

# Management of Sliceable Transponder with NETCONF and YANG

Matteo Dallaglio\*, Nicola Sambo\*, Filippo Cugini<sup>†</sup> and Piero Castoldi\*

\*Scuola Superiore Sant'Anna, Pisa, Italy

Email: m.dallaglio@sssup.it

<sup>†</sup>CNIT, Pisa, Italy

**Abstract**—NETCONF is emerging as Software Defined Networking (SDN) protocol for the control and management of optical networks. It enables data plane device configuration and access to monitoring information. NETCONF may exploit YANG data model to describe network elements to be controlled/managed. YANG and NETCONF are of interest for operators since they provide a standard way to control and manage network elements, independently from the vendor.

This paper presents and demonstrates a management plane based on NETCONF protocol. In particular, a YANG model describing optical transponders supporting slice-ability, variable rate, and monitoring functionalities is proposed. NETCONF experimental demonstrations are carried out to validate the proposed model and to prove the control and management capabilities of these technologies applied to elastic optical networks (EONs).

**Index Terms**—NETCONF, YANG, Sliceable Transponder, EON, SBVT, slice-ability, elastic.

## I. INTRODUCTION

Data and control plane of optical networks have experienced relevant advances in the recent years. Considering data plane technologies, sliceable transponders are emerging as a technology meeting the requirements of operators to support variable bit-rate, optimizing the spectral efficiency based on the required optical reach — thus supporting multiple modulation formats or forward error correction (FEC) — and also enabling slice-ability (i.e., the capability of generating independent optical flows to be directed toward different paths and destinations) [1]–[4]. Such transponders, thanks to coherent detection, also support monitoring of transmission parameters (e.g., pre-FEC bit error rate —pre-FEC BER) [5]. These technologies aims to increase both the scalability and agility of the transport network, allowing resource optimization and scaling of bandwidth as demands change and increase. However, while data and control planes have experienced such advances, the innovations in the management plane still need improvements [6] to develop management mechanisms to reduce deployment and operational complexity and maximize benefits of EONs capabilities.

Network Configuration Protocol (NETCONF) [7] is emerging as an SDN protocol standardized by the Internet Engineering Task Force (IETF), providing both control (e.g., data plane device configuration) and management functionalities (e.g., access to monitoring information). NETCONF protocol provides mechanisms to install, manipulate, and delete management

states and information of network devices. NETCONF may rely on the Yet Another Next Generation (YANG) modelling language [8], [9] to describe network devices in a standard way. YANG and NETCONF are gaining interest for operators and research since they provide a standard way to control and manage network elements, independently from the vendor [10].

NETCONF and YANG have been introduced in several networking scenarios. For example, in [11], authors propose a NETCONF agent for link state monitoring compared with other management technologies such as Simple Network Management Protocol (SNMP). In [12], authors highlight the benefits provided by NETCONF in terms of security and scalability, compared with SNMP and REpresentational State Transfer (REST). In [13], authors uses YANG and NETCONF to manage a 10G-PON. Preliminary works of YANG modelling applied to EONs are mentioned in [14]. However, the YANG model has not been reported, and NETCONF is not fully supported.

All the mentioned works do not address a definition of a YANG model for sliceable transponders in EONs employing NETCONF protocol.

This paper presents and demonstrates a control and management plane based on NETCONF protocol exploiting YANG model. In particular, the proposed YANG model describes transponders based on multi-carrier technology, enabling slice-ability and variable baud rate, bit rate, number of carriers, FEC, and modulation format, and supporting the monitoring of several physical parameters such as chromatic dispersion, BER, Q-factor, and others. The paper presents a management experiment using NETCONF exploiting the proposed YANG model. In particular, the controller performs device state discovery, and monitoring information polling.

## II. YANG AND NETCONF

YANG is a data modelling language that can be used to express the structure and semantics of a device information in a vendor-neutral format [8]. Two types of information are present in a YANG model: configuration data and state (operational) data. Configuration data are explicitly set by an external entity on the system. State data reflects the parameters that cannot be set by an external entity (some of these data are fixed by the vendor).

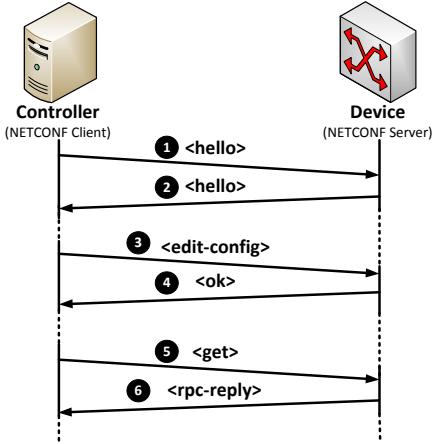


Fig. 1. NETCONF message exchange.

NETCONF is a network configuration and management protocol standardized by the IETF [7]. NETCONF is based on a simple Remote Procedure Call (RPC) layer running over secure transport protocols (typically SSH) to foster interoperability and security. Compared to Simple Network Management Protocol (SNMP), NETCONF has been standardized later, thus it has been designed with additional features to overcome SNMP lacks. Some of the limitation of SNMP have been in RFC 3535 [15] after the IETF workshop held by the Internet Architecture Board on Network Management in 2002. NETCONF provides all the benefit of SNMP as well as many useful features. In particular, NETCONF is transaction based, i.e. when some of the network devices successfully upload the configuration but others fail, NETCONF allows a managed device to rollback to a known-state configuration. Configuration and operational states are well defined and clearly separated. NETCONF supports multiple configuration data stores (candidate, running, startup). Moreover, streaming and playback of event notifications are also supported. The main NETCONF message exchange is illustrated in Fig. 1. Initially, controller (client) and device (server) create a NETCONF session and exchange the list of their capabilities by sending `<hello>` messages (1,2). A capability describes a supported data model. Once the session is opened, the NETCONF peers exchange `<rpc>` and `<rpc-reply>` messages. The `<rpc>` message is used to enclose a NETCONF command sent from the controller (client) to the device (server). The `<get>` command of the `<rpc>` message is used to retrieve the running configuration and state information of the device (3). The `<edit-config>` request is used to write a specific configuration on the device (5) (e.g. to set the maximum transmission unit of a specific Ethernet interface). The `<rpc-reply>` message is sent from the device to the controller in response to an `<rpc>` message. The response data for the given method invoked is encoded as one or more child elements enclosed in the `<rpc-reply>` message.

### III. TRANSPONDER MODEL

In this section, the proposed YANG model for a sliceable transponder with monitoring capabilities is described. The sliceable transponder presented in [2] has been used as a reference architecture to design the YANG model. Standardization guidelines on data modelling suggested by IETF in [16] and by the OpenConfig working group in [17] have been considered. The model has been published online on a public repository [18]. A schematic view of the proposed YANG model describing the transponder is organized in the tree diagram shown in Fig. 2. In particular, a generic transponder is modelled as a list of sub-carrier modules, each one generating or detecting a sub-carrier and a list of connections. In case more than one sub-carrier modules are installed, the Boolean field `slice-ability-support` is used to indicate whether the transponder supports slice-ability or not [2]. The configuration data of the sub-carrier module are the following: e.g., bit rate, baud rate, modulation, FEC, central frequency, and bandwidth. Some additional fields may appear depending whether the sub-carrier is used in reception (e.g., local oscillator, sampling rate, and analog bandwidth) or in transmission (e.g., the launch power). The state data of the sub-carrier module are the following: supported bit rates, baud rates, modulation formats, and FEC. Then, an additional field is proposed for monitoring capabilities, i.e. listing the parameters that can be monitored by the transponder itself: e.g., input power at the receiver. Regarding the connections, the configuration data are the following: a list of sub-carriers module ids referring to those sub-carriers involved in the connection, and the frequency slot composed by the nominal central frequency (`n`) and the slot width (`m`), according to ITU-T [19].

Listing 1 shows the proposed YANG model. In the next sections, the proposed YANG model code will be detailed.

#### A. Sub-carrier module

In the model, each sub-carrier module (line 290) has a configurable direction (line 156) which is either transmission (TX) or reception (RX). Several parameters of the sub-carrier module can be configured, e.g. the bit rate (line 159), baud rate (line 163), modulation format (line 167), FEC (line 171), central frequency (line 172), and bandwidth (line 175). In addition, some other parameters can be configured depending whether the sub-carrier is used in reception or in transmission. In reception, those parameters are the local oscillator (line 104), sampling rate (line 107), and analog bandwidth (line 111), while in transmission, the launch power (line 94). The state data is composed by a replica of the configuration data (as explained in section II), plus the supported functionalities and some additional monitoring information. The supported functionality are organized in lists, in particular: list of supported bit rates (line 181), baud rates (line 186), modulation formats (line 191), and FEC (line 85). Then, an additional field is proposed for monitoring functionalities, i.e. listing the parameters that can be monitored by the transponder itself: input power at the receiver (line 117), pre-FEC BER

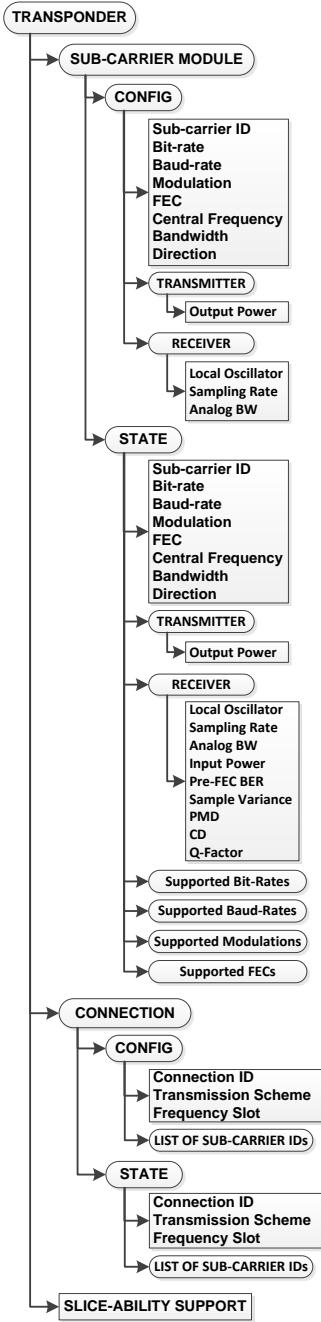


Fig. 2. Transponder YANG Tree

(line 121), polarization mode dispersion (line 134), chromatic dispersion (line 141), and Q-factor (line 147).

### B. Connections

In the model, each connection (line 270) is identified with a unique *connection-id* (line 272), uses a specific *transmission scheme* (e.g. NWDM) (line 232), is supported by one or more sub-carriers specified by the *sub-carrier list* (line 235), and occupies a certain amount of bandwidth specified by the *frequency-slot* (line 243). The sub-carrier list contains the IDs of those sub-carriers involved in the connection while the

frequency slot is described by the nominal central frequency index ( $n$ ) (line 253) and the slot width ( $m$ ) (line 257) according to ITU-T [19]. All the connection parameters introduced so far are configuration data. The state data is composed for the connections by just the replica of the configuration data.

Listing 1. Transponder YANG Model

```

2 module transponder {
3   namespace "http://ssup.it/transponder";
4   prefix tran;
5   import modulation-formats {
6     prefix mdfrms;
7   }
8   import fec-types {
9     prefix fec;
10 }
11 import ietf-yang-types {
12   prefix yang;
13 }
14
15 typedef transmission-type {
16   type enumeration {
17     enum NWDM;
18     enum O-OFDM;
19     enum TFP; // Time-frequency packing
20   }
21 }
22 typedef direction-type {
23   type enumeration {
24     enum TX;
25     enum RX;
26   }
27 }
28 typedef bit-rate-type {
29   type decimal64 {
30     fraction-digits 3;
31     range "0..max";
32   }
33   units "Gb/s";
34 }
35 typedef baud-rate-type {
36   type decimal64 {
37     fraction-digits 3;
38     range "0..max";
39   }
40   units "Gbaud";
41 }
42 typedef modulation-type {
43   type identityref {
44     base mdfrms:modulation-format;
45   }
46 }
47 typedef fec-type {
48   type identityref {
49     base fec:fec-type;
50   }
51 }
52 typedef frequency-ghz-type {
53   type decimal64 {
54     fraction-digits 8;
55     range "0..max";
56   }
57   units "GHz";
58 }
59
60 grouping fec-config {
61   container fec-in-use {
62     presence "Enables FEC";
63     leaf name {
64       type fec-type;
65     }
66     container rate {
67       leaf message-length {
68         type int16 {
69           range "1..max";
70         }
71       }
72       leaf block-length {
73         type int16 {
74           range "1..max";
75         }
76       }
77       must "block-length >= message-length" {
78         error-message "block-length must be >= message-length";
79       }
80     } // rate
81   } // fec-in-use
82 } // fec-config
83
84 grouping fec-state {
85   container supported-fec {
86     description "List of supported FEC schemes";
87     leaf-list fec {
88       type fec-type;
89     }
90   }
91 }
92
93 grouping fec-parameters {
94   container fec-parameters {
95     description "FEC parameters";
96     leaf fec-type {
97       type fec-type;
98     }
99   }
100 }
101
102 grouping fec-configuration {
103   container fec-configuration {
104     description "FEC configuration";
105     leaf fec-type {
106       type fec-type;
107     }
108   }
109 }
110
111 grouping fec-operations {
112   container fec-operations {
113     description "FEC operations";
114     leaf fec-type {
115       type fec-type;
116     }
117   }
118 }
119
120 grouping fec-statistics {
121   container fec-statistics {
122     description "FEC statistics";
123     leaf fec-type {
124       type fec-type;
125     }
126   }
127 }
128
129 grouping fec-monitors {
130   container fec-monitors {
131     description "FEC monitors";
132     leaf fec-type {
133       type fec-type;
134     }
135   }
136 }
137
138 grouping fec-augments {
139   container fec-augments {
140     description "FEC augments";
141     leaf fec-type {
142       type fec-type;
143     }
144   }
145 }
146
147 grouping fec-interfaces {
148   container fec-interfaces {
149     description "FEC interfaces";
150     leaf fec-type {
151       type fec-type;
152     }
153   }
154 }
155
156 grouping fec-entities {
157   container fec-entities {
158     description "FEC entities";
159     leaf fec-type {
160       type fec-type;
161     }
162   }
163 }
164
165 grouping fec-entities {
166   container fec-entities {
167     description "FEC entities";
168     leaf fec-type {
169       type fec-type;
170     }
171   }
172 }
173
174 grouping fec-entities {
175   container fec-entities {
176     description "FEC entities";
177     leaf fec-type {
178       type fec-type;
179     }
180   }
181 }
182
183 grouping fec-entities {
184   container fec-entities {
185     description "FEC entities";
186     leaf fec-type {
187       type fec-type;
188     }
189   }
190 }
191
192 grouping fec-entities {
193   container fec-entities {
194     description "FEC entities";
195     leaf fec-type {
196       type fec-type;
197     }
198   }
199 }
200
201 grouping fec-entities {
202   container fec-entities {
203     description "FEC entities";
204     leaf fec-type {
205       type fec-type;
206     }
207   }
208 }
209
210 grouping fec-entities {
211   container fec-entities {
212     description "FEC entities";
213     leaf fec-type {
214       type fec-type;
215     }
216   }
217 }
218
219 grouping fec-entities {
220   container fec-entities {
221     description "FEC entities";
222     leaf fec-type {
223       type fec-type;
224     }
225   }
226 }
227
228 grouping fec-entities {
229   container fec-entities {
230     description "FEC entities";
231     leaf fec-type {
232       type fec-type;
233     }
234   }
235 }
236
237 grouping fec-entities {
238   container fec-entities {
239     description "FEC entities";
240     leaf fec-type {
241       type fec-type;
242     }
243   }
244 }
245
246 grouping fec-entities {
247   container fec-entities {
248     description "FEC entities";
249     leaf fec-type {
250       type fec-type;
251     }
252   }
253 }
254
255 grouping fec-entities {
256   container fec-entities {
257     description "FEC entities";
258     leaf fec-type {
259       type fec-type;
260     }
261   }
262 }
263
264 grouping fec-entities {
265   container fec-entities {
266     description "FEC entities";
267     leaf fec-type {
268       type fec-type;
269     }
270   }
271 }
272
273 grouping fec-entities {
274   container fec-entities {
275     description "FEC entities";
276     leaf fec-type {
277       type fec-type;
278     }
279   }
280 }
281
282 grouping fec-entities {
283   container fec-entities {
284     description "FEC entities";
285     leaf fec-type {
286       type fec-type;
287     }
288   }
289 }
290
291 grouping fec-entities {
292   container fec-entities {
293     description "FEC entities";
294     leaf fec-type {
295       type fec-type;
296     }
297   }
298 }
299
300 grouping fec-entities {
301   container fec-entities {
302     description "FEC entities";
303     leaf fec-type {
304       type fec-type;
305     }
306   }
307 }
308
309 grouping fec-entities {
310   container fec-entities {
311     description "FEC entities";
312     leaf fec-type {
313       type fec-type;
314     }
315   }
316 }
317
318 grouping fec-entities {
319   container fec-entities {
320     description "FEC entities";
321     leaf fec-type {
322       type fec-type;
323     }
324   }
325 }
326
327 grouping fec-entities {
328   container fec-entities {
329     description "FEC entities";
330     leaf fec-type {
331       type fec-type;
332     }
333   }
334 }
335
336 grouping fec-entities {
337   container fec-entities {
338     description "FEC entities";
339     leaf fec-type {
340       type fec-type;
341     }
342   }
343 }
344
345 grouping fec-entities {
346   container fec-entities {
347     description "FEC entities";
348     leaf fec-type {
349       type fec-type;
350     }
351   }
352 }
353
354 grouping fec-entities {
355   container fec-entities {
356     description "FEC entities";
357     leaf fec-type {
358       type fec-type;
359     }
360   }
361 }
362
363 grouping fec-entities {
364   container fec-entities {
365     description "FEC entities";
366     leaf fec-type {
367       type fec-type;
368     }
369   }
370 }
371
372 grouping fec-entities {
373   container fec-entities {
374     description "FEC entities";
375     leaf fec-type {
376       type fec-type;
377     }
378   }
379 }
380
381 grouping fec-entities {
382   container fec-entities {
383     description "FEC entities";
384     leaf fec-type {
385       type fec-type;
386     }
387   }
388 }
389
390 grouping fec-entities {
391   container fec-entities {
392     description "FEC entities";
393     leaf fec-type {
394       type fec-type;
395     }
396   }
397 }
398
399 grouping fec-entities {
400   container fec-entities {
401     description "FEC entities";
402     leaf fec-type {
403       type fec-type;
404     }
405   }
406 }
407
408 grouping fec-entities {
409   container fec-entities {
410     description "FEC entities";
411     leaf fec-type {
412       type fec-type;
413     }
414   }
415 }
416
417 grouping fec-entities {
418   container fec-entities {
419     description "FEC entities";
420     leaf fec-type {
421       type fec-type;
422     }
423   }
424 }
425
426 grouping fec-entities {
427   container fec-entities {
428     description "FEC entities";
429     leaf fec-type {
430       type fec-type;
431     }
432   }
433 }
434
435 grouping fec-entities {
436   container fec-entities {
437     description "FEC entities";
438     leaf fec-type {
439       type fec-type;
440     }
441   }
442 }
443
444 grouping fec-entities {
445   container fec-entities {
446     description "FEC entities";
447     leaf fec-type {
448       type fec-type;
449     }
450   }
451 }
452
453 grouping fec-entities {
454   container fec-entities {
455     description "FEC entities";
456     leaf fec-type {
457       type fec-type;
458     }
459   }
460 }
461
462 grouping fec-entities {
463   container fec-entities {
464     description "FEC entities";
465     leaf fec-type {
466       type fec-type;
467     }
468   }
469 }
470
471 grouping fec-entities {
472   container fec-entities {
473     description "FEC entities";
474     leaf fec-type {
475       type fec-type;
476     }
477   }
478 }
479
480 grouping fec-entities {
481   container fec-entities {
482     description "FEC entities";
483     leaf fec-type {
484       type fec-type;
485     }
486   }
487 }
488
489 grouping fec-entities {
490   container fec-entities {
491     description "FEC entities";
492     leaf fec-type {
493       type fec-type;
494     }
495   }
496 }
497
498 grouping fec-entities {
499   container fec-entities {
500     description "FEC entities";
501     leaf fec-type {
502       type fec-type;
503     }
504   }
505 }
506
507 grouping fec-entities {
508   container fec-entities {
509     description "FEC entities";
510     leaf fec-type {
511       type fec-type;
512     }
513   }
514 }
515
516 grouping fec-entities {
517   container fec-entities {
518     description "FEC entities";
519     leaf fec-type {
520       type fec-type;
521     }
522   }
523 }
524
525 grouping fec-entities {
526   container fec-entities {
527     description "FEC entities";
528     leaf fec-type {
529       type fec-type;
530     }
531   }
532 }
533
534 grouping fec-entities {
535   container fec-entities {
536     description "FEC entities";
537     leaf fec-type {
538       type fec-type;
539     }
540   }
541 }
542
543 grouping fec-entities {
544   container fec-entities {
545     description "FEC entities";
546     leaf fec-type {
547       type fec-type;
548     }
549   }
550 }
551
552 grouping fec-entities {
553   container fec-entities {
554     description "FEC entities";
555     leaf fec-type {
556       type fec-type;
557     }
558   }
559 }
560
561 grouping fec-entities {
562   container fec-entities {
563     description "FEC entities";
564     leaf fec-type {
565       type fec-type;
566     }
567   }
568 }
569
570 grouping fec-entities {
571   container fec-entities {
572     description "FEC entities";
573     leaf fec-type {
574       type fec-type;
575     }
576   }
577 }
578
579 grouping fec-entities {
580   container fec-entities {
581     description "FEC entities";
582     leaf fec-type {
583       type fec-type;
584     }
585   }
586 }
587
588 grouping fec-entities {
589   container fec-entities {
590     description "FEC entities";
591     leaf fec-type {
592       type fec-type;
593     }
594   }
595 }
596
597 grouping fec-entities {
598   container fec-entities {
599     description "FEC entities";
600     leaf fec-type {
601       type fec-type;
602     }
603   }
604 }
605
606 grouping fec-entities {
607   container fec-entities {
608     description "FEC entities";
609     leaf fec-type {
610       type fec-type;
611     }
612   }
613 }
614
615 grouping fec-entities {
616   container fec-entities {
617     description "FEC entities";
618     leaf fec-type {
619       type fec-type;
620     }
621   }
622 }
623
624 grouping fec-entities {
625   container fec-entities {
626     description "FEC entities";
627     leaf fec-type {
628       type fec-type;
629     }
630   }
631 }
632
633 grouping fec-entities {
634   container fec-entities {
635     description "FEC entities";
636     leaf fec-type {
637       type fec-type;
638     }
639   }
640 }
641
642 grouping fec-entities {
643   container fec-entities {
644     description "FEC entities";
645     leaf fec-type {
646       type fec-type;
647     }
648   }
649 }
650
651 grouping fec-entities {
652   container fec-entities {
653     description "FEC entities";
654     leaf fec-type {
655       type fec-type;
656     }
657   }
658 }
659
660 grouping fec-entities {
661   container fec-entities {
662     description "FEC entities";
663     leaf fec-type {
664       type fec-type;
665     }
666   }
667 }
668
669 grouping fec-entities {
670   container fec-entities {
671     description "FEC entities";
672     leaf fec-type {
673       type fec-type;
674     }
675   }
676 }
677
678 grouping fec-entities {
679   container fec-entities {
680     description "FEC entities";
681     leaf fec-type {
682       type fec-type;
683     }
684   }
685 }
686
687 grouping fec-entities {
688   container fec-entities {
689     description "FEC entities";
690     leaf fec-type {
691       type fec-type;
692     }
693   }
694 }
695
696 grouping fec-entities {
697   container fec-entities {
698     description "FEC entities";
699     leaf fec-type {
700       type fec-type;
701     }
702   }
703 }
704
705 grouping fec-entities {
706   container fec-entities {
707     description "FEC entities";
708     leaf fec-type {
709       type fec-type;
710     }
711   }
712 }
713
714 grouping fec-entities {
715   container fec-entities {
716     description "FEC entities";
717     leaf fec-type {
718       type fec-type;
719     }
720   }
721 }
722
723 grouping fec-entities {
724   container fec-entities {
725     description "FEC entities";
726     leaf fec-type {
727       type fec-type;
728     }
729   }
730 }
731
732 grouping fec-entities {
733   container fec-entities {
734     description "FEC entities";
735     leaf fec-type {
736       type fec-type;
737     }
738   }
739 }
740
741 grouping fec-entities {
742   container fec-entities {
743     description "FEC entities";
744     leaf fec-type {
745       type fec-type;
746     }
747   }
748 }
749
750 grouping fec-entities {
751   container fec-entities {
752     description "FEC entities";
753     leaf fec-type {
754       type fec-type;
755     }
756   }
757 }
758
759 grouping fec-entities {
760   container fec-entities {
761     description "FEC entities";
762     leaf fec-type {
763       type fec-type;
764     }
765   }
766 }
767
768 grouping fec-entities {
769   container fec-entities {
770     description "FEC entities";
771     leaf fec-type {
772       type fec-type;
773     }
774   }
775 }
776
777 grouping fec-entities {
778   container fec-entities {
779     description "FEC entities";
780     leaf fec-type {
781       type fec-type;
782     }
783   }
784 }
785
786 grouping fec-entities {
787   container fec-entities {
788     description "FEC entities";
789     leaf fec-type {
790       type fec-type;
791     }
792   }
793 }
794
795 grouping fec-entities {
796   container fec-entities {
797     description "FEC entities";
798     leaf fec-type {
799       type fec-type;
800     }
801   }
802 }
803
804 grouping fec-entities {
805   container fec-entities {
806     description "FEC entities";
807     leaf fec-type {
808       type fec-type;
809     }
810   }
811 }
812
813 grouping fec-entities {
814   container fec-entities {
815     description "FEC entities";
816     leaf fec-type {
817       type fec-type;
818     }
819   }
820 }
821
822 grouping fec-entities {
823   container fec-entities {
824     description "FEC entities";
825     leaf fec-type {
826       type fec-type;
827     }
828   }
829 }
830
831 grouping fec-entities {
832   container fec-entities {
833     description "FEC entities";
834     leaf fec-type {
835       type fec-type;
836     }
837   }
838 }
839
840 grouping fec-entities {
841   container fec-entities {
842     description "FEC entities";
843     leaf fec-type {
844       type fec-type;
845     }
846   }
847 }
848
849 grouping fec-entities {
850   container fec-entities {
851     description "FEC entities";
852     leaf fec-type {
853       type fec-type;
854     }
855   }
856 }
857
858 grouping fec-entities {
859   container fec-entities {
860     description "FEC entities";
861     leaf fec-type {
862       type fec-type;
863     }
864   }
865 }
866
867 grouping fec-entities {
868   container fec-entities {
869     description "FEC entities";
870     leaf fec-type {
871       type fec-type;
872     }
873   }
874 }
875
876 grouping fec-entities {
877   container fec-entities {
878     description "FEC entities";
879     leaf fec-type {
880       type fec-type;
881     }
882   }
883 }
884
885 grouping fec-entities {
886   container fec-entities {
887     description "FEC entities";
888     leaf fec-type {
889       type fec-type;
890     }
891   }
892 }
893
894 grouping fec-entities {
895   container fec-entities {
896     description "FEC entities";
897     leaf fec-type {
898       type fec-type;
899     }
900   }
901 }
902
903 grouping fec-entities {
904   container fec-entities {
905     description "FEC entities";
906     leaf fec-type {
907       type fec-type;
908     }
909   }
910 }
911
912 grouping fec-entities {
913   container fec-entities {
914     description "FEC entities";
915     leaf fec-type {
916       type fec-type;
917     }
918   }
919 }
920
921 grouping fec-entities {
922   container fec-entities {
923     description "FEC entities";
924     leaf fec-type {
925       type fec-type;
926     }
927   }
928 }
929
930 grouping fec-entities {
931   container fec-entities {
932     description "FEC entities";
933     leaf fec-type {
934       type fec-type;
935     }
936   }
937 }
938
939 grouping fec-entities {
940   container fec-entities {
941     description "FEC entities";
942     leaf fec-type {
943       type fec-type;
944     }
945   }
946 }
947
948 grouping fec-entities {
949   container fec-entities {
950     description "FEC entities";
951     leaf fec-type {
952       type fec-type;
953     }
954   }
955 }
956
957 grouping fec-entities {
958   container fec-entities {
959     description "FEC entities";
960     leaf fec-type {
961       type fec-type;
962     }
963   }
964 }
965
966 grouping fec-entities {
967   container fec-entities {
968     description "FEC entities";
969     leaf fec-type {
970       type fec-type;
971     }
972   }
973 }
974
975 grouping fec-entities {
976   container fec-entities {
977     description "FEC entities";
978     leaf fec-type {
979       type fec-type;
980     }
981   }
982 }
983
984 grouping fec-entities {
985   container fec-entities {
986     description "FEC entities";
987     leaf fec-type {
988       type fec-type;
989     }
990   }
991 }
992
993 grouping fec-entities {
994   container fec-entities {
995     description "FEC entities";
996     leaf fec-type {
997       type fec-type;
998     }
999   }
1000 }
1001
1002 grouping fec-entities {
1003   container fec-entities {
1004     description "FEC entities";
1005     leaf fec-type {
1006       type fec-type;
1007     }
1008   }
1009 }
1010
1011 grouping fec-entities {
1012   container fec-entities {
1013     description "FEC entities";
1014     leaf fec-type {
1015       type fec-type;
1016     }
1017   }
1018 }
1019
1020 grouping fec-entities {
1021   container fec-entities {
1022     description "FEC entities";
1023     leaf fec-type {
1024       type fec-type;
1025     }
1026   }
1027 }
1028
1029 grouping fec-entities {
1030   container fec-entities {
1031     description "FEC entities";
1032     leaf fec-type {
1033       type fec-type;
1034     }
1035   }
1036 }
1037
1038 grouping fec-entities {
1039   container fec-entities {
1040     description "FEC entities";
1041     leaf fec-type {
1042       type fec-type;
1043     }
1044   }
1045 }
1046
1047 grouping fec-entities {
1048   container fec-entities {
1049     description "FEC entities";
1050     leaf fec-type {
1051       type fec-type;
1052     }
1053   }
1054 }
1055
1056 grouping fec-entities {
1057   container fec-entities {
1058     description "FEC entities";
1059     leaf fec-type {
1060       type fec-type;
1061     }
1062   }
1063 }
1064
1065 grouping fec-entities {
1066   container fec-entities {
1067     description "FEC entities";
1068     leaf fec-type {
1069       type fec-type;
1070     }
1071   }
1072 }
1073
1074 grouping fec-entities {
1075   container fec-entities {
1076     description "FEC entities";
1077     leaf fec-type {
1078       type fec-type;
1079     }
1080   }
1081 }
1082
1083 grouping fec-entities {
1084   container fec-entities {
1085     description "FEC entities";
1086     leaf fec-type {
1087       type fec-type;
1088     }
1089   }
1090 }
1091
1092 grouping fec-entities {
1093   container fec-entities {
1094     description "FEC entities";
1095     leaf fec-type {
1096       type fec-type;
1097     }
1098   }
1099 }
1100
1101 grouping fec-entities {
1102   container fec-entities {
1103     description "FEC entities";
1104     leaf fec-type {
1105       type fec-type;
1106     }
1107   }
1108 }
1109
1110 grouping fec-entities {
1111   container fec-entities {
1112     description "FEC entities";
1113     leaf fec-type {
1114       type fec-type;
1115     }
1116   }
1117 }
1118
1119 grouping fec-entities {
1120   container fec-entities {
1121     description "FEC entities";
1122     leaf fec-type {
1123       type fec-type;
1124     }
1125   }
1126 }
1127
1128 grouping fec-entities {
1129   container fec-entities {
1130     description "FEC entities";
1131     leaf fec-type {
1132       type fec-type;
1133     }
1134   }
1135 }
1136
1137 grouping fec-entities {
1138   container fec-entities {
1139     description "FEC entities";
1140     leaf fec-type {
1141       type fec-type;
1142     }
1143   }
1144 }
1145
1146 grouping fec-entities {
1147   container fec-entities {
1148     description "FEC entities";
1149     leaf fec-type {
1150       type fec-type;
1151     }
1152   }
1153 }
1154
1155 grouping fec-entities {
1156   container fec-entities {
1157     description "FEC entities";
1158     leaf fec-type {
1159       type fec-type;
1160     }
1161   }
1162 }
1163
1164 grouping fec-entities {
1165   container fec-entities {
1166     description "FEC entities";
1167     leaf fec-type {
1168       type fec-type;
1169     }
1170   }
1171 }
1172
1173 grouping fec-entities {
1174   container fec-entities {
1175     description "FEC entities";
1176     leaf fec-type {
1177       type fec-type;
1178     }
1179   }
1180 }
1181
1182 grouping fec-entities {
1183   container fec-entities {
1184     description "FEC entities";
1185     leaf fec-type {
1186       type fec-type;
1187     }
1188   }
1189 }
1190
1191 grouping fec-entities {
1192   container fec-entities {
1193     description "FEC entities";
1194     leaf fec-type {
1195       type fec-type;
1196     }
1197   }
1198 }
1199
1200 grouping fec-entities {
1201   container fec-entities {
1202     description "FEC entities";
1203     leaf fec-type {
1204       type fec-type;
1205     }
1206   }
1207 }
1208
1209 grouping fec-entities {
1210   container fec-entities {
1211     description "FEC entities";
1212     leaf fec-type {
1213       type fec-type;
1214     }
1215   }
1216 }
1217
1218 grouping fec-entities {
1219   container fec-entities {
1220     description "FEC entities";
1221     leaf fec-type {
1222       type fec-type;
1223     }
1224   }
1225 }
1226
1227 grouping fec-entities {
1228   container fec-entities {
1229     description "FEC entities";
1230     leaf fec-type {
1231       type fec-type;
1232     }
1233   }
1234 }
1235
1236 grouping fec-entities {
1237   container fec-entities {
1238     description "FEC entities";
1239     leaf fec-type {
1240       type fec-type;
1241     }
1242   }
1243 }
1244
1245 grouping fec-entities {
1246   container fec-entities {
1247     description "FEC entities";
1248     leaf fec-type {
1249       type fec-type;
1250     }
1251   }
1252 }
1253
1254 grouping fec-entities {
1255   container fec-entities {
1256     description "FEC entities";
1257     leaf fec-type {
1258       type fec-type;
1259     }
1260   }
1261 }
1262
1263 grouping fec-entities {
1264   container fec-entities {
1265     description "FEC entities";
1266     leaf fec-type {
1267       type fec-type;
1268     }
1269   }
1270 }
1271
1272 grouping fec-entities {
1273   container fec-entities {
1274     description "FEC entities";
1275     leaf fec-type {
1276       type fec-type;
1277     }
1278   }
1279 }
1280
1281 grouping fec-entities {
1282   container fec-entities {
1283     description "FEC entities";
1284     leaf fec-type {
1285       type fec-type;
1286     }
1287   }
1288 }
1289
1290 grouping fec-entities {
1291   container fec-entities {
1292     description "FEC entities";
1293     leaf fec-type {
1294       type fec-type;
1295     }
1296   }
1297 }
1298
1299 grouping fec-entities {
1300   container fec-entities {
1301     description "FEC entities";
1302     leaf fec-type {
1303       type fec-type;
1304     }
1305   }
1306 }
1307
1308 grouping fec-entities {
1309   container fec-entities {
1310     description "FEC entities";
1311     leaf fec-type {
1312       type fec-type;
1313     }
1314   }
1315 }
1316
1317 grouping fec-entities {
1318   container fec-entities {
1319     description "FEC entities";
1320     leaf fec-type {
1321       type fec-type;
1322     }
1323   }
1324 }
1325
1326 grouping fec-entities {
1327   container fec-entities {
1328     description "FEC entities";
1329     leaf fec-type {
1330       type fec-type;
1331     }
1332   }
1333 }
1334
1335 grouping fec-entities {
1336   container fec-entities {
1337     description "FEC entities";
1338     leaf fec-type {
1339       type fec-type;
1340     }
1341   }
1342 }
1343
1344 grouping fec-entities {
1345   container fec-entities {
1346     description "FEC entities";
1347     leaf fec-type {
1348       type fec-type;
1349     }
1350   }
1351 }
1352
1353 grouping fec-entities {
1354   container fec-entities {
1355     description "FEC entities";
1356     leaf fec-type {
1357       type fec-type;
1358     }
1359   }
1360 }
1361
1362 grouping fec-entities {
1363   container fec-entities {
1364     description "FEC entities";
1365     leaf fec-type {
1366       type fec-type;
1367     }
1368   }
1369 }
1370
1371 grouping fec-entities {
1372   container fec-entities {
1373     description "FEC entities";
1374     leaf fec-type {
1375       type fec-type;
1376     }
1377   }
1378 }
1379
1380 grouping fec-entities {
1381   container fec-entities {
1382     description "FEC entities";
1383     leaf fec-type {
1384       type fec-type;
1385     }
1386   }
1387 }
1388
1389 grouping fec-entities {
1390   container fec-entities {
1391     description "FEC entities";
1392     leaf fec-type {
1393       type fec-type;
1394     }
1395   }
1396 }
1397
1398 grouping fec-entities {
1399   container fec-entities {
1400     description "FEC entities";
1401     leaf fec-type {
1402       type fec-type;
1403     }
1404   }
1405 }
1406
1407 grouping fec-entities {
1408   container fec-entities {
1409     description "FEC entities";
1410     leaf fec-type {
1411       type fec-type;
1412     }
1413   }
1414 }
1415
1416 grouping fec-entities {
1417   container fec-entities {
1418     description "FEC entities";
1419     leaf fec-type {
1420       type fec-type;
1421     }
1422   }
1423 }
1424
1425 grouping fec-entities {
1426   container fec-entities {
1427     description "FEC entities";
1428     leaf fec-type {
1429       type fec-type;
1430     }
1431   }
1432 }
1433
1434 grouping fec-entities {
1435   container fec-entities {
1436     description "FEC entities";
1437     leaf fec-type {
1438       type fec-type;
1439     }
1440   }
1441 }
1442
1443 grouping fec-entities {
1444   container fec-entities {
1445     description "FEC entities";
1446     leaf fec-type {
1447       type fec-type;
1448     }
1449   }
1450 }
1451
1452 grouping fec-entities {
1453   container fec-entities {
1454     description "FEC entities";
1455     leaf fec-type {
1456       type fec-type;
1457     }
1458   }
1459 }
1460
1461 grouping fec-entities {
1462   container fec-entities {
1463     description "FEC entities";
1464     leaf fec-type {
1465       type fec-type;
1466     }
1467   }
1468 }
1469
1470 grouping fec-entities {
1471   container fec-entities {
1472     description "FEC entities";
1473     leaf fec-type {
1474       type fec-type;
1475     }
1476   }
1477 }
1478
1479 grouping fec-entities {
1480   container fec-entities {
1481     description "FEC entities";
1482     leaf fec-type {
1483       type fec-type;
1484     }
1485   }
1486 }
1487
1488 grouping fec-entities {
1489   container fec-entities {
1490     description "FEC entities";
1491     leaf fec-type {
1492       type fec-type;
1493     }
1494   }
1495 }
1496
1497 grouping fec-entities {
1498   container fec-entities {
1499     description "FEC entities";
1500     leaf fec-type {
1501       type fec-type;
1502     }
1503   }
1504 }
1505
1506 grouping fec-entities {
1507   container fec-entities {
1508     description "FEC entities";
1509     leaf fec-type {
1510       type fec-type;
1511     }
1512   }
1513 }
1514
1515 grouping fec-entities {
1516   container fec-entities {
1517     description "FEC entities";
1518     leaf fec-type {
1519       type fec-type;
1520     }
1521   }
1522 }
1523
1524 grouping fec-entities {
1525   container fec-entities {
1526     description "FEC entities";
1527     leaf fec-type {
1528       type fec-type;
1529     }
1530   }
1531 }
1532
1533 grouping fec-entities {
1534   container fec-entities {
1535     description "FEC entities";
1536     leaf fec-type {
1537       type fec-type;
1538     }
1539   }
1540 }
1541
1542 grouping fec-entities {
1543   container fec-entities {
1544     description "FEC entities";
1545     leaf fec-type {
1546       type fec-type;
1547     }
1548   }
1549 }
1550
1551 grouping fec-entities {
1552   container fec-entities {
1553     description "FEC entities";
1554     leaf fec-type {
1555       type fec-type;
1556     }
1557   }
1558 }
1559
1560 grouping fec-entities {
1561   container fec-entities {
1562     description "FEC entities";
1563     leaf fec-type {
1564       type fec-type;
1565     }
1566   }
1567 }

```

```

89     }
90   } // supported
91 } // fec-state
92
93 grouping transmitter-config {
94   leaf output-power {
95     type int16;
96     units "dBm";
97   }
98 } // transmitter-config
99
100 grouping transmitter-state {
101 }
102
103 grouping receiver-config {
104   leaf local-oscillator {
105     type frequency-ghz-type;
106   }
107   leaf sampling-rate {
108     type uint32;
109     units "GS/s";
110   }
111   leaf analog-bw {
112     type frequency-ghz-type;
113   }
114 } // receiver-config
115
116 grouping receiver-state {
117   leaf input-power {
118     type int16;
119     units "dBm";
120   }
121   leaf pre-fec-ber {
122     type decimal64 {
123       fraction-digits 18;
124       range "0..max";
125     }
126   }
127   leaf sample-variance {
128     type decimal64 {
129       fraction-digits 18;
130       range "0..max";
131     }
132     reference "http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=7035536";
133   }
134   leaf pmd {
135     type decimal64 {
136       fraction-digits 8;
137       range "0..max";
138     }
139     units "ps/(km)^0.5";
140   }
141   leaf cd {
142     type decimal64 {
143       fraction-digits 5;
144     }
145     units "ps/nm/km";
146   }
147   leaf q-factor {
148     type decimal64 {
149       fraction-digits 5;
150     }
151     units "dB";
152   }
153 } // receiver-state
154
155 grouping subcarrier-module-config {
156   leaf direction {
157     type direction-type;
158   }
159   leaf bit-rate {
160     description "The bit-rate in use";
161     type bit-rate-type;
162   }
163   leaf baud-rate {
164     description "The baud-rate in use";
165     type baud-rate-type;
166   }
167   leaf modulation {
168     description "Modulation format in use";
169     type modulation-type;
170   }
171   uses fec-config;
172   leaf central-frequency {
173     type frequency-ghz-type;
174   }
175   leaf bandwidth {
176     type frequency-ghz-type;
177   }
178 } // subcarrier-module-config
179
180 grouping subcarrier-module-state {
181   container supported-bit-rates {
182     leaf-list bit-rate {
183       type bit-rate-type;
184     }
185   }
186   container supported-baud-rates {
187     leaf-list baud-rate {
188       type baud-rate-type;
189     }
190   }
191   container supported-modulations {
192     leaf-list modulation {
193       type modulation-type;
194     }
195   }
196   uses fec-state;
197 } // subcarrier-module-state
198
199 grouping subcarrier-module {
200   container config {
201     uses subcarrier-module-config;
202     container transmitter {
203       when "../direction=:TX";
204       uses transmitter-config;
205     }
206     container receiver {
207       when "../direction=:RX";
208       uses receiver-config;
209     }
210   }
211   container state {
212     config false;
213     uses subcarrier-module-config;
214     uses subcarrier-module-state;
215     container transmitter {
216       when "../direction=:TX";
217       uses transmitter-config;
218       uses transmitter-state;
219     }
220     container receiver {
221       when "../direction=:RX";
222       uses receiver-config;
223       uses receiver-state;
224     }
225   }
226 } // subcarrier-module
227
228 grouping connection-config {
229   leaf connection-id {
230     type uint32;
231   }
232   leaf transmission-scheme {
233     type transmission-type;
234   }
235   list subcarrier {
236     key "subcarrier-id";
237     leaf subcarrier-id {
238       type leafref {
239         path "/tran:transponder/tran:subcarrier-module
240           /tran:subcarrier-id";
241       }
242     }
243     container frequency-slot {
244       reference "draft-ietf-ccamp-flexi-grid-fwk-07";
245       leaf nominal-central-frequency-granularity {
246         type frequency-ghz-type;
247         default 6.25;
248       }
249       leaf slot-width-granularity {
250         type frequency-ghz-type;
251         default 12.5;
252       }
253       leaf n {
254         type int16;
255         mandatory true;
256       }
257       leaf m {
258         type int16 {
259           range "1..max";
260         }
261         mandatory true;
262       }
263     }
264   }
265 } // connection-config
266
267 grouping connection-state {
268 }
269
270 grouping connections {
271   list connection {
272     key "connection-id";
273     leaf connection-id {
274       type leafref {
275         path "../config/connection-id";
276       }
277     }
278     container config {
279       uses connection-config;
280     }
281     container state {
282       config false;
283       uses connection-config;
284       uses connection-state;
285     }
286   }
287 } // connection

```

```

286 } // connections
287 //----- MAIN TREE -----
288 container transponder {
289     list subcarrier-module {
290         key "subcarrier-id";
291         leaf subcarrier-id {
292             type uint32;
293         }
294         uses subcarrier-module;
295     }
296
297     leaf slice-ability-support {
298         when "count(.. / subcarrier-module) >= 1";
299         type boolean;
300         config false;
301     }
302
303     leaf node-id {
304         type uint16;
305     }
306
307     leaf add-drop-id {
308         type uint16;
309     }
310
311     container connections {
312         uses connections;
313     }
314 }
315 }
316 } // transponder

```

#### IV. EXPERIMENTAL DEMONSTRATION

The controller runs a python implementation of a NETCONF client. The device is emulated using a virtual machine with 4GB RAM, 1 processor at 3.4GHz, and Ubuntu Linux as operating system. The virtual machine runs ConfD, a NETCONF server implementation made by Tail-f (CISCO), and a C program we made to emulate the monitoring capabilities of the device (Fig. 3). Our program communicates with ConfD through specific APIs provided by the latter. The transponder YANG model, as in section III, has been included in the capabilities of both client and server. In the experiment, controller and devices are under the same local area network and connected to the same fast Ethernet switch. The experiment is divided into two parts. First, the controller discovers the device state and functionality. Then, the controller monitors an active connection by issuing NETCONF `<get>` command.

##### A. Transponder discovery

In this part of the experiment, the controller interrogates the device to retrieve its current state (e.g. installed sub-carriers modules, supported transmission parameters). In particular, the controller issues an RPC `<get>` command asking for the full state information of the transponder (Fig. 1 (3)). Listing 2 shows the content of the message captured with Wireshark.

Listing 2. NETCONF RPC `<get>` message.

```

<?xml version="1.0" encoding="UTF-8" ?>
<rpc xmlns="urn:ietf:params:xml:ns:netconf:base:1.0"
      message-id="1">
    <get><filter type='xpath' select=' /transponder' /></get>
</rpc>

```

The device replies with a `<rpc-reply>` message containing the full transponder state information (Fig. 1 (4)). Listing 3 shows the content of the captured message (some portions of the message have been skipped by adding "..."). Among all data, it is possible to evince that the transponder has four sub-carrier modules installed. The first sub-carrier module (ID equal to 1) supports the following transmission parameters: four bit rates (112, 124, 224, 248 Gbps), two baud rates (28,

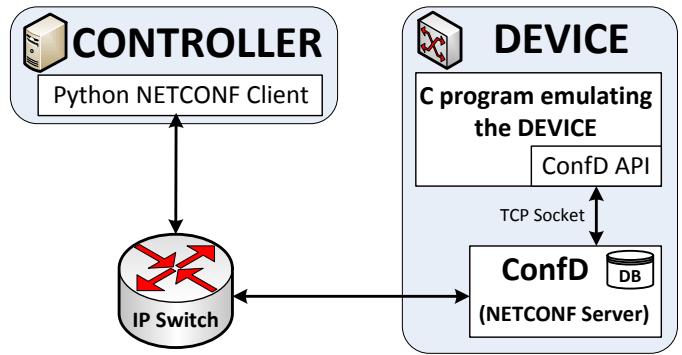


Fig. 3. Experimental setup.

31 Gbaud), two possible modulation formats (dual polarization quadrature phase shift keying —DP-QPSK— and dual polarization 16 quadrature amplitude modulation —DP-16QAM), and two FEC schemes (Low-Density Parity-Check – LDPC, Golay). In particular, 112 and 124 Gbps are associated to DP-QPSK with 28 and 31 Gbaud, respectively, while 224 and 248 Gbps to DP-16QAM with 28 and 31 Gbaud, respectively.

Listing 3. NETCONF RPC reply message with the requested data.

```

<?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>
          <supported-bit-rates>
            <bit-rate>112.0</bit-rate>
            <bit-rate>124.0</bit-rate>
            <bit-rate>224.0</bit-rate>
            <bit-rate>248.0</bit-rate>
          </supported-bit-rates>
          <supported-baud-rates>
            <baud-rate>28.0</baud-rate>
            <baud-rate>31.0</baud-rate>
          </supported-baud-rates>
          <supported-modulations>
            <modulation xmlns:mdfrms="http://sssup.it/modulation-formats">mdfrms:dp-qpsk</modulation>
            <modulation xmlns:mdfrms="http://sssup.it/modulation-formats">mdfrms:dp-16qam</modulation>
          </supported-modulations>
          <supported-fec>
            <fec xmlns:fec="http://sssup.it/fec-types">fec:ldpc</fec>
            <fec xmlns:fec="http://sssup.it/fec-types">fec:golay</fec>
          </supported-fec>
        </state>
      </subcarrier-module>
      <subcarrier-module>
        <subcarrier-id>2</subcarrier-id>
        <state>
          <supported-bit-rates>
            <bit-rate>112.0</bit-rate>
            <bit-rate>124.0</bit-rate>
            <bit-rate>224.0</bit-rate>
            <bit-rate>248.0</bit-rate>
          </supported-bit-rates>
          <supported-baud-rates>
            <baud-rate>28.0</baud-rate>
            <baud-rate>31.0</baud-rate>
          </supported-baud-rates>
          <supported-modulations>
            <modulation xmlns:mdfrms="http://sssup.it/modulation-formats">mdfrms:dp-qpsk</modulation>
            <modulation xmlns:mdfrms="http://sssup.it/modulation-formats">mdfrms:dp-16qam</modulation>
          </supported-modulations>
          <supported-fec>
            <fec xmlns:fec="http://sssup.it/fec-types">fec:ldpc</fec>
            <fec xmlns:fec="http://sssup.it/fec-types">fec:golay</fec>
          </supported-fec>
        </state>
      </subcarrier-module>
    </transponder>
  </data>
</rpc-reply>

```

```

<subcarrier-module>
  <subcarrier-id>3</subcarrier-id>
  ...
</subcarrier-module>
<subcarrier-module>
  <subcarrier-id>4</subcarrier-id>
  ...
</subcarrier-module>
<slice-ability-support>true </slice-ability-support>
<node-id>1</node-id>
<add-drop-id>1</add-drop-id>
<connections>
</connections>
</transponder>
</data>
</rpc-reply>

```

### B. Retrieve monitoring information

In the last part of the experiment, the controller monitors the Q-Factor of a specific connection by periodically issuing a `<get>` command (polling). The considered monitored connection was previously installed and assigned to the first sub-carrier module (sub-carrier id=1) in reception. More in details, it is a 112 Gbps (28 Gbaud) connection, modulated with DP-QPSK modulation and a 14/15 LDPC code. Listing 4 shows the content of the `<get>` command encapsulated in the RPC message issued by the controller.

**Listing 4.** NETCONF RPC `<get>` message to retrieve the Q-Factor.

```

<?xml version="1.0" encoding="UTF-8"?>
<rpc xmlns="urn:ietf:params:xml:ns:netconf:base:1.0"
      message-id="1">
  <get>
    <filter type='xpath' select=' /transponder/subcarrier-
        module[subcarrier-id=1]/state/receiver/q-factor' />
  </get>
</rpc>

```

The reply of the device is showed in Listing 5. In particular, the returned message shows that the monitored connection has a Q-Factor value equal to 6dB.

**Listing 5.** NETCONF RPC `<rpc-reply>` message containing the Q-Factor.

```

<?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>
            <q-factor>6.0</q-factor>
          </receiver>
        </state>
      </subcarrier-module>
    </data>
  </rpc-reply>

```

During the experiment we measured the time from the instant when the `<get>` command is issued to the instant that the `<rpc-reply>` returns to the controller. This time is around 11 milliseconds. We also measured the time required to the device to process the request, i.e. from the instant when the `<get>` command is received to the instant when the `<rpc-reply>` is sent. This processing time is around 8.5 milliseconds.

## V. CONCLUSION

This paper proposed and demonstrated a NETCONF protocol including YANG model describing a transponder with monitoring capabilities. The proposed YANG model of a sliceable transponder is fully detailed in the paper. An experiment is successfully conducted considering two use cases: transponder discovery and monitoring of a connection. In

both use cases, extremely fast performance has been achieved. NETCONF message captures are provided for `<get>`, and `<rpc-reply>` messages.

## REFERENCES

- [1] M. Jinno, H. Takara, Y. Sone, K. Yonenaga, and A. Hirano, "Multiflow optical transponder for efficient multilayer optical networking," *Communications Magazine, IEEE*, vol. 50, no. 5, pp. 56–65, 2012.
- [2] N. Sambo and et al., "Next generation sliceable bandwidth variable transponders," *Communications Magazine, IEEE*, vol. 53, no. 2, 2015.
- [3] M. Svaluto Moreolo, J. Fabrega, L. Nadal, F. Vilchez, V. Lopez, and J. Fernandez-Palacios, "Cost-effective data plane solutions based on OFDM technology for flexi-grid metro networks using sliceable bandwidth variable transponders," in *Proc. of ONDM*, May 2014.
- [4] A. Napoli and at al., "Novel dac digital pre-emphasis algorithm for next-generation flexible optical transponders," in *Proc. of OFC*, March 2015.
- [5] A. Napoli and et al., "Next generation elastic optical networks: The vision of the european research project IDEALIST," *Communications Magazine, IEEE*, vol. 53, no. 2, pp. 152–162, Feb 2015.
- [6] A. Martinez, M. Yannuzzi, V. Lopez, D. Lopez, W. Ramirez, R. Serral-Gracia, X. Masip-Bruin, M. Maciejewski, and J. Altmann, "Network management challenges and trends in multi-layer and multi-vendor settings for carrier-grade networks," *Communications Surveys Tutorials, IEEE*, vol. 16, no. 4, 2014.
- [7] R. Enns, M. Bjorklund, J. Schoenwaelder, and A. Bierman, "Network configuration protocol (NETCONF)," IETF RFC 6241, June 2011.
- [8] M. Bjorklund, "YANG - a data modeling language for the network configuration protocol (NETCONF)," IETF RFC 6020.
- [9] J. Vergara and et al., IETF draft-vergara-flexgrid-yang-00, Oct. 2014.
- [10] J. Schonwalder, M. Bjorklund, and P. Shafer, "Network configuration management using NETCONF and YANG," *Communications Magazine, IEEE*, vol. 48, no. 9, pp. 166–173, Sept 2010.
- [11] D. Loureiro, P. Goncalves, and A. Nogueira, "NETCONF agent for link state monitoring," in *Communications (ICC), 2012 IEEE International Conference on*, June 2012, pp. 6565–6569.
- [12] P. da Paz Ferraz Santos, R. Pereira Esteves, and L. Zambenedetti Granville, "Evaluating SNMP, NETCONF, and RESTful web services for router virtualization management," in *Proc. of IFIP/IEEE IM*, May 2015.
- [13] A. Berti Sassi, M. Gavidia, E. Leao Fernandes, and M. Ribeiro Nascimento, "Integrated management of 10G-PON network element using NETCONF and OpenFlow," in *Proc. of CNSM*, Nov 2014.
- [14] J. Oliveira, M. Siqueira, G. Curiel, A. Hirata, F. van't Hooft, D. Macedo, M. Colazza, and C. Rothenberg, "Experimental testbed of reconfigurable flexgrid optical network with virtualized GMPLS control plane and autonomic controls towards SDN," in *Microwave Optoelectronics Conference (IMOC), 2013 SBMO/IEEE MTT-S International*, Aug 2013, pp. 1–5.
- [15] J. Schoenwaelder, "Overview of the 2002 iab network management workshop," IETF RFC 3535.
- [16] A. Bierman, "Guidelines for authors and reviewers of yang data model documents," IETF RFC 6087.
- [17] in <http://www.openconfig.net>.
- [18] in <https://github.com/mattedallo/ssb/blob/master/yang-models/transponder.yang>.
- [19] "Draft revised G.694.1 version 1.3," Unpublished ITU-T Study Group 15, Question 6.