and Casellas, R., "Advanced Optical Transceiver and Switching Solutions for Next-Generation Optical Networks", accepted in *JOCN*, 16, 8, 2024.
- [ONF] Open Networking Foundation, "ONF transport API. [online] available: <https://github.com/opennetworkingfoundation/tapi>".
- [OpenConfig] OpenConfig, "Openconfig web site. [online] available: <http://www.openconfig.net/>," (2018).
- [OpenROADM] OpenROADM, "The open roadm multi-source agreement (MSA)". [online] available: <http://www.openroadm.org>.
- [OpenROADMMSA24] OpenROADM MSA, "OpenROADM Multi-Source Agreement Public Repository," GitHub, Release 16.0, Oct. 18, 2024. Available: https://github.com/OpenROADM/OpenROADM_MSA_Public. Accessed: Nov. 13, 2024.
- [Sal20] F. Saliou et al., "5G optics in 2020 - Where are we now? What did we learn?" in *ECOC*, (2020).
- [Wel21] D. Welch et al., "Point-to-multipoint optical networks using coherent digital subcarriers," *J. Light. Technol.* 39 (2021).
-