



## **Joint Power-Efficient Traffic Shaping and Service Provisioning for Metro Elastic Optical Networks**

Downloaded from: <https://research.chalmers.se>, 2025-06-18 03:16 UTC

Citation for the original published paper (version of record):

Hadi, M., Agrell, E. (2019). Joint Power-Efficient Traffic Shaping and Service Provisioning for Metro Elastic Optical Networks. *Journal of Optical Communications and Networking*, 11(12): 578-587.  
<http://dx.doi.org/10.1364/JOCN.11.000578>

N.B. When citing this work, cite the original published paper.

© 2019 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, or reuse of any copyrighted component of this work in other works.

# Joint Power-Efficient Traffic Shaping and Service Provisioning for Metro Elastic Optical Networks

Mohammad Hadi, *Member, IEEE* and Erik Agrell, *Fellow, IEEE*

**Abstract**—Considering the time-averaged behavior of a metro elastic optical network, we develop a joint procedure for resource allocation and traffic shaping to exploit the inherent service diversity among the requests for a power-efficient network operation. To support quality of service diversity, we consider minimum transmission rate, average transmission rate, maximum burst size, and average transmission delay as the adjustable parameters of a general service profile. The work evolves from a stochastic optimization problem, which minimizes the power consumption subject to stability, physical, and service constraints. The optimal solution of the problem is obtained using a complex dynamic programming method. To provide a near-optimal fast-achievable solution, we propose a sequential heuristic with a scalable and causal software implementation, according to the basic Lyapunov iterations of an integer linear program. The heuristic method has a negligible optimality gap and a considerable shorter runtime compared to the optimal dynamic programming, and reduces the consumed power by 72% for an offered traffic with unit variation coefficient. The adjustable trade-offs of the proposed scheme offer a typical 10% power saving for an acceptable amount of excess transmission delay or drop rate.

**Keywords**—Traffic shaping, Quality of service, Metro elastic optical networks, Dynamic programming, Lyapunov optimization.

## I. INTRODUCTION

**P**OWER consumption in metro networks is an increasingly important issue, and mainly motivated by considerable traffic growth, dynamic traffic patterns, and diverse quality of service (QoS) requirements [1]–[3]. If the power efficiency does not improve proportionally to the unprecedented metro traffic growing rate, then the metro will consume an unacceptably large amount of energy [2]. Fixed resource allocation techniques are not suitable solutions to support dynamic metro traffic patterns since they waste power when the overall traffic volume is far from its peak [4]–[7]. Traditional resource allocation methods without QoS management cannot be used to offer a power-efficient network operation because they do not provide different levels of QoS and dissipate power by a resource allocation beyond the actual need of a service level agreement [5], [8]. Software-defined networking (SDN) is a promising management framework that offers a suitable control plane with the required state information to manage the flexibility of the tunable parameters of the data plane [7], [9]. To efficiently exploit the offered SDN manageability,

we need a power-efficient resource allocation scheme to be fast enough to reallocate available resources according to the dynamic network behavior, and to be sufficiently general to support diverse QoS requirements [9]–[11].

Traffic shaping is a well-studied topic of computer science, whose applications for power-efficient resource allocation in optical networks can be promising. Unfortunately, typical traffic shaping methods in computer science cannot be employed for optical networks because they do not consider the physical network interactions [12, sec. 5.4]. Furthermore, the available SDN-based traffic engineering techniques are developed over a simple abstraction of the data plane flexibility and cannot support intensive flexibility of the elastic optical networks [13]. There are some research works on power-efficient resource allocation in the core optical networks [14]–[20]; however, they do not consider the fast traffic dynamism and QoS diversity of the metro elastic optical networks (MEONs). Some researchers have studied the problem of resource allocation in MEONs, but without a detailed attention to the deep interactions between power consumption, service requirements, and network status [4], [21], [22]. We proposed a stochastic QoS-aware algorithm for dynamic resource allocation in MEONs in [23], where the transmission delay is not directly set in the service profile.

In this paper, transmission delay is included explicitly among the service-characterizing parameters. This provides a general service profile with adjustable parameters of minimum transmission rate, average transmission rate, maximum burst size, and especially, the average transmission delay. We consider an elastic and practical transponder structure with configurable modulation, tunable spectrum, and accurate signal-to-noise ratio (SNR) commitments, which is fed by a traffic shaper having limited buffer length and possible data drop. Hitless fast reconfiguration of elastic transponders is a promising research area with several experimental demonstrations [2], [11]. Our focus is on an SDN-based cross-layer network management, where network management occurs in the centralized control plane, and traffic shapers are connecting elements of the packet to optical sublayers in the data plane. We concentrate on the importance of traffic shaping and develop an optimization problem, which jointly configures traffic shapers and elastic transponders for a minimized amount of total time-averaged transponder power consumption subject to stability, physical, and QoS constraints. Considering the average delay, the solution of the formulated optimization problem cannot be derived using the Lyapunov drift optimization; unlike [23], where the results are obtained by direct application of the Lyapunov optimization tool. Although the optimal solution is obtained by dynamic programming, soon the curse of dimensionality appears. To alleviate this

---

M. Hadi and E. Agrell are with the Department of Electrical Engineering, Chalmers University of Technology, SE-41296 Gothenburg, Sweden (e-mail: mohadi@chalmers.se). This work was supported in part by Vinnova under grant no. 2017-05228 and the Knut and Alice Wallenberg Foundation under grant no. 2013.0021.

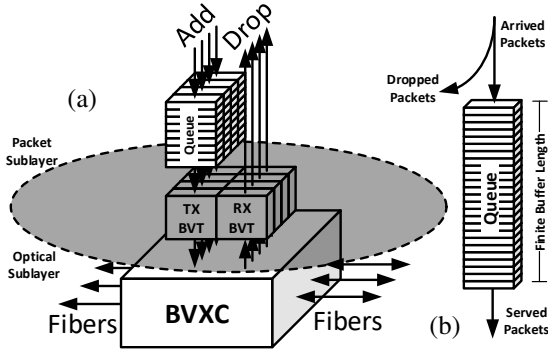


Fig. 1: System model architecture. (a) Optical node. (b) Traffic shaper. BVXC, bandwidth-variable cross-connect; BVT, bandwidth-variable transponder.

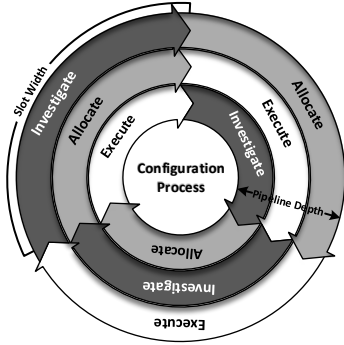


Fig. 2: Pipelined three-stage configuration process.

curse, a fast and accurate heuristic algorithm is proposed based on the basic Lyapunov drift technique. The heuristic needs an iterative optimization of an integer linear program (ILP) to shape the traffic streams according to an adaptation between the transponder configuration and network status. The heuristic yields almost the optimal solution and its convergence speed is fast enough to track fluctuating metro traffic patterns adaptively. This fine adaptation improves the power efficiency by offering a power consumption proportional to the average traffic load, not its peak. The consumed power can also be traded for the tolerable amount of transmission delay or drop rate.

The paper continues with describing the system model in Sec. II. The proposed joint power-efficient traffic shaping and resource provisioning scheme is formulated in Sec. III and its useful extensions are discussed in Sec. IV. The optimal and near-optimal solutions of the formulated optimization problem are derived in Secs. V and VI, respectively. Simulation results are included in Sec. VII. Finally, the paper is concluded in Sec. VIII.

## II. SYSTEM MODEL

Let  $\mathbb{Z}_a^b$  be the set of integers from  $a$  to  $b$ , inclusively. We consider an MEON characterized by an arbitrary topology graph that operates in discrete time intervals  $n \in \mathbb{Z}_0^\infty$  with interval duration  $T$ . The optical fiber bandwidth is divided

into  $N$  equal-bandwidth frequency slots with a granularity of  $W$ . Topology links in an MEON are short enough to enable optical transmission without any intermediate amplification along fiber links. Therefore, we assume that each fiber link has only a pair of input and output amplifiers, whose gains equal to the switching and fiber loss, respectively. As shown in Fig. 1(a), each topology node is equipped with a buffer-less cross-connect and a bank of transponders in the optical layer. The cross-connects are bandwidth-variable and can switch an arbitrary range of the fiber spectrum. Transmit and receive transponders are elastic and their modulation format, spectrum width, and central frequency can be repeatedly reconfigured. The transmit transponder is fed by a limited-length buffer queue in the packet sublayer, where the traffic shaping process occurs.

A centralized SDN controller starts a pipelined configuration process in each interval  $n$ . As shown in Fig. 2, the configuration process includes three consecutive stages of investigate, allocate, and execute, and therefore, it takes three intervals provided that each stage ends within  $T$ . The controller begins the configuration process with the investigate stage in interval  $n$  and queries all optical nodes to update network state information. Next, the controller goes to the allocate stage in interval  $n+1$  and uses the collected network status information of interval  $n$  to run a resource allocation algorithm and configure the network elements. Finally, in the execute stage, the controller reconfigures the network to operate within interval  $n+2$  according to the new configuration calculated in the interval  $n+1$ . As illustrated in Fig. 2, different stages of the last three consecutive configuration processes can proceed in parallel, *i.e.*, the process has a pipeline depth of 3. The stages may have different time durations but they should complete their actions within an interval width  $T$  [23]. The time duration of each stage has its limiting factors and cannot be reduced below a certain bound. In view of the demonstrated hitless transponders and proposed fast SDN-based control signalling methods [10], [11], the allocation stage usually has the dominant time duration, which is limited by the computational complexity of the resource allocation algorithm [23].

There are  $I$  connection requests, each having its own service profile  $\mathbf{S}_i, i \in \mathbb{Z}_1^I$ . The service profile is an array  $\mathbf{S}_i = (M_i, R_i, B_i, D_i)$ , whose elements denote minimum transmission rate, average transmission rate, maximum burst size, and average transmission delay, respectively. The actual transmission rate of connection  $i$  is time-varying, but its instantaneous values are lower-limited by  $M_i$ , and its time-average equals  $R_i$ .  $B_i$  denotes the available buffering space for a burst of traffic that arrives suddenly, but is still transmitted by the average rate  $R_i$ . Transmitted bits can experience different transmission delay, but on average, they have a delay of  $D_i$ .

As enlarged in Fig. 1(b), the traffic shaping process of connection  $i$  settles in its transmit queue with a backlog size  $q_i[n] \in \mathbb{Z}_0^{Q_i}$  and a limited storage size  $Q_i$ . The queue is filled by an arbitrary random process  $a_i[n]$ , which denotes the number of arrived bits in interval  $n$ , and is emptied by a connecting light-path, which guarantees the required minimum and average rates. If the input arrival violates the agreed service profile parameters,  $d_i[n]$  arrived bits will be dropped in interval

$n$ . According to Little's theorem, for a stable steady state, the average number of queued bits is upper-bounded by  $D_i R_i$  [24, sec. 1], [25]. The queue size should be enough to accommodate the average number of the queued bits  $D_i R_i$  and have a free space for the burst size of  $B_i$ ; therefore, we implicitly have  $Q_i \geq D_i R_i + B_i$ .

Connection  $i$  is routed over one of its available  $J$  paths  $\mathbf{P}_i = (P_{i,1}, \dots, P_{i,J})$ . Each connection  $i$  has a pair of transmit and receive transponders in its source and destination nodes, respectively, with reconfigurable parameters of  $f_i[n]$  and  $s_{i,k}[n]$ , which denote the start slot and the number of servicing slots for modulation  $k$  in interval  $n$ . We force connection  $i$  to select at most one of  $K$  available modulations and therefore,  $s_{i,k}[n] \neq 0$  for at most one value of  $k$ . Modulation  $k$  has spectral efficiency  $C_k$  and requires a spectrum width narrower than  $U_{i,j,k}$  along path  $P_{i,j}$ . Considering transmission attenuation, switching loss, amplified spontaneous emission noise, approximated nonlinear interference, SNR margin, and different SNR threshold of each modulation format [26], the values of  $U_{i,j,k}$  can be precomputed according to [27, Sec. IV-A]. In fact, the SNR relates inversely to the spectrum width and therefore,  $U_{i,j,k}$  is the spectrum width for which the received SNR over path  $P_{i,j}$  equals the sum of the SNR margin and SNR threshold of modulation  $k$ . There should be at least  $G$  guard slots between the assigned slots of two connections with intersecting paths.

Transponders, cross-connects, and amplifiers are the main sources of power consumption in an MEON. The power consumption of the amplifiers and cross-connects mainly depends on the network topology and has no significant traffic-dependency. Therefore, we focus on the power consumption of the elastic transponders, which is directly affected by the traffic dynamism [28]. The power consumption of the transmit and receive transponders for connection  $i$  with modulation  $k$  in interval  $n$  is modelled as  $p_{i,k}[n] = s_{i,k}[n](E + FC_k)$ , where the constant  $E$  denotes the fixed consumed power per slot and the slope coefficient  $F$  relates the consumed power to the selected modulation  $k$  [19], [28], [29]. The parameters being used in the paper are summarized in Tab. I.

### III. MATHEMATICAL PROBLEM FORMULATION

For a power-efficient network management, we need to consider full state information of the network over time and configure traffic shapers and elastic transponders according to the QoS requirements and physical limitations. This interconnected resource allocation process is the solution of the stochastic optimization problem

$$\min_{\mathbf{x}[n], \mathbf{t}[n], \mathbf{d}[n], \mathbf{f}[n], \mathbf{s}[n]} \sum_{i=1}^I \sum_{k=1}^K \overline{s_{i,k}}(E + FC_k) + V \sum_{i=1}^I \overline{d_i} \quad \text{s.t.} \quad (1a)$$

$$\overline{q_i} - \frac{D_i}{T}(\overline{a_i} - \overline{d_i}) \leq 0, \quad i \in \mathbb{Z}_1^I \quad (1b)$$

$$W \sum_{k=1}^K C_k \overline{s_{i,k}} \geq R_i, \quad i \in \mathbb{Z}_1^I \quad (1c)$$

TABLE I: Constants, variables, and operators along with their corresponding definitions.

Type	Notation	Definition
Indices	$i, i' \in \mathbb{Z}_1^I$	Connection index
	$j, j' \in \mathbb{Z}_1^J$	Path index
	$k \in \mathbb{Z}_1^K$	Modulation index
	$n, n' \in \mathbb{Z}_0^\infty$	Interval index
	$r \in \mathbb{Z}_0^\infty$	Frame index
	$I \in \mathbb{Z}_1^\infty$	Number of connections
Input Parameters	$J \in \mathbb{Z}_1^\infty$	Number of paths
	$K \in \mathbb{Z}_1^\infty$	Number of modulations
	$N \in \mathbb{Z}_1^\infty$	Number of slots
	$G \in \mathbb{Z}_1^\infty$	Minimum guard slots
	$T \in \mathbb{R}^{\geq 0}$ [s]	Interval width
	$W \in \mathbb{R}^{\geq 0}$ [Hz]	Slot bandwidth
	$V \in \mathbb{R}^{\geq 0}$	Drop penalty coefficient
	$L \in \mathbb{R}^{\geq 0}$	Lyapunov coefficient
	$E \in \mathbb{R}^{\geq 0}$ [W]	Power bias constant
	$F \in \mathbb{R}^{\geq 0}$ [W]	Power slope coefficient
	$M_i \in \mathbb{R}^{\geq 0}$ [bit/s]	Minimum rate
	$R_i \in [M_i, \infty)$ [bit/s]	Average rate
	$B_i \in \mathbb{R}^{\geq 0}$ [bit]	Maximum burst
	$D_i \in \mathbb{R}^{\geq 0}$ [s]	Average delay
	$\mathbf{S} = (\mathbf{S}_i) = ((M_i, R_i, B_i, D_i))$	Service profile array
	$\mathbf{P} = (\mathbf{P}_i) = ((P_{i,j}))$	Shortest path array
	$\mathbf{Q} = (Q_i \in \mathbb{Z}_0^\infty)$ [bit]	Buffer length array
	$\mathbf{C} = (C_k \in \mathbb{R}^{\geq 0})$ [bit/s/Hz]	Spectral efficiency array
	$\mathbf{U} = (U_{i,j,k} \in \mathbb{R}^{\geq 0})$ [Hz]	Maximum spectrum array
	$\mathbf{a}[n] = (a_i[n] \in \mathbb{R}^{\geq 0})$ [bit]	Arrival rate array
Output Parameters	$\mathbf{f}[n] = (f_i[n])$	Start slot array
	$\mathbf{p}[n] = (p_{i,k}[n])$ [W]	Power consumption array
	$\mathbf{y}[n] = (y_i[n])$	Delay virtual queue array
	$\mathbf{z}[n] = (z_i[n])$	Rate virtual queue array
	$\mathbf{q}[n] = (q_i[n])$ [bit]	Actual queue array
	$\mathbf{d}[n] = (d_i[n])$ [bit]	Dropped bit array
	$\mathbf{s}[n] = (s_{i,k}[n])$	Servicing slot array
	$\mathbf{t}[n] = (t_{i,i'}[n])$	Relative location array
	$\mathbf{x}[n] = (x_{i,j,k}[n])$	Path & modulation array
	$\Delta$	Actual queue space
Other Parameters	$\Gamma$	Virtual queue space
	$\Omega$	Event space
	$\Delta(\omega, \mathbf{q})$	Action space
	$\Phi \in (0, 1]$	Renewal probability
	$\varphi[n]$	Renewal random process
	$\alpha[r]$	Renewal start interval
	$\gamma[n]$	Virtual queue array
	$\omega[n]$	Event array
	$\delta[n]$	Action array
	$\mu(\mathbf{q})$	Optimal cost
	$\mu$	Optimal cost vector
	$\eta(\delta, \omega, \gamma, \mathbf{q})$	Incurred cost
	$\eta(\delta, \omega, \gamma)$	Incurred cost vector
	$\rho(\delta, \omega, \mathbf{q}, \mathbf{q}')$	Transition probability
	$\rho(\delta, \omega)$	Transition matrix
operators	$\overline{x}$	Time-average
	$\mathcal{O}(x)$	Order
	$\mathcal{E}\{x\}$	Expectation

$$W \sum_{k=1}^K C_k s_{i,k}[n] \geq M_i, \quad i \in \mathbb{Z}_1^I, n \in \mathbb{Z}_0^\infty \quad (1d)$$

$$a_i[n] + q_i[n] - Q_i - TW \sum_{k=1}^K C_k s_{i,k}[n] \leq d_i[n], \quad i \in \mathbb{Z}_1^I, n \in \mathbb{Z}_0^\infty \quad (1e)$$

$$W s_{i,k}[n] \leq \sum_{j=1}^J U_{i,j,k} x_{i,j,k}[n], \quad i \in \mathbb{Z}_1^I, k \in \mathbb{Z}_1^K, n \in \mathbb{Z}_0^\infty \quad (1f)$$

$$f_i[n] + \sum_{k=1}^K s_{i,k}[n] \leq N, \quad i \in \mathbb{Z}_1^I, n \in \mathbb{Z}_0^\infty \quad (1g)$$

$$\sum_{j=1}^J \sum_{k=1}^K x_{i,j,k}[n] \leq 1, \quad i \in \mathbb{Z}_1^I, n \in \mathbb{Z}_0^\infty \quad (1h)$$

$$t_{i,i'}[n] + t_{i',i}[n] = 1, \quad i, i' \in \mathbb{Z}_1^I, n \in \mathbb{Z}_0^\infty : i \neq i' \quad (1i)$$

$$f_i[n] + \sum_{k=1}^K s_{i,k}[n] + G \leq f_{i'}[n] + N(3 - t_{i,i'}[n] - \sum_{k=1}^K x_{i,j,k}[n] - \sum_{k=1}^K x_{i',j',k}[n]), \quad i, i' \in \mathbb{Z}_1^I, \\ j, j' \in \mathbb{Z}_1^J, n \in \mathbb{Z}_0^\infty : i \neq i', P_{i,j} \cap P_{i',j'} \neq \emptyset \quad (1j)$$

$$q_i[n+1] = \max \left\{ 0, a_i[n] + q_i[n] - d_i[n] - TW \sum_{k=1}^K C_k s_{i,k}[n] \right\}, \quad i \in \mathbb{Z}_1^I, n \in \mathbb{Z}_0^\infty \quad (1k)$$

where the bar signs denote the time-averaged values with the definition  $\bar{x} = \lim_{n \rightarrow \infty} \frac{1}{n} \sum_{n' \in \mathbb{Z}_0^{n-1}} \mathcal{E}\{x[n']\}$  [25], [30]. The optimization is over the binary arrays  $\mathbf{x}[n]$  and  $\mathbf{t}[n]$ , and nonnegative integer arrays  $\mathbf{d}[n]$ ,  $\mathbf{f}[n]$ , and  $\mathbf{s}[n]$ , which denote the tunable parameters of the traffic shapers and elastic transponders, such as path, modulation, start slot, number of servicing slots, and number of dropped bits.

The objective function (1a) is a mixed expression of the total time-averaged transponder power consumption and number of dropped bits with an adjustable weighting coefficient  $V$ . Increasing  $V$  pushes the optimization to reduce drop rate, while a lower value of  $V$  gives the optimization priority to the consumed power. Constraint (1b) is a simple adoption of Little's theorem and guarantees the required average delay  $D_i$  [24, sec. 1], [30]. Constraints (1c) and (1d) assure the prescribed average rate  $R_i$  and minimum rate  $M_i$ , respectively. Constraint (1e) drops the incoming bits that cannot be accommodated in the finite buffer length  $Q_i$ . SNR limitations are considered in constraints (1f) by keeping the assigned spectrum widths below their permitted upper-bound values  $U_{i,j,k}$ , where the binary variable  $x_{i,j,k}[n]$  equals 1 if connection  $i$  uses path  $P_{i,j}$  and modulation  $k$  in interval  $n$ , and otherwise 0. For an infeasible modulation  $k$  over path  $P_{i,j}$ ,  $U_{i,j,k} = 0$ , which in turn forces  $s_{i,k}[n] = 0$  in constraint (1f) to avoid selection of an infeasible modulation. This constraint may force  $s_{i,k}[n] = 0$  for all modulations  $k$ , then it means that connection  $i$  is inactive in interval  $n$ . The assigned slots settle in the acceptable range of the fiber spectrum by constraints (1g). According to constraints (1h), at most one modulation and one path can be selected for each connection. Constraint (1i) determines the relative locations of the assigned slots, where the Boolean variable  $t_{i,i'}[n]$  equals 1 if  $f_i[n] \leq f_{i'}[n]$ , and 0 otherwise. Constraint (1j) prevents two neighboring spectrum widths from overlapping by keeping  $G$  guard slots between them, where  $P_{i,j} \cap P_{i',j'} \neq \emptyset$  means that paths  $P_{i,j}$  and  $P_{i',j'}$  have at least one common link. Assuming  $q_i[0] = 0$ , the queue backlogs are updated recursively by constraint (1k).

The optimization problem (1) is a purely theoretical con-

struction, which jointly optimizes the variables for all  $n \in \mathbb{Z}_0^\infty$ . It has  $I(JK + K + I + 1)$  variables and  $\mathcal{O}(I^2 J^2 + IK + I^2 + 6I)$  constraints in each interval. Furthermore, the number of variables and constraints evolve linearly with the number of intervals. The future state of the network is needed to solve (1), and this is not possible for a causal implementation. Even if (1) is truncated to a finite number of intervals with known state information, the resulting complexity and runtime will impede a practical deployment. Therefore, a causal and sequential algorithm is needed to follow the dynamic state of the network and derive the optimal solution of (1).

#### IV. EXTENSIONS AND VARIATIONS

Clearly, a more accurate model with less computational complexity provides more power efficiency but the accuracy and complexity are usually traded for each other. For example, a complex SNR expression improves accuracy and allows to reduce SNR margin and improve power efficiency while its computational complexity entangles the configuration process and leads to power deficiency by degrading the adaptation capability to track traffic fluctuations. Although the optimization problem (1) can cover a wide range of operational scenarios, it can also be extended to take various additional conditions and constraints into account. Here, we list a few such possibilities and discuss their implementations and implications. These extensions usually influence the accuracy and complexity in opposite directions, where the overall impact depends on the input parameters listed in Tab. I.

##### A. Reconfiguration Constraints

Sometimes, reconfiguration is a hard process. For instance, rerouting can lead to service interruption and therefore, some operators may desire no or rare rerouting. Such reconfiguration limitations can be considered by adding extra constraints to (1), as illustrated in [23]. The reconfiguration constraints usually fix part of the optimization variables such as  $\mathbf{x}[n]$  and consequently, reduce the computational complexity. Moreover, different reconfiguration possibilities can have distinct cost values. A clear example is the number of dropped bits  $d_i[n]$ , whose time-averaged value appears with the cost penalty  $V$  in the objective function (1a). Similarly, the cost of other reconfigurations, such as changing modulation formats or paths, can be included in the objective function through distinct penalty coefficients and arbitrary mathematical dependencies to the variables in Tab. I. Depending on the mathematical format of the cost penalty terms, the extended formulation may be more or less complex.

##### B. More Complex SNR Constraints

As mentioned in Sec. II, the coefficients  $U_{i,j,k}$  were precomputed according to [27, Sec. IV-A] by considering the additive noise and an approximated nonlinear interface as well as a generic SNR margin to account for the worst-case mismatch between calculated and real SNR values. This gives the SNR constraint (1f) a very simple, linear form, which allows fast optimization even for large networks. As an alternative, (1f)

can be replaced with a more accurate expression, such as [26, Eq. (7)], which accurately models the nonlinear interference depending on the actual bandwidth allocation. This approach can give a more efficient resource utilization and improve the power efficiency, but it may also change the structure of the problem (1) such that its optimization can only be possible for relatively small networks, or large networks with a long reconfiguration period [15]. The impact of the SNR margin on power efficiency will be discussed in Sec. VII.

### C. Adaptive Guard Bandwidth

Guard bandwidth can mitigate nonlinear impairments at the price of more resource usage [27]. Although we fix the guard bandwidth  $G$  in (1), it can be adaptively selected in each interval to meet SNR requirements for a lower amount of allocated resources. To do this, we can define different guard bandwidths and select the best one by adding extra constraints to (1). As a simpler method, constraint (1j) can be revised intelligently to adaptively tune guard bandwidths. Clearly, the guard bandwidth should be selected proportional to the power of nonlinear interferences, which in turn is a function of  $x_{i,j,k}[n]$ ,  $s_{i,k}[n]$ , and  $f_i[n]$  [26], [27]. Roughly, the guard bandwidth in constraint (1j) can be approximated by a linear combination of the optimization variables to adaptively be tuned without changing the complexity order of the formulation.

### D. More Complex Power Models

The power consumption of the transponders, which severely depends on the traffic, was directly included in (1) [28]. Alternatively, the objective function (1a) can be extended to have more terms, such as [29, Eqs. (3) and (4)], to take the power consumption of optical amplifiers and cross-connects into account. The new power terms can complicate the problem (1) by injecting more variables to the optimization process such as node, add, and drop degrees, which are not significantly influenced by traffic fluctuations. Such a more accurate power model can allow to optimize the operation of other network elements and improve power efficiency, especially for relatively small networks, where the optimization complexity is affordable. For large networks, the excess amount of traffic-dependent power efficiency may not worth the imposed computational complexity [15].

In the next sections, without loss of generality, we focus on the optimization problem (1) and show how its solution can be derived. Undoubtedly, the proposed solution methods are general and can be applied to any variations of the problem (1) described above.

## V. OPTIMAL SOLUTION

Lyapunov drift optimization is a lucrative tool for solving stochastic optimization problems with time-averaged expressions, which partitions the complex main problem into simpler subproblems such that the iterative solution of the subproblems achieves the optimum response of the main problem. In the basic Lyapunov optimization, where the queue update equation

(1k) is the only queue-dependent constraint, a simple deterministic optimization subproblem is involved in each iteration. Since (1) is further constrained by the queue states through (1b) and (1e), we cannot use the basic form of the Lyapunov optimization [23], [30]. An efficient extension of the basic Lyapunov optimization that can cope with this problem is renewal theory [25], where the involved subproblems require more complex non-deterministic optimizations. In this theory, the time line is decomposed into successive renewal frames  $n = \alpha[r], \dots, \alpha[r+1] - 1$  for  $r \in \mathbb{Z}_0^\infty$ , where  $\alpha[r]$  denotes the start interval of frame  $r$ , when all the queues are simultaneously empty, and  $\alpha[0] = 0$ . To force the queues to repeatedly visit a renewal state of being simultaneously empty, we drop all unserved bits in all queues and increment  $r$  at the end of every interval  $n$  with probability  $\Pr\{\varphi[n] = 1\} = \Phi > 0$ . The renewal process  $\varphi[n]$  is an independent and identically distributed (i.i.d.) Bernoulli random process over intervals and independent of the arrival random processes, which are also assumed to be i.i.d. over intervals in the renewal theory. While these forced renewals create inefficiency in the network operation by unwanted dropped bits, they make the network mathematically analyzable. Furthermore, the unwanted renewal drops can be made arbitrarily low with a small choice of  $\Phi$ . It can be shown that a system optimized without forced renewals has a performance that is no better than a system with forced renewals, but where all drops from forced renewals are counted as delivered throughput [30].

Now, let the actual queue space  $\mathbb{A}$  include all possible values for  $\mathbf{q}[n]$ . Define the event array  $\omega[n] = (\mathbf{a}[n], \varphi[n]) \in \Omega$ , where the event space  $\Omega$  contains all possible events, and the action array  $\delta[n] = (\mathbf{x}[n], \mathbf{t}[n], \mathbf{d}[n], \mathbf{f}[n], \mathbf{s}[n]) \in \Delta(\omega[n], \mathbf{q}[n])$ , whose corresponding space set  $\Delta(\omega[n], \mathbf{q}[n])$  is a function of  $\omega[n]$  and  $\mathbf{q}[n]$ . Assuming a renewal operation, we can follow Algorithm 1 to approach arbitrarily closely the optimum value of (1) [30]. In this algorithm, a virtual queue is assigned to each time-averaged constraint and is updated using a similar recursive equation as the actual queues in (1k). We refer to  $\gamma[n] = (\mathbf{y}[n], \mathbf{z}[n]) \in \mathbb{F}$  as a virtual queue array, whose elements are devoted to time-averaged constraints (1b) and (1c), respectively, and  $\mathbb{F}$  is the corresponding space set. The constant  $L$ , named the Lyapunov penalty coefficient, is used to control the proximity to the optimal solution. Algorithm 1 initializes the actual and virtual queues by a proper-size zero array  $\mathbf{0}$  and then starts a renewal frame at  $\alpha[0] = 0$ . For each frame  $r \in \mathbb{Z}_0^\infty$ , we observe actual and virtual queues and define the objective function

$$\sum_{n=\alpha[r]}^{\alpha[r+1]-1} \eta(\delta[n], \omega[n], \gamma[\alpha[r]], \mathbf{q}[n]) \quad (2)$$

where the incurred cost

$$\begin{aligned} \eta(\delta, \omega, \gamma, \mathbf{q}) = & \sum_{i=1}^I \mathcal{E} \left\{ y_i \left( q_i - \frac{D_i}{T} (a_i - d_i) \right) \right. \\ & \left. + z_i \left( TR_i - TW \sum_{k=1}^K C_k s_{i,k} \right) + L \left( \sum_{k=1}^K p_{i,k} + V d_i \right) \middle| \gamma \right\} \quad (3) \end{aligned}$$

and  $\delta$ ,  $\omega$ ,  $\gamma$ , and  $\mathbf{q}$  are given values of the sets  $\Delta(\omega, \mathbf{q})$ ,  $\Omega$ ,  $\mathbb{I}$ , and  $\mathbb{A}$ , respectively. For each interval  $n$ , actual and virtual queues are updated by (1k) and

$$y_i[n+1] = \max \left\{ 0, y_i[n] + q_i[n] - \frac{D_i}{T} (a_i[n] - d_i[n]) \right\}, \quad i \in \mathbb{Z}_1^I \quad (4)$$

$$z_i[n+1] = \max \left\{ 0, z_i[n] + TR_i - TW \sum_{k=1}^K C_k s_{i,k}[n] \right\}, \quad i \in \mathbb{Z}_1^I. \quad (5)$$

Minimizing (2) is a weighted stochastic shortest-path problem, which can be solved by dynamic programming [25], [31, sec. 2]. We let  $\rho(\delta, \omega, \mathbf{q}, \mathbf{q}')$  denote the transition probability from state  $\mathbf{q} \in \mathbb{A}$  to state  $\mathbf{q}' \in \mathbb{A}$  given known values of  $\omega \in \Omega$  and  $\delta \in \Delta(\omega, \mathbf{q})$ , and let  $\mu(\mathbf{q})$  be the optimum expected sum cost to the end interval of the frame, provided that we start in state  $\mathbf{q} \in \mathbb{A}$ . By basic dynamic programming, the optimal action for given values of  $\omega[n]$  and  $\mathbf{q}[n]$  in an interval  $n \in \{\alpha[r], \dots, \alpha[r+1] - 1\}$  is [31, sec. 2]

$$\min_{\delta \in \Delta(\omega[n], \mathbf{q}[n])} \left\{ \eta(\delta, \omega[n], \gamma[\alpha[r]], \mathbf{q}[n]) + \sum_{\mathbf{q}' \in \mathbb{A}} \rho(\delta, \omega[n], \mathbf{q}[n], \mathbf{q}') \mu(\mathbf{q}') \right\}. \quad (6)$$

To solve (6), we need the values of  $\mu(\mathbf{q})$  for frame  $r$ , which are in turn obtained by Bellman's equation [31, sec. 1]

$$\mu = \mathcal{E} \left\{ \min_{\delta \in \Delta(\omega, \mathbf{q})} [\eta(\delta, \omega, \gamma[\alpha[r]], \mathbf{q}) + \rho(\delta, \omega) \mu] \right\} \quad (7)$$

where  $\mu$  and  $\eta(\delta, \omega, \gamma[\alpha[r]], \mathbf{q})$  are the optimal and incurred cost vectors, respectively, whose entries include  $\mu(\mathbf{q})$  and  $\eta(\delta, \omega, \gamma[\alpha[r]], \mathbf{q})$  for all possible values of  $\mathbf{q} \in \mathbb{A}$ . Further,  $\rho(\delta, \omega)$  is transition probability matrix and the expectation is over the distribution of  $\omega$ .

Unfortunately, the size of the state space in (6) and (7) grows exponentially with the number of state variables. Known as the curse of dimensionality, this phenomenon renders dynamic programming intractable in practical implementations. Even for a limited state space, (6) and (7) are complex nonlinear optimization problems, which can often only be solved for small network topologies using numerical methods [30].

## VI. NEAR-OPTIMAL SOLUTION

Although Algorithm 1 yields the optimum solution of (1), the involved dynamic programming is very time-consuming and therefore, we cannot finely track the instantaneous fluctuations of the traffic. This directly contradicts the agile and real-time network management required for the metro networks. To tackle this problem, we propose the heuristic Algorithm 2 based on the basic Lyapunov drift theory. The idea behind Algorithm 2 is that we can ignore renewal events and their corresponding complex dynamic programming, and take a step-wise routine to update the network status using the basic Lyapunov iterations [23], [25]. Algorithm 2 involves an ILP in each interval, which can be solved fast enough to follow

---

### Algorithm 1: Optimal Dynamic Program Algorithm

---

**input:**  $I, J, K, N, G, T, W, V, L, E, F, \mathbf{S}, \mathbf{P}, \mathbf{Q}, \mathbf{C}, \mathbf{U}, \mathbf{a}[n], \Phi$   
**output:**  $\mathbf{x}[n], \mathbf{t}[n], \mathbf{d}[n], \mathbf{f}[n], \mathbf{s}[n], \mathbf{p}[n], \mathbf{q}[n], \mathbf{y}[n], \mathbf{z}[n]$   
 initialize renewal process by  $r = -1$  and  $\varphi[-1] = 1$   
 initialize queues by  $\mathbf{q}[0] = \mathbf{0}$ ,  $\mathbf{y}[0] = \mathbf{0}$ , and  $\mathbf{z}[0] = \mathbf{0}$   
**for**  $n = 0, 1, 2, \dots$  **do**  
   **if**  $\varphi[n-1] = 1$  **then**  
      $r \leftarrow r + 1$   
     set frame start interval  $\alpha[r]$  to  $n$   
     update incurred cost terms in (2)  
     find optimal costs by solving (7)  
   **end**  
   update  $\varphi[n]$  by a random binary with  $\Pr\{\varphi[n] = 1\} = \Phi$   
   find optimal action by solving (6)  
   update  $\mathbf{q}[n]$ ,  $\mathbf{y}[n]$ , and  $\mathbf{z}[n]$  using (1k), (4), and (5)  
**end**

---



---

### Algorithm 2: Fast Lyapunov Drift Algorithm

---

**input:**  $I, J, K, N, G, T, W, V, L, E, F, \mathbf{S}, \mathbf{P}, \mathbf{Q}, \mathbf{C}, \mathbf{U}, \mathbf{a}[n]$   
**output:**  $\mathbf{x}[n], \mathbf{t}[n], \mathbf{d}[n], \mathbf{f}[n], \mathbf{s}[n], \mathbf{p}[n], \mathbf{q}[n], \mathbf{y}[n], \mathbf{z}[n]$   
 initialize queues by  $\mathbf{q}[0] = \mathbf{0}$ ,  $\mathbf{y}[0] = \mathbf{0}$ , and  $\mathbf{z}[0] = \mathbf{0}$   
**for**  $n = 0, 1, 2, \dots$  **do**  
   minimize objective (8) constrained to (1d)–(1j)  
   update  $\mathbf{q}[n]$ ,  $\mathbf{y}[n]$ , and  $\mathbf{z}[n]$  using (1k), (4), and (5)  
**end**

---

dynamic metro traffic streams. Algorithm 2 achieves a near-optimal solution at the cost of consuming a bit more power. However, it is more capable to track the dynamic patterns of the metro traffic and shape the traffic streams more effectively for a power-efficient operation and therefore, its optimality gap is negligible. The algorithm begins with initializing the queues by zero. Then, for each interval  $n \in \mathbb{Z}_0^\infty$ , we observe the network arrival and queuing status, and solve an ILP including the constraints (1d)–(1j) in interval  $n$  and the objective [23]

$$\min_{\substack{\mathbf{x}[n], \mathbf{t}[n] \\ \mathbf{d}[n], \mathbf{f}[n], \mathbf{s}[n]}} \sum_{i=1}^I \left[ L \left( \sum_{k=1}^K p_{i,k}[n] + V d_i[n] \right) + y_i[n] (q_i[n] - \frac{D_i}{T} (a_i[n] - d_i[n])) + z_i[n] \left( TR_i - TW \sum_{k=1}^K C_k s_{i,k}[n] \right) \right]. \quad (8)$$

After (8) is solved, the queues are updated by (1k), (4), and (5). In the next intervals, the sequence of solving the ILP and updating the queues continues, as explained in Algorithm 2.

## VII. NUMERICAL RESULTS

We use simulation results to validate the performance of Algorithms 1 and 2 in different scenarios. Simulations run on a desktop computer with a Corei7-7700K processor and 16 GB installed physical memory. We employ MATLAB and YALMIP for general coding development, while GUROBI and CPLEX are used for solving the optimization problems. Optimal costs and actions in Algorithm 1 are calculated using





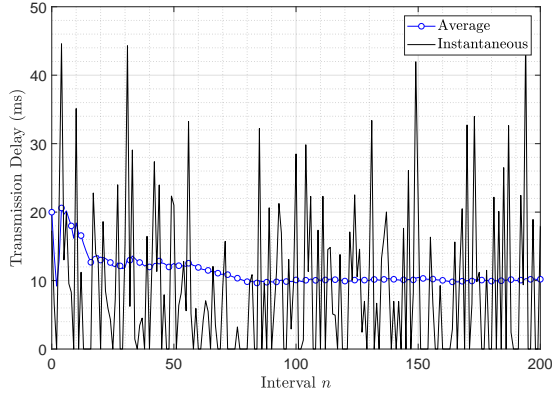


Fig. 9: Example of time-varying transmission delay for a connection with service profile  $\mathbf{S}_i = (25 \text{ Gbit/s}, 50 \text{ Gbit/s}, 0 \text{ Gbit}, 10 \text{ ms})$ .

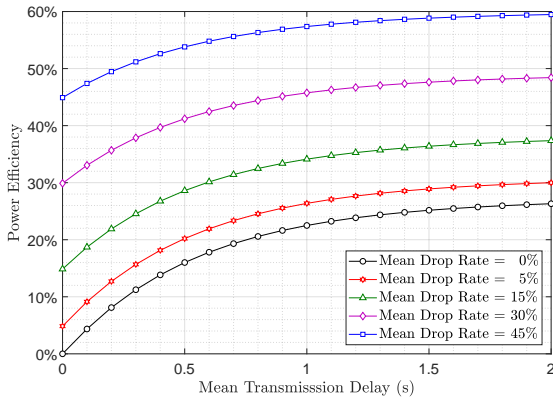


Fig. 10: Power efficiency compared with a scenario without transmission delay and drop rate for different values of mean transmission delay  $\sum_{i=1}^I D_i/I$  and mean drop rate  $\sum_{i=1}^I \bar{d}_i/\bar{a}_i/I$ .

respectively, to precompute  $U_{i,j,k}$ . The network topology in Fig. 3(a), which is loosely modelled after a real commercial network topology covering the city of Stockholm and its surrounding urban areas [32], is used for practical evaluation of Algorithm 2. For the most time-consuming computations, the artificial small network topology in Fig. 3(b) is employed. Unless otherwise specified, the average rates  $R_i$  and average delays  $D_i$  are uniformly selected from  $[0, 100]$  Gbit/s and  $[0, 1000]$  ms, respectively. Furthermore,  $M_i = 0$ ,  $B_i = 0$ , and  $Q_i = D_i R_i$ . Arrival processes  $a_i[n]$  are i.i.d. over all intervals and connections, with a log-normal distribution of an average  $R_i T$  and unit variation coefficient in each interval [33].

We compare the accuracy and runtime of Algorithms 1 and 2 in Figs. 4 and 5, respectively, using the small network in Fig. 3(b). The objective function (1a) is a mixture of two time-averaged terms for the consumed power and drop rate, and therefore, we need to consider both terms for comparing the accuracy. Fig. 4 shows the distributions of the consumed power and drop rate relative errors for a high number of simulation runs. To provide a fair comparison, the same service profiles and traffic flows were used for both algorithms in

each simulation run. The average consumed power and drop rate relative errors are 0.37% and 0.31%, respectively. Thanks to these very low relative errors, Algorithm 1 can be simply replaced with Algorithm 2 without any noticeable performance degradation. According to the runtime curves in Fig. 5, this brings a speed improvement of more than three orders of magnitude, which is vital for a scalable implementation of the algorithm and its ability to track the dynamic traffic behavior of the metro network. Also, Algorithm 2 can be solved more than 100 times faster than its counterparts in backbone networks, where the SNR constraints are the main source of computational complexity [15], [20], [26]. We also consider the SNR requirements in the proposed formulation, but describe them using the linear constraint (1f) and move a main part of the complexity to the precomputation stage to considerably reduce the runtime of Algorithm 2. The short runtime of Algorithm 2 allows to decide on the network configuration based on the latest status information, offers a reconfiguration period of a couple of seconds, and enables adaptive tracking of dynamic traffic patterns with a burst duration less than a minute [15], [34].

Fig. 6 illustrates the power efficiency versus traffic variation coefficient for the real network topology in Fig. 3(a). Each curve represents a different ratio  $M_i/R_i$ . Here, the power efficiency is the amount of saved power by Algorithm 2 compared with a traditional resource allocation scheme that considers the worst operational conditions and fixes the configuration of the network elements [23]. Higher values of the power efficiency indicate more saved power. In a typical scenario with the traffic variation 1, the power efficiency is 72%, which means that Algorithm 2 consumes 72% less power than the traditional resource allocation scheme. This improvement results from the adaptation mechanism embedded in the proposed algorithms. In fact, our methods follow the average behaviors of the network and not its worst operational conditions, which allows allocating the available resource according to the actual needs of the connections in each interval. Increasing the minimum rates  $M_i$  fixes part of the allocated resources and consequently, reduces the adaptation mechanism and power efficiency.

The proposed formulation supports adaptive modulation selection and considers physical SNR limitations using the linear constraint (1f). This constraint provides a cross-layer tool to analyze the physical behavior of the network and its linear structure is a novel property that expedites the convergence speed of Algorithm 2, as illustrated in Fig. 5. To further solidify this claim, we report in Fig. 7 the amount of excess power consumed to increase the SNR margin from 0 dB to a desired value. Each curve corresponds to a different set of selectable modulations, each having the  $K$  lowest-order modulations from the introduced modulation formats. Clearly, increasing the SNR margin results in more consumed power. This implies that the power efficiency is improved when using more sensitive receive transponders with a lower required SNR margin. The sensitivity of the transponders is more important when a higher number of modulations is available, as can be seen in Fig. 7. In fact, high-order modulations are more prone to noise and require a higher SNR threshold, which is not achieved for higher SNR margins, especially for longer

light-paths. This limits the possibility of adaptive modulation selection and consequently, increases the assigned bandwidth and power consumption.

A significant benefit of the proposed algorithms is their ability to support a general set of QoS parameters. In Figs. 8 and 9, we show how Algorithm 2 guarantees the QoS for a randomly selected connection  $i$  with the service profile  $\mathbf{S}_i = (25 \text{ Gbit/s}, 50 \text{ Gbit/s}, 0 \text{ Gbit}, 10 \text{ ms})$ . As shown in Fig. 8, the transmission rate changes over time, but its values stay above the required minimum rate of  $M_i = 25 \text{ Gbit/s}$ . The average transmission rate approaches the desired average rate of  $R_i = 50 \text{ Gbit/s}$  after a short transient. The same statement holds for the average transmission delay and its prescribed average value  $D_i = 10 \text{ ms}$  is achieved after a short transient, as can be seen in Fig. 9. Note that the transmission rate behaves like the one reported in [23], but investigating the behavior of the transmission delay is a new contribution.

Algorithm 2 offers adjustable parameters  $L$  and  $V$  to trade the consumed power against transmission delay and drop rate, respectively. Such trade-offs enhance the flexibility of the network management and the network administrator can use them to reduce operational cost or improve subscription experience. These useful trade-offs are illustrated in Fig. 10, where the power efficiency is defined as the amount of saved power compared with a reference scenario having no transmission delay  $D_i = 0, \forall i$  and drop rate  $\bar{d}_i = 0, \forall i$ . According to the numerical results, we can reduce the consumed power by 10% if the network can tolerate a mean transmission delay  $\frac{1}{I} \sum_{i \in \mathcal{Z}_1^I} D_i$  and a mean drop rate  $\frac{1}{I} \sum_{i \in \mathcal{Z}_1^I} \bar{d}_i / \bar{a}_i$  of 125 ms and 5%, or 250 ms and 0%, respectively. Note that the trade-offs concern the mean values and therefore, some connections with strict QoS, such as  $D_i = 0$  or  $\bar{d}_i = 0$ , may still exist.

## VIII. CONCLUSIONS

We apply the concept of traffic shaping to resource allocation in metro elastic optical networks. We develop a stochastic optimization problem to dynamically shape input traffic flows and provide an adaptation between allocated resources and actual needs of the shaped traffic flows. The proposed formulation considers physical-layer interactions to minimize the total time-averaged transponder power consumption for a set of defined QoS requirements. A general collection of minimum transmission rate, average transmission rate, average transmission delay, and maximum burst volume is used to describe the QoS requirements of a connection request. Although the optimum solution of the problem can be obtained by dynamic programming, the resulting complexity and runtime will be too large for practical deployment. To provide a causal and practical implementation, a heuristic solution is proposed using the Lyapunov drift theory. The near-optimal solution of the heuristic is achieved more than several orders of magnitude faster with a negligible optimality gap than with dynamic programming. Therefore, the heuristic is accurate and fast enough to offer a power-efficient operation with a typical 72% of power efficiency by tracking the dynamic traffic behavior of the metro network. The consumed power can be traded for transmission delay and drop rate using the controllable

parameters of the proposed method. In a typical scenario, these trade-off relationships reduce the consumed power by 10% if the service profiles can tolerate an overall average transmission delay and drop rate of 125 ms and 5%, respectively.

## REFERENCES

- [1] G. Shen, Y. Zhang, X. Zhou, Y. Sheng, N. Deng, Y. Ma, and A. Lord, "Ultra-dense wavelength switched network: A special EON paradigm for metro optical networks," *IEEE Communications Magazine*, vol. 56, no. 2, pp. 189–195, 2018.
- [2] P. Layec, A. Dupas, D. Verchère, K. Sparks, and S. Bigo, "Will metro networks be the playground for (true) elastic optical networks?" *Journal of Lightwave Technology*, vol. 35, no. 6, pp. 1260–1266, 2017.
- [3] Y. Ye, F. J. Arribas, J. Elmirghani, F. Idzikowski, J. L. Vizcaíno, P. Monti, F. Musumeci, A. Pattavina, and W. Van Heddeghem, "Energy-efficient resilient optical networks: Challenges and trade-offs," *IEEE Communications Magazine*, vol. 53, no. 2, pp. 144–150, 2015.
- [4] C. Rottondi, M. Tornatore, F. Puleio, S. Raavi, A. Pattavina, and G. Gavioli, "On the benefits of elastic transponders in optical metro networks," in *Optical Fiber Communication Conference and National Fiber Optic Engineers Conference (OFC/NFOEC)*, 2012.
- [5] Z. Zhong, N. Hua, M. Tornatore, Y. Li, H. Liu, C. Ma, Y. Li, X. Zheng, and B. Mukherjee, "Energy efficiency and blocking reduction for tidal traffic via stateful grooming in IP-over-optical networks," *Journal of Optical Communications and Networking*, vol. 8, no. 3, pp. 175–189, 2016.
- [6] R. Alvizu, X. Zhao, G. Maier, Y. Xu, and A. Pattavina, "Energy-efficient dynamic optical routing for mobile metro-core networks under tidal traffic patterns," *Journal of Lightwave Technology*, vol. 35, no. 2, pp. 325–333, 2017.
- [7] R. Alvizu, S. Troia, G. Maier, and A. Pattavina, "Matheuristic with machine-learning-based prediction for software-defined mobile metro-core networks," *Journal of Optical Communications and Networking*, vol. 9, no. 9, pp. D19–D30, 2017.
- [8] B. C. Chatterjee, S. Ba, and E. Oki, "Fragmentation problems and management approaches in elastic optical networks: a survey," *IEEE Communications Surveys & Tutorials*, vol. 20, no. 1, pp. 183–210, 2018.
- [9] F. Paolucci, A. Sgambelluri, F. Cugini, and P. Castoldi, "Network telemetry streaming services in SDN-based disaggregated optical networks," *Journal of Lightwave Technology*, vol. 36, no. 15, pp. 3142–3149, 2018.
- [10] A. Dupas, P. Layec, E. Dutisseuil, S. Belotti, S. Bigo, E. H. Salas, G. Zervas, and D. Simeonidou, "Elastic optical interface with variable baudrate: Architecture and proof-of-concept," *Journal of Optical Communications and Networking*, vol. 9, no. 2, pp. A170–A175, 2017.
- [11] A. Dupas, P. Layec, D. Verchère, Q. P. Van, and S. Bigo, "Ultra-fast hitless 100Gbit/s real-time bandwidth variable transmitter with SDN optical control," in *Optical Fiber Communications Conference and Exposition (OFC)*, 2018.
- [12] A. S. Tanenbaum and D. J. Wetherall, *Computer Networks*, 5th ed. USA: Prentice Hall, 2011.
- [13] H. Huang, S. Guo, P. Li, B. Ye, and I. Stojmenovic, "Joint optimization of rule placement and traffic engineering for QoS provisioning in software defined network," *IEEE Transactions on Computers*, vol. 64, no. 12, pp. 3488–3499, 2015.
- [14] M. Hadi and M. R. Pakravan, "Energy-efficient fast configuration of flexible transponders and grooming switches in OFDM-based elastic optical networks," *Journal of Optical Communications and Networking*, vol. 10, no. 2, pp. 90–103, 2018.
- [15] M. Hadi and M. R. Pakravan, "Energy-efficient service provisioning in inter-data center elastic optical networks," *IEEE Transactions on Green Communications and Networking*, vol. 3, no. 1, pp. 180–191, 2019.
- [16] H. Khodakarami, B. S. G. Pillai, B. Sedighi, and W. Shieh, "Flexible optical networks: An energy efficiency perspective," *Journal of Lightwave Technology*, vol. 32, no. 21, pp. 3356–3367, 2014.

- [17] H. Khodakarami, B. S. G. Pillai, and W. Shieh, "Quality of service provisioning and energy minimized scheduling in software-defined flexible optical networks," *Journal of Optical Communications and Networking*, vol. 8, no. 2, pp. 118–128, 2016.
- [18] A. Fallahpour, H. Beyranvand, and J. A. Salehi, "Energy-efficient manycast routing and spectrum assignment in elastic optical networks for cloud computing environment," *Journal of Lightwave Technology*, vol. 33, no. 19, pp. 4008–4018, 2015.
- [19] J. Zhang, Y. Zhao, X. Yu, J. Zhang, M. Song, Y. Ji, and B. Mukherjee, "Energy-efficient traffic grooming in sliceable-transponder-equipped IP-over-elastic optical networks," *Journal of Optical Communications and Networking*, vol. 7, no. 1, pp. A142–A152, 2015.
- [20] L. Yan, E. Agrell, M. N. Dharmaweera, and H. Wymeersch, "Joint assignment of power, routing, and spectrum in static flexible-grid networks," *Journal of Lightwave Technology*, vol. 35, no. 10, pp. 1766–1774, 2017.
- [21] B. Yan, Y. Zhao, X. Yu, W. Wang, Y. Wu, Y. Wang, and J. Zhang, "Tidal-traffic-aware routing and spectrum allocation in elastic optical networks," *Journal of Optical Communications and Networking*, vol. 10, no. 11, pp. 832–842, 2018.
- [22] C. Rottondi, M. Tornatore, A. Pattavina, and G. Gavioli, "Routing, modulation level, and spectrum assignment in optical metro ring networks using elastic transceivers," *Journal of Optical Communications and Networking*, vol. 5, no. 4, pp. 305–315, 2013.
- [23] M. Hadi, M. R. Pakravan, and E. Agrell, "Dynamic resource allocation in metro elastic optical networks using Lyapunov drift optimization," *Journal of Optical Communications and Networking*, vol. 11, no. 6, pp. 250–259, 2019.
- [24] L. Kleinrock, *Queueing Systems: Computer Applications*, 5th ed. New York: Wiley-Interscience, 1976, vol. II.
- [25] M. J. Neely, "Stochastic network optimization with application to communication and queueing systems," *Synthesis Lectures on Communication Networks*, vol. 3, no. 1, pp. 1–211, 2010.
- [26] L. Yan, E. Agrell, H. Wymeersch, and M. Brandt-Pearce, "Resource allocation for flexible-grid optical networks with nonlinear channel model," *Journal of Optical Communications and Networking*, vol. 7, no. 11, pp. B101–B108, 2015.
- [27] P. Poggiolini, G. Bosco, A. Carena, V. Curri, Y. Jiang, and F. Forghieri, "The GN-model of fiber non-linear propagation and its applications," *Journal of Lightwave Technology*, vol. 32, no. 4, pp. 694–721, 2014.
- [28] J. L. Vizcaíno, Y. Ye, and I. Tafur Monroy, "Energy efficiency analysis for flexible-grid OFDM-based optical networks," *Computer Networks*, vol. 56, no. 10, pp. 2400–2419, 2012.
- [29] M. Ju, F. Zhou, S. Xiao, and Z. Zhu, "Power-efficient protection with directed  $p$ -cycles for asymmetric traffic in elastic optical networks," *Journal of Lightwave Technology*, vol. 34, no. 17, pp. 4053–4065, 2016.
- [30] M. J. Neely, "Stochastic optimization for Markov modulated networks with application to delay constrained wireless scheduling," in *Conference on Decision and Control (CDC)*, 2009, pp. 4826–4833.
- [31] D. P. Bertsekas, *Dynamic Programming and Optimal Control*. Belmont: Athena Scientific, 1995, vol. II.
- [32] AB Stokab, <https://www.stokab.se/Vart-Fibernet/Geografisk-tackning>, 2019.
- [33] M. Jinno, H. Takara, B. Kozicki, Y. Tsukishima, Y. Sone, and S. Matsuoka, "Spectrum-efficient and scalable elastic optical path network: architecture, benefits, and enabling technologies," *IEEE Communications Magazine*, vol. 47, no. 11, pp. 66–73, 2009.
- [34] L. Velasco, A. Asensio, J. Berral, A. Castro, and V. López, "Towards a carrier SDN: An example for elastic inter-datacenter connectivity," *Optics express*, vol. 22, no. 1, pp. 55–61, 2014.