

Autonomous and Energy Efficient Lightpath Operation based on Digital Subcarrier Multiplexing

Luis Velasco, Sima Barzegar, Diogo Sequeira, Alessio Ferrari, Nelson Costa, Vittorio Curri, João Pedro, Antonio Napoli, and Marc Ruiz

Abstract—The massive deployment of 5G and beyond will require high capacity and low latency connectivity services, so network operators will have either to overprovision capacity in their transport networks or to upgrade the optical network controllers to make decisions nearly in real time; both solutions entail high capital and operational expenditures. A different approach could be to move the decision making toward the nodes and subsystems, so they can adapt dynamically the capacity to the actual needs and thus reduce operational costs in terms of energy consumption. To achieve this, several technological challenges need to be addressed. In this paper, we focus on the autonomous operation of Digital Subcarrier Multiplexing (DSCM) systems, which enable the transmission of multiple and independent subcarriers (SC). Herein, we present several solutions enabling the autonomous DSCM operation, including: *i*) SC quality of transmission estimation; *ii*) autonomous SC operation at the transmitter side and blind SC configuration recognition at the receiver side; and *iii*) intent-based capacity management implemented through Reinforcement Learning. We provide useful guidelines for the application of autonomous SC management supported by the extensive results presented.

Index Terms—Network Automation, Digital Subcarrier Multiplexing, Energy Savings, Intent-based Networking, Reinforcement Learning.

I. INTRODUCTION

DIGITAL Subcarrier Multiplexing (DSCM) systems, based on advanced digital signal processing (DSP) modules, represent a key technology to transparently route heterogenous data in an efficient and cost-effective way [1], [2]. The key-aspect of DSCM with

respect to single wavelength transmission is the usage of one single laser to digitally generate multiple Nyquist subcarriers (SC), e.g., 4, 8 or more, instead of a single one [3], [4]. One of the major advantages of using DSCM in optical transport networks is to keep high data-rates (e.g., 400 Gb/s) while using lower symbol rates (SR) per SC (e.g., 8 or 11 GBaud). As an example, a 32 GBaud system can be implemented using 4×8 GBaud SCs, multiplexed at near Nyquist sub-carrier spacing. This lowers the penalties caused by fiber propagation impairment, such as dispersion and nonlinear Kerr effects [5]. The DSCM is realized at the transmitter (Tx) side and each SC is individually detected and post-processed at the coherent receivers (Rx); SCs with different modulation format (MF), SR, and forward error correction (FEC) overhead can coexist. For an introduction to DSCM, we refer the reader to [4] and the guide in [6]. Note that although every SC could be independently routed through the network [7], in this paper, we assume that all the SCs in the optical connection (*lightpath*) are routed together from transmitter to receiver.

The flexibility provided by DSCM systems can be used to substantially reduce the energy consumption; for example, in case the actual amount of traffic that the lightpath needs to transport is low, the number of SCs that are active can be reduced. This represents a step further in terms of flexibility with respect to the sliceable-bandwidth variable transponder proposed in [8], by achieving a higher granularity and increasing the flexibility thanks to the digital generation of the SCs; this might be useful, especially for metro applications [9]. In fact, of the total power consumption of a coherent optical transponder, over 38% comes from subsystems that could be switched off in case a SC can be de-activated [10]. In addition, the authors in [11] show power increments of ~28% for an optical transponder when increasing the order of the Quadrature Amplitude Modulation (QAM) over Quadrature Phase-Shift Keying (QPSK) and about the same ratio when increasing the symbol rate from 32 to 43 GBaud. Therefore, a proper dynamic configuration of the SCs in terms of MF and SR to adapt the capacity of the

Manuscript received July 10, 2020.

The research leading to these results has received funding from the European Community's through the MSCA REAL-NET project (G.A. 813144), by the AEI/FEDER through the TWINS project (TEC2017-90097-R), and by the ICREA institution.

Luis Velasco (luis.velasco@upc.edu), Sima Barzegar, Diogo Sequeira, and Marc Ruiz are with the Optical Communications Group at Universitat Politècnica de Catalunya, Barcelona, Spain. Alessio Ferrari and Vittorio Curri are with Politecnico di Torino, Turin, Italy. Nelson Costa is with Infinera, Lisbon, Portugal. João Pedro is with Infinera and with Instituto de Telecomunicações, Lisbon, Portugal. Antonio Napoli is with Infinera, Munich, Germany.

lightpath to the actual traffic can lead to non-negligible energy savings, since the power consumption of several subsystems, including those that are always active, depends on the actual configuration of the SCs.

Some works can be found in the literature exploiting the inherent capability of Elastic Optical Networks (EON) [12] to adapt lightpaths as a function of the actual traffic, and to create multiple parallel lightpaths to offer a combined capacity as seen from the virtual link (vlink) perspective in the packet layer (see e.g., [13], [14]). In this work, we target at extending EON capabilities by adopting dynamic capacity management of SC-operated lightpaths. To implement such dynamic configuration of the SCs, both the Tx and the Rx sides must be synchronized to properly encode and decode the optical signals of every SC. To that end, the Software Defined Networking (SDN) controller can be in charge of configuring each SC of every lightpath by programming both sides with the specific configuration. However, this would significantly increase the number of tasks to be performed within the centralized controller (i.e., analyzing measured traffic, finding optimal SC configuration, and re-tuning Tx and Rx when needed for every single lightpath) and, because it requires near real-time operation, it would introduce more complexity in the whole network.

In this paper, we propose a decentralized approach where the intelligence is distributed [15]; the Tx decides autonomously the SC configurations, i.e., the SC's MF and SR, and the Rx can detect and recognize such configurations. Previous works in the literature have proposed methods for the blind recognition of the MF based on the received constellation, see, e.g., [16], [17]. Nonetheless, such methods require high signal-to-noise ratio (SNR) for accurate recognition, e.g., $\text{SNR} > 20$ dB and/or they do not consider the SR configuration, which limits its applicability. Alternatively, one could implement a communication protocol between the Tx and the Rx, so that the Tx, e.g., by first configuring the new SC with a robust and fixed configuration (e.g., QPSK and 8 GBaud), sends messages to the Rx with the selected configuration for the SC. In this paper, however, we propose an improved method based on analyzing the received constellation that was shown to provide extraordinary accuracy even in a low SNR regime. To keep the SDN controller informed about the current configuration, both Tx and Rx sides can issue notifications of SC-related events.

Nevertheless, the role of the SDN controller is of paramount importance as it solves the routing and spectrum allocation (RSA) [18] for the request, thus providing a lightpath in the optical network topology that satisfies not only the needed capacity but also the required Quality of Transmission (QoT) for every SC. Note that individual SCs are impaired differently when they cross Wavelength Selective Switches (WSS),

which are part of Reconfigurable Optical Add-Drop Multiplexers (ROADM). To this end, software tools like the open source GNPY [19], being developed within the Telecom Infra Project, can be used but need to be upgraded to properly consider the filter penalties.

Another issue to be addressed is how the lightpath can be managed to provide enough capacity to the packet layer without detrimental effects, like added delays or packet losses, and at the same time minimize energy consumption. In this regard, advances in network automation [20] are receiving considerable attention from the industry as the complexity of the network increases and the requirements from the services become more stringent and diverse. Network automation entails the collection of performance monitoring data that are analyzed (e.g., using Artificial Intelligence (AI) / Machine Learning (ML) techniques), and the extracted knowledge is used to make decisions, thus defining control loops. However, several aspects and practices related to network automation require standardization and in fact novel concepts, such as for example, the Intent-based Networking (IBN) [21], are currently under study within standardization bodies. IBN allows the definition of operational objectives that a network entity, e.g., a lightpath, has to meet without specifying how to achieve them. IBN is in charge of implementing and enforcing those goals, often with the help of AI/ML. In this paper, we go beyond network automation by demonstrating an IBN solution based on Reinforcement Learning (RL) [22]. RL fits well with IBN as it entails learning on how to map situations to actions to maximize rewards, without specifically programming the learner.

The remainder of the paper is organized as follows. Section II introduces the proposed solution for autonomous SC operation, where an intent, with the operational objective to allocate enough capacity, is in charge of autonomously making decisions to configure and activate or deactivate SCs at the Tx side. Section III focuses on lightpath provisioning and includes an estimation of per-SC QoT, which is later used to solve the RSA with SC (RSA-SC) configuration problem. Section IV proposes algorithms to manage the SC at the Tx and Rx sides; the Transponder Agent at the Tx side is in charge of managing the capacity of the lightpath by properly configuring the different SCs to minimize energy consumption, whereas the agent at the Rx side needs to recognize the configuration of each SC. Section V is devoted to our intent-based solution and two approaches are proposed to anticipate future traffic conditions thus minimizing or totally eliminating undesirable effects at the packet layer. The first solution uses a *short-term* traffic forecast, whereas the second one is based on RL. The discussion is supported by extensive illustrative results in Section VI, including exhaustive simulations over realistic scenarios. Finally, Section VII draws the main conclusions of our work.

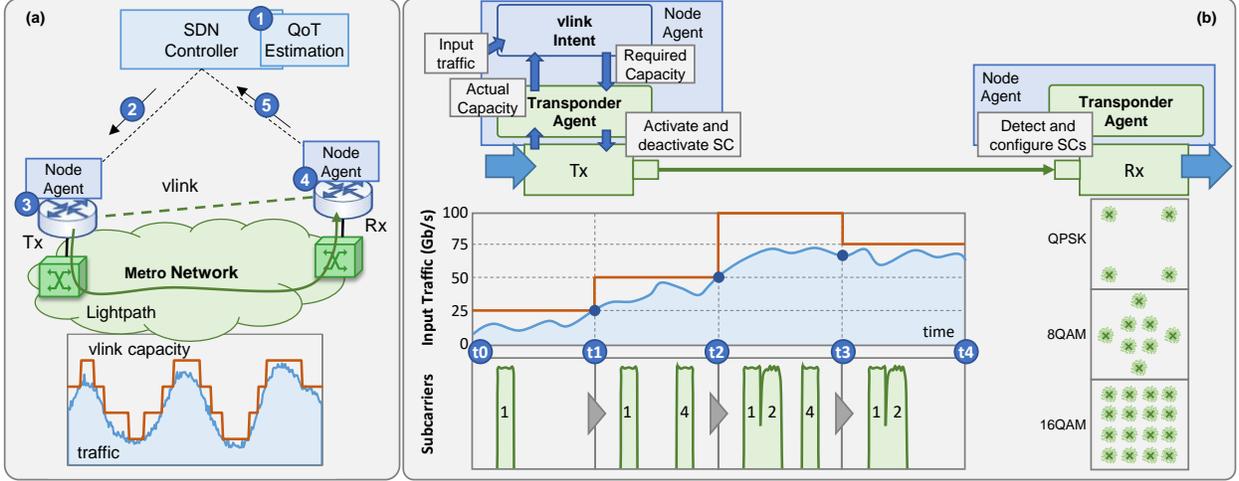


Fig. 1. Vlink workflow overview (a) and intent-based autonomous vlink operation to allocate capacity for the input traffic (b).

II. AUTONOMOUS CAPACITY MANAGEMENT

As previously stated, we propose the autonomous operation of the vlinks performed locally at their end nodes and being able to recognize in one end the decisions made at the remote end. This is particularly important at the optical layer, as the Rx optical transponder of the lightpath supporting the vlink needs to realize the actual configuration of the signal that was transmitted. From the packet layer perspective, it is also desirable to devise solutions that: *i*) do not introduce delays and traffic losses; and *ii*) minimize the changes in the capacity of the vlink to facilitate the management of such capacity at the packet nodes.

Fig. 1a presents the key elements in the control and data planes and overviews the envisioned workflow of vlinks. At the set-up time, the SDN controller receives a request to create a vlink between two end nodes with a given maximum capacity. The SDN controller computes the best route of the underlying lightpath and estimates the QoT. Based on such estimation, the SDN controller determines the number of SCs and a *configuration map* with the set of feasible configurations for each SC s in terms of pairs $\langle MF_s, SR_s \rangle$ that can be used; this would allow the autonomous SC operation and will support the given maximum capacity (step 1 in Fig. 1a). Then, the SDN controller establishes the lightpath by allocating the needed optical spectrum along the path connecting Tx and Rx, and sends the configuration map to the vlink intent in charge of the autonomous vlink operation (step 2). The SDN then relies on the vlink intent to manage the capacity of the vlink as a function of the input traffic (step 3). An illustrative example of such capacity management is displayed in the inset in Fig. 1a, where the capacity of the vlink (orange line) increases and decreases to serve the input traffic (blue area). The vlink intent communicates the desired capacity of the vlink to the *Transponder Agent*, which activates or deactivates

SCs in the underlying optical signal. Such decision making is performed within the Tx and consequently, a mechanism is needed for the Rx to detect and configure SCs' MF and SR, as well as to detect the absence of a SC (step 4). Finally, notifications are sent to the SDN controller informing about changes in the vlink and the optical signal (step 5).

Fig. 1b presents an example of the evolution over time of the input traffic and the active configuration of the optical signal, as decided by the Tx, to meet the capacity requirements of the vlink intent. Once the connection is set up by the SDN controller at time t_0 , the vlink intent observes the amount of input traffic and makes decisions consequently. For instance, based on required capacity, the transponder agent within the Tx configures the first 8 GBaud QPSK SC (SC1) at t_0 , which entails that the connection capacity is 25 Gb/s, considering a FEC overhead of 20%. Based on periodical capacity updates, received from the vlink intent, the Tx increases connection's capacity to 50 Gb/s at t_1 by activating SC4 with the same configuration as for SC1. The capacity is increased to 100 Gb/s at time t_2 by activating SC2 configured as 11 GBaud 8QAM. If the input traffic would continue increasing, the last SC (SC3) would be activated, e.g., with an 11 GBaud 16QAM configuration to increase capacity up to 170 Gb/s. On the contrary, if the capacity requirements would decrease (as in t_3), one or more SCs could be turned off, thus saving energy.

Note that the decisions made in the SC configuration can limit the maximum capacity of the vlink. For instance, if all SCs can be configured with QPSK, 8QAM or 16QAM MFs and 8 or 11 GBaud, the maximum capacity of the vlink would lie in the range [100-280] Gb/s depending on the selected configuration. Therefore, assuming that SCs cannot be reconfigured (i.e., to change the configuration of a SC, the SC needs to be deactivated and then activated again with a

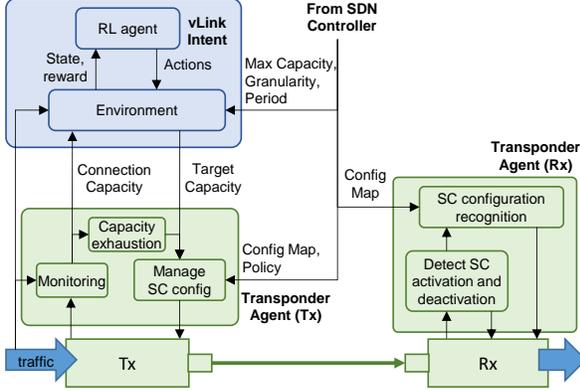


Fig. 2. Relationship between vlink intent and Transponder Agent.

different configuration, which would cause packet losses if the SC is supporting traffic), such decision-making process need to be intelligently carried out to maximize the potential capacity of the connection without traffic disruption, while saving as much energy as possible. For instance, in view of the evolution of the traffic at t_1 , the vlink intent could have requested a capacity of 75 Gb/s, so that the Tx would have configured SC2 as a 11 GBaud 16QAM signal and then, deactivated SC4, thus resulting in the same performance but with lower total energy and configuration changes. In addition, note that not all SCs can support all different configurations, as this depends on the expected SNR at the receiver. For example, a cascade of filters will affect the “external” SCs more severely than the inner ones, thus limiting the MF order of the former to just QPSK. This introduces additional complexity in the decision-making process. Therefore, we propose a vlink intent based on RL, whereas the algorithm in the transponder, making decisions on the configuration of the SCs, can be based on policies received from the SDN controller (Fig. 2). At the Rx side, the transponder agent is in charge of detecting the presence or absence of SCs and properly configuring the new ones.

The modules that make possible the envisioned autonomous vlink operation are presented next.

III. LIGHTPATH PROVISIONING

In this section, we first focus on abstracting the relevant information for QoT estimation, and then we introduce the problem to be solved in the SDN controller during lightpath provisioning as a response to a connection request. We name this as the RSA-SC problem, which includes the estimation of the QoT used to derive the configuration map with the feasible configurations of the SCs.

We assume a state-of-the-art transparent SC-operated optical network, whose topology can be represented by a graph $G(N,E)$, where N is the set of ROADMs and E is the set of bidirectional amplified Optical Line Systems (OLS) connecting the ROADMs. The ROADMs

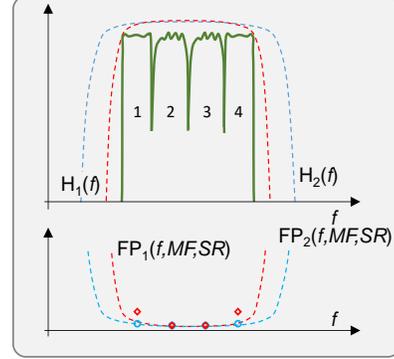


Fig. 3. Filter transfer functions $H_{1,2}(f)$ over a SC and respective filter penalty $FP_{1,2}(f, MF, SR)$.

perform transparent routing of selected lightpaths, introducing optical filtering penalties.

A. Subcarrier QoT Estimation

If the symbol rate and fiber dispersion are sufficiently high, which is usually the case for practical scenarios, the transparent propagation of each SC s in the set S of SCs belonging to an optical connection is affected by two different sources of additive noise: *i*) the amplified spontaneous emission (ASE) noise introduced by Optical Amplifiers (OA); and *ii*) the Non-Linear Interference (NLI) generated by cross- and self- channel nonlinear crosstalk triggered by the fiber Kerr effect during propagation over each fiber span [23]. As the ASE noise, NLI is also a Gaussian distributed interference. Since the phase noise part of the NLI is typically compensated by DSP, the transmission channel for each SC propagating over a given lightpath can be modelled as an additive Gaussian Noise Nonlinear channel [24]. When affected by additive Gaussian noise, the Bit Error Rate (BER) for MFs with coherent receivers is univocally set by the SNR [25]. The ASE noise introduced by each intermediate OA depends on its gain and noise-figure. The NLI introduced by each fiber span depends on the launch power, on fiber characteristics, and on those of each active SC. For a spectrally resolved NLI evaluation, models must consider also the simultaneous effect of stimulated Raman scattering [26].

The QoT for SC s in frequency slot f , propagating over a given lightpath, is determined by the generalized signal-to-noise ratio (GSNR) including the accumulated ASE noise $P_{ASE;s,f}$ and NLI $P_{NLI;s,f}$. Each OLS controller determines the working point of the line by setting the gain of each OA and, consequently, the power levels, NLI, and ASE noise. So, given the OLS controller strategy and using data of fiber spans, performance impairments on the SC (s,f) of each OLS within the network topology is summarized (*abstracted*) by $GSNR_{e;s,f}$ in Eq. (1):

$$\begin{aligned} \text{GSNR}_{e:s,f} &= 10 \log_{10} \left(\frac{P_{s,f}}{P_{\text{ASE};s,f} + P_{\text{NLI};s,f}} \right) \\ &= 10 \log_{10} \left(\frac{1}{\text{OSNR}_{s,f}^{-1} + \text{SNR}_{\text{NL};s,f}^{-1}} \right), \end{aligned} \quad (1)$$

where $P_{s,f}$ is the received SC power, Optical SNR (OSNR) represents SNR due to the ASE noise, whereas SNR_{NL} is the NLI contribution. Note that Eq. (1) defines the GSNR impairing the SC(s,f) if propagating on the OLS e only. In order to avoid spectral-load-dependent performance degradation, $\text{GSNR}_{e:s,f}$ is evaluated supposing full spectral load.

Assuming a typical root-raised-cosine spectral shape, every individual SC s has a spectral occupation $B_s = (1+\rho) \cdot SR_s$, where ρ is the roll-off factor. Besides, a guard-band B_{GB} separates the SCs. Then, the spectral occupation for the overall set of SCs S :

$$B_{ch} = (|S| - 1) \cdot B_{GB} + \sum_{s \in S} (1 + \rho) \cdot SR_s \quad (2)$$

Optical filtering leads to signal distortion, which is modelled as an equivalent SNR penalty. Given a slot width $b \geq B_{ch}$, and presuming a uniform WSS transfer function $H(f)$ for each frequency slot and slot width, the amount of SNR filtering penalty FP_s , depends on the SC index s and the SC configuration $\langle MF, SR \rangle$ (Fig. 3). Hence, ROADMs can be abstracted by a list $FP_s(b, MF, SR)$ of values (in dB), for each different slot width and MF to be supported; $FP_s(b, MF, SR)$ values can be experimentally characterized or analytically evaluated supposing matched-filtering at the Rx [25].

The described scenario enables a full abstraction of the QoT information for a SC-operated transparent network topology described by a graph $G(N, E)$, where every OLS e can be characterized by the matrix $[\text{GSNR}_{e:s,f}]$, while every ROADM n is characterized by the matrix $[FP_{n:s}(b, MF, SR)]$ with a vector of penalties for different frequency slot widths and SC configurations. Relying on such abstraction, the estimation of the QoT for every SC to be transmitted through the lightpath $l = \langle p, f, b \rangle$, that consists of a path p and a frequency slot f of width b , can be computed as described by Eq. (3). Table I summarizes the notation.

Equation (3) can be used to solve the RSA-SC problem by evaluating the OSNR and the SNR_{NL} contributions; the open-source library GNPY [19] can be used for that purpose.

$$\begin{aligned} \text{GSNR}_s(l, MF, SR) &= \\ &10 \log_{10} \left(\frac{1}{\sum_{e \in E(l(p))} \text{OSNR}_{e:s,l(f)}^{-1} + \text{SNR}_{\text{NL};e:s,l(f)}^{-1}} \right) \\ &- \sum_{n \in N(l(p))} FP_{n:s}(l(b), MF, SR) \end{aligned} \quad (3)$$

TABLE I: NOTATION

N	Set of optical nodes, including transponders and ROADMs. Index n .
E	Set of optical line systems. Index e .
l	Lightpath, defined by the tuple $\langle p, f, b \rangle$
p	Route of the lightpath (ordered list of nodes).
f	Frequency slot.
b	Bandwidth of the frequency slot.
S	Set of subcarriers in a lightpath. Index s .
ρ	Roll-off factor

B. The RSA-SC problem

Given a request with the lightpath end ROADMs and the maximum capacity to be supported, we can define the RSA-SC provisioning problem that finds a proper lightpath $l = \langle p, f, b \rangle$, as well as the *SC definition*; the latter includes: *i*) the tuple $\langle |S|, B_{ch} \rangle$ with the number of SCs and the channel spectral width; and *ii*) the maximum configuration of each SC s in terms of MF and SR, with the objective of minimizing or maximizing a defined cost function, e.g., to minimize spectrum usage. The problem can be formally stated as:

Given:

- a set L of precomputed lightpaths over a connected directed graph $G(N, E)$; ROADMs and OLSs are defined with the abstracted QoT data needed to enable lightpath QoT estimation given by Eq. (3);
- the availability of every frequency slice in the optical spectrum of every OLS e in E ;
- a connection request to be served, defined by the tuple $\langle origin, termination, capacity \rangle$;
- a function that verifies whether a candidate lightpath l connecting origin and termination ROADMs meets the QoT requirements for the selected frequency slot and spectrum width, given a selected SC configuration that supports the required capacity;
- the set of *SC definitions* enabled by the end transponders that can support the given capacity and the required QoT.

Output: the lightpath and the SC definition.

Objective: Minimize the used resources.

In view of the complexity of the RSA-SC to be solved in real time, we propose to use an approximate approach like the Routing, Modulation and Slot Assignment (RMSA) problem in [27], where a set of k distinct routes connecting origin and destination ROADMs can be precomputed on $G(N, E)$. Then, given: *i*) the set of SC definitions; *ii*) the minimum SNR for every SC configuration; and *iii*) the maximum required capacity, every candidate lightpath can be verified in terms of estimated QoT and capacity, for a set of possible spectral widths.

ALGORITHM I. LIGHTPATH VERIFICATION AND SC DEFINITION

Input: $l = \langle p, f, \max B \rangle$, $SCDefs$, $\min SNR[]$, B , $capacity$
Output: $AllowedSCDefs$

```

1:  $AllowedSCDefs \leftarrow \emptyset$ 
2: for each  $scd$  in  $SCDefs$  do
3:   if  $capacity(scd) < capacity$  OR
      $scd.B_{ch} > l.maxB$  then continue
4:    $scdOK \leftarrow TRUE$ 
5:   for each  $b$  in  $B$  do
6:     if  $b < scd.B_{ch}$  then continue
7:      $lp \leftarrow \langle l.p, l.f, b \rangle$ 
8:     for each  $s$  in  $scd.S$  do
9:        $conf \leftarrow scd[s].config$ 
10:      if  $GSNR(lp, conf) < \min SNR[conf]$  then
11:         $scdOK \leftarrow FALSE$ ; break
12:      if  $scdOK$  then  $AllowedSCDefs \leftarrow$ 
         $AllowedSCDefs \cup \{ \langle lp, scd \rangle \}$ 
13: return  $AllowedSCDefs$ 

```

Algorithm I presents such a function; it initializes the set of allowed SC definitions to be returned as output (line 1). Then, for each SC definition supported by the transponders, the configuration is checked to verify whether: *i*) it supports the required capacity; and *ii*) the required spectral width does not exceed the maximum available in the route of l (lines 2-4). Next, the QoT of the candidate lightpath is verified for the different spectrum widths in set B , provided that such width supports the one required by the SC configuration (lines 5-6); a specific lightpath for the selected spectrum width is created (line 7). Then, the QoT of every SC in the examined SC configuration is evaluated for the specific lightpath and, if all SCs in the SC configuration satisfy the QoT (lines 8-11), it is added to the set of allowed configurations (line 12); the set is eventually returned (line 13).

Once the RSA-SC problem is solved, the SDN controller establishes the lightpath by allocating the selected frequency slot and spectrum width along the defined path and programming the optical transponders with the SC definition, specifying the configuration of every SC, and the maximum capacity to be ensured.

IV. TRANSPONDER AGENT

When the lightpath is established, the actual capacity that is provided by the connection can be managed by activating and configuring each SC independently as a function of the traffic to be conveyed. In this section, we first present the procedures for the Tx to select the configuration of the optical connection, including which SCs are active and their configuration to provide the needed capacity. Then, we discuss different options for synchronizing such configuration with the Rx. We assume that the SC definition contains a SC configuration map with all possible options, from those allowed by the transponder, that do not exceed the maximum configuration received from the controller.

ALGORITHM II. MAXSC CAPACITY MANAGEMENT

Input: $conn$, $configMap$, $targetCapacity$
Output: s , $config$

```

1:  $diffCap \leftarrow targetCapacity - conn.capacity$ 
2: if  $diffCap > 0$  then
3:    $\langle s, config \rangle \leftarrow \operatorname{argmax} \{ s.getMaxConfig(configMap) \mid$ 
      $\forall s \in conn \mid NOT s.isActive() \}$ 
4: return  $\langle s, config \rangle$ 
5:  $config \leftarrow \emptyset$ 
6:  $s \leftarrow \operatorname{argmin} \{ s.capacity \mid \forall s \in conn \mid$ 
      $s.isActive() \text{ AND } s.capacity < -diffCap \}$ 
7: return  $\langle s, config \rangle$ 

```

ALGORITHM III. ADAPTIVE CAPACITY MANAGEMENT

Input: $conn$, $configMap$, $targetCapacity$
Output: s , $config$

```

1:  $diffCap \leftarrow targetCapacity - conn.capacity$ 
2: if  $diffCap > 0$  then
3:    $activeSC \leftarrow \operatorname{argmax} \{ sc.upgradability \mid \forall s \in SC \mid$ 
      $sc.upgradability = sc.getMaxConfig(configMap) -$ 
      $diffCap \}$ 
4:   if  $activeSC \neq \emptyset$  then
5:      $\langle s, config \rangle \leftarrow \operatorname{findNonActiveSC}(conn, configMap,$ 
      $activeSC.capacity + diffCap)$ 
6:     if  $s \neq \emptyset$  then return  $\langle s, config \rangle$ 
7:      $\langle s, config \rangle \leftarrow \operatorname{findNonActiveSC}(conn, configMap,$ 
      $diffCap)$ 
8:     if  $s \neq \emptyset$  then return  $\langle s, config \rangle$ 
9:      $\langle s, config \rangle \leftarrow \operatorname{argmax} \{ s.getMaxConfig(configMap) \mid$ 
      $\forall s \in conn \mid NOT s.isActive() \}$ 
10:    return  $\langle s, config \rangle$ 
11:    $config \leftarrow \emptyset$ 
12:    $s \leftarrow \operatorname{argmax} \{ s.capacity \mid \forall s \in conn \mid$ 
      $s.isActive() \text{ AND } s.capacity < -diffCap \}$ 
13:   if  $s = \emptyset$  AND  $conn.numActiveSCs = 1$  then
14:      $\langle s, config \rangle \leftarrow \operatorname{findNonActiveSC}(conn,$ 
      $configMap, targetCapacity)$ 
15:   return  $\langle s, config \rangle$ 

```

ALGORITHM IV. FINDNONACTIVESC() FUNCTION

Input: $conn$, $configMap$, $capacity$
Output: s , $config$

```

1:  $SC = \{ s \} \leftarrow \operatorname{argmin} \{ s.overProvision \mid \forall s \in conn \mid$ 
      $NOT s.isActive() \text{ AND } s.overProvision =$ 
      $s.getMinConfig(configMap, capacity) - capacity \}$ 
2:  $\langle s, config \rangle \leftarrow \operatorname{argmin} \{ s.upgradability \mid \forall s \in SC \mid$ 
3:    $s.upgradability = s.getMaxConfig(configMap) -$ 
      $s.getMinConfig(configMap, capacity) \}$ 
4: if  $s \neq \emptyset$  AND  $conn.numNonActiveSCs = 1$  then
5:   return  $\langle s, s.getMaxConfig(configMap) \rangle$ 
6: return  $\langle s, config \rangle$ 

```

A. Capacity Management at the Transmitter Side

We propose two different policies for the autonomous management of the optical connection: *i*) the *maxSC* connection management policy, which allocates the SC with maximum possible capacity, thus potentially minimizing the changes in the connection's configuration; and *ii*) the *adaptive* connection management policy, which allocates the minimum capacity to satisfy the input traffic needs, targeting at

minimizing the excess of capacity with respect to the actual traffic with the goal of eventually minimizing the total energy consumption. Both algorithms return one single SC change in the connection configuration. Therefore, they must be executed within an outer loop until they cannot perform more changes (i.e., returning $\langle s=\emptyset, config=\emptyset \rangle$).

The pseudocode for *maxSC* connection management policy is presented in Algorithm II. The algorithm receives as input the state of the connection in terms of the configuration of the SCs, the map of feasible configurations for each SC, and the target capacity for the connection. It returns the SC to be activated/deactivated, if any, and its target configuration. The algorithm computes the difference of capacity between the current and the requested ones (line 1). In case the capacity of the connection needs to be increased, the SC providing the largest capacity among those not yet active is selected (lines 2-3). On the other hand, the capacity of the connection is reduced by selecting the active SC with the lowest capacity, whose deactivation leaves enough capacity in the connection (lines 4-6). Finally, the selected SC, if any, and the target configuration are returned (line 7).

This policy enables minimizing the number of configurations to be supported thus potentially reducing the cost of the transponders. It selects the SC with the largest capacity among the non-active ones; this minimizes the number of changes in the capacity of the connection, which facilitates its management at the packet layer. Nonetheless, this policy might lead to large energy consumption due to the overprovisioning, since the selection of the SC to be activated does not depend on the actual capacity needs.

Algorithm III presents the *adaptive* connection management policy, which targets at minimizing the energy consumption by configuring the connection capacity as close as possible to the actual needs. In case that an increment of capacity is requested, the algorithm identifies the active SC with the largest potential capacity increase (lines 2-3 in Algorithm III), so another SC can be activated supporting the requested capacity plus the identified active SC (lines 4-6). In that case, a consecutive execution of the algorithm will find that the first SC can be deactivated. If any active SC is identified, a non-active SC needs to be activated, which is configured to support the requested increment of capacity (lines 7-8). In case that the increment of capacity cannot be supported by just one single SC, the non-active SC with the maximum capacity is returned (lines 9-10). In the case of reducing capacity, a target SC is identified, so the total capacity of the connection still supports the requested one (lines 11-12). If no SC can be deactivated and there is just one single SC activated, the algorithm tries to activate one SC so all

ALGORITHM V. CONFIGURATION RECOGNITION ALGORITHM

Input: $s, configList$

Output: $foundConfig$

```

1:  $foundConfig \leftarrow \langle MF = \emptyset, SR = \emptyset \rangle$ 
2:  $minDistance \leftarrow \infty$ 
3:  $SRList = getSRList(configList)$ 
4: for each  $SR$  in  $SRList$  do
5:    $constellation \leftarrow getConstellation(s, SR)$ 
6:    $\langle MF, d \rangle \leftarrow findFittestModFormat(constellation,$ 
                                      $getMFList(configList, SR))$ 
7:   if  $d < minDistance$  then
8:      $foundConfig \leftarrow \langle MF, SR \rangle$ ;  $minDistance \leftarrow d$ 
9: return  $foundConfig$ 

```

target capacity can be served (lines 13-14). Again, a consecutive execution of the algorithm will find that the first SC can be deactivated, so just one single SC will remain active.

The algorithm for the *adaptive* connection management policy calls the `findNonActiveSC()` function. This identifies the non-active SC providing the minimum overprovisioning (line 1 in Algorithm IV) and among them, the one that will be closer to its maximum capacity (line 2). The configuration that better fits the required capacity is returned, except if there is just one single non-active SC, where the maximum configuration is returned (lines 4-6).

B. SC Configuration Recognition at the Receiver Side

The Rx oversees the detection of presence or absence of SCs and of configuring properly the new ones. The approach proposed in this paper relies on matching the received constellation with the MF, which is characterized by a defined set of constellation points. Therefore, the distance between the received and the expected constellation points can be computed as averaged summation of the minimum squared Euclidean distances; this is similar as in a traditional clustering problem [20] but in this case the *centroids* are predefined by the constellation points of the selected MF. To find the right SC configuration, such computation is performed for all possible configurations for a given SC, and the one that minimizes the distance is selected.

Algorithm V presents the pseudocode for the SC configuration recognition at the Rx side; given SC s and a list of possible configurations for that SC, it returns the most likely configuration. After the initialization (lines 1-2 in Algorithm V), the signal is decoded and the list of possible symbol rates is obtained from the configuration list (line 3). Then, for each possible SR, the signal is processed, and the observed constellation of symbols in the complex space is obtained (lines 4-5). Using that processed constellation, the MF that minimizes the distance is selected from the list of possible MFs for the selected SR (line 6). Then, if the distance is smaller than the minimum obtained so far,

the configuration is stored (lines 7-8). Finally, the most likely configuration is returned.

V. VIRTUAL LINK INTENT-BASED CAPACITY MANAGEMENT

The capacity management algorithms, introduced in the previous section, require measuring the input traffic to determine the capacity that needs to be ensured. However, such monitoring reflects the capacity that was needed at the time the measurement was performed, so in case that the target capacity exceeded the one of the optical connections, some traffic might need to be queued waiting for more capacity to be available; this is at the cost of introducing delay or even traffic losses in case the queue capacity is exceeded. In addition, this approach is purely reactive, as if the traffic volume reduces, the capacity of the connection reduces as well, which can derive into capacity oscillations making difficult for the packet nodes managing it. Therefore, it is desirable to devise solutions that not only reduce the energy consumption by managing the configuration of the SCs, but also predict: *i*) how to configure the SCs before the capacity is *exhausted* to avoid introducing delays and traffic losses; and *ii*) how to reduce the changes in the vlink capacity. To this end, we propose a capacity management to be carried out at the vlink level and keep the capacity management at the transponders as a back-up in case the prediction fails and the current capacity is exhausted, so that the measured traffic volume is always ensured provided that is under the maximum capacity to be guaranteed.

The first solution is to use *short-term* traffic forecasts, instead of the actual measured traffic, to *predict* future capacity needs. This short-term prediction can be obtained by projecting forward the last monitored traffic samples, i.e., interpolating traffic in the last w measurements and extrapolating for next δ time units [28]. Although this approach can help to anticipate potential capacity exhaustion by allocating capacity in advance, when the traffic volume decreases, the reduction of capacity can lead to undesired capacity fluctuations and to increase queuing.

In view of the above, we propose a RL model for vlink capacity management based on Q-learning [22]. This approach aims at managing the capacity of the vlink as a function of the current and predicted traffic volume, while avoiding queuing (and traffic losses) and keeping moderated SC configuration changes by removing oscillating capacity allocation actions. Q-learning is a model-free discrete RL technique that involves n^s states and n^a actions, and it can learn the optimal policy represented by a Q-table of pairs $\langle \text{state}, \text{action} \rangle$. After taking an action, the RL agent observes the new state and gained reward, and updates the Q-table accordingly [22]. In our approach, the computation

TABLE II: STATE AND REWARD COMPONENTS

$y(t)$	max input traffic at t (b/s)
$y(t+\delta)$	Expected max input traffic at $t+\delta$ (b/s)
$y'(t)$	$\max(y(t), y(t+\delta))$
$z(t)$	current capacity limit (b/s)
$\Delta y(t)$	Derivative of max input traffic (b/s)
$\Delta z(t)$	Derivative of capacity limit (b/s)
k	Capacity unit size (b/s)
α	Capacity overprovisioning factor

TABLE III: REWARD FUNCTION COMPONENTS

i	$f_i(\mathcal{X}(t))$	Comment	φ_i
0	-	<i>null component</i>	0
1	$y(t) > z(t)$	<i>Capacity limit violation</i>	-1000
2	$z(t) - y(t) > \alpha \cdot k$	<i>Cap. overprovisioning</i>	-100
3	$\text{sgn}(\Delta y(t)) \neq \text{sgn}(\Delta z(t))$	<i>Trend mismatch</i>	-10
4	$\text{sgn}(\Delta y(t)) = \text{sgn}(\Delta z(t))$	<i>Trend match</i>	100

of state and reward, as well as the actions to be taken, are evaluated periodically (e.g., every minute) when a new set $\mathcal{X}(t)$ of measurements, statistics i.e., *max*, *min*, *avg* of the last monitoring samples, and parameters are available for the connection (in RL, this is traditionally called *episode*). Table II presents the main notation that will be consistently used for state and reward function definition.

The state at time t has been defined to reflect how close the used capacity is to its current limit either at time t or to the expectation at time $t+\delta$ computed using the aforementioned traffic prediction method. For the sake of simplicity, we assume that $\mathcal{X}(t)$ contains the trend (slope) computed with the last m time periods, so linear extrapolation to forecast expected traffic at $t+\delta$ is a straightforward operation. Eq. (4) presents a formal definition of the state, where $y' \in \mathcal{X}$ is the selected measurement for traffic and $z \in \mathcal{X}$ is the current capacity.

$$s(t) = \min([n^s \cdot y'(t)/z(t)], n^s) \quad (4)$$

The reward function r is generalized as follows:

$$r(t) = \varphi_0 + \sum_{i \in m} \varphi_i \cdot f_i(\mathcal{X}(t)) \quad (5)$$

where the Boolean function $f_i(\cdot)$ returns 1 if the condition is met and 0 otherwise, and the coefficient φ_i penalizes (<0) or promotes (>0) the result of the operation; besides, φ_0 ensures a non-zero result. Table III describes all components by means of both a brief description and a formula, as well as it defines the values for each coefficient. Capacity violation is strongly penalized, much more than provisioning excess of capacity; these two conditions assure a proper capacity allocation avoiding queuing and traffic loss. In addition, trend mismatch component aims at eliminating undesired capacity fluctuations by penalizing cases where the evolution of traffic and capacity (i.e., the sign of the derivative) follows opposite trends. On the

contrary, trend match promotes with a positive reward the case when both traffic and capacity evolve in the same direction.

With the state and the reward computed in the so-called *environment*, the RL agent learns and takes actions that maximize the expected total reward over all successive episodes. The RL agent selects the action with highest total reward with probability $1-\varepsilon$ (*exploitation*) or randomly with probability ε (*exploration*) [22], where exploration is encouraged during an initial learning phase and a decay strategy is implemented. Specifically, actions are defined as the number of k -size capacity units to be added/subtracted to the current vlink capacity; such operation is performed in the environment and the result is requested as target capacity to the transponder agent.

VI. ILLUSTRATIVE RESULTS

In this section, we first introduce the power model used for comparing the proposed SC management solutions, and then we study the performance of different SC configurations and the QoT estimation, with the aim of finding the operational limits of DSCM for autonomous capacity management. Next, the results of the SC configuration recognition algorithms are presented, being the basis to analyze afterwards the performance of the capacity management performed at the transponder or by the vlink intent.

A. Power Model

The results in the following subsections present power savings that have been computed assuming a power model aiming to abstract the actual realization of a DSCM transponder. The power model is based on the conclusions from [10] and [11] that can be summarized as follows: *i*) the power consumption of a transponder is related to its actual configuration in terms of MF and SR, and *ii*) the architecture of a DSCM transponder contains subsystems that are always active independently of the number and configuration of active SCs (e.g., DAC / ADC), and others that can be switched off in case a SC is not active.

With the above considerations, the power consumption of a DSCM transponder can be modeled as eq. (6), where P_{base} is the minimum power consumption and $\beta(\cdot)$ is a multiplier that depends on the actual configuration of the SCs. Equation (7) defines $\beta(\cdot)$ as the summation of two components, those coming from *always active* subsystems and those from subsystems that are associated to a particular SC and thus, can be switched off if that SC is not active. The ratio $R(\cdot)$, defined in eq. (8), computes the proportion of power from subsystems in each group, and $\alpha(\cdot)$ computes the incremental power as a function of the MF and SR configured in a given SC, being 0 if the SC is not active.

Then, from eq. (7), the power consumption of *always active* subsystems starts from a minimum one given by the minimum configuration supported (e.g., <QPSK,8>) and increases depending on the actual configuration, whereas that of the subsystems associated to a particular SC can be 0 if that particular SC is not active.

$$P = P_{base} * \sum_{SC} \beta(MF_{SC}, SR_{SC}) \quad (6)$$

$$\beta(MF_{SC}, SR_{SC}) = R(\text{always on}) * \max(\alpha_{min}, \alpha(MF_{SC}, SR_{SC})) + \quad (7)$$

$$R(\text{per SC}) * \alpha(MF_{SC}, SR_{SC})$$

$$R(\cdot) = \frac{P_{base}(\cdot)}{P_{base}} \quad (8)$$

In this paper, we assume: $R(\text{always on}) = 62\%$, and the values of $\alpha(\cdot)$ in Table VI. This leads to energy savings in the range [56.6%-46.8%] when only one SC is active.

B. DSCM performance and QoT estimation

Let us first study the minimum OSNR for each configuration defined by the pair <MF, SR> assuming a back-to-back scenario. We have considered QPSK, 8QAM and 16QAM, and 8 and 11 GBaud per SC and assumed no guard-band between SCs. The results have been obtained through Monte-Carlo simulation carried out in MATLAB, where a 2^{13} pseudo-random binary sequence was generated and shaped by a root-raised cosine filter with 0.15 roll-off factor; no NLI was considered. Fig. 4 presents the obtained BER, computed through direct error counting, as a function of the OSNR. Assuming a pre-FEC BER threshold of $4 \cdot 10^{-3}$, which includes about one order of magnitude to cope with system margins, the minimum OSNR for every combination is also highlighted.

Let us now study the QoT of an optical system with 4 SCs that is transmitted over a 100 km lightpath, which route includes several ROADMs and OLSs. The ROADMs consist of one WSS for add/drop and one WSS per degree and direction, which results in the optical signal crossing 2 WSSs per ROADM. In line with [29], in this study we compute an approximate value of the $FP_{n;s}(b, SR)$ that does not depend on the MF for the number of ROADM nodes considered. The approximation considers the fundamental and MF-independent penalty due to the noise enhancement introduced by the equalizer at the receiver as it compensates also for filter spectral distortions; at the same time, it neglects all implementation-dependent effects, mostly driven by the effective number of quantization bits of the DAC/ADC, to keep the

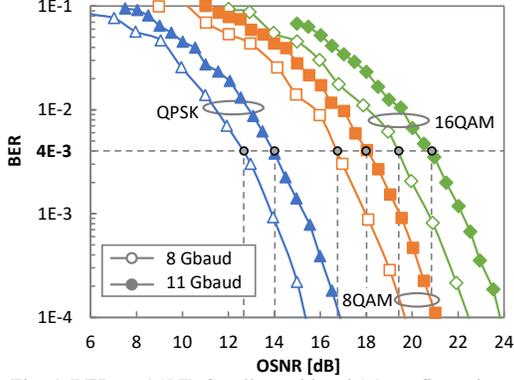


Fig. 4. BER vs OSNR for all considered SC configurations.

implementation analysis agnostic with respect to the MF. Hence, it is computed as the quotient between the area of the inverse of the filtering transfer function of the WSS normalized to the maximum value, and the area of the SC power spectral density. For this analysis, we consider slot widths of 37.5, 50, and 62.5 GHz. OLSs include OAs with a noise figure equal to 5 dB and ITU-T G.652.D fibers with attenuation equal to 0.2 dB/km, dispersion coefficient equal to 16.7 ps/(nm·km), effective area equal to 80 μm^2 , and a nonlinear coefficient of 1.31 (W/km) $^{-1}$. We also assume that OAs are configured to optimize QoT by maximizing the GSNR at the receiver [31].

Fig. 5 reports the evaluated $FP_{n,s}$ for all four SCs and for 8 and 11 GBaud. As expected, penalties are negligible for the internal SCs (SC2 and SC3), while are larger for the external ones (SC1 and SC4). In particular, the penalty is remarkable for 37.5 GHz slot width as the external SCs are significantly distorted by the transition band of the transfer function (32.5 GHz); in the case of 11 GBaud the penalty exceeds 4 dB.

Fig. 6 presents the GSNR and the OSNR computed

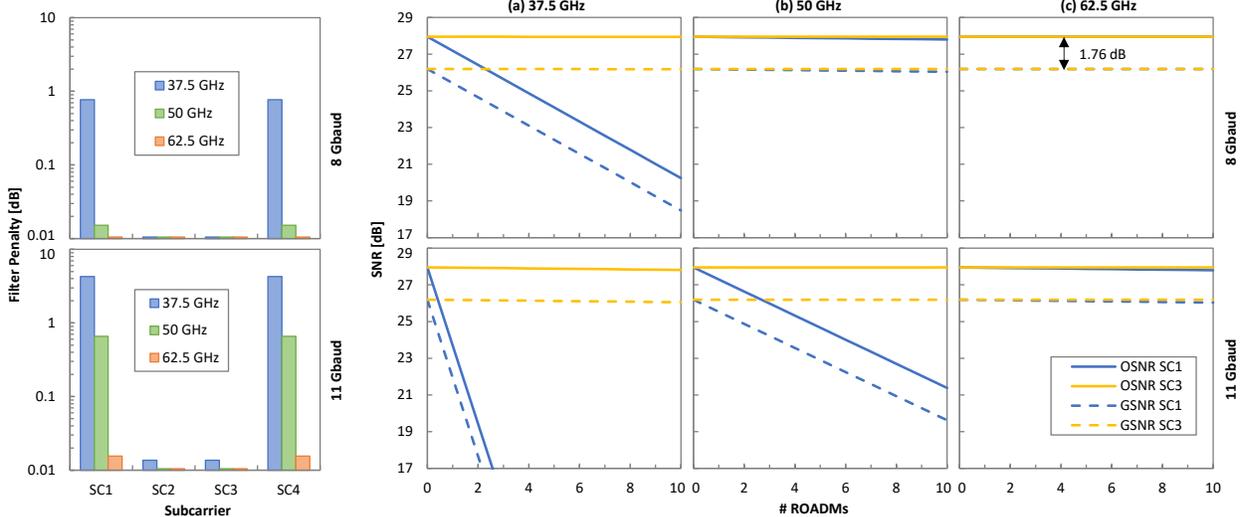


Fig. 5. Filter penalties vs SC

Fig. 6. OSNR and GSNR at the Rx for different slot widths and SRs.

TABLE IV: MIN OSNR FOR THRESHOLD BER AND MAXIMUM #ROADMs DUE TO FILTERING PENALTY FOR EXTERNAL SCs

Config. (<MF, SR>)	OSNR [dB] (BER = $4 \cdot 10^{-3}$)	Slot width (GHz)		
		37.5	50	62.5
QPSK, 8	12.7	10	10	10
QPSK, 11	14.0	2	10	10
8QAM, 8	16.8	8	10	10
8QAM, 11	18.1	2	10	10
16QAM, 8	19.5	8	10	10
16QAM, 11	20.8	0	8	10

TABLE V: AVG. DISTANCES FOR SYMBOL RATE DETECTION

TX / RX [Gbaud]	QPSK			8QAM			16QAM		
	8	11	diff	8	11	diff	8	11	diff
8	0.57	0.63	11%	0.53	0.60	13%	0.46	0.55	20%
11	0.75	0.63	19%	0.81	0.60	35%	0.76	0.54	41%

using an extended GNP version [19], for one external and one internal SC vs. the number of ROADMs in the route of the lightpath for different configurations of SR and slot width. SC2 and SC4 are not reported here as their performance is approximately the same of SC3 and SC1, respectively. We consider a typical line control that aims at maximizing the QoT at the worst-case of full spectral load [30], so, given the OSNR, the GSNR is 2/3 of its value in linear units, corresponding to 1.76 dB difference in dB units, as the OLSs are working at the optimum point [31]. Moreover, the propagation impairment for the internal SC is always dominant, being not affected by the number of filters. For the external SC, ASE noise and NLI are the dominant impairments only for 62.5 GHz slot width and 8 GBaud and 50 GHz slot width. In the other scenarios, filtering plays a key role and becomes the dominant impairment limiting the SC performance.

Finally, Table IV summarizes the results for the operational limits of each SC configuration, in terms of:

i) the minimum OSNR to guarantee pre-FEC BER threshold equal to $4 \cdot 10^{-3}$; and *ii)* the maximum number of ROADMs that can be crossed assuming that pre-FEC BER threshold. Here, we observe that the 62.5 GHz slot width never limits the route of the lightpath. In the case of 50 GHz slot width, the maximum number of ROADMs is limited to 8 only with 16QAM and 11 GBaud. Finally, for 37.5 GHz, filters limit the maximum number of ROADMs, especially for 11 GBaud. The results in Table IV will be consistently used along the rest of this section.

C. SC Configuration Recognition at the Receiver Side

Let us now analyze the accuracy of Algorithm V proposed for SC configuration recognition in Section IV.B. Note that the target is to reach the maximum accuracy of the algorithm for scenarios with the ROADMs and OSNR values summarized in Table IV, for the different configurations.

Comparing the average distances between the received constellation and the sets of expected constellation points representing three different MFs will reveal at the Rx side the actual MF used by the Tx. However, one key aspect of the algorithm is the assumption that two configurations with the same MF and different SR will produce different average distance between the received constellation and the expected constellation points, being that with the lowest distance the one actually selected by the Rx. Table V shows the distances resulting from evaluating the received constellation under the conditions in Table IV. We observe significant differences in the computed distances between the SRs, being the average distance for the right one the shortest in all cases. Note that the minimum difference is as high as 11% for $\langle \text{QPSK}, 8 \rangle$ evaluated under OSNR = 12.7 dB and 10 ROADMs with 50 GHz slot width, which allows for accurate recognition even in the worst conditions. Note also that the difference increases when the OSNR increases, e.g., the difference of the distances is almost double for $\langle 16\text{QAM}, 8 \rangle$ evaluated under OSNR = 19.5 dB.

Fig. 7 presents the results of the accuracy of Algorithm V obtained for all configurations as a function of the OSNR. We observe that the algorithm recognizes the received configuration with 100% accuracy well below the minimum required OSNR in Table IV.

D. Transponder Agent Capacity Management

From the viewpoint of the pure optical layer, all the configurations studied can be used, especially for 50 and 62 GHz slot widths. However, transponders not supporting all such configurations might be more cost effective. With this in mind, we now focus on the

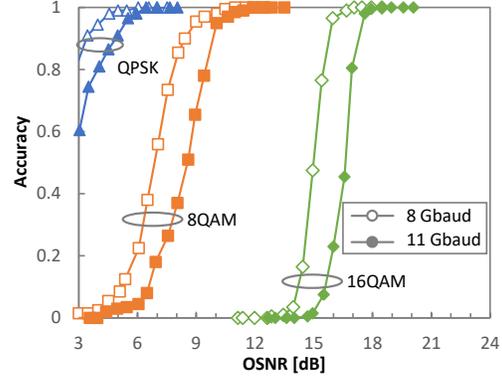


Fig. 7. Accuracy of SC configuration recognition vs OSNR.

TABLE VI: CONFIGURATIONS, POWER, AND SETS

Config. ($\langle \text{MF}, \text{SR} \rangle$)	$\alpha(\langle \text{MF}, \text{SR} \rangle)$	Capacity [Gb/s]	Config. Set		
			all	sel	16QAM
QPSK, 8	1.00	25.6	x	x	
QPSK, 11	1.29	38.4	x		
8QAM, 8	1.14	35.2	x	x	
8QAM, 11	1.47	51.2	x		
16QAM, 8	1.28	52.8	x	x	x
16QAM, 11	1.65	70.4	x	x	x

capacity management and study whether reducing the number of configurations available would be also convenient from the packet layer perspective. In this subsection, we consider that the transponder agent at the Tx side adapts the configuration of the optical connection to provide enough capacity for the input traffic, by running the algorithms in Section IV.A.

For this analysis, we define three configuration sets for the transponders (see Table VI): *i) all*: transponders implement all possible configurations; *ii) selected*: transponders implement some configurations with the maximum diversity in capacity and lightpath scenarios; and *iii) 16QAM*: transponders implement only 16QAM for all SRs. The normalized power consumption and capacity (assuming 20% FEC overhead) of every configuration are detailed in Table VI. Besides the power related to the SCs, another 15% of the maximum configuration supported by the transponder for each SC (active or not) needs to be accounted.

Two different policies have been carefully selected to concentrate numerical evaluation under relevant conditions: *i)* the *maxSC* with only the 16QAM configuration set; and *ii)* the *adaptive* connection management with the three configuration sets. Two traffic profiles lasting for 2.5 days are used to compare the aforementioned policies, namely *High Traffic* and *Low Traffic*. In the *High Traffic* profile, the traffic varies from 20 to 240 Gb/s, whereas in the *Low Traffic* case it is limited to 60 Gb/s. The traffic profiles have been especially tailored to induce many configuration changes, as they contain traffic variations around the

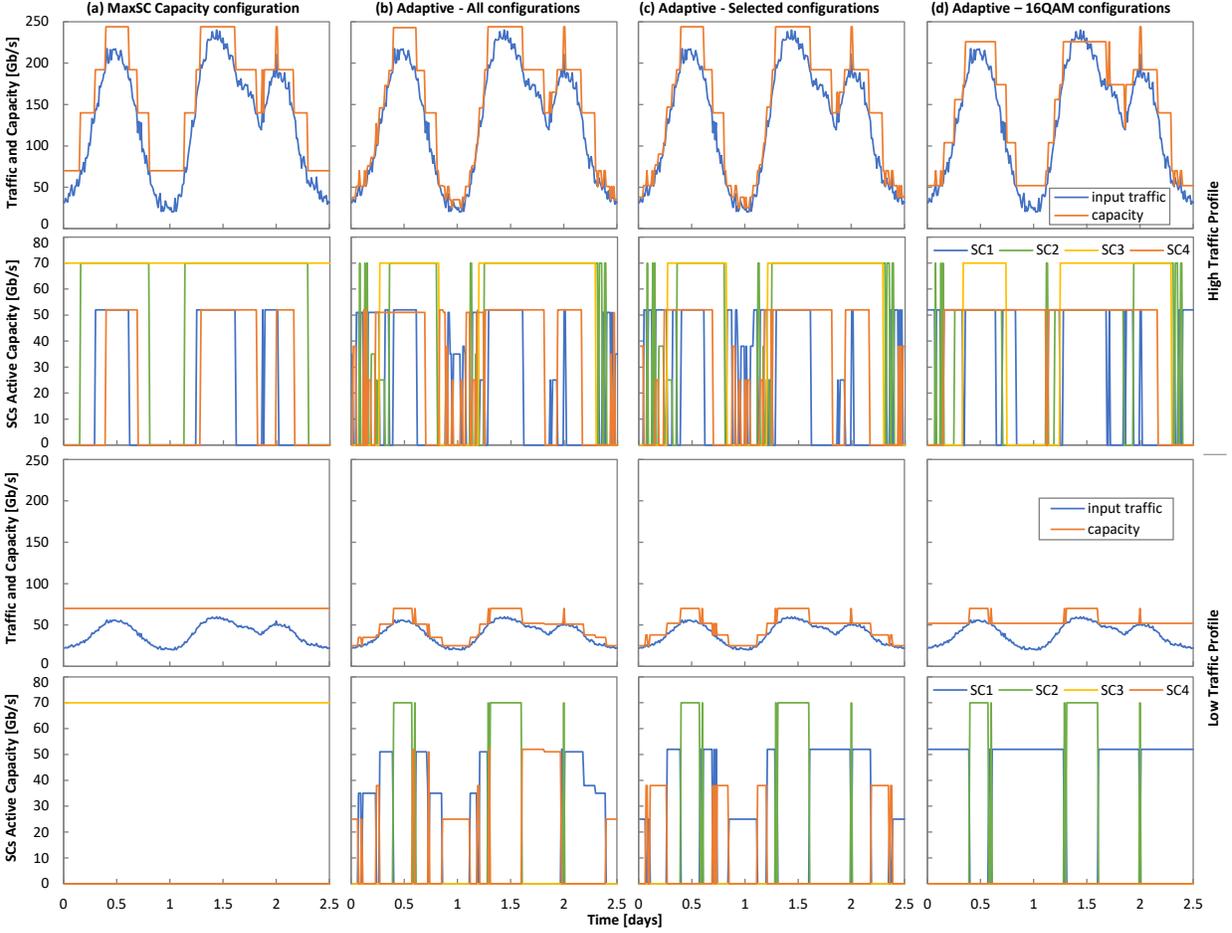


Fig. 8. Optical connection managed by the Transponder Agent. Total and SCs capacity vs time for two traffic profiles.

TABLE VII: SUMMARY OF RESULTS FOR OPTICAL CONNECTION CONFIGURATION MANAGED BY THE TRANSPONDER AGENT

	High Traffic Profile				Low Traffic Profile			
	maxSC	adapt-all	adapt-sel	adapt-16Q	maxSC	adapt-all	adapt-sel	adapt-16Q
#Changes per day	8.8	51.2	51.2	23.2	0	68	82.4	44
Energy savings [%]	28	30	31	32	47	51	53	51
Packet Loss [MB]	4.8	14.9	11.4	14.9	0	0	0	0
Queue (max) [MB]	16	16	16	16	0	9	8	8
Queue (avg) [MB]	0.3	1.4	1.4	1	0	1.2	1	0.7

values of the capacity of the SC configurations, as well as sudden trend changes that might cause poor performance at the packet layer.

A simulation environment based on CURSA-SQ [32]-[34] was used to study the impact of the vlink capacity management of the packet layer. The results in Fig. 8 have been obtained for a connection with a configuration map where the external SC can be configured with any MF but only with 8 GBaud, whereas the internal ones could be configured with all possible configurations supported by the transponder.

Fig. 8 shows the input traffic and the allocated capacity (upper row), and the active capacity per SC (lower row) for the two profiles and for all four

combinations of policies and configuration sets. In the case of the *High Traffic* profile, we observe that although all four combinations manage the capacity of the connection as a function of the input traffic, the *maxSC* policy needs a lower number of changes at the cost of a larger capacity overprovisioning and energy consumption (Fig. 8a). On the opposite, the *adaptive* policy adapts the capacity almost perfectly (Fig. 8b-d) independently of the configuration set, at the cost of a larger number of changes. We also observe that the reactive nature of the algorithms might result in capacity exhaustion as some SCs are not configured to their maximum capacity (Fig. 8d)

This is more evident for the *Low Traffic* profile, where although all different combinations keep active

just one single SC simultaneously, the *adaptive* policy performs many changes targeting a perfect adaptation of the capacity of the connection to the input traffic to minimize overprovisioning; the *maxSC* policy keeps the SC configuration constant at the cost of a larger overprovisioning. Table VII summarizes the results, where we observe that all combinations result in virtually the same energy consumption and reach energy savings higher than 28% under the *High Traffic* profile and 47% under the *Low Traffic* one.

Although in general, the different combinations achieve a noticeable capacity adaptation and large energy savings, two main disadvantages can be observed from this solution: *i*) it is purely reactive, so the capacity is allocated or released as a function of the current input traffic, without considering any requirement (e.g., time) to properly configure the connection, and consequently induces packet loss and added delay as a result of the time packets spent in the vlink queue; and *ii*) it does not consider the traffic evolution, so it produces unnecessary fluctuations in the configuration of the connections.

E. Intent-based vlink operation

Let us now consider the architecture depicted in Fig. 2, where the vlink intent takes the role of client to manage the capacity of the vlink and the Transponder Agent serves such capacity by properly managing the configuration of the SCs. In view of the results obtained in the last subsection, for this study we consider the two policies with only the 16QAM configuration set for the transponders.

For the predictive solution, we exhaustively explored configurations of δ and w ranging between 1 and 20

minutes. The obtained results for the configuration with best performance (6 and 10 min for δ and w , respectively) are presented in Fig. 9a-b and Table VIII (left). We observe that no packet loss has been produced and the time in the vlink queue has been drastically reduced by using a short-term prediction of the evolution of the traffic. Such queue is produced as a result of the no perfect accuracy of the prediction, which produces some capacity exhaustion, and thus the capacity of the vlink is increased automatically by the transponder agent. Besides, we observe a slight reduction in the number of configuration changes and on the energy savings, that are now over 26% and 45% for the *High* and *Low Traffic* profiles, respectively.

Finally, the results obtained when the RL-based vlink intent is used, are presented in Fig. 9c-d and Table VIII (right). Recall that the reward function has been configured according to the values in Table III; moreover, initial exploration probability ϵ was set to 1, reducing every episode by a multiplicative factor of 0.98. The Q-learning algorithm was configured with a learning rate of 0.05 and a discount factor of 0.95. In view of the results, we can confirm that packet loss and time in queue have been totally eliminated, while energy savings are still over 26% and 46% for the *High* and *Low Traffic* profiles, respectively.

VII. CONCLUDING REMARKS

The autonomous operation of SCs has been presented in this paper from a holistic perspective, with the objective to reduce energy consumption at the optical layer, motivated by the fact that SCs can be configured independently with the desired MF and SR. This,

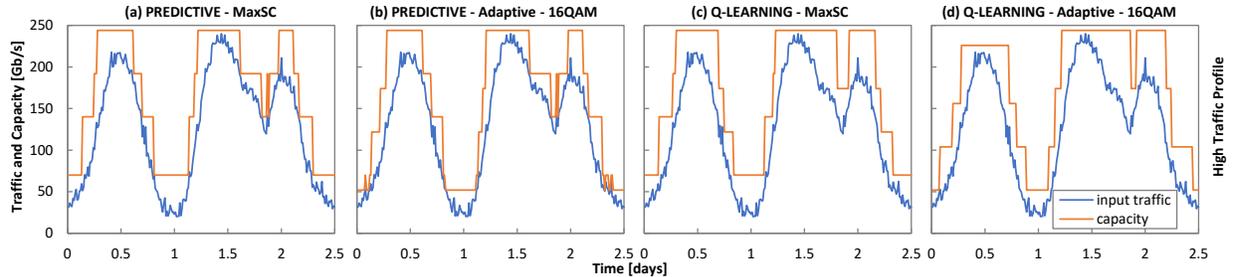


Fig. 9. Optical connection managed by the vLink intent. Total capacity vs time for the *High Traffic* profile.

TABLE VIII: SUMMARY OF RESULTS FOR OPTICAL CONNECTION CONFIGURATION MANAGED BY VLINK INTENT

	Predictive				Q-Learning			
	High Traffic Profile		Low Traffic Profile		High Traffic Profile		Low Traffic Profile	
	maxSC	adapt-16Q	maxSC	adapt-16Q	maxSC	adapt-16Q	maxSC	adapt-16Q
#Changes per day	7.2	18.8	1.6	20.8	7.2	5.6	0	1.6
Energy savings [%]	26	28	45	49	31	26	47	46
Packet Loss [MB]	0	0	0	0	0	0	0	0
Queue (max) [MB]	4.5	10.3	0	5	0	0	0	0
Queue (avg) [MB]	0	0.2	0	0.1	0	0	0	0
Cap. Exhaust. per day	1	5	0	4	0	0	0	0

TABLE IX: MAIN LESSONS LEARNT

Features	Main observations
per-SC QoT estimation	We observed that the penalties are different not only for the different SC configurations, but also for the position of the SC in the channel. Therefore, accurate per-SC QoT estimation is of paramount importance: <i>i</i>) for finding a lightpath and a slot width ($l = \langle p, f, b \rangle$) that satisfies the required capacity and minimizes the selected cost function; and <i>ii</i>) for the autonomous SC configuration management at the Tx side by defining the SC configuration map.
Set of supported SC configurations	Having a large variety of SC configurations might increase the cost of the transponders unnecessarily, as they do not show additional energy savings and their use might result in variations in the capacity of the vlink, which might add additional complexity at the packet layer.
Capacity management at the Tx side	Algorithms that try to tightly adapt the capacity of the vlink to the current or near future input traffic, did not show large energy savings. Increments in capacity of one SC configured to the maximum capacity resulted in effective alternatives, which also simplifies the algorithms. Capacity management uniquely at the optical layer, resulted in poor performance at the packet layer.
SC Configuration Recognition	Several alternatives might be implemented, like the simple and efficient algorithm proposed in this paper. Being able to recognize the right SC configuration with 100% accuracy allows bring the intelligence to the data plane thus, liberating the control plane from collecting monitoring and making configuration decision near real-time.
vlink intent	Intent-based vlink capacity management based on RL resulted in a good solution for smooth capacity evolution, eliminating completely packet loss and time in vlink queue.

together with the possibility to switch off parts of the transponder directly related to each of the SC and the fact that the power consumption of a transponder is related to its actual configuration in terms of MF and SR, opened an opportunity to save energy by activating only those SCs which are needed to support the upper layer packet traffic.

The QoT was firstly analyzed, and the need to perform a per-SC estimation was introduced. The QoT estimation needs to consider not only the path and the specific SC configuration, but also the filtering penalties; the filter penalty function needs to be defined for each slot width to be allowed and depends of the specific frequency slot of the channel and the SC configuration; analytical formulae for the per-SC QoT estimation were developed. QoT estimation was used afterward to solve the RSA problem during the lightpath provisioning phase and to define the SC configuration map that will be used afterwards for the autonomous capacity management.

The autonomous capacity management problem was faced following an incremental approach. First, the algorithms in the transponder agent were built with the objective to provide the required capacity for the lightpath and reduce energy consumption by activating SCs when their capacity is needed, while not adding detrimental effect neither for the packet traffic (packet loss and time spent in the vlink queue) nor for the packet nodes (excessive changes in the capacity of the vlink). At the Tx side, two policies for capacity management were discussed and the required algorithms designed. Specifically, the *maxSC* targeting at minimizing the changes in the SC configuration and the *adaptive* focuses on minimizing the energy consumption by adjusting the capacity of the lightpath to the input

traffic as much as possible. At the Rx side, an algorithm for blind SC configuration was proposed; it is based on computing the average distances between a predefined set of constellation points for every MF and the received constellation. The algorithm is based on the observation that the computed distance is the shortest one when the tested MF and SR are the ones configured for the analyzed SC.

The capacity management implemented in the transponder agent is purely reactive, which would impact on the operation and performance of the packet layer. To solve that, an intent-based solution was proposed to manage the vlink capacity at the packet layer, where such capacity is actually used. The intent-based solution targets to ensure the packet layer performance indicators, like packet loss, time in queue, and configuration changes, while relying on the transponder agent for the SC configuration management targeting at minimizing energy consumption. Two solutions were developed, one based on short-term prediction and the other based on RL.

Table IX summarizes the main lessons learnt in this work, which we believe that might help telecom operators and system vendors to introduce autonomous SC management in their networks and portfolios.

REFERENCES

- [1] Y. Zhang *et al.*, "Digital subcarrier multiplexing for flexible spectral allocation in optical transport network," OSA Optics Express, vol. 19, pp. 21880-21889, 2011.
- [2] T. Rahman *et al.*, "Digital Subcarrier Multiplexed Hybrid QAM for Data-rate Flexibility and ROADM Filtering Tolerance," in Proc. OFC, 2016.
- [3] D. Krause *et al.*, "Design considerations for a digital subcarrier coherent optical modem," in Proc. OFC, 2017.

- [4] H. Sun *et al.*, “800G DSP ASIC Design Using Probabilistic Shaping and Digital Sub-Carrier Multiplexing,” *IEEE/OSA J. of Lightwave Technology*, vol. 38, pp. 4744-4756, 2020.
- [5] M. Qiu, *et al.* “Digital subcarrier multiplexing for fiber nonlinearity mitigation in coherent optical communication systems,” *OSA Optics Express*, vol. 22, pp. 18770-18777, 2014.
- [6] Infinera: “The Ultimate Guide to Nyquist Subcarriers,” [on-line] <https://www.infinera.com/wp-content/uploads/The-Ultimate-Guide-to-Nyquist-Subcarriers-0208-WP-RevA-0719.pdf>.
- [7] J. Bäck *et al.*, “CAPEX Savings Enabled by Point-to-Multipoint Coherent Pluggable Optics Using Digital Subcarrier Multiplexing in Metro Aggregation Networks,” accepted in ECOC 2020.
- [8] N. Sambo *et al.*, “Next Generation Sliceable Bandwidth Variable Transponders,” *IEEE Communications Magazine*, vol. 53, pp. 163-171, 2015.
- [9] L. Velasco *et al.*, “Saving CAPEX by Extending Flexgrid-based Core Optical Networks towards the Edges,” *IEEE/OSA J. of Optical Communications and Networking*, vol. 5, pp. A171-A183, 2013.
- [10] C. Fludger, “Performance oriented DSP design for flexible coherent transmission,” in *Proc. OFC*, 2020.
- [11] F. Frey *et al.*, “Estimation of Trends for Coherent DSP ASIC Power Dissipation for different bitrates and transmission reaches,” in *Proc. Photonic Networks*, 2017.
- [12] V. López and L. Velasco, *Elastic Optical Networks: Architectures, Technologies, and Control*, in *Optical Networks book series*, ISBN 978-3-319-30173-0, Springer, 2016.
- [13] M. Klinkowski *et al.*, “Elastic Spectrum Allocation for Time-Varying Traffic in FlexGrid Optical Networks,” *IEEE J. on Selected Areas in Communications*, vol. 31, pp. 26-38, 2013.
- [14] F. Paolucci *et al.*, “Multipath Restoration and Bitrate Squeezing in SDN-based Elastic Optical Networks,” *Photonic Network Communications*, vol. 28, pp. 45-57, 2014.
- [15] A. P. Vela *et al.*, “Distributing Data Analytics for Efficient Multiple Traffic Anomalies Detection,” *Computer Communications*, vol. 107, pp. 1-12, 2017.
- [16] G. Jajoo, Y. Kumar, and S. Yadav, “Blind Signal PSK/QAM Recognition Using Clustering Analysis of Constellation Signature in Flat Fading Channel,” *IEEE Communication Letters*, vol. 23, pp. 1853-1856, 2019.
- [17] Z. Zhao *et al.*, “A Modulation Format Identification Method Based on Amplitude Deviation Analysis of Received Optical Communication Signal,” *IEEE Photonics Journal*, vol. 11, 2019.
- [18] L. Velasco *et al.*, “Modeling the Routing and Spectrum Allocation Problem for Flexgrid Optical Networks,” *Photonic Network Comm*, vol. 24, pp. 177-186, 2012.
- [19] A. Ferrari *et al.*, “GNPy: an open source application for physical layer aware open optical networks,” *IEEE/OSA J. of Optical Communications and Networking*, vol. 12, pp. C31-C40, 2020.
- [20] D. Rafique and L. Velasco, “Machine Learning for Optical Network Automation: Overview, Architecture and Applications,” *IEEE/OSA J. of Optical Communications and Networking*, vol. 10, pp. D126-D143, 2018.
- [21] A. Clemm, L. Ciavaglia, L. Granville, J. Tantsura (Eds.), “Intent-Based Networking - Concepts and Definitions,” IRTF draft work-in-progress, Mar. 2020.
- [22] R. Sutton and A. Barto, *Reinforcement learning: an introduction*, MIT Press, 1998.
- [23] A. Carena *et al.*, “EGN model of non-linear fiber propagation,” *OSA Optics Express*, vol. 22, pp. 16335-16362, 2014.
- [24] R. Dar *et al.*, “Accumulation of nonlinear interference noise in fiber-optic systems,” *OSA Optics Express*, vol. 22, pp. 14199-14211, 2014.
- [25] J. Proakis and M. Salehi, *Digital communications*, 5th Ed., New York McGraw-hill, 2007.
- [26] M. Cantono *et al.*, “On the interplay of nonlinear interference generation with stimulated Raman scattering for QoT estimation,” *IEEE/OSA J. of Lightwave Technology*, vol. 36, pp. 3131-3141, 2018.
- [27] L. Velasco *et al.*, “Designing, Operating and Re-Optimizing Elastic Optical Networks,” (Invited Tutorial) *IEEE/OSA J. of Lightwave Technology*, vol. 35, pp. 513-526, 2017.
- [28] S. Ishak and H. Al-Deek, “Performance Evaluation of Short-Term Time-Series Traffic Prediction Model,” *J. of Transportation Engineering*, vol. 128, pp. 490-498, 2002.
- [29] T. Rahman *et al.*, “On the mitigation of optical filtering penalties originating from ROADM cascade,” *IEEE Photonics Tech. Letters*, vol. 26, pp. 154-157, 2013.
- [30] V. Curri *et al.*, “Elastic All-Optical Networks: A New Paradigm Enabled by the Physical Layer. How to Optimize Network Performances?” *IEEE/OSA J. of Lightwave Technology*, vol. 35, pp. 1211-1221, 2017.
- [31] V. Curri *et al.*, “Design strategies and merit of system parameters for uniform uncompensated links supporting Nyquist-WDM transmission,” *IEEE/OSA J. of Lightwave Technology*, vol. 33, pp. 3921-3932, 2015.
- [32] M. Ruiz, F. Coltraro, and L. Velasco, “CURSA-SQ: A Methodology for Service-Centric Traffic Flow Analysis,” *IEEE/OSA J. of Optical Communications and Networking*, vol. 10, pp. 773-784, 2018.
- [33] A. Bernal *et al.*, “Near Real-Time Estimation of End-to-End Performance in Converged Fixed-Mobile Networks,” *Computer Communications*, vol. 150, pp. 393-404, 2020.
- [34] L. Velasco and M. Ruiz, “Supporting Time-Sensitive and Best-Effort Traffic on a Common Metro Infrastructure,” *IEEE Comm. Letters*, vol. 24, pp. 1664-1668, 2020.