Experimental Demonstration of a Fully Disaggregated and Automated White Box Comprised of Different Types of Transponders and Monitors

Nicola Sambo, Kostas Christodoulopoulos, Nikos Argyris, Pietro Giardina, Camille Delezoide, Andrea Sgambelluri, Aristotelis Kretsis, Giannis Kanakis, Francesco Fresi, Giacomo Bernini, Hercules Avramopoulos, Emmanuel Varvarigos, and Piero Castoldi

***Abstract*—White boxes, originally introduced in data centers, are expected to soon also penetrate the market of wide area networks, since operators see them as a way to drive down capital expenditure. Indeed, white boxes can be composed of modules from different vendors, removing the traditional vendor lock-in and creating more competition in the market, thus reducing hardware costs. In this paper, we describe the experimental demonstration of a *fully disaggregated* white box, composed of two different types of transponders, monitors (including filtering effect parameter monitors), add-drop multiplexers, and switches. Toward this end, we developed an appropriate control and management plane for the white box based on NETCONF and YANG. Using that, we demonstrated the automatic reconfiguration of the white box so as to maintain the service in the presence of multiple unexpected degradations: signal-to-noise ratio decrease and filtering distortion. We also demonstrated and presented here a vendor-neutral procedure to compute the thresholds used for raising alarms in response to physical layer degradations.**

***Index Terms*—Disaggregation, inter-operability, NETCONF, white box, YANG.**

I. Introduction

WHITE boxes are attracting increasing interest from service providers and network operators [1], [2] and are expected to penetrate the wide area networks market in the following years, as a way to reduce capital expenditure [2]. White boxes provide disaggregation of software from hardware (e.g., for control) and can be assembled with inter-operating modules from different vendors. In order to support the control and management of white boxes, effort has to be done to achieve standardized data models shared by vendors and operators. In addition, the migration towards multi-vendor networks and towards inter-operability imposes the redefinition of several procedures – which are now vendor locked – for operating the network, for management, and maintenance in a vendor-neutral way. YANG [3] has been identified by operators and service providers as the data modeling language to enable the interfacing with the control and the management system. YANG is supported by the emerging NETCONF protocol standardized by Internet Engineering Task Force (IETF) [4].

A great deal of research is currently being carried out on white boxes and, more generally, on disaggregation. Several consortiums and projects, such as OpenConfig [5], OpenROADM [6], Telecom Infra Project (TIP) [7], and IETF [8], [9], that include operators, service providers, and vendors, are working to define vendor-neutral and disaggregated networks (describing the related media channels, nodes, amplifiers, interfaces, and so on). Several activities within these consortiums are focused on the definition of YANG models. Such topic is ongoing also within research in general [10]–[13]. In particular, such works propose data models for transponders [10]–[12], for programming recovery actions in Elastic Optical Networks (EON) [12], and for space division multiplexed optical networks [13]. Some experimental demonstrations of optical white boxes have also been performed [14]–[18]. The works in [15], [16] demonstrated the control and management of a white box, showing capabilities to automatically react against unexpected physical layer degradations. In [17], the disaggregation of node functionalities was studied against disasters. The authors in [18] focused on telemetry applications for the streaming of monitoring information, also having reliability as their main objective. Moreover, quality of transmission (QoT) was studied within the TIP project [19] for vendor-neutral networks, aiming to deliver an open source and vendor-agnostic tool for QoT estimation.

This paper contributes to the research field of inter-operability and vendor-agnostic optical networks by demonstrating: i) the inter-operability of several node components provided by dif-

Manuscript received July 18, 2018; revised October 31, 2018; accepted November 12, 2018. Date of publication November 16, 2018; date of current version February 21, 2019. This work was supported by the EC through the Horizon 2020 ORCHESTRA Project (g.a. 645360). This paper was presented in part at the Optical Fiber Communications Conference and Exposition, San Diego, CA, USA, March 2018. *(Corresponding author: Nicola Sambo.)*

N. Sambo, A. Sgambelluri, and P. Castoldi are with Scuola Superiore Sant'Anna, Pisa 56127, Italy (e-mail: n.sambo@sssup.it; a.sgambelluri@sssup.it; castoldi@sssup.it).

K. Christodoulopoulos, A. Kretsis, and E. Varvarigos are with CTI, Patras 26504, Greece (e-mail: kchristo@mail.ntua.gr; akretsis@ceid.upatras.gr; vmanos@central.ntua.gr).

N. Argyris, G. Kanakis, and H. Avramopoulos are with NTUA, Athens 15780, Greece (e-mail: nikosa@mellanox.com; giankan@mail.ntua.gr; hav@mail.ntua.gr).

P. Giardina and G. Bernini are with Nextworks, Pisa 56122, Italy (e-mail: p.giardina@nextworks.it; g.bernini@nextworks.it).

C. Delezoide is with Nokia Bell Labs, Paris 91620, France (e-mail: camille.delezoide@nokia-bell-labs.com).

F. Fresi is with CNIT, Pisa 00133, Italy (e-mail: francesco.fresi@cnit.it).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org

Digital Object Identifier 10.1109/JLT.2018.2881537

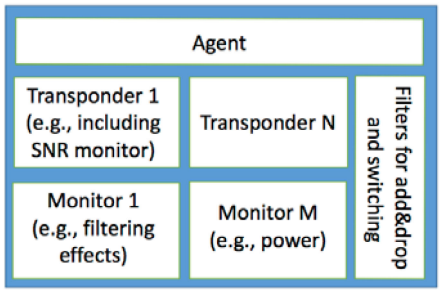


Fig. 1. White box architecture.

(a) Parameter	Value	Unit
Bit rate	112	Gb/s
Modulation format	PM-QPSK	-
FEC	7	%
Baud rate	28	Gbaud
Launch power	0	dBm
Central frequency	193.1	THz

(b) Parameter	Value	Unit
pre-FEC BER	1.7 x 10-5	-
B		GHz
OSNR		dB
CD		ps/nm
PMD		ps
SNR		dB

Fig. 2. Part of database recording: (a) configuration parameters; (b) monitored parameters.

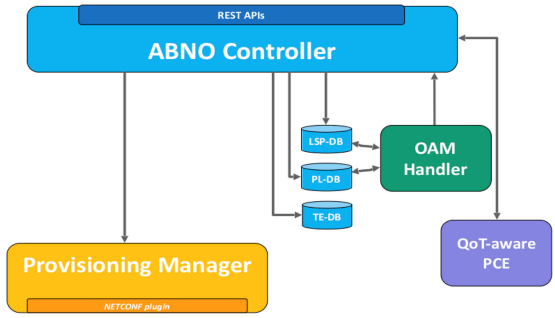


Fig. 3. Control and management plane architecture.

ferent partners and ii) the control and management of white box including vendor-agnostic procedures guaranteeing service maintenance in the presence of multiple degradations. In particular, we experimentally demonstrate a fully disaggregated white box composed of the following cooperating modules, each provided by a different partner: two types of transponders, monitors (including a filtering effect monitor), switches, add-drop multiplexers, and an agent for interfacing with the control and management plane. We demonstrate the configuration of the white box through NETCONF and YANG, as well as its dynamic reconfiguration (baud rate adaptation, filter reconfiguration, and signal frequency shifting) upon physical layer degradations (e.g., filtering effects and signal to noise ratio degradation) identified through its monitors. This work is an extension of [15]. With respect to [15], we provide in the current paper more implementation details. Additionally, a contribution of this paper is the description and the experimental demonstration of a vendor-agnostic procedure to determine the QoT threshold that should be used to identify a physical layer degradation. Such threshold is computed to enable an operator to discern between normal QoT fluctuations (e.g., on the bit error rate – BER – values) and soft failures (i.e., unexpected physical layer degradations). It can also be used for anticipating the outage of a connection due to unacceptable QoT, thus guaranteeing the proper operation, administration, and maintenance (OAM) of the connection.

II. White Box's Architecture and Specifications

The white box architecture (disaggregated from the control and management software) is shown in Fig. 1. It consists of modules possibly provided by different partners: N transponders, also including end-to-end performance (e.g., BER) monitors as well as other monitors (such as a filtering effect monitor [20] and a power monitor), and filters for add and drop multiplexing and switching based on flexible grid. Finally, the white box includes an agent interfacing with the software-based control and management plane.

More specifically, in our experiments, the following modules were considered and realized:

- *Transponder 1* supports polarization multiplexing 16 and 8 quadrature amplitude modulation (PM-16QAM and PM-8QAM), and polarization multiplexing quadrature phase shift keying (PM-QPSK) modulation formats in a single carrier scheme, with 200, 150, and 100 Gb/s net rates, respectively. The transponder also supports 12% and 28% code rate, 28 and 32 Gbaud, end-to-end monitor of signal to noise ratio (SNR), optical SNR (OSNR), chromatic dispersion (CD), polarization mode dispersion (PMD), and bit error rate (BER).
- *Transponder 2* supports PM-16QAM and PM-QPSK modulation formats, with 200 and 100 Gb/s net rates, respectively. The transponder supports 7% code rate, 28 Gbaud, and end-to-end monitor of CD, PMD, and BER.
- *Monitor 1* monitors filter effects through the measurement of the 10 dB-bandwidth (B) of the signal before digital signal processing (DSP).
- *Monitor 2* consists of a commercial power monitor.
- *Add and drop* and *switching* are based on commercial Bandwidth Variable Wavelength Selective Switches (BV-WSSs) supporting the ITU-T flex-grid. An amplifier stage is also included (booster in the add and pre-amp in the drop).
- Finally, the white box contains an *agent* interfacing with the NETCONF protocol. Given the nature of NETCONF which operates on a client-server base, the agent includes two databases (shown in Fig. 2) storing the values of *configuration* parameters (e.g., transponder bit rate and modulation format) and *state* (used for monitoring purposes) parameters (including values such as B and BER). Configuration data is decided by the control plane, it is carried by NETCONF and it is written in the database to configure the white box accordingly. Monitoring data is sent via NETCONF to the management system.

III. Control and Management of White Box

The control and management plane, shown in Fig. 3, is based on the IETF ABNO [21] architecture, and includes a stateful

ABNO controller, databases (e.g., traffic engineering database – TED), the Provisioning Manager, and the Operation, Administration, and Maintenance (OAM) Handler. NETCONF is used for configuration and for carrying monitoring information.

The *ABNO controller* is extended to take decisions using the monitoring information, and closing the so called observe-decide-act control loop envisioned in the ORCHESTRA project [22]. Within this framework, actions (e.g., code adaptation, rerouting, spectrum shifting) are taken based on (re)optimization decisions that are themselves triggered by the observation of the monitored parameters (e.g., SNR, filtering effects). Opendaylight tool is used as Provisioning Manager after being extended for NETCONF.

The ABNO-controller exploits a QoT-aware Path Computation Element (PCE) based on the DEPLOY optimization tool [22]. DEPLOY performs path computation (path, spectrum and transmission parameters selection) and re-optimization (e.g., code adaptation) calculations taking into account the QoT/physical layer. In addition, it performs some application specific calculations, such as QoT estimation, failure localization, and soft-failure threshold calculation (discussed in Section IV). To carry out these calculations it leverages three databases implemented in ABNO: the *Traffic Engineering Database* (*TE-DB*); the *Label Switch Path DataBase* (*LSP-DB*) including information on connections' state (e.g., used path, spectrum, rate, and modulation format), which makes the control plane *stateful*; the *Physical Layer DataBase* (*PL-DB*) storing information related to the physical layer. In particular, PL-DB includes nominal values of monitored parameters (e.g., amplifier noise figure from datasheets), as well as monitored parameter values (e.g., end-to-end BER). The latter are collected through NETCONF, as it will be described below.

Given a request for a connection between two nodes at a specific rate, the PCE decides the transponder configuration parameters (gross rate, baud rate, code rate, and modulation format), and the routing and spectrum assignment. Then, the ABNO controller triggers the *Provisioning Manager*, which is connected to the white box's agent, to configure the data plane. The Provisioning Manager acts as an SDN-controller exploiting NETCONF protocol. Data plane configuration includes the setting of the selected transponder at the proper configuration parameters' values, and the BV-WSSs for the add-drop and switching functions. In case of problems such as faults or physical layer degradations (e.g., due to increased attenuation or unexpected filtering effects), the ABNO controller reconfigures the white box to adapt the transmission parameters or re-route the entire connection.

NETCONF protocol supports the YANG model detailed in [10] for controlling and managing the transponders. Moreover, a YANG model including the configuration of filter central frequency, bandwidth, and in/out ports is adopted to configure the BV-WSSs' add-drop and switching functions. The NETCONF <edit-config> message is sent by the Provisioning Manager to the white box agent in order to configure or reconfigure transponders and BV-WSSs. Configuration values are written in the agent's database and are used to configure the hardware.

Moving now from the control plane to the management plane, the OAM Handler of ABNO is responsible for ensuring the proper operation, administration and maintenance of the services. The OAM Handler is connected to the white box's agent. To be more specific, the developed OAM Handler and the white box agent are fully programmable to provide i) periodic and on-demand polling of monitored parameters, and ii) threshold-based alarm handling. These are implemented with the NETCONF protocol still pointing to the YANG model in [10]. The monitored values are stored in the PL-DB of ABNO, and can be used for various optimization operations by the PCE, such as QoT estimation, threshold calculations, etc. The alarms initiate certain ABNO operations, such as the identification and localization of failures (e.g., fiber cuts, filtering effect, or signal attenuation), and also trigger the ABNO-controller to perform white box reconfiguration or service rerouting for recovery.

The OAM Handler can be programmed to periodically request a set of monitored parameters from the agents residing in the white boxes. The programmable parameters include the period, the white box(es) agent(s) to be polled, and the set of monitored parameters to be obtained from the ones available. To obtain monitoring information, the NETCONF <get> message is sent by the OAM Handler to poll the related agent. The agent receiving such a message retrieves the requested monitored values from its local database and sends them through the NETCONF <rpc-reply> message to the OAM Handler.

The same process can be initiated on demand by ABNO or the PCE, as opposed to the periodic monitoring operation discussed above. In the case that a specific parameter is required (for example, to localize a failure or to improve QoT estimation accuracy), the OAM handler can directly require certain monitored parameters from certain agents.

The white box agents can also be programmed to create alarms related to specific events. An agent can be programmed by the OAM Handler to raise an alarm when some specified monitored parameter exceeds a pre-specified threshold. Note that this feature differs from periodic or on demand monitoring. Even though the monitored values that are collected (e.g., periodically) in the OAM Handler/ABNO can be checked against thresholds, such an approach introduces delays. Typically, the monitoring polling period would be set much higher than the execution time of the DSP monitoring algorithm, while complete power loss e.g., due to a fiber cut, can be immediately sensed and needs to be immediately conveyed. Moreover, relying on centralized failure identification creates a bottleneck. In our approach, the developed white box agents can be programmed to raise an alarm when an event/problem is identified locally. Appropriate alarms can be set to report sudden deteriorations of the QoT, typically referred to as *soft-failures*, up to the complete loss of signal, corresponding to a *hard failure*. The related alarm is forwarded to the OAM Handler that is then responsible, along with ABNO/DEPLOY, for handling the event.

Regarding the implementation of the alarm feature, the NETCONF <create-subscription> message is used by the OAM Handler to inform the agent of the monitored parameter and its alarm threshold. If the threshold is exceeded, the agent creates a NETCONF <notification> message to actually implement the alarm, and sends it to the OAM Handler.

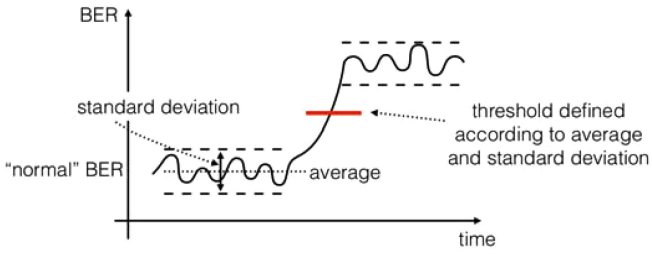


Fig. 4. Detection of a soft failure as an unnatural variation of the QoT monitoring.

IV. Vendor-Neutral Soft-Failure Threshold Identification

In this section, we describe the procedure to identify a sudden malfunction/soft-failure on a lightpath, independently of the system in which the lightpath is operating. To do so, we make full use of the programmable features of the proposed management plane: periodical monitoring, threshold/alarm setup, and alarm creation and handling.

When monitoring the QoT of a lightpath (e.g., through BER or OSNR), a number of fluctuations may be observed due to short-time physical effects, such as polarization effects and/or increased network load inducing higher nonlinear noise level, which could cause control overhead and ping pong effects if unnecessary actions are triggered due to each of them. Thus, an important issue is to discern between normal QoT fluctuations and a soft failure causing a rapid decrease of QoT, as shown in Fig. 4.

The technique we propose for the threshold calculation assumes that the normal QoT fluctuations are Gaussian. Thus, we monitor the QoT parameter and its average in order to understand the normal QoT fluctuations. Let μ and σ be the average and the standard deviation of the QoT monitored parameter of interest, respectively. We define any QoT deviation as abnormal if it exceeds the threshold $TH = \mu + k{\cdot}\sigma$, where k is a real number selected according to the targeted confidence level and the FEC limit to avoid an outage. In our experiments, we used $k = 5$ to achieve 0.99999 confidence. Since the average value μ of the QoT metric changes due to long and medium-term effects (network equipment ageing, weather), the threshold calculation process (in particular, the estimation of μ) should be refreshed/changed periodically (for example, every month) so as to capture the current state of the network and follow long and medium-term effects. Few measurements, e.g., every 10 minutes over a few days, could capture the short-term fluctuations when calculating σ.

Regarding the implementation, at a given network state, the OAM Handler polls BER values to compute an estimation of μ and σ, via the NETCONF <get> message. Then, the TH is computed accordingly and NETCONF <create-subscription> is used to inform the agent about the threshold generating alarms. If the threshold is exceeded, the NETCONF <notification> message implements the alarm. The experimental verification of this concept will be described in the next section.

Some considerations are discussed in the following about the "root cause analysis" of a failure [23], i.e., the process for identifying the causes of problems and failure events. Indeed, besides the prompt reaction to guarantee the proper maintenance of a service, the identification of the causes of a degradation, such as the malfunctioning device or devices, would also be beneficial. Soft failure detection can be very useful because performance degradation can be the symptom that a specific component/device is going to fail within a certain time. Thus, the detection of an anomalous behavior of physical layer parameters such as BER together with a root cause analysis can enable a network operator to intervene (e.g., by replacing the responsible components) before a more substantive failure. A possibility is the examination of various types of real-life network fault use cases, which can be used for training of a fault detection mechanism based on machine learning [24]. However, a problem with similar approaches is that they can only track specific failure patterns that they were trained for. In a real network, there could be many failures or even the trained ones could evolve differently. These activities should be investigated in future research studies.

V. Experimental Demonstration

We experimentally demonstrated the white box configuration, monitoring, and reconfiguration in the testbed of Fig. 5. In particular, the control and management plane guarantee the proper (i.e., below the BER threshold) operation of services by reconfiguring the white box's transmission parameters and filters in case of physical layer degradations.

In this section, first, we will show a reliability experiment performed in Pisa lab in the presence of two simultaneous physical layer degradations: (1) an OSNR decrease (e.g., due to fiber or amplifier aging) achieved by varying the attenuation of the Variable Optical Attenuator (VOA) and (2) filtering effects (e.g., laser and filter misalignment in a cascade of filters, due to laser and/or filter aging) achieved by narrowing the filter in the intermediate node (second BV-WSS). Then, we will show the computation of the BER threshold to identify a soft failure according to the procedure presented in Sec. IV, as performed in Paris lab.

Regarding the reliability experiment, three connection requests were considered, one at 150 Gb/s and two at 200 Gb/s net rate. Transponder 1 was configured at 150 Gb/s, PM-8QAM, 28 Gbaud, and 12% code rate, with central frequency 193.6 THz (i.e., $n = 80$ in the ITU-T flex-grid). Transponder 2, which supports two carriers, was configured with central frequencies of 193.5625 THz ($n = 74$) and 193.6375 THz ($n = 86$), respectively, each at 200 Gb/s, PM-16QAM, 28 Gbaud, and 7% code rate. All the signals were switched with 37.5 GHz bandwidth ($m = 3$ according to ITU-T flex-grid recommendations). Then, OSNR degradation was introduced through VOA, as well as filter distortions affecting just the portion of the bandwidth related to the signal generated by Transponder 1.

Fig. 6 shows the evolution of Transponder 1's (electrical) SNR (Fig. 6a), BER (Fig. 6b), and signal bandwidth B (Fig. 6c), as monitored by the white box. To be more specific, BER and SNR are monitored by Transponder 1 itself, while B is monitored through the external monitor of filtering effects. As seen in Fig. 6, before the failures, SNR is above 12 dB, BER

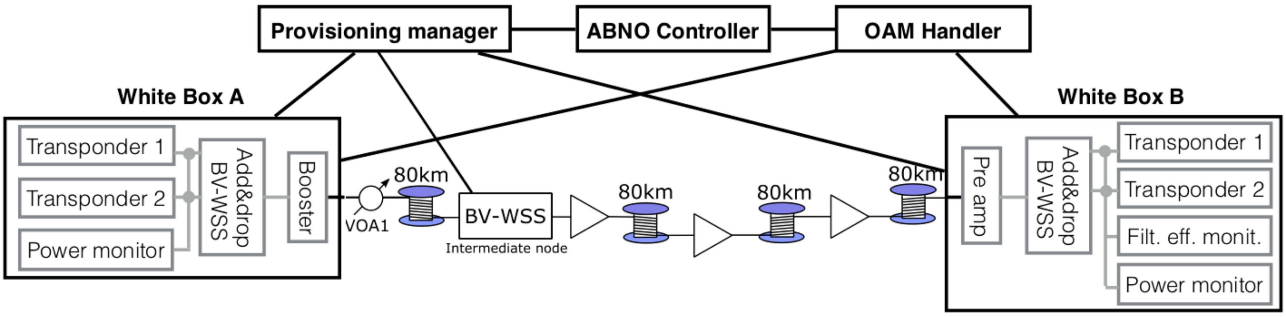


Fig. 5. Experimental setup.

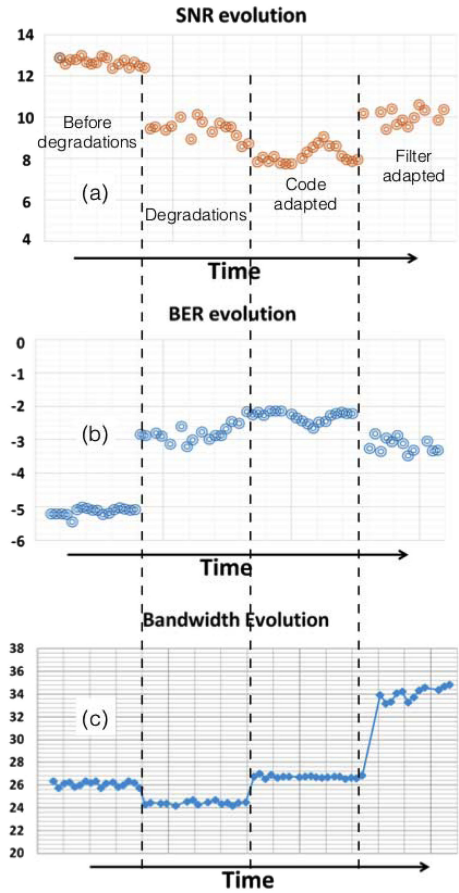


Fig. 6. SNR (a), BER (b), and B (c) evolution during the reliability experiment.

```
Received Notification:
<pre-fec-ber-change xmlns="http://sssup.it/transponder">
    <subcarrier-module-id>1</subcarrier-module-id>
    <transponder-id>801</transponder-id>
    <pre-fec-ber>0.0035085</pre-fec-ber>
  </pre-fec-ber-change>
```

Fig. 7. NETCONF <notification> message.

is around 10^{-5}, and B = 26 GHz. When the two degradations occur, BER jumps to around 10^{-3}, SNR falls by about 3.5 dB, and B = 24 GHz. This BER increase triggers an alarm shown in Fig. 7 that is sent to the OAM Handler, reporting the unique identifier of the transponder (801) and the value of the BER (0.0035085). This alarm is sent because of the violation of the BER threshold, which was set to 2×10^{-3}. In response to this alarm, a first reconfiguration is automatically performed.

Fig. 8 shows the log of the DEPLOY PCE engine reporting, in "Soft Failures Request 1", the affected lightpath connection (with id 801) and, in "Soft Failures Response 1", the "transmission_parameters_adaptation" decision. In particular, we see that baudrate is increased to 32 Gbaud to support a more redundant code (28%) and a higher BER threshold of 0.0135. Fig. 9 shows the NETCONF <edit-config> message sent for transponder reconfiguration. The increase of baudrate, however, implies an increase of the signal bandwidth, thus worsening even more the detrimental filtering effects. Indeed, BER increases further and is measured to reach around 0.01 as shown in Fig. 6b. As a result of this, the filtering effect monitor raises a new alarm, in response to which, the DEPLOY engine computes an appropriate reconfiguration of the filters. The related log is reported in Fig. 10. The filter enlargement can only be enabled by shifting the third connection (id 803) from $n = 86$ to $n = 88$. The "push-pull" technique is exploited to perform this signal shift [26] in a hitless way. Fig. 11 shows the NETCONF message for filter reconfiguration, with parameter $m = 4$ (50 GHz) instead of $m = 3$ (37.5 GHz). This operation improves the QoT and the monitored BER returns to acceptable values (shown in Fig. 6b). Reconfiguration time is in the order of few seconds, and is dominated by the time required for laser synthonization and BV-WSS reconfiguration. After reconfiguration, the bandwidth parameter B in Fig. 6c becomes approximately 32 GHz, approaching the corresponding baudrate value.

Further experiments were performed in Nokia Bell Labs in Paris where the white box was moved and reassembled in an experimental setup similar to the one described in Fig. 5. First, reliability experiments similar to the ones in Pisa were performed to verify their repeatability and the capability of the white box to adapt to a different system (testbed). The repeated experiments (not reported here for space reason) were successful. In addition, the computation of the threshold for soft failures was performed and reported in the following.

In this experiment we programmed the OAM Handler to periodically communicate with the agent of the egress node to obtain monitored BER values related to a lightpath established

Soft Failures Request 1

```
{ "Affected_Lightpaths":[801], "Reconfigurable_Lightpaths":[801, 802, 803], "Monitoring":[{ "lightpath_id":
801, "tsp_monitor": [ {"pre_fec":0.0035085}], "additional_monitors": [{"monitor_id":1, "bandwidth":23.38}] }]}
```

Soft Failures Response 1

```
{ "Recovery_Status": [1], "Recovery_Actions": [ { "reconfigured_lightpath_id": 801, "action_type":
"transmission_parameters_adaptation", "baud_rate":32.0, "fec_threshold":0.0135}]}
```

Fig. 8. Log of the DEPLOY for the first recovery request.

```
<baud-rate>32.0</baud-rate>
<modulation xmlns:mf="http://sssup.it/modulation-formats">mf:pm-qpsk</modulation>
<fec-in-use>
  <name xmlns:fec="http://sssup.it/fec-types">fec:ldpc</name>
  <rate>
    <message-length>3</message-length>
    <block-length>15</block-length>
  </rate>
</fec-in-use>
```

Fig. 9. Reconfiguration of baudrate and FEC through the NETCONF <edit-config> message.

Soft Failures Request 2

```
{ "Affected_Lightpaths":[801], "Reconfigurable_Lightpaths ":[801, 802, 803], "Monitoring":[{ "lightpath_id":
801, "tsp_monitor": [ {"pre_fec":0.0136815}], "additional_monitors": [{"monitor_id":1, "bandwidth":25.98}] }]}
```

Soft Failures Response 2

```
{ "Recovery_Status": [1], "Recovery_Actions": [{ "reconfigured_lightpath_id": 801, "action_type":
"filter_adaptation", "n": 81, "m": 4 }, { "reconfigured_lightpath_id": 803, "action_type": "push-pull", "n":
88, "m": 3 } ]}
```

Fig. 10. Log of the DEPLOY for the second recovery request.

```
<filter xmlns="sssup:filter" xmlns:nc="urn:ietf:params:xml:ns:netconf:base:1.0">
  <connections nc:operation="replace">
    <connection-id>1</connection-id>
    <n>81</n>
    <m>4</m>
    <output-port-id>1</output-port-id>
    <input-port-id>0</input-port-id>
  </connections>
</filter>
```

Fig. 11. NETCONF <edit-config> message reconfiguring the filter.

```
<?xml version="1.0" encoding="UTF-8"?>
<rpc-reply xmlns="urn:ietf:params:xml:ns:netconf:base:1.0" message-id="1">
  <data>
    <transponder xmlns="http://sssup.it/transponder">
      <subcarrier-module>
        <subcarrier-id>1</subcarrier-id>
        <state>
          <receiver>
            <pre-fec-ber>0.0001128324</pre-fec-ber>
          </receiver>
        </state>
      </subcarrier-module>
    </transponder>
  </data>
</rpc-reply>
```

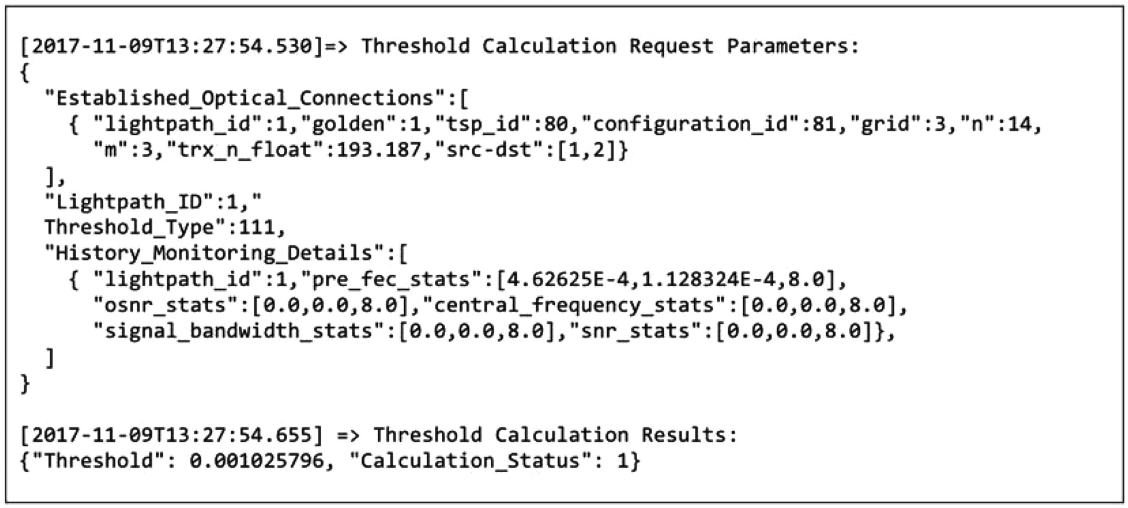
Fig. 12. NETCONF <rpc-reply> message reporting a value of monitored BER.

at 100 Gb/s with PM-QPSK and 28% FEC. Fig. 12 shows a NETCONF <rpc-reply> message. The monitored values were stored in ABNO PL-DB.

To emulate the short-term effects at the data plane, we introduced random attenuation in the VOA following a Gaussian distribution. After monitoring several values, ABNO called the DEPLOY-PCE to calculate the soft failure threshold TH for the lightpath. The calculation was done according to the definition in Section IV using $k = 5$. The monitored BER average was $\mu = 4.62625 \cdot 10^{-4}$ and the standard deviation was $\sigma = 1.128324 \cdot 10^{-4}$ for the 8 monitored BER values. Fig. 13 shows the log of the DEPLOY-based PCE for the threshold computation request. The computation result for the considered lightpath was $\mathrm{TH} = 1.025 \cdot 10^{-3}$. Fig. 14 shows the NETCONF <create-subscription> message sent by the OAM Handler to inform the agent about the newly calculated alarm threshold.

VI. Conclusion

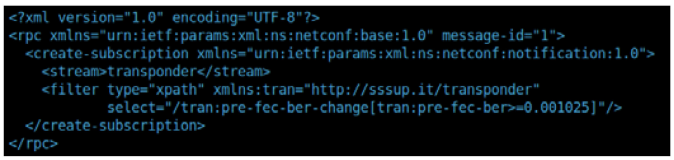
We experimentally demonstrated a fully disaggregated white box, where hardware was separated by the control and management plane. We showed the continued proper operation of services even in the presence of Quality of Transmission problems due to increased attenuation and unexpected filtering effects. Our results support the ability of Software Defined Networking for white boxes to provide automation and business continuity. The white box was composed of transponders, performance monitors (including a filtering effect monitor), and add-drop multiplexers provided by different partners, as well an agent for interfacing with the control and management plane. NETCONF



```
[2017-11-09T13:27:54.530]=> Threshold Calculation Request Parameters:
{
  "Established_Optical_Connections":[
    { "lightpath_id":1,"golden":1,"tsp_id":80,"configuration_id":81,"grid":3,"n":14,
      "m":3,"trx_n_float":193.187,"src-dst":[1,2]}
  ],
  "Lightpath_ID":1,"
  Threshold_Type":111,
  "History_Monitoring_Details":[
    { "lightpath_id":1,"pre_fec_stats":[4.62625E-4,1.128324E-4,8.0],
      "osnr_stats":[0.0,0.0,8.0],"central_frequency_stats":[0.0,0.0,8.0],
      "signal_bandwidth_stats":[0.0,0.0,8.0],"snr_stats":[0.0,0.0,8.0]},
  ]
}

[2017-11-09T13:27:54.655] => Threshold Calculation Results:
{"Threshold": 0.001025796, "Calculation_Status": 1}
```


Fig. 13. DEPLOY log for threshold computation request.



```
<?xml version="1.0" encoding="UTF-8"?>
<rpc xmlns="urn:ietf:params:xml:ns:netconf:base:1.0" message-id="1">
  <create-subscription xmlns="urn:ietf:params:xml:ns:netconf:notification:1.0">
    <stream>transponder</stream>
    <filter type="xpath" xmlns:tran="http://sssup.it/transponder"
            select="/tran:pre-fec-ber-change[tran:pre-fec-ber>=0.001025]"/>
  </create-subscription>
</rpc>
```


Fig. 14. NETCONF <create-subscription> message to subscribe to alarms according to the computed threshold.

protocols and YANG models were used to control and manage the white box. A procedure for computing the threshold used for identifying a soft failure was also presented and experimentally demonstrated.

REFERENCES

[1] 2016. [Online]. Available: https://code.facebook.com/posts/1977308282496021/an-open-approach-for-switching-routing-and-transport/

[2] E. Riccardi, P. Gunning, O. Gonzalez de Dios, M. Quagliotti, V. Lopez, and A. Lord, "An operator's view on introduction of White Boxes in optical networks," *J. Lightw. Technol.*, vol. 36, no. 15, pp. 3062–3072, Aug. 2018.

[3] M. Bjorklund, "YANG—A data modeling language for the network configuration protocol (NETCONF)," IETF RFC 6020, Oct. 2010.

[4] R. Enns, M. Bjorklund, J. Schoenwaelder, and A. Bierman, "Network configuration protocol (NETCONF)," IETF RFC 6241, Jun. 2011.

[5] [Online]. Available: http://www.openconfig.net/

[6] [Online]. Available: http://www.openroadm.org/home.html

[7] [Online]. Available: https://telecominfraproject.com/

[8] J. Vergara *et al.*, "YANG data model for flexi-grid optical networks," IETF draft-vergara-flexigrid-yang-05, Jul. 2017.

[9] R. Kunze *et al.*, "A framework for management and control of DWDM optical interface parameters," IETF draft-ietf-ccamp-dwdm-if-mng-ctrl-fwk-06, Jun. 2017.

[10] M. Dallaglio, N. Sambo, F. Cugini, and P. Castoldi, "Control and management of transponders with NETCONF and YANG," *J. Opt. Commun. Netw.*, vol. 9, no. 3, pp. B43–B52, Mar. 2017.

[11] M. Dallaglio, N. Sambo, F. Cugini, and P. Castoldi, "YANG models for vendor-neutral optical networks, reconfigurable through state machine," *IEEE Commun. Mag.*, vol. 55, no. 8, pp. 170–178, Aug. 2017.

[12] N. Sambo, A. Giorgetti, F. Cugini, and P. Castoldi, "Sliceable transponders: Pre-programmed OAM, control, and management," *J. Lightw. Technol.*, vol. 36, no. 7, pp. 1403–1410, Apr. 1, 2018.

[13] R. Muñoz *et al.*, "SDN-enabled sliceable multi-dimensional (spectral and spatial) transceiver controlled with YANG/NETCONF," in *Proc. Opt. Fiber Commun. Conf.*, 2018, Paper M2A.5.

[14] N. Sambo, I. Tomkos, A. Shaikh, S. Bigo, M. Suzuki, and H. J. Schmidtke, "Guest editorial: Optical networks supporting interoperability and white boxes," *J. Lightw. Technol.*, vol. 36, no. 15, pp. 3058–3061, Aug. 1, 2018.

[15] N. Sambo *et al.*, "Experimental demonstration of fully disaggregated white box including different types of transponders and monitors, controlled by NETCONF and YANG," in *Proc. Opt. Fiber Commun. Conf.*, 2018, Paper M4A.3.

[16] L. Velasco *et al.*, "Building autonomic optical whitebox-based networks," *J. Lightw. Technol.*, vol. 36, no. 15, pp. 3097–3104, Aug. 2018.

[17] M. Shiraiwa *et al.*, "Experimental demonstration of disaggregated emergency optical system for quick disaster recovery," *J. Lightw. Technol.*, vol. 36, no. 15, pp. 3083–3096, Aug. 2018.

[18] F. Paolucci, A. Sgambelluri, F. Cugini, and P. Castoldi, "Network telemetry streaming services in SDN-based disaggregated optical networks," *J. Lightw. Technol.*, vol. 36, no. 15, pp. 3142–3149, Aug. 2018.

[19] M. Filer, M. Cantono, A. Ferrari, G. Grammel, G. Galimberti, and V. Curri, "Multi-vendor experimental validation of an open source QoT estimator for optical networks," *J. Lightw. Technol.*, vol. 36, no. 15, pp. 3073–3082, Aug. 2018.

[20] C. Delezoide, P. Ramantanis, and P. Layec, "On the performance prediction of optical transmission systems in presence of filtering," in *Proc. 19th Int. Conf. Transparent Opt. Netw.*, 2017.

[21] IETF RFC 7491, IETF, Fremont, CA, USA, Mar. 2015.

[22] K. Christodoulopoulos *et al.*, "ORCHESTRA—Optical performance monitoring enabling flexible networking," in *Proc. 17th Int. Conf. Transparent Opt. Netw.* 2015.

[23] G. Ellinas, D. Papadimitriou, J. Rak, D. Staessens, J. P. G. Sterbenz, and K. Walkowiak, "Practical issues for the implementation of survivability and recovery techniques in optical networks," *Opt. Switching Netw.*, vol. 14, no. 2, p. 179–193, 2014.

[24] D. Rafique, T. Szyrkowiec, H. Grießer, A. Autenrieth, and J.-P. Elbers, "Cognitive assurance architecture for optical network fault management," *J. Lightw. Technol.*, vol. 36, no. 7, pp. 1443–1450, Apr. 2018.

[25] M. Dallaglio *et al.*, "YANG model and NETCONF protocol for control and management of elastic optical networks," in *Proc. Opt. Fiber Commun. Conf. Exhib.*, 2016, Paper W3F.5.

[26] F. Cugini *et al.*, "Push-pull defragmentation without traffic disruption in flexible grid optical networks," *J. Lightw. Technol.*, vol. 31, no. 1, pp. 125–133, Jan. 2013.

Authors' biographies not available at the time of publication.