Bringing Disaggregated Telemetry and ML to the Transceiver for Autonomic Signal Adaptation

M. Felipe Silva[1], A. Pacini[1], A. Sgambelluri[1], L. Valcarenghi[1], F. Paolucci[2]

(1) Scuola Superiore Sant'Anna, Pisa, Italy, (2) CNIT, Pisa, Italy

alessandro.pacini@santannapisa.it

Abstract: Soft failure localization is performed at SDN transceiver agents with peer-to-peer optical telemetry and lightweight ML-based algorithm. Results on a real disaggregated testbed dataset show the effectiveness in terms of accuracy and computational complexity.

1. Introduction

Next generation networking segments will be designed with special care on full, open and online monitoring data extraction and analysis. This trend is evident in the optical networking ecosystem. In fact, optical networks data, control and management systems (e.g., Software defined Networking - SDN) are evolving to support autonomic optical networking disclosed by network awareness in a closed loop fashion. Detailed optical performance monitoring and efficient platform and systems providing awareness are becoming a crucial aspect opening the way to the era of Zero Touch Networking [1]. Optical telemetry is hot topic in the context of disaggregated optical networks [2], thanks to the availability of open data models allowing the exchange of status information between optical node components belonging to different vendors [3] and centralized controllers and monitor handlers. Such platforms facilitate the introduction of Machine Learning (ML) engines to process online the optical data lake. Typically, agents are configured to send monitoring data to centralized controllers.

Peer-to-peer telemetry (P2PT) has been proposed recently as an alternative monitoring technique to complement telemetry at central collectors in Elastic Optical Networks handled by SDN control [4]. Such technique allows optical elements to autonomously detect issues related to their transmitted lightpaths (e.g., soft failures) and take local adaptation decisions without the involvement of the SDN controller, thus speeding up the adaptation process. However, the main issue deals with the complexity of ML-based processing, typically requiring neural networks of several layers and parallel computation processing (e.g., resorting to Graphics Processing Units - GPU) to achieve fast processing latency. Thus, so far, ML have been run in centralized dedicated edge nodes.

In this paper, we propose and evaluate a ML model, that is deployable into a general purpose CPU to achieve fast soft failure detection and localization. The scheme is suitable for being placed inside the SDN agent of the transceiver, that may implement countermeasures (e.g., lightpath adaptation) to react against Quality of Transmission (QoT) degradation, or may notify the SDN controller more rapidly with respect to network-wide collectors. The experimental results report the evidence of using a simple unsupervised ML-based approach to detect and localize soft failures using only data from normal network conditions for training.

2. Peer-to-peer telemetry scenario and operation

In typical disaggregated telemetry architectures, the SDN controller enables a set of device agents to stream physical parameter data measured by the device hardware at regular time periods. In optical networks, coherent xponders provide the OSNR and the pre-FEC BER to monitor the QoT of the received lightpath, while Optical Line Systems (OLS) agents report, for example, power inputs, outputs as a function of time, to detect power level anomalies. Data are stored at centralized handlers running computational intensive ML-based algorithms to detect QoT anomalies, localize affected devices and proactively reconfiguring the network.

The proposed P2PT allows a selected and limited set of QoT streams to be received directly by other device agents, enabling autonomic adaptations. In the case of xPonder, QoT data related to the lightpath currently served by its transmission transceiver is evaluated, that is the data monitored by the disaggregated devices crossed by that lightpath. In addition, a limited set of other data may be used to refine the analysis, e.g. related to spectrally adjacent lightpaths, that should not be affected by a possible signal adaptation in case of soft failure. Fig. 1a shows the P2PT steps for xPonder xP1 collector. First, the controller triggers P2PT stream requests to selected agents indicating xP1 as data collector. P2PT data streams are sent to xP1 (step 1). Samples are processed by a ML module, located at the xP1 agent or at the co-located ROADM controller (e.g., R1). If the module detects QoT anomalies, it identifies the affected devices along the lightpath devices chain and configures the local transmission module for the most appropriate signal adaptation, provided that the frequency slots assigned by the controller must not be exceeded (step 2). Typical adaptations at the cards are central frequency tuning, FEC change or update, launch power variations. Finally, the agent notifies the controller providing information about its detection and the enforced adaptation.

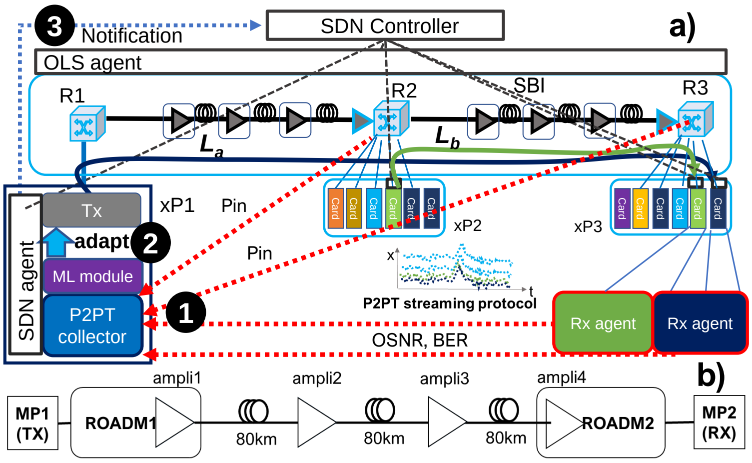


Fig. 1: Peer-to-peer telemetry at the xPonder agent: operation (a); experimental setup of the dataset (b).

Fig. 1b shows the experimental setup considered in the paper for the creation of P2PT dataset used to evaluate ML algorithms at the source agent. In particular, the experimental setup includes 2 ROADMs with one xPonder each. The ROADMs are connected by a OLS composed by a three-spans link. All the EDFA amplifiers are configured in constant gain mode and gain value in order to enter the fiber with 0dBm of optical power. A telemetry monitoring system has been considered in order to collect real time data from the optical devices. The data is collected with a 5s period and includes: (i) the input and output power of the booster amplifier of ROADM1 (ampli1), (ii) the input and output power of the EDFA amplifiers included in the OLS system (ampli2, ampli3 and ampli4) along the three-spans link, (iii) the input and output power at the receiver amplifier of ROADM2 (ampli4) and (iv) the BER and OSNR at the receiver (MP2).

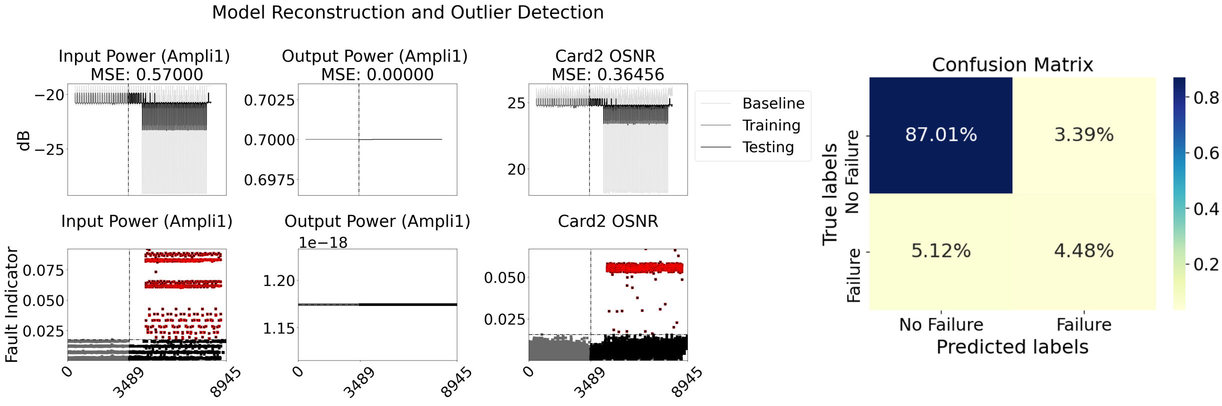
3. Lightweight ML-based solution at the transceiver

A lightweight unsupervised ML-based approach is here described. Alongside its low computational requirements for deploy in real monitoring systems, it has the advantage of using only data from normal operation conditions for training without the need for collecting failure data, which is easier to obtain from production networks.

Principal components analysis (PCA) is a classical multivariate statistical procedure that aims to estimate a linear static relationship between the data in its input space and a small unknown number of latent variables that retain most of the variance in the data [5]. This subspace mapping is reached by reducing the dimensionality of the original input data through a linear projection. Although PCA has been used for several different purposes, varying from feature extraction to manifold learning, here we apply the technique in a novelty detection fashion. Assuming the training data matrix $\mathbf{X} \in \mathbb{R}^{n \times m}$ is composed of m telemetry parameters collected n times from several different network devices under normal working conditions, $\mathbf{X}$ can be decomposed into $\mathbf{X} = \mathbf{T}\mathbf{U}^T$, where $\mathbf{T}$ is called the scores matrix and $\mathbf{U}$ is a set of m orthogonal vectors, also called the loadings matrix (analogous to the eigenvectors). The orthogonal vectors can be obtained by decomposing the covariance matrix of $\mathbf{X}$. The first d components from $\mathbf{U}$ are associated with the higher eigenvalues and are the principal components of the data matrix, and correspond to the dimensions with the largest variability.

With the d principal components estimated from the training data $\mathbf{X}$ collected under normal working conditions of the network, to detect failures from unseen data, it needs to perform this mapping and demapping operation using the main orthogonal vectors $\mathbf{U}_d$. Assuming that a test matrix, $\mathbf{Z}$, containing data from both normal and failure conditions, $\mathbf{Z}$ is mapped to the principal components space $\mathbb{R}^d$ and reversed back to the feature space, $\mathbb{R}^m$, while the residual error, $\mathbf{E}$, is computed as the difference between the original and the reconstructed test matrix using, $\mathbf{E} = \mathbf{Z} - (\mathbf{Z}\mathbf{U}_d)\,\mathbf{U}_d^T$. Finally, to establish a quantitative measure of failure and to allow its detection, a failure indicator (FI) is computed and classified through linear thresholds. In this case, the FI is simply the Euclidean norm over the matrix of residual errors, $\mathbf{E}$. Failure classification and localization is performed through linear thresholds estimated for each telemetry parameter based on 99% of confidence over the training data.

The basic intuition behind this PCA-based approach is that as the orthogonal vectors have been learned to linear map only telemetry data from normal working conditions, when the mapping/demapping occurs with data from failure conditions the residual error will grows proportionally to the level of discrepancy between the telemetry from the normal conditions and the actual failure, allowing its direct detection. Note that as the number of principal components d to be retained is usually very small, the computational loads to perform this mapping/demapping over the test data are minimal as the number of parameters involved in this inner product are very small, allowing fast and online evaluation of soft failures.



(a) (b)

Fig. 2: (a) Model reconstruction (top row) and failure detection (bottom row). In (b) the confusion matrix is shown.

4. Evaluation and results

Fig. 2 introduces the performance of the proposed approach in terms of data reconstruction and failure detection (Fig. 2a) along with the confusion matrix (Fig. 2b). The dataset consists of two phases. In the training phase (portion of data before the vertical line in Fig. 2a) the key parameters are observed in normal conditions, with no failures. The second phase (portion of data after the vertical line in Fig. 2a) introduces soft failures, obtained by adding 10dB attenuation at the output WSS of ROADM1, impacting the input power of ampli1. All the EDFA present a mute power of 0.4dBm, thus the input power variation is observed only at ampli1. By considering the training part in the top row of Fig. 2a, it can be verified that the model adequately reproduces the normal pattern of the system, which is corroborated by the small values of the mean squared error for the training data. However, for the test data, the model does not reach the same level of performance in data reconstruction for the part of data related to the soft failures, which is the expected behavior as we aim to track failures through the residual errors. That is also seen in the bottom row of Fig. 2a, showing the corresponding failure indicators for each sample during the outlier detection. To check the correspondence between a detected soft failure and the actual condition of the system the confusion matrix is used. After verifying the values at the main diagonal the model performs accurate predictions of failure and no failure conditions, with a solid 91.49% of accuracy in the predictions.

Regarding the CPU time and memory usage, we compared the PCA-based approach to a 1D convolutional neural network approach with 132 filters distributed along 9 layers. The neural network was implemented similarly to PCA, using the same input/output data. The ML module functionality has been experimentally validated considering a Linux PC equipped with CPU Intel® Core™ i7-8750H, and 32 GB of RAM. After 1,000,000 trials of both of the compared approaches over the test data, the average CPU time was 52,343.8 nanoseconds for PCA whereas the neural-based approach was 3,367,020 nanoseconds. Regarding the memory usage, in average, PCA used 9,772.9 KB of memory while the neural net used 58,153.4 KB. In general, the PCA-based approach is 64x faster and consumes approximately 6x less memory than using a neural network considering only the evaluation of the test data (the training time was not considered in this study once it should be performed offline).

5. Conclusions

We proposed an unsupervised ML approach that aims to reach 91% accuracy in soft failure detection and localization running 64x faster with 6x less memory than the corresponding neural network. The solution is evaluated with P2PT at the SDN xPonder agent, drastically improving system responsiveness with limited CPU resources.

References

1. ETSI, "Zero-touch network and service management (ZSM); reference architecture," ETSI GS ZSM 002 V1.1.1, 2019.
2. A. Sgambelluri, A. Pacini, F. Paolucci, P. Castoldi, and L. Valcarenghi, "Reliable and scalable kafka-based framework for optical network telemetry," *IEEE/OSA J. Opt. Commun. Netw.*, vol. 13, no. 10, pp. E42–E52, 2021.
3. L. Velasco, A. Sgambelluri, R. Casellas, L. Gifre, J.-L. Izquierdo-Zaragoza, F. Fresi, F. Paolucci, R. Martínez, and E. Riccardi, "Building autonomic optical whitebox-based networks," *JLT*, vol. 36, no. 15, pp. 3097–3104, 2018.
4. F. Paolucci, A. Sgambelluri, P. Castoldi, and F. Cugini, "Telemetry solutions in disaggregated optical networks: an experimental view," in *OFC 2021*, 2021, pp. 1–3.
5. I. Jolliffe, *Principal Component Analysis*. 2nd Edition. Springer, 2002.