Soft-Failure Detection, Localization, Identification, and Severity Prediction by Estimating QoT Model Input Parameters

Sima Barzegar, Marc Ruiz, Andrea Sgambelluri, Filippo Cugini, Antonio Napoli, and Luis Velasco

Abstract—The performance of optical devices can degrade because of aging and external causes like, for example, temperature variations. Such degradation might start with a low impact on the Quality of Transmission (QoT) of the supported lightpaths (soft-failure). However, it can degenerate into a hard-failure if the device itself is not repaired or replaced, or if an external cause responsible for the degradation is not properly addressed. In this work, we propose comparing the QoT measured in the transponders with the one estimated using a QoT tool. Those deviations can be explained by changes in the value of input parameters of the QoT model representing the optical devices, like noise figure in optical amplifiers and reduced Optical Signal to Noise Ratio in the Wavelength Selective Switches. By applying reverse engineering, the value of those modeling parameters can be estimated as a function of the observed QoT of the lightpaths. Experiments reveal high accuracy estimation of modeling parameters, and results obtained by simulation show large anticipation of soft-failure detection and localization, as well as accurate identification of degradations before they have a major impact on the network.

Index Terms—Soft-Failure Identification and Localization, Network Automation.

I. INTRODUCTION

The massive application of the optical technology [1] in the core, metro, and access segments [2], is a clear consequence of its high bandwidth, low latency, and high reliability, which enables the deployment of 5G and beyond. Because of the growing complexity of optical systems, it is critical to assess the Quality of Transmission (QoT) of optical connections (lightpath). This can be quantified in terms of Signal to Noise Ratio (SNR) and measured within Optical Transponders (TRX).

The QoT is related to the linear and nonlinear (NL) optical noise, and it can be estimated based on a model describing the physics of propagation, e.g., the generalized Gaussian Noise (GN) model [3]. Additionally, effects such as aging of optical devices might severely affect the QoT. Aging effects are usually considered by means of costly system margins [4]. Examples include: i) the degradation of Optical Amplifiers (OA), which can be quantified as increased Noise Figure (NF); and ii) detuning of the lasers in the TRXs or frequency drift of the filters in Wavelength Selective Switches (WSS), which can lead to misalignments. If those degradations (soft-failures) are not properly handled (e.g., by retuning, repairing, or replacing the related optical device), they can degenerate into hard-failures when the SNR reduces, and zero post-Forward Error Correction (FEC) error cannot be achieved (FEC limit); this could affect a large portion of network services. Therefore, it is of paramount importance not only to detect any QoT degradation, but also to identify the cause and localize the device causing the degradation.

Further, a considerable effort is being paid towards disaggregating the optical layer to enrich the offer of available solutions and to enable the deployment of solutions that better fit optical network operators’ needs [5]. Such disaggregation, however, tends to make network surveillance and maintenance more complex in general, due to the absence of vendors providing support of vertically integrated network equipment.

To support failure management, the control plane of the optical network needs to be enriched with Monitoring and Data Analytics (MDA) capabilities [6], [7]. Once monitoring data, notifications, and alarms have been collected from the data plane, data analytics algorithms (e.g., based on Machine Learning (ML) techniques [8]) can analyze them to proactively detect the degradation, identify and localize the cause, and anticipate hard-failures before they occur. Once detected, identified, and localized, recommendations can be issued to the network controller so it can decide about rerouting and/or reconfiguring the network [9], as well as notifying the management plane for maintenance. Note that ML-based approaches require training and validation datasets, which makes their practical application difficult due to key drawbacks,
The topic of failure management in optical networks (including anticipated detection, identification, and localization) has been extensively explored and several related works can be found in the literature, in particular using ML techniques (see the tutorial in [12]). For soft-failure detection and identification, the authors in [13] proposed to analyze the evolution of the Bit Error Rate (BER) in the transponders and issue notifications when the BER increases; an algorithm running in the centralized network manager anticipates degradations and identifies the most probable cause of failure and its probability. The work in [14] focused on detecting and identifying filter-related failures by analyzing the spectrum of optical signals at the receiver. An autoencoder-based solution to detect and identify soft-failures in optical links was proposed in [15], while in [16] the authors proposed a convolutional neural network running in the TRXs that estimates the probabilities for four types of soft-failures with good performance for single failure scenarios. The work in [17] presented a solution for failure prediction that scores the features related to failures based on their importance. The authors in [18] proposed a classifier to predict the failure probability of optical links complemented with a heuristic for failure localization under the single link-failure assumption. Methods for localization of filter-related failures were proposed in [19] by analyzing the optical spectrum in intermediate locations. Finally, a method based on graph neural networks was proposed in [20] that analyzes the alarms received to identify the root alarm and localize the failure.

Apart from alarm correlation, several of these works centered on analyzing the QoT represented by the measured BER, spectrum, etc. for detecting failures, or correlating alarms for localization. In contrast, the status of optical devices was analyzed in [21] through measures related to device parameters like optical power, gain, temperature, etc. to proactively detect and localize potential faults and determining the likely root-cause. This approach to failure detection, derived from the analysis of devices’ parameters, is key to really identify and localize the failure itself.

However, it is not always possible to obtain the right value of those devices’ parameters that can be related to the QoT, in particular in disaggregated scenarios. Note that any QoT model uses a set of input parameters to describe the specific characteristics of the different optical devices that participate in the optical layer, like WSSs as building blocks of Reconfigurable Optical Add / Drop Multiplexers (ROADM), TRXs, and In-Line OAs, e.g., Erbium Doped Fiber Amplifier (EDFA). The authors in [22], [23] proposed ML methods for finding the right value of such QoT model’s parameters aiming at improving the QoT estimation. Our approach applies reverse engineering from the real QoT values—collected periodically from the TRXs—to derive the evolution of the value of QoT model’s parameters (referred to as modeling parameters in the rest of the paper) that explain such QoT observations. We believe that, by analyzing such evolution, it is possible to anticipate more precisely future degradations, and enable failure localization.

This paper extends our previous work in [24] and proposes the MESARTHIM methodology that targets at: i) detecting and localizing the optical device responsible for the soft-failure; ii) identifying the modeling parameters that explain the observed effects in the QoT; and iii) estimating the evolution of the value of such parameters to find whether the soft-failure will degenerate into a hard-failure. This advanced network performance analysis procedure facilitates diagnosis and network maintenance. Furthermore, because the relation between monitored SNR and modeling parameters is not linear, the analysis carried out in the later space (i.e., modeling parameters) can accelerate soft-failure detection, identification, and localization. Specifically, the contributions are:

1. The MESARTHIM methodology for soft-failure detection, identification, and localization, modeling parameter estimation and severity estimation. Section II details its building blocks.

2. Network surveillance and soft-failure localization based on device modeling parameter estimation, which is a key part of the MESARTHIM methodology. Starting from [24], algorithms based on the analysis of the SNR and optical device modeling parameters are detailed in Section III. Moreover, this section presents a procedure that combines the analysis of SNR values with the expected ones, obtained with the open source GNPy QoT tool [25], to find the most likely value of modeling parameters.

3. Procedures for identification and severity estimation, once a soft-failure has been detected and localized. Section IV applies time series forecasting techniques to provide answers to critical questions, such as whether and when an SNR threshold will be violated for a given soft-failure.

The discussion is supported by the experiments and numerical results presented in Section V.

II. THE MESARTHIM METHODOLOGY

Among the effects degrading the QoT within optical systems, in this work, we consider degradations arising
from ROADMs and In-Line OAs, where a ROADM consists of WSSs and OAs. Both building blocks face aging and non-ideal conditions. For example, although OAs are considered robust devices, they also suffer time-varying effects like Noise Figure (NF) which might increase over time due to the aging of the amplifier building blocks. The NF is also frequency-dependent and, as the allocation of the spectrum might become time-dependent. Therefore, the NF can be modeled as a time-frequency variation. The pump lasers of the EDFAs also present degradation, which can be adjusted thanks to internal control loops, but which still reduces the EDFA efficiency. For what concerns the WSSs, they might suffer temperature-dependent variations, which might lead to frequency shift over time; furthermore, individual channels can drift as well, and both effects can be highly detrimental in terms of QoT. In the context of this work, we consider gradual time-varying device degradations on OA and add/drop (A/D) WSSs in the ROADMs. Specifically, we consider that soft-failures can be explained by one of the following events in the modeling parameters: a) NF increase; b) maximum optical output power (P-max) decrease; and c) Optical SNR (OSNR) degradation caused by frequency drifts of the WSSs due to temperature fluctuation.

Our proposed architecture for soft-failure analysis is illustrated in Fig. 1. The optical layer consists of a disaggregated set of ROADMs and TRXs, and a set of optical links with a number of In-Line OAs interconnecting ROADMs. The control plane includes: i) a Network Controller to program the network devices, which is coordinated by the Network Management plane, and includes network maintenance; ii) an MDA system [6] that collates measurements from the data plane, analyses the data and issues recommendations to the network controller, as well as notifications regarding failures; and iii) a QoT tool based on GNPy that estimates the SNR of the lightpaths and it is used for connection provisioning and for failure analytics.

The MDA system stores a replica of the operational databases (DB) that are synchronized from the network controller. In addition, it collects measurements from the optical devices with a given periodicity and stores them in a Monitoring DB; in this work, we assume that the MDA collects SNR samples from the TRXs every 15 minutes. These measurements are used by MESARTHIM to: i) estimate those modeling parameters related to optical devices (resources); ii) analyze the evolution of the measured SNR and that of the modeling parameters to detect any degradation as soon as it appears; and iii) determine the severity of the degradation based on the foreseen impact on the performance of the lightpaths.

Fig. 1 also sketches the MESARTHIM methodology implemented in the MDA system. Specifically, the following building blocks can be identified: (1) the Surveillance block that analyzes the SNR measurements and the value of modeling parameters to detect any meaningful degradation (e.g., by threshold crossing); (2) the Localization block that localizes the soft-failure; (3) the Find Modeling Configuration block that finds the most likely value of the modeling parameters of a given resource, so it results into SNR values of the lightpaths being supported by such resources similar to those that have been actually measured; (4) the soft-failure Identification block that, assuming a resource has been localized as the source of the soft-failure, finds what is the modeling parameter responsible for such failure; and (5) the Severity Estimation block that estimates whether and when the soft-failure will degenerate into a hard-failure. In addition, two internal repositories are used: i) the Device Modeling Config DB with the evolution of the value of modeling parameters along time for every resource; and ii) the Network Diagnosis DB that stores historical data for analysis purposes. The MESARTHIM manager coordinates those blocks to achieve intelligent QoT analysis, as well as manages the interface with the
TABLE 1. NOTATION

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$G$</td>
<td>Graph representing the network topology.</td>
</tr>
<tr>
<td>$P$</td>
<td>Set of all lightpaths</td>
</tr>
<tr>
<td>$P'$</td>
<td>Subset of lightpaths ($P' \subseteq P$)</td>
</tr>
<tr>
<td>$R$</td>
<td>Set of optical devices, index $r$.</td>
</tr>
<tr>
<td>$C$</td>
<td>Set of clusters of lightpaths with found same behavior (${&lt;behavior, P'&gt;}$)</td>
</tr>
<tr>
<td>$SR$</td>
<td>Set of resources suspicious. Each element identifies the resource $r$ and the lightpaths that it supports ($SR = {&lt;r, P'&gt;}$).</td>
</tr>
<tr>
<td>$SF$</td>
<td>Set of Soft-Failures. ($SF = {&lt;r, P'&gt;}$)</td>
</tr>
</tbody>
</table>

The main procedures for the different blocks of MESARTHIM are detailed next.

III. SURVEILLANCE AND LOCALIZATION

In this section, we describe two different approaches for the surveillance block, named SNR-wise that analyzes the evolution of the SNR, and Modeling-wise that analyzes the evolution of the value of modeling parameters. Resources affected by a soft-failure procedure are localized. Additionally, the main procedure for the Find Modeling Configuration block is presented. Table 1 introduces the used notation. We assume that surveillance is carried out periodically, e.g., after at least one new SNR measurement has been collected for every lightpath in the network. Both algorithms return the resources with the found likely modeling configuration, each with a subset of lightpaths, indicating that some soft-failure has been detected.

A. SNR-wise Surveillance

This approach focuses on the analysis of SNR measurements and compares them to the SNR estimated by the QoT tool for every lightpath, to detect any meaningful deviation (exceeding a differential threshold). The lightpaths that exceed the differential threshold are considered degraded and are further analyzed in terms of the behavior of the measured SNR evolution to find a correlation among them; behavior is the result of stationarity analysis [26] of the SNR evolution. Non-stationary patterns (e.g., trend, periodicity), if found, are quantified (e.g., a period interval in case of seasonality) to compose the behavior. For illustrative purposes, Fig. 2 shows three examples of behavior: a) stationary (typical for lightpaths that do not exceed the differential SNR threshold); b) gradual decay; and c) cyclic. In the case of finding groups of lightpaths with similar behavior, common underlying resources are analyzed to localize the responsible for such degradation. It is worth highlighting that grouping lightpaths with similar behavior enables localizing multiple soft-failures. In such a case, the likely configuration parameters are estimated using the Find...

Algorithm I describes the procedure used to find the behavior of the evolution in time of a time series data. The algorithm receives an object $x$, which includes: i) the time series to be analyzed (e.g., the SNR of a path $p$); ii) a window size $w$ used for smoothing purposes; and iii) a threshold $thr$ used to detect a significant non-stationary behavior. The data series ($Y$) is smoothed ($\hat{Y}$) by computing a moving average in non-overlapped windows of size $w$ (lines 1-2 in Algorithm I). Then, the difference between maximum and minimum in $\hat{Y}$ is computed and compared with the threshold; if it is lower than the threshold, no behavior is returned (line 3). Otherwise, the algorithm carries out an analysis to characterize the type of behavior by clearly distinguishing between gradual degradation (gradual) and cyclic fluctuation (cyclic), as well as any other undefined evolution (e.g., random peaks). Specifically, if $\hat{Y}$ presents an incremental or decremental monotonic evolution, the fittest trendline (in terms of Pearson correlation coefficient) among linear, polynomial, exponential, and logarithmic trends is computed and returned as a parameter of the identified gradual degradation (lines 4-6). Otherwise, the periodicity of $\hat{Y}$ is computed based on the results of automated periodogram power spectral density analysis [26]. In case that a significant period is found, cyclic fluctuation with that period is returned (lines 8-9), whereas other behavior is returned if neither gradual nor cyclic behavior is found (line 10).

Algorithm II details the pseudocode of the SNR-wise Surveillance algorithm; it receives as input the network...
**Algorithm II. SNR-wise Surveillance Algorithm**

**INPUT:** \( G, P, T, \text{current}_t \)  
**OUTPUT:** \( SR \)

1. \( P' \leftarrow \emptyset \)
2. for each \( p \) in \( P \) do
3. if getMonitoringData\((p, \text{current}_t)\) < \( \text{SNR\_threshold}(p) \) then
4. \( P' \leftarrow P' \cup \{ p \} \)
5. if \( P' = \emptyset \) then return \( \emptyset \)
6. \( C = \{ \langle \text{behavior}, P' \rangle \} \leftarrow \emptyset \)
7. for each \( p \) in \( P' \) do
8. \( p_{\text{snr}} \leftarrow \text{getMonitoringData}(p, T) \)
9. hasBehavior \( \leftarrow \text{findBehavior}(p) \)
10. if hasBehavior then addBehaviorSimilarity\((C, p)\)
11. if \( C = \emptyset \) then return \( \emptyset \) else \( SR \leftarrow \emptyset \)
12. for each \( c \) in \( C \) do
13. \( R = \{ \langle \text{resource}, P' \rangle \} \leftarrow \)
14. \( \text{FindCommonResources}(G, c, P') \)
15. \( SR \leftarrow SR \cup R \)
16. for each \( r \) in \( SR \) do
17. \( \text{r.evolve} \leftarrow \emptyset \)
18. for each \( t \) in \( T \) do
19. \( \text{findLikelyModelingConfig}(r, t) \)
20. return \( SR \)

In this surveillance approach analyzles the evolution of the value of modeling parameters of the resources. In this case, the SNR measurements of all the lightpaths in the network supported by a resource are used to estimate the most likely modeling configuration of such resource using the Find Modeling Config block. The found modeling configuration is stored and its evolution is analyzed to detect any meaningful degradation, e.g., a significant trend and/or variation.

Algorithm III details the pseudocode of the Modeling-wise Surveillance algorithm; it first initializes the \( SR \) data structure (line 1 in Algorithm III) and creates the set of resources with the lightpaths that each one supports (line 2). Next, for every single resource, the algorithm uses the last SNR measurements to find the current value of the parameters modeling the resource, which are stored in the Device Modeling Config DB by the Find Modeling Configuration block (lines 3-6). The behavior of the modeling parameters evolution is analyzed (by calling Algorithm I) and, if a non-stationary pattern is found, the resource is added to the \( SR \) set (lines 7-9). It is worth noting that this analysis could detect soft-failures that have not yet had a relevant impact on the lightpaths (i.e., the SNR degradation threshold has not been exceeded yet), as parameters and SNR are not linearly related.

**Algorithm IV. Soft-Failure Localization Algorithm**

**INPUT:** \( SR \)  
**OUTPUT:** \( iSF, mSF \)

1. for each \( r \) in \( SR \) do
2. \( R_r \leftarrow \emptyset \)
3. for each \( r' \) in \( SR \mid r \neq r' \) do
4. if \( r.P = r'.P \) then \( R_r \leftarrow R_r \cup \{ r, r' \} \)
5. \( SR \leftarrow SR - R_r \)
6. \( iSF \leftarrow \emptyset \)
7. for each \( r \) in \( SR \) do
8. if \( r.P \cap r'.P = \emptyset \forall r \mid r \neq r' \) then
9. \( iSF \leftarrow iSF \cup \{ r \} \)
10. \( SR \leftarrow SR - \{ r \} \)
11. return \( iSF, SR \)

C. Soft-Failure Localization

In case that the surveillance phase has identified a set of suspicious resources, Algorithm IV localizes the soft-failures. The algorithm first removes the suspicious resources that explain the very same set of lightpaths (lines 1-6 in Algorithm IV), as localization is not yet possible in those cases. Next, the resources that explain the SNR of complete subsets of lightpaths are localized and classified as independent soft-failures, \( iSF \) (lines 6-10). The rest of the suspicious resources are related to

---

Graph \( G \), the list \( P \) of lightpaths currently established in the network, the number \( T \) of historical monitoring samples to be considered, and the current time \( \text{current}_t \). The algorithm first retrieves and examines the last monitoring data available for every lightpath looking for those with degraded SNR (lines 1-5 in Algorithm II). In case some SNR degradation is found, SNR-wise Surveillance proceeds with an in-depth SNR analysis carried out in two steps. During the first step, the set of clusters \( C \), capturing the behavior observed in the lightpaths, is found (lines 6-10); for this analysis, the last \( T \) monitoring samples are considered. Note that by considering the evolution of lightpaths’ SNR, spurious measurements in one lightpath can be detected and ignored. In the case that, at least, one set of lightpaths presents a similar behavior, e.g., decay or periodicity (as in Fig. 2), the algorithm continues with the second step. The common resources supporting the lightpaths in each cluster \( c \) are computed and added to the set \( SR \) of resources that are suspicious of being affected by a soft-failure (lines 12-14). Moreover, for each common resource, a likely evolution of the input parameters is found (lines 15-17); the estimated configuration is stored in the Device Modeling Config DB for further analysis. The resources with the found likely configuration, each with a subset of lightpaths, are eventually returned (line 18).

B. Modeling-wise Surveillance

This surveillance approach analyzes the evolution of the resources that explain the most likely modeling configuration of such resource using the Find Modeling Config block. The
soft-failures that affect common subsets of lightpaths, so a given lightpath can be affected by more than one soft-failure.

D. Finding the Most Likely Modeling Configuration

The above surveillance approaches use the FindModeling Configuration block, which estimates the most likely modeling configuration of a given resource \( r \). Given the ranges of feasible configuration values \( \mathbb{V} \) of \( r \), the configuration estimation problem consists in finding the most likely values \( v \), by minimizing the error (computed as mean squared error - \( \text{MSE} \)) between the measured \( \langle S \rangle \) and the estimated \( \langle \hat{S} \rangle \) SNR for the set of lightpaths being supported by \( r \), \( P(r) \), i.e.:

\[
\min_{v \in \mathbb{V}} \text{MSE} \left( \langle S(P(r)) \rangle, \hat{S}(P(r)|v) \right)
\] (1)

The SNR estimation \( \hat{S}(\cdot) \) is obtained by calling the QoT tool and then, the optimization problem in Eq. (1) cannot be solved by traditional methods like Steepest Descent, which are based on computing the gradient of the function to be minimized. In view of that, the findLikelyModelingConfig() procedure uses the modelingConfigSearch() one (Algorithm V) to solve the optimization problem in Eq. (1). The algorithm interrogates the QoT tool with different values of the parameters and the configuration entailing the lowest error with respect to the SNR values measured is returned. The procedure assumes that: \( i \) the function is convex in \( \mathbb{V} \), i.e., there is just one minimum that is the global minimum (tests supporting this assumption will be carried out); and \( ii \) there is just one of the modeling parameters of \( r \) with a value different than the initially found. The procedure fixes the value(s) of the parameter(s) and requests the QoT tool to compute, with such configuration, the SNR values of the lightpaths being supported by \( r \). By computing the \( \text{mse}(\cdot) \) function, the procedure determines if such configuration is likely enough or if more queries to the QoT tool are needed.

As calls to the QoT tool are time consuming, modelingConfigSearch() targets at minimizing them; instead of using the brute force and request the estimation of the SNR for the whole range of possible values, the procedure uses the projection of two points to determine the next value of the parameters to be used for the estimation. As parameters typically evolve smoothly in time when devices are affected by a soft-failure, it also stores the last configuration(s) in the Device Modeling Config DB; every time it is called, it uses such configurations as starting points for the search aiming at finding the optimal solution fast.

IV. SOFT-FAILURE IDENTIFICATION AND SEVERITY ESTIMATION

Let us now focus on the Soft-Failure Identification and the Severity Estimation blocks, which are assumed to be executed as soon as degradation is detected and localized (via SNR and/or device configuration analysis). These blocks make MESARTHIM able to generate notifications to the network controller containing not only the list of degraded lightpaths, but also their expected evolution and the estimated time for the soft-failure to degrade into a hard-failure.

Let us imagine that the surveillance and localization analysis in Section III detected some degradation at time \( t \) and therefore, non-empty ISR and/or MSF sets were found. In addition, let us define \( \text{state} \) as the combination of the estimated modeling configuration of the resources that are involved, and the SNR experienced by the supported lightpaths. With the aim of an accurate diagnosis, it is interesting not only to analyze the current state but also to predict its future evolution. It is for this very reason that historical data (within a pre-defined time window) are gathered from monitoring and device config DBs and stored in the Network Diagnosis DB. The evolution of each parameter (lightpaths’ SNR and device modeling config parameters) is analyzed individually using time series forecasting techniques. By using different techniques with different parametrization, several expected evolutions can be obtained as a practical way to generate different likely projections for parameter degradation.

For illustrative purposes, let us analyze an example of the evolution of three parameters selected from a hypothetical state: the SNR of a given lightpath affected by an SNR degradation, as well as the NF and P-max parameters modeling an OA traversed by that lightpath. Fig. 3 presents three time-evolutions of the OA modeling parameters and SNR, and their analysis at current time \( t \). At this time, the failure has been correctly localized in the selected OA; however, the cause of the degradation (either NF or P-max) is still

---

**Algorithm V. Modeling Config Search**

**Input:** \( P, r, \mathbb{V}_m, \mathbb{V}_v \)  
**Output:** \( <v, m> \)

1: \( <v, m> \leftarrow \text{configDB}.\text{getMin}(); \text{numIters} \leftarrow 0 \)
2: **while** \( m > \epsilon \text{<MaxIters>} \) **do**
3: \( <v, m> \leftarrow \text{configDB}.\text{getNeighbours}(v) \)
4: \( v_a \leftarrow \text{intersect}(v, m, \mathbb{V}_m, \mathbb{V}_v, \text{Value}) \)
5: \( v_b \leftarrow \text{intersect}(v, m, \mathbb{V}_m, \mathbb{V}_v, \text{Value}) \)
6: **if** \( v_a = v_b \) **then** \( v_a \leftarrow (v_a + v_b) / 2 \)
7: **if** \( v_a = v_b \) **then** \( v_b \leftarrow (v_a + v_b) / 2 \)
8: \( m_a \leftarrow \text{configDB}.\text{getConfig}(v_a) \)
9: **if** \( m_a = \text{None} \) **then**
10: \( m_a \leftarrow \text{MSE}(P, r, \text{QTool}(P)), \text{getConfig}(v_a) \)
11: \( \text{configDB}.\text{addConfig}(v_a, m_a) \)
12: \( m_b \leftarrow \text{configDB}.\text{getConfig}(v_b) \)
13: **if** \( m_b = \text{None} \) **then**
14: \( m_b \leftarrow \text{MSE}(P, r, \text{QTool}(P)), \text{getConfig}(v_b) \)
15: \( \text{configDB}.\text{addConfig}(v_b, m_b) \)
16: \( <v, m> \leftarrow \text{configDB}.\text{getMin}() \)
17: **return** \( <v, m> \)
unclear. This is the reason behind performing the estimation and evolution analysis for all modeling parameters; such analysis is carried out following forecasting techniques. Three curves representing the upper ($f_u$), centered ($f_c$), and lower ($f_l$) projected possible evolutions are depicted for each parameter. Based on such projections, monitored data are evaluated and an indicator ($\phi$) is computed and used to discard modeling parameters while identifying the one that is the most likely to be the real cause of the observed performance degradation. Hence, the indicator provides large values when the modeling parameter is far from the expected behavior of an affected parameter. The obtained indicator is accumulated with the values obtained in the previous computations, and identification is positive when the parameter with the lowest accumulated indicator is far from all the rest of the parameters with a significantly higher accumulated one. This evidence can be easily observed by setting up a threshold allowing the discrimination of low and high accumulated indicator modeling parameters.

Once the Soft-Failure Identification block has found enough evidence of the modeling parameter explaining the soft-failure, the Severity Estimation block uses the centered projection of the evolution of such modeling parameter to estimate the likely evolution of the SNR for the affected lightpaths (i.e., those being supported by the localized resource). Analyzing such evolution, it is easy to check whether any lightpath would exceed a given threshold. For instance, the minimum SNR defined by its modulation format and bit rate. The next subsections detail these two MESARTHIM blocks.

### A. Soft-failure Identification

Algorithm VI is used to identify the most likely modeling parameter(s) explaining the observed performance degradation; it receives as input the resource responsible for the soft-failure and the number $T$ of historical monitoring samples, and, if sufficient evidence is found, it returns the modeling parameter identified as a potential failure.

The algorithm starts by retrieving the modeling parameters $v$ that characterize the state of the resource, as well as the evolution of each modeling parameter in the last $T$ measurements (lines 1-2 in Algorithm VI). Then, each parameter $i$ in $v$ is evaluated independently to compute a score indicating how likely is that such parameter is responsible for the resource failure. Specifically, the time series of the parameter $i$ is selected and split into two portions: one with the first $T-\delta$ data points ($Y_{\delta}$) used for fitting and a second one ($Y_{\text{eval}}$), with the last $\delta$ measurements, used for evaluation and indicator computation (lines 3-6) (dark grey area in Fig. 3). Using the $Y_{\delta}$ segment, a set of time-dependent functions $F$, where the parameter value is modeled as a function of time, is obtained by applying Holt-Winters exponentially-based modeling and polynomial fitting in a wide range of degrees (e.g., from 1 to 7) [30] (line 7). The set $F$ contains all functions with similar goodness-of-fit, i.e., +/- 5% of variation in terms of Pearson correlation coefficient. Moreover, functions are evaluated in the time interval of $Y$ to verify that trend is always monotonic; otherwise, the function is discarded. Then, three of those functions are selected to be the likely projections illustrated in Fig. 3 (line 8). This selection contains: i) $f_l$ (lower) and $f_u$ (upper), with the functions with the lowest and highest value at the current time, respectively; and ii) $f_c$ (centered), with the function with the smallest number of coefficients providing maximum Pearson correlation coefficient (+/- 5%), which is assumed to be the most likely parameter evolution.

After obtaining the likely projections, they are compared with the trendline (computed using the same methodology of $f_c$) in $Y_{\text{eval}}$. This comparison is carried out using an additive multi-factorial indicator ($\phi$) that combines several Boolean and continuous variables.

**Algorithm VI. Soft Failure Identification**

**INPUT**: $r, T$  
**OUTPUT**: param $
\begin{align*}
1: & \quad v \leftarrow \text{getModelingParameters}(r) \\
2: & \quad r.\text{eval} \leftarrow \text{configDB.SELECT}(r, T) \\
3: & \quad \text{for each } i \in v \text{ do} \\
4: & \quad \quad Y \leftarrow \text{getTimeSeries}(r.\text{eval}, i) \\
5: & \quad \quad Y_{\delta} \leftarrow Y[1..T-\delta] \\
6: & \quad \quad Y_{\text{eval}} \leftarrow Y[T-\delta+1..T] \\
7: & \quad \quad F \leftarrow \text{fitTimeDependentFunctions}(Y_{\delta}, \delta) \\
8: & \quad \quad F^* = \{f_l, f_c, f_u\} \leftarrow \text{select Likely Evolution}(F, T) \\
9: & \quad \quad r.\phi[i] \leftarrow \text{selectHighIndicatorParameters}(r) \\
10: & \quad v^* \leftarrow \text{selectHighIndicatorParameters}(r) \\
11: & \quad \text{for each } i \in v^* \text{ do} \\
12: & \quad \quad r.\text{accum}[i] += r.\phi[i] \\
13: & \quad \quad \text{if } r.\text{accum}[i] > \text{thr then } v \leftarrow v^i \\
14: & \quad \text{if } |v| = 1 \text{ then return } v.\text{getFirst()} \\
15: & \quad \text{return } \emptyset
\end{align*}
TABLE 2. INDICATOR FUNCTION COMPONENTS

<table>
<thead>
<tr>
<th>i</th>
<th>Description ($x_i$)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>[boolean] Projections $f_k$, $f_c$, $f_u$ do not show parameter degradation</td>
</tr>
<tr>
<td>2</td>
<td>[continuous] $\text{rMSE}(Y_{eval}, f_i)$</td>
</tr>
<tr>
<td>3</td>
<td>[continuous] $\min(\text{rMSE}(Y_{eval}, f_i))$ for $i = {l,c,u}$</td>
</tr>
<tr>
<td>4</td>
<td>[boolean] $f_i = f_c$ OR $f_u = f_c$</td>
</tr>
</tbody>
</table>

according to Eq. (2), where $x_i$ represents a variable and $b_i$ the weighting coefficient for that variable.

$$\varphi = \sum_{i=1,n} b_i \cdot x_i$$

Table 2 briefly describes the considered variables, sorted in the descendend degree of importance. Two Boolean functions are used: $x_l$ returns True if there is no evidence of degradation in the parameter, whereas $x_u$ checks if any of the upper and lower projections is the same function as the centered one. In addition, two continuous variables are defined: $x_c$ accounts for the relative Mean Square Error (rMSE) of the data in the evaluation portion with respect to the centered projection, whereas $x_u$ returns the minimum rMSE of the data with one of the projections. Note that if measurements in the evaluation portion follow a trend showing a parameter degradation similar to the projections (preferably, the centered one), the indicator remains low; otherwise, the indicator increases indicating that the observed behavior differes from the expected one and therefore, the likelihood of the parameter not to be the cause of the failure increases.

Once the indicator is computed for all modeling parameters characterizing the state of the resource, the selection of parameters, whose indicator is significantly higher than the rest, is conducted (line 10). Thus, considering $\varphi_{\text{min}}$ as the minimum indicator value of a parameter, the limit $\varphi_{\text{min}} + \Delta \varphi$ is defined as the reasonable limit for parameters with low indicator, being all parameters above $\varphi_{\text{min}} + \Delta \varphi$ selected as those with high indicator. For the resultant set of parameters (if not empty), the cumulative indicator is updated by adding the current one (lines 11-12). Finally, the cumulative indicator is compared with an identification threshold $thr$ and, if exceeded, the parameter is removed from the candidate parameter set (line 13). The identification is considered positive when just one parameter remains as a candidate and it is eventually returned (line 14). Otherwise, the algorithm returns no identification (line 15). It is worth noting that the value of $thr$ needs to be high enough to clearly identify the parameter without false positives.

B. Severity Estimation

Once a modeling parameter has been identified as the cause of the observed degradation, the severity estimation algorithm is executed. This method uses the projection of the modeling parameters to estimate when some affected lightpath will cross the FEC limit.

Algorithm VII. Severity Estimation

INPUT: $r, P, T, \text{param}$

OUTPUT: $\text{estimatedTime}$

1. $r$, $\text{evol} \leftarrow \text{modelingConfigDB}.\text{SELECT}(r, T)$
2. $f_c \leftarrow \text{getCenteredProjection}(r, \text{evol}[\text{param}], \text{timeWindow})$
3. for $i$ in $[1, \text{timeWindow}]$ do
4.   $\text{SNR} \leftarrow S(P | f_i(t))$
5.   for each $p$ in $P$ do
6.     if $\text{SNR}[p] < p.\text{minSNR}$ then return $t$
7. return INF

Although the severity estimation could be based on the projected evolution of the modeling parameters, defining a threshold for each one is not easy as different lightpaths are impacted differently by its degradation.

Algorithm VII details the pseudocode; it receives as input the resource responsible for the soft-failure, the list of lightpaths supported by such device, the number $T$ of historical monitoring samples, and the identified modeling parameter, and it returns the estimated time that the soft-failure will cause a major impact on, at least, one of the lightpaths. The algorithm first retrieves the last $T$ configuration values from the Device Modeling Config DB, which are used to compute the centered projection $f_i$ for the considered input parameter on a given time window (lines 1-2 in Algorithm VII). The projection $f_i$ is used as the input parameter to estimate the SNR for the list of lightpaths; in case that the estimated QoT for any of the lightpaths falls behind the minimum SNR, the algorithm returns the estimated time of this event to happen (lines 3-6); otherwise, it returns a large value that exceeds the considered time window (line 7).

V. RESULTS

A. Experimental assessment

Being the basis of the MESARTHIM methodology, the Find Modeling Configuration has been evaluated experimentally in the testbed depicted in Fig. 4. Two commercial coherent transponders (labeled TRX-1 and TRX-2), with optical line interface at 100G (32-GBd Quadrature Phase-Shift Keying -QPSK), have been connected using an optical multi-span link. The pair of transponders has been equipped with a specifically designed driver enabling both the configuration and the real-time monitoring of the SNR; the generated signal is filtered by a WSS Wave Shaper device. The optical link consists of 4 spans, each realized by an 80 km single-mode-fiber spool, for a total distance of 320 km. Five EDFAs have been used to compensate for the power attenuation; all being the single stage with gain in the range 15-25 dB. OA1 was configured with a constant 17.5 dB gain to compensate for the filter insertion losses and the other OAs with a constant gain of 16 dB to compensate entirely for the fiber losses within the span.

Two different experiments were carried out. In the first, we reproduced the effect of a filter detuning. We
configured the TRXs at 193.9 THz and the WSS with a bandwidth of 50 GHz, collecting the estimated SNR at TRX-2. Then, we reconfigured the WSS filter, reducing the bandwidth with steps of 2 GHz, until the operational status of TRX-2 card was down. GNPy was used as a tool to estimate the expected QoT for the lightpath. In order to estimate the QoT, together with the scenario that includes fiber types and span length, modeling parameters of the WSS, OAs and the TRXs are provided as input to GNPy; it estimates the QoT using the generalized GN model [3], which considers both the ASE noise and NL Interference (NLI) accumulation.

Assuming that the device responsible for the SNR measured in TRX-2 is unknown, we run the Find Modeling Configuration block of MESARTHIM to estimate the most likely modeling configuration of the WSS and one of the OAs in the optical link. Fig. 5 presents the results obtained when the observed SNR is explained by a reduction in the OSNR of the A/D WSS (Fig. 5a) and by an increased value of the NF in one of the OAs (Fig. 5a). We observe that changes in both modeling parameters could explain the evolution of the observed SNR in the lightpath with minimal error, being the resulting values of the modeling parameters within a feasible range. Note that with just one lightpath in this experiment, it is not possible to make any localization, as any of the devices could be responsible for the observed reduction in the SNR.

In the second experiment, we slightly changed the multi-span link scenario with respect to that in Fig. 4, by adding 5 dB attenuators before spans 2-4 emulating an additional 25 km in each span. Therefore, the gain of OAs 3-5 had to increase to 21 dB to compensate for the increased losses that also affected the NF of our amplifiers which is inversely proportional to the configured gain. The experiment was carried out in three steps, where one span was modified at a time. For each step, we continuously collected the estimated SNR at TRX-2, detecting the variation of the transmission metrics during the testbed evolution. Fig. 6 presents the results from the Find Modeling Configuration block when the length of the spans 2-4 was increased to an equivalent of 105 km and consequently, the gain of OAs 3-5. At each step, the module was able to explain the increment in the SNR of the lightpath by a reduction in the NF of the related OA. For illustrative purposes, the estimated SNR of the lightpath that would be obtained by keeping the NF constant is also represented in Fig. 6. These two experiments assess the high accuracy of the Modeling Config Search for estimating the modeling parameters of the devices.

The next subsections evaluate the MESARTHIM methodology through simulation.

B. Simulation environment

For the simulation, we selected a German-like network topology with 17 nodes and 26 bidirectional links (see Fig. 7). 136 bidirectional lightpaths, representing all the origin-destination pairs, were established through the shortest route in terms of hops. Fig. 8 plots the number of lightpaths that every link in the network is supporting (which is particularly important for failure localization). For the sake of simplicity in the analysis of the results, we assume that all signals use the QPSK modulation format.

The optical data plane was simulated by a GNPy instance. We generated SNR measurements for every lightpath by varying every modeling parameter of every intermediate OAs and A/D WSSs in the ROADMs in the network, independently. With these measurements, a set of realistic types of failures (use cases) affecting optical devices were reproduced by forcing the modeling parameters of the selected devices (NF and P_\text{max} in the OAs and OSNR in the WSSs) to vary over time. From the different variations that might happen, we focus on gradual variations, i.e., those soft-failures that can eventually degenerate into hard-failures. Among all possible gradual degradations, we focus on the exponential increase (NF) and logarithmic decay (P_\text{max} and OSNR), since both types of degradations accelerate in time and hence, it is crucial to anticipate their detection and localization as much as possible. Finally, variability to the SNR samples was added in the form of random noise.
The resulting samples were stored in the simulated control plane and fed the module implementing the MESARTHIM methodology. In the case of the SNR-wise Surveillance algorithm, the SNR threshold (line 3 in Algorithm II) was set to the expected SNR for each given lightpath minus a fixed value that exceeds the random variations introduced by the monitoring generator.

The next subsections present the obtained results for the different procedures of the MESARTHIM methodology based on this simulation setup.

C. Surveillance and Device Configuration Estimation

Let us first illustrate the convergence of the Modeling Config Search algorithm with an example entailing two sets of lightpaths. We are interested in finding the most likely modeling config for the OSNR of an A/D WSS and for the NF of an OA (each supporting one of the sets of lightpaths), given its monitored SNR. Fig. 9 plots the MSE as a function of the configuration value, as well as those values explored by the algorithm for the two optical devices. The inset tables specify the MSE values, where the configuration that gives the minimum MSE is finally selected. From these results, as well as from those in the experimental assessment, we conclude that the algorithm converges in the whole range of the true soft-failure origin, regardless of the selected QoT parameter.

We now focus on the evolution of the SNR over time for the defined use cases. The graphs in the upper row in Fig. 10 present such evolution, where, for the sake of clarity, we plot only one sample of the affected lightpath. Note that only the lightpaths affected by the failure will experience an evolution in their SNR, whereas the rest of the lightpaths will show no variation over time other than a random one plus some uncorrelated spurious measurements introduced by the monitoring generator. The time in the graphs is normalized, as the time-scales for the considered soft-failures are different, ranging from days to months or even years. The evolution of the modeling parameters is shown in the bottom-row graphs, where the actually programmed value and the interval of values [max, min] estimated by the Find Configuration block is plotted.

We observe that the range of possible values of the modeling parameters is tighter when the value of the parameter deviates from its nominal one. In addition, the range of possible values for the modeling parameters is different for the different parameters, being the P-max of the OAs the one with the largest range. This might have a clear impact if the detection of the soft-failure is performed by tracking the evolution of that parameter. Fig. 11 complements Fig. 10 by plotting the maximum and average error in the estimation of the modeling parameters as a function of the magnitude of the degradation. We observe in Fig. 11a that both maximum and average P-max estimation errors are high for low degradation magnitudes (15.6% and 7.8%, respectively). In contrast, the average error for NF and A/D WSS (Fig. 11b,c) are remarkably low and almost constant for the degradation magnitudes studied. Fig. 11 also confirms the observation regarding the error in the estimation of the value of the modeling parameters greatly reduces with the magnitude of the degradation, which is a very promising result and it can be exploited for soft-failure localization and identification.

Some conclusions can be drawn from the results obtained so far: 1) the proposed method for estimating the value of modeling parameters of the devices has shown remarkable accuracy in the experimental tests, which has been confirmed by simulation for all failure use cases; 2) in general, the estimation interval is tighter
when the impact of the value of the parameter on the observed SNR is higher; 3) in the specific case of the maximum power of the OAs, the range of values that result in the SNR values observed is large when the observed SNR remains around the nominal value. However, when SNR degrades with evident trend, the correlation between P-max and SNR becomes larger.

To help developing intuition about the differences that can be expected by analyzing the SNR of the lightpaths and the value of the modeling parameters of optical devices, Fig. 10 compares the time to detect a degradation by analyzing the measured SNR and the value of the modeling parameters. For the sake of simplicity, let us assume that degradation detection is performed by threshold crossing; the threshold was set to 1 dB below the nominal SNR value for the lightpaths, not below a minimum SNR resulting in a pre-FEC BER over $4 \cdot 10^{-3}$ for QPSK signals. The thresholds related to modeling parameters were defined as a percentage of the variation range given by the nominal value and the extreme value. Values are selected to reduce detecting false degradations: i) for P-max it was set to 40% in the interval between the nominal value (20 dBm) and the extreme value (10 dBm) due to the large range of variation observed; ii) for NF of the OAs, the percentage was set to 20% in the interval between the nominal (5 dB) and the extreme value (15 dB); and iii) for the OSNR of the A/D WSSs, the percentage was set also to 20% in the interval between the nominal (38 dB) and the extreme value (20 dB).

With these values, the detection of the degradation in the case of the SNR of the lightpath happened at normalized times 0.92, 0.86, and 0.86 for the P-max, NF, and A/D WSS OSNR gradual soft-failure use cases, respectively. This contrasts with the detection at times 0.78, 0.47, and 0.63 when the analysis was in the value of the P-max, NF, and OSNR of the A/D WSSs, respectively, which results in anticipation between 15% and 45%. Note that such anticipation is enabled by the different evolution of modeling parameters and their non-linear impact on the SNR of the lightpaths.

D. Soft-Failure Localization

The above discussion considered the time for the detection only. Note that soft-failure location (Algorithm IV in Section III.C) requires several lightpaths to find the common resources in the network topology. When the evolution of the monitored SNR changes suddenly, SNR-wise surveillance (Algorithm II) collects enough lightpaths to easily localize the failure; however, under a gradual degradation, the lightpaths exceeding the threshold might be not enough for the localization.
### Table 3. Examples of Soft-Failure Localization

<table>
<thead>
<tr>
<th>Failure in OA in link Frankfurt-Mannheim</th>
<th>Time</th>
<th>Degraded paths</th>
<th>Common Resources</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0.86</td>
<td>1 TRXs, 2 A/Ds, 1 Link</td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.90</td>
<td>2 A/Ds</td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.93</td>
<td>4 Links (Failure in Frankfurt-Mannheim)</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Failure in A/D WSS Dusseldorf</th>
<th>Time</th>
<th>Degraded paths</th>
<th>Common Resources</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0.86</td>
<td>1 TRXs, 2 A/Ds, 1 Link</td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.91</td>
<td>1 A/D WSS (Failure in Dusseldorf)</td>
<td></td>
</tr>
</tbody>
</table>

Fig. 8 includes a study of the maximum number of distinct routes that need to be considered to unambiguously identify every link as responsible for a soft-failure, considering that longer lightpaths will be more affected by device degradations. For the study, we selected all the links in the network, together with an incremental number of lightpaths selected by their total length, and fed Algorithm IV for the (multiple) soft-failure localization (i.e., 26 soft-failures were localized with a single execution of Algorithm IV). For the localization, not only the number of lightpaths is important, but also their routes. We repeated the experiments with the lightpaths sorted in inverse order, i.e., assuming that shorter lightpaths would exceed the (relative) threshold first. The results showed that all soft-failures could be perfectly localized when the resources of the two shortest lightpaths were analyzed.

With the above in mind, let us now show how the SNR-wise surveillance and localization evolves over time. Table 3 presents the results from two different failures. The first failure was in an OA in the link Frankfurt-Mannheim (supporting 41 distinct lightpaths), and the second in the A/D WSS in the Dusseldorf ROADM (supporting 16 distinct lightpaths). For each failure, each row shows the detection time when the measured SNR of some new lightpaths is below the threshold (bear in mind that the threshold is relative to the expected SNR for that specific lightpath). However, the localization of the soft-failure is not successful until just one resource (assuming that it is responsible for the failure) can be identified. Such identification happens at normalized times 0.93 and 0.91 for the failures in the OA and the A/D WSS, respectively. Note that the number of lightpaths needed to localize the soft-failure, although small, adds some extra time that could be of paramount importance for the impact on the network.

In contrast, soft-failure localization under the Modeling-wise approach considers all the lightpaths supported by such resource, as it analyzes the estimated evolution of the modeling parameters by resource. In consequence, Algorithm IV under the Modeling-wise approach can localize the cause of degradations in their very early stages (between 31% and 49% with respect to the SNR-wise approach).

As a final remark, it should be noted that even though the Modeling-wise approach considers all the lightpaths supporting a given resource, unambiguous localization might still not be possible if few lightpaths with distinct routes are established. Fig. 8 shows that, for some links in the considered scenario, unambiguous localization of a soft-failure is only possible when all supported lightpaths are analyzed. Therefore, in case that not all these paths are established, the localization procedure (Algorithm IV) would be unable to localize the failure. E.g., let us consider the soft-failure in the link Frankfurt-Mannheim in Table 3 and imagine that only one lightpath is supported by such link. Then, a degradation in that lightpath can be explained by degradation in more than one resource. This highlights the need for additional procedures to be applied for those scenarios which result in ambiguous localization.

### E. Identification and Severity Estimation

Once the degradation has been localized, let us focus on the identification of the modeling parameter responsible for such degradation. We assume that the device explaining the observed degradation is an OA. According to the notation in Section IV.A, identification performance is clearly dependent on the configuration of all coefficients related to indicator $\phi$. Several configurations were tested to find the one giving the desired importance to every component, ensuring that parameters with a high indicator are correctly selected, while guaranteeing no false positives. Such configuration is $<b_1, b_2, b_3, b_4, \Delta \phi, thr> = <20, 10, 2, 1, 5, 30>$, which will be used hereafter.

Fig. 12 shows the obtained results for a gradual degradation caused by NF, whereas Fig. 13 shows the results for a gradual degradation caused by P-max. In both cases, Fig. 12a and Fig. 13a plot the evolution of the accumulative indicator with time, and the decision threshold $thr$. Both indicators remain clearly under the threshold until time around 0.32, where enough evidence of the cause of the degradation is found. Note that the time of such identification represents 32% of anticipation compared to the earliest time obtained for degradation detection (at time 0.47) using a threshold on the evolution of the input parameter and 64% compared to the earliest detection time using a threshold on the evolution of the SNR. Fig. 12b-c and Fig. 13b-c show the computed projections for the two modeling parameters under study and for both failures; note that such figures are similar to Fig. 3. The projections are plotted for a long window (around 0.25 time units) for clarity purposes, although the value used for fitting and evaluation was $\delta = 0.03$ normalized time units.

In the case that the failure is a consequence of the NF degradation (Fig. 12b-c), we observe that the centered projection $f_1$ for the NF parameter is highly accurate and clearly between the upper and lower ones, which results in the minimum indicator parameter. On the contrary, P-max indicates an evident but less accurate projection and moreover, $f_1$ overlaps with $f_a$, which increases its
indicator above 5 units from the one of NF. As a consequence of this, P-max is selected as a parameter with a high indicator, so its accumulative indicator increased. Conversely, in Fig. 13b-c the minimum indicator is that of P-max, where there is an evident degradation for f_i. Note that this indicator is much lower than that computed for NF, where no degradation is observed, thus exceeding by far the indicator limit. In conclusion, we see how the proposed methodology allows discriminating the actual failing parameter and perform a fine failure identification.

We can take advantage of the identification method to
IMPLEMENT AN ALTERNATIVE LOCALIZATION METHOD THAT CAN BE APPLIED WHEN NOT ENOUGH LIGHTPATHS WITH DISTINCT ROUTES ARE ESTABLISHED IN THE NETWORK. THIS ALLOWS THE UNAMBIGUOUS LOCALIZATION OF A SOFT-FAILURE, AS MOTIVATED AT THE END OF SECTION V.D. IN THIS CASE, WE ASSUME THAT ONLY LIGHTPATHS BETWEEN THE TWO END ROADMS OF THE LINK FRANKFURT-MANNHEIM ARE ESTABLISHED AND HAVE CONSIDERED SOFT-FAILURES IN ONE OF THE SUPPORTING DEVICES (OA, TRX OR A/D WSS), IN LINE WITH TABLE 3. IN THIS CASE, WE EXECUTE ALGORITHM VI CONSIDERING A SINGLE VIRTUAL RESOURCE THAT ABSTRACTS THE SUPPORTING OPTICAL DEVICE TYPES. THE ALGORITHM RETURNS THE MOST PROBABLE MODELING PARAMETER AMONG THOSE OF OA, TRX AND A/D WSS, WHICH HELPS TO REDUCE THE NUMBER OF DEVICES TO BE ANALYZED MANUALLY.

Fig. 14 shows the performance of the identification procedure for soft-failure localization. Three scenarios are considered for the real cause of the soft-failure: i) gradual NF degradation in an OA (Fig. 14a), ii) gradual OSNR degradation in an A/D WSS (Fig. 14b), and iii) gradual OSNR degradation in a TRX (Fig. 14c). We observe that the accumulated score clearly increases for the two types of devices that are not the real cause of the failure for all the analyzed scenarios. This happens at time 0.3 in all the cases.

Finally, Fig. 15 presents the obtained results for severity estimation for gradual degradation of P-max, NF and A/D WSS OSNR as a function of the time. The plots show the evolution of the estimated time with the real-time, where the area in grey color highlights the real-time when the soft-failure degenerates into a hard-failure ±5%. We observe that the estimated time takes a large value when the estimation does not observe a major impact in the selected timeWindow for any of the affected lightpaths, and rapidly converges to the real degeneration time. In fact, assuming that the severity is accurately estimated after two consecutive executions returning estimated times close enough one to the other, anticipation over 42% to the degeneration time are obtained, which leave enough time to plan the adequate maintenance operations.

TABLE IV: SUMMARY OF THE MESARTHIM METHODOLOGY

<table>
<thead>
<tr>
<th>Method</th>
<th>Degradation Detection and Failure Localization</th>
<th>Cause Identification and Severity Estimation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Analysis in the lightpaths’ SNR space</td>
<td>• SNR-wise surveillance finds common resources in sets of affected lightpaths by analyzing its SNR. • Analyzes all lightpaths. For failure localization, the algorithm needs that several lightpaths to be affected.</td>
<td>• No cause can be identified. • QoT can be estimated based on the projected evolution of the SNR.</td>
</tr>
<tr>
<td>Analysis in the devices’ modeling parameters space</td>
<td>• Modeling-wise surveillance analyzes the value of the modeling parameters for all network devices. • The algorithm detects degradations and localizes their sources very ahead in time.</td>
<td>• Cause identification based on the projected evolution of the modeling parameter of the device where the failure is localized. • Severity estimation difficult to estimate by analyzing the evolution of the input parameters.</td>
</tr>
</tbody>
</table>

VI. CONCLUDING REMARKS

QoT estimation is typically carried out during the provisioning phase and in-operation planning to ensure that computed lightpaths will provide zero post-FEC errors, assuming some values for the QoT model input parameters related to the optical devices in the network. In this paper, QoT estimation was used for the reverse process, i.e., given the real measured QoT of a set of lightpaths, we were interested in estimating the value of the modeling parameters of the optical devices.

Because of the non-linear relation between lightpaths QoT and the value of the modeling parameters, the ability to estimate the value of such parameters opens the opportunity to analyze its evolution, which can be as a result of, e.g., aging, temperature variations, etc. The proposed MESARTHIM methodology combines analysis of the evolution of the monitoring QoT and their transformation into the estimated modeling parameters space, not only for the degradation detection, but also for its localization, identification, and severity estimation.

After the experimental assessment of the method for estimating the modeling parameters of the devices, the MESARTHIM methodology was evaluated through simulation. The methodology demonstrated remarkable anticipation in failure detection and localization by analyzing the estimation of the value of the modeling parameters of the devices. A simple example showed the reason behind such potentials in the different evolution of the modeling parameters and the lightpaths SNR. In addition, accurate cause identification based on the analysis of the projected evolution of the modeling parameters was demonstrated, which enabled the estimation of the severity in terms of the time when the soft-failure degrades into a hard-failure. Such severity estimation allows planning maintenance, as it largely anticipates degradation.

Table IV summarizes the main characteristics with pros and cons of the MESARTHIM methodology.
REFERENCES


