Internet Engineering Task Force
Internet-Draft
Intended status: Experimental
Expires: January 9, 2020

G. Galimberti, Ed.
Cisco
R. Kunze
Deutsche Telekom
D. Hiremagalur, Ed.
G. Grammel, Ed.
Juniper
July 8, 2019

A YANG model to manage the optical interface parameters for an external transponder in a WDM network
draft-ietf-ccamp-dwdm-if-param-yang-01

Abstract

This memo defines a Yang model related to the Optical Transceiver parameters characterising coherent 100G and above interfaces. 100G and above Transceivers support coherent modulation, multiple modulation formats, multiple FEC codes including some not yet specified (or by in phase of specification by) ITU-T G.698.2 [ITU.G698.2] or any other ITU-T recommendation. More context about the state of the Coherent transceivers is described in draft-many-coherent-DWDM-if-control. Use cases are described in RFC7698.

The Yang model defined in this memo can be used for Optical Parameters monitoring and/or configuration of the endpoints of a multi-vendor IaDI optical link. The use of this model does not guarantee interworking of transceivers over a DWDM. Optical path feasibility and interoperability has to be determined by means outside the scope of this document. The purpose of this model is to program interface parameters to consistently configure the mode of operation of transceivers.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at https://datatracker.ietf.org/drafts/current/.

Copyright Notice

Table of Contents

1. Introduction

This memo defines a Yang model that translates and obsolete the SNMP mib module defined in draft-galikunze-ccamp-dwdm-if-snmp-mib for managing single channel optical interface parameters of DWDM applications, using the approach specified in G.698.2. This model supports parameters to characterize coherent transceivers found in current implementations to specify the mode of operation. As application identifiers like those specified in ITU-T G.874.1 [ITU.G874.1] are not available we use mode templates instead. A mode template describes transceiver characteristics in detail and can be identified by a mode-id.

This draft refers and supports the RFC7698 and draft-many-coherent-DWDM-if-control.

The YANG model describing and extending the optical parameters allows different vendors and operators to retrieve, provision and exchange information across the multi-vendor IaDI interfaces in an abstract manner.

The they concept introduced by this YANG model is the notion of a mode. A mode is a combination of parameters or parameter ranges that is supported by a transceiver. As an example, operating a device in QPSK mode may use a different FEC and requires less OSNR to reach the FEC limit than the same transceiver operating in QAM16 mode. Given the number of parameters and their possible combinations it is important for vendors to be able to qualify a set of combinations which is the basis to define a mode. The YANG model furthermore provides means to selecting one mode as current-mode from that pre-defined list of modes supported by the transceiver module. Once selected, current-opt-if-och-mode-params provide the means to configure specific parameters at run time and retrieve actual parameters from the module. For example, the frequency is a parameter that can be set within min/max boundaries set by the current mode. Laser Temperature however is a ro parameter available at run-time that can be checked against the mode boundaries and may trigger an event.

2. The Internet-Standard Management Framework

For a detailed overview of the documents that describe the current Internet-Standard Management Framework, please refer to section 7 of RFC 3410 [RFC3410].

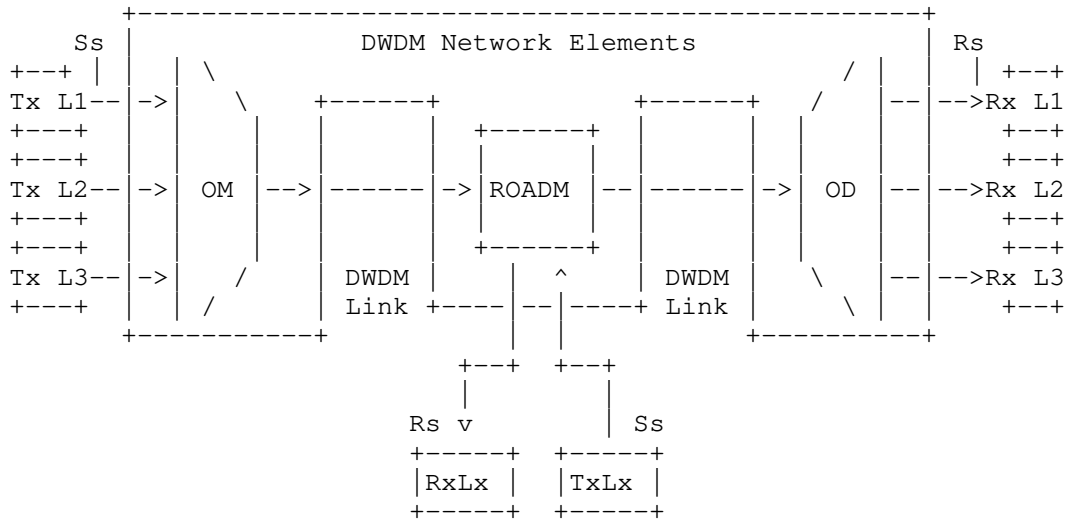
This memo specifies a Yang model for optical interfaces.

3. Conventions

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119] In the description of OIDs the convention: Set (S) Get (G) and Trap (T) conventions will describe the action allowed by the parameter.

4. Overview

Figure 1 shows a set of reference points, for single-channel connection between transmitters (Tx) and receivers (Rx). Here the DWDM network elements include an OM and an OD (which are used as a pair with the opposing element), one or more optical amplifiers and may also include one or more OADMs.



Ss = reference point at the DWDM network element tributary output
Rs = reference point at the DWDM network element tributary input
Lx = Lambda x
OM = Optical Mux
OD = Optical Demux
ROADM = Reconfigurable Optical Add Drop Mux

from Fig. 5.1/G.698.2

Figure 1: External transponder in WDM netwoks

4.1. Optical Parameters Description

The link between the external transponders through a WDM network media channels are managed at the edges, i.e. at the transmitters (Tx) and receivers (Rx) attached to the S and R reference points respectively.

Definitions of the optical parameters are provided below to increase the readability of the document.

4.1.1. Parameters at Ss

output-power:
The mean launched power at Ss is the average power (in dBm) of a pseudo-random data sequence coupled into the DWDM link.

central frequency:
This parameter indicates the Central frequency value that Ss and Rs will be set to work (in THz)

4.1.2. Interface at point Rs

input-power:
The average received power (in dBm) at point Rs.

Curr-OSNR:
Current Optical Signal to Noise Ratio (OSNR) estimated at Rx Transceiver port.

Curr-q-factor:
"Q" factor estimated at Rx Transceiver port.

4.2. Use Cases

The use cases are described in draft-ietf-ccamp-dwdm-if-mng-ctrl-fwk

4.3. Optical Interface for external transponder in a WDM network

The ietf-ext-xponder-wdm-if is an augment to the ietf-interface. It allows the user to set the operating mode of transceivers as well as other operational parameters. The module provides also treshold settings and notifications to supervise measured parameters and notify the client.

```
module: ietf-ext-xponder-wdm-if
 augment /if:interfaces/if:interface:
    +--rw optIfOChRsSs
       +--rw if-current-mode
```

```
|   +--ro mode-id?                        string
|   +--ro application-identifier?         string
|   +--ro min-central-frequency?          uint32
|   +--ro max-central-frequency?          uint32
|   +--ro min-input-power?                dbm-t
|   +--ro max-input-power?                dbm-t
|   +--ro min-output-power?               dbm-t
|   +--ro max-output-power?               dbm-t
|   +--ro min-osnr-margin?                int32
|   +--ro min-q-margin?                   int32
|   +--ro fec-info?                       string
|   +--ro fec-bitrate?                    string
|   +--ro fec-gain?                       string
|   +--ro fec-ber-mantissa-threshold?     uint32
|   +--ro fec-ber-exponent-threshold?     int32
|   +--ro number-of-lanes?                uint32
|   +--ro min-laser-temperature?          int32
|   +--ro max-laser-temperature?          int32
|   +--ro max-total-rx-optical-power?     dbm-t
|   +--ro max-chromatic-dispersion?       int32
|   +--ro max-diff-group-delay?           int32
|   +--ro modulation-format?              string
|   +--ro bits-per-symbol?                uint32
|   +--ro num-symbols-in-alphabet?        uint32
|   +--ro symbols-index?                  uint32
+--ro if-supported-mode
|   +--ro number-of-modes-supported?     uint32
|   +--ro mode-list* [mode-id]
|      +--ro mode-id?                    string
|      +--ro application-identifer?      string
|      +--ro min-central-frequency?      uint32
|      +--ro max-central-frequency?      uint32
|      +--ro min-channel-input-power?    dbm-t
|      +--ro max-channel-input-power?    dbm-t
|      +--ro min-channel-output-power?   dbm-t
|      +--ro max-channel-output-power?   dbm-t
|      +--ro min-osnr-margin?            int32
|      +--ro min-q-margin?               int32
|      +--ro fec-info?                   string
|      +--ro fec-bitrate?                string
|      +--ro fec-gain?                   string
|      +--ro pre-fec-ber-mantissa-threshold? uint32
|      +--ro pre-fec-ber-exponent-threshold? int32
|      +--ro number-of-lanes?            uint32
|      +--ro min-laser-temperature?      int32
|      +--ro max-laser-temperature?      int32
|      +--ro max-total-rx-optical-power? dbm-t
|      +--ro max-chromatic-dispersion?   int32
```

```
          |      +--ro max-diff-group-delay?     int32
          |      +--ro modulation-format?        string
          |      +--ro baud-rate?                string
          |      +--ro bits-per-symbol?          uint32
          |      +--ro num-symbols-in-alphabet?  uint32
          |      +--ro symbols-index?            uint32
          +--rw current-opt-if-och-mode-params
             +--rw mode-id?                             string
             +--rw central-frequency?                   uint32
             +--rw channel-output-power?                int32
             +--ro channel-input-power?                 int32
             +--ro total-input-power?                   int32
             +--rw min-fec-ber-mantissa-threshold?      uint32
             +--rw min-fec-ber-exponent-threshold?      int32
             +--rw max-fec-ber-mantissa-threshold?      uint32
             +--rw max-fec-ber-exponent-threshold?      int32
             +--rw number-of-tcas-supported?            uint32
             +--rw mode-list* [tca-type]
             |  +--rw tca-type        opt-if-och-tca-types
             |  +--rw min-threshold?                    int32
             |  +--rw max-threshold?                    int32
             +--ro cur-osnr?                            int32
             +--ro cur-q-factor?                        int32
             +--ro uncorrected-words?                   uint64
             +--ro pre-fec-ber-mantissa?                uint32
             +--ro pre-fec-ber-exponent?                int32


   notifications:
     +---n opt-if-och-central-frequency-change
     |  +--ro if-name?   -> /if:interfaces/interface/name
     |  +--ro new-opt-if-och-central-frequency
     |     +--ro central-frequency?               uint32
     +---n opt-if-och-mode-change
     |  +--ro if-name?   -> /if:interfaces/interface/name
     |  +--ro mode-id?                            string
     +---n opt-if-och-min-tca
        +--ro if-name?   -> /if:interfaces/interface/name
        +--ro tca-type?   opt-if-och-tca-types
```

5. Structure of the Yang Module

ietf-ext-xponder-wdm-if is a top level model for the support of this feature.

6. Yang Module

The ietf-ext-xponder-wdm-if is defined as an extension to ietf interfaces.

<CODE BEGINS> file "ietf-ext-xponder-wdm-if.yang"

```
module ietf-ext-xponder-wdm-if {
    namespace "urn:ietf:params:xml:ns:yang:ietf-ext-xponder-wdm-if";
    prefix ietf-ext-xponder-wdm-if;

    import ietf-interfaces {
      prefix if;
     }

     organization
       "IETF CCAMP
       Working Group";

     contact
       "WG Web:   <http://tools.ietf.org/wg/ccamp/>
        WG List:  <mailto:ccamp@ietf.org>


        Editor:   Dharini Hiremagalur
                  <mailto:dharinih@juniper.net>";

     description
       "This module contains a collection of YANG definitions for
        configuring Optical interfaces.

        Copyright (c) 2016 IETF Trust and the persons identified
        as authors of the code.  All rights reserved.

        Redistribution and use in source and binary forms, with or
        without modification, is permitted pursuant to, and
        subject to the license terms contained in, the Simplified
        BSD License set forth in Section 4.c of the IETF Trust's
        Legal Provisions Relating to IETF Documents
        (http://trustee.ietf.org/license-info).";

    revision "2019-07-08" {
           description
               "Revision 1.2";
           reference
               "";
```

```
       revision "2018-10-22" {
              description
                  "Revision 1.2";
              reference
                  "";
       }

       revision "2018-03-06" {
              description
                  "Revision 1.1";
              reference
                  "";
       }

       revision "2017-03-06" {
              description
                  "Revision 1.0";
              reference
                  "";
       }
       revision "2016-03-17" {
              description
                  "Initial revision.";
              reference
                  "";
       }



        typedef dbm-t {
          type decimal64 {
            fraction-digits 2;
            range "-50..-30 | -10..5 | 10000000";
          }
          description "
            Amplifier Power in dBm ";
        }
        typedef opt-if-och-tca-types {

            type enumeration {
                enum max-laser-linewdt {
                  description " The maximum laser linewidth";
                }
                enum min-tx-power-tca {
                  description " The min tx power tca";
                }
                enum max-tx-power-tca {
                  description " The min tx power tca";
```

```
                }
                enum min-rx-power-tca{
                  description " The min tx power tca";
                }
                enum max-rx-power-tca{
                  description " The min tx power tca";
                }
                enum max-pol-power-diff-tca{
                  description " The power diff. between polariz. tca";
                }
                enum max-pol-skew-diff-tca{
                  description " The Skew between the two polariz. tca";
                }
                enum min-frequency-offset-tca{
                  description " Min Frequency offset tca";
                }
                enum max-frequency-offset-tca{
                  description " Max Frequency offset tca";
                }
                enum min-osnr-tca{
                  description " Min OSNR tca";
                }
                enum max-osnr-tca{
                  description " Max OSNR tca";
                }
                enum min-laser-temperature-tca{
                  description " The min tx power tca";
                }
                enum max-laser-temperature-tca{
                  description " Temperature tca";
                }
                enum min-fec-ber-tca{
                  description " Min Pre Fec BER tca";
                }
                enum max-fec-ber-tca{
                  description " Max Pre Fec BER tca";
                }
                enum min-q-tca{
                  description "Min Q tca";
                }
                enum max-q-tca {
                  description "Max Q tca";
                }
            }
            description " The different types of TCA's";
        }
```

```
        grouping opt-if-och-power {
           description "Interface optical Power";
           leaf channel-output-power {
               type int32;
               units ".01dbm";
               description "The output power for this interface
                            in .01 dBm.
                            The setting of the output power is
                             optional";
           }

           leaf channel-input-power {
                type int32;
                units ".01dbm";
                config false;
                description "The current channel input power of this
                      interface";
           }
           leaf total-input-power {
                type int32;
                units ".01dbm";
                config false;
                description "The total input power of this
                      interface";
           }
        }

        grouping opt-if-och-tca-thresholds {
             description "Thresholds for TCA's";
             leaf  tca-type {
                type opt-if-och-tca-types;
                description  "type of the TCA eg TX Power";
             }
             leaf  min-threshold {
               type int32;
               description " A TCA is generated if the variable is
                            less than this value";

              }
              leaf  max-threshold {
               type int32;
               description " A TCA is generated if the variable is
                            more than this value";
              }
         }

         grouping opt-if-och-fec {
             description "Fec info";
```

```
             leaf fec-info {
                type string {
                   length "1..255";
                }
                config false;
                description
                         "Fec Type - eg GFEC";
             }
             leaf fec-bitrate {
                 type string {
                     length "1..255";
                 }
                 config false;
                 description
                         "Fec Overhead rate ";
             }
             leaf fec-gain {
                type string {
                    length "1..255";
                }
                config false;
                description
                         "Fec Overhead rate ";
              }
              leaf  pre-fec-ber-mantissa-threshold {
                type uint32;
                description " Mantissa of the FEC BER threshold";

               }
               leaf  pre-fec-ber-exponent-threshold {
                 type int32;
                 description " Exponent of the FEC BER threshold";
               }
         }

         grouping opt-if-och-central-frequency {
             description "Interface Central Frequency";
             leaf  central-frequency {
               type uint32;
               description " This parameter indicates the frequency
                             of this interface ";
              }
         }


         }

         grouping opt-if-och-modulation-params {
```

```
            description "Optical modulation parameters for the lane";
            leaf modulation-format {
               type string {
                  length "1..255";
               }
               config false;
               description
                        "Modulation format for this mode";
            }
            leaf baud-rate {
               type uinte32
               description
                        "Baud-rate or symbol rate";
            }
            leaf  bits-per-symbol {
              type uint32;
              description " This parameter the bits per symbol for
                           this mode.";

            }
            leaf  num-symbols-in-alphabet {
              type uint32;
              description " This parameter the bits per symbol for
                           this mode.";

             }
             leaf  symbols-index {
              type uint32;
              description " This parameter is the symbol index this
                           mode.";

             }


        grouping opt-if-och-lane-param {
            description "Optical parameters for the lane";
            leaf number-of-lanes {
               type  uint32;
               config false;
               description
                      "Number of optical lanes of this interface";
            }
            leaf min-laser-temperature {
               type  int32;
               units ".01C";
               config false;
               description
                      "Minimum Laser Temperature of this mode for
```

```
                    this interface";
            }
            leaf max-laser-temperature {
                type  int32;
                units ".01C";
                config false;
                description
                       "Maximum Laser Temperature of this mode for
                          this interface";
            }
            leaf max-total-rx-optical-power {
                type dbm-t;
                config false;
                description
                        "Maximum rx optical power of this mode for
                          this interface";
            }
            leaf max-chromatic-dispersion {
                type int32;
                config false;
                description
                         "Maximum chromatic dispersion of this
                           mode for this interface";
            }
            leaf max-diff-group-delay  {
                type int32;
                config false;
                description
                         "Maximum Differential group delay of this
                           mode for this interface";
            }
             uses opt-if-och-modulation-params;
          }



         grouping opt-if-och-tca-list {
           description "List of TCA's.";
           leaf number-of-tcas-supported {
               type uint32;
               description "Number of tcas
                            supported by this interface";
            }
           list mode-list {
               key "tca-type";
               description "List of the tcas";
               uses opt-if-och-tca-thresholds;
           }
```

```
        }

        grouping opt-if-och-fec-tca-thresholds {
            description "Pre FEC BER Thresholds for TCA's";
            leaf  min-fec-ber-mantissa-threshold {
              type uint32;
              description " Min Mantissa of the FEC BER threshold";

             }
             leaf  min-fec-ber-exponent-threshold {
              type int32;
              description " Min Exponent of the FEC BER threshold";

             }
             leaf  max-fec-ber-mantissa-threshold {
              type uint32;
              description " Max Mantissa of the FEC BER threshold";

             }
             leaf  max-fec-ber-exponent-threshold {
              type int32;
              description " Max Exponent of the FEC BER threshold";

             }
        }


        grouping opt-if-och-mode-params  {
            description "OCh mode parameters.";
            leaf mode-id {
                  type string {
                        length "1..255";
                  }
                  description
                        "Id for the OCh mode template";
            }
            leaf  min-osnr-margin {
              type int32;
              units "dB";
              config false;
              description " OSNR margin to FEC threshold";
             }
             leaf  q-margin {
              type int32;
              units "dB";
              config false;
              description " Q-Factor margin to FEC threshold";
             }
```

```
            uses opt-if-och-central-frequency;
            uses opt-if-och-power;
            uses opt-if-och-fec-tca-thresholds;
            uses opt-if-och-tca-list;

        }

        grouping opt-if-och-statistics {
            description "OCh statistics.";
            leaf  cur-osnr {
              type int32;
              units "dB";
              config false;
              description " OSNR margin to FEC threshold";
             }
             leaf  cur-q-factor {
              type int32;
              units "dB";
              config false;
              description " Q-Factor of the interface";
             }
             leaf  uncorrected-words {
              type uint64;
              config false;
              description " Post FEC errored words";
             }
             leaf  pre-fec-ber-mantissa {
              type uint32;
              config false;
              description " Pre fec FEC errored words mantissa";
             }
             leaf  pre-fec-ber-exponent {
              type int32;
              config false;
              description " Pre fec FEC errored words exponent";
             }
        }

        grouping opt-if-och-mode  {
                description "OCh mode template.";
                leaf mode-id {
                  type string {
                        length "1..255";
                  }
                  config false;
                  description
                        "Id for the OCh mode template";
                 }
```

```
                leaf application-identifier {
                    type uint32;
                    config false;
                    description "This parameter indicates the
                                 application identifier according
                                 toG.698.2";
                }
                leaf min-central-frequency {
                    type uint32;
                    config false;
                    description "This parameter indicates the minimum
                                  frequency for this template  ";
                 }
                 leaf max-central-frequency {
                    type uint32;
                    config false;
                    description "This parameter indicates the minimum
                                  frequency for this template  ";
                 }
                 leaf min-channel-input-power {
                    type dbm-t;
                    config false;
                    description "The minimum input power of this
                                 interface";
                 }
                 leaf max-channel-input-power {
                    type dbm-t;
                    config false;
                    description "The maximum input power of this
                                 interface";
                 }
                 leaf min-channel-output-power {
                    type dbm-t;
                    config false;
                    description "The minimum output power of this
                                 interface";
                 }
                 leaf max-channel-output-power {
                    type dbm-t;
                    config false;
                    description "The maximum output power of this
                                 interface";
                 }
                 leaf  osnr-margin {
                    type int32;
                    units "dB";
                    config false;
                    description "OSNR margin to FEC threshold";
```

```
                }
                leaf  q-margin {
                   type int32;
                   units "dB";
                   config false;
                   description "Q-Factor margin to FEC threshold";
                }
                uses opt-if-och-fec;
                uses opt-if-och-lane-param;

         }



        grouping opt-if-och-mode-list {
           description "List of Mode list group.";
           leaf number-of-modes-supported {
               type uint32;
               description "Number of modes
                            supported by this interface";
            }
           list mode-list {
               key "mode-id";
               description "List of the modes ";
               uses opt-if-och-mode;
           }
         }


         notification opt-if-och-central-frequency-change {
            description "A change of Central Frequency has been
                         detected.";
            leaf "if-name" {
                type leafref {
                    path "/if:interfaces/if:interface/if:name";
                }
                description "Interface name";
            }
            container new-opt-if-och-central-frequency {
            description "The new Central Frequency of the
                         interface";
            uses opt-if-och-central-frequency;
            }
         }

         notification opt-if-och-mode-change {
            description "A change of Mode Template has been
```

```
                    detected.";
            leaf "if-name" {
                type leafref {
                    path "/if:interfaces/if:interface/if:name";
                }
                description "Interface name";
            }
            leaf mode-id {
                type string {
                    length "1..255";
                }
                description  "Id for the OCh mode template";
            }

        }

        notification opt-if-och-min-tca {
           description "A min output TCA notification.";
           leaf "if-name" {
               type leafref {
                   path "/if:interfaces/if:interface/if:name";
               }
               description "Interface name";
           }
           leaf tca-type {
               type opt-if-och-tca-types;
               description "Type of TCA for eg min tx power TCA";
           }
        }

        augment "/if:interfaces/if:interface" {
          description "Parameters for an optical interface";
          container optIfOChRsSs {
             description "RsSs path configuration for an interface";
             container if-current-mode {
                  description "Current mode template of the
                               interface";
                  uses opt-if-och-mode;
              }

              container if-supported-mode {
                  config false;
                  description "Supported mode list of
                               this interface";
                  uses opt-if-och-mode-list;
              }
              container current-opt-if-och-mode-params {
                  description "Current parameters of
```

```
                                this interface";
                    uses opt-if-och-mode-params;
                    uses opt-if-och-statistics;
                }


            }
        }
    }
```

<CODE ENDS>

7. Security Considerations

The YANG module defined in this memo is designed to be accessed via the NETCONF protocol [RFC6241]. he lowest NETCONF layer is the secure transport layer and the mandatory-to-implement secure transport is SSH [RFC6242]. The NETCONF access control model [RFC6536] provides the means to restrict access for particular NETCONF users to a pre-configured subset of all available NETCONF protocol operation and content.

8. IANA Considerations

This document registers a URI in the IETF XML registry [RFC3688]. Following the format in [RFC3688], the following registration is requested to be made:

URI: urn:ietf:params:xml:ns:yang:ietf-interfaces:ietf-ext-xponder-wdm-if

Registrant Contact: The IESG.

XML: N/A, the requested URI is an XML namespace.

This document registers a YANG module in the YANG Module Names registry [RFC6020].

This document registers a YANG module in the YANG Module Names registry [RFC6020].

prefix: ietf-ext-xponder-wdm-if reference: RFC XXXX

9. Acknowledgements

10. Contributors

Dean Bogdanovic
Westford
U.S.A.
email ivandean@gmail.com

Bernd Zeuner
Deutsche Telekom
Darmstadt
Germany
email B.Zeuner@telekom.de

Massimiliano Salsi
Juniper Networks
1133 Innovation Way
Sunnyvale, CA, 94089
U.S.A.
+1 408936847
email msalsi@google.com

Arnold Mattheus
Deutsche Telekom
Darmstadt
Germany
email a.mattheus@telekom.de

Manuel Paul
Deutsche Telekom
Berlin
Germany
email Manuel.Paul@telekom.de

Walid Wakim
Cisco
9501 Technology Blvd
ROSEMONT, ILLINOIS 60018
UNITED STATES
email wwakim@cisco.com

Kam Lam
Nokia
USA
+1 732 331 3476
kam.lam@nokia.com

11. References

11.1. Normative References

[ITU.G694.1]
International Telecommunications Union, "Spectral grids for WDM applications: DWDM frequency grid", ITU-T Recommendation G.694.1, February 2012.

[ITU.G698.2]
International Telecommunications Union, "Amplified multichannel dense wavelength division multiplexing applications with single channel optical interfaces", ITU-T Recommendation G.698.2, November 2009.

[ITU.G709]
International Telecommunications Union, "Interface for the Optical Transport Network (OTN)", ITU-T Recommendation G.709, June 2016.

[ITU.G7710]
International Telecommunications Union, "Common equipment management function requirements", ITU-T Recommendation G.7710, August 2017.

[ITU.G798]
International Telecommunications Union, "Characteristics of optical transport network hierarchy equipment functional blocks", ITU-T Recommendation G.798, Devember 2017.

[ITU.G8201]
International Telecommunications Union, "Error performance parameters and objectives for multi-operator international paths within the Optical Transport Network (OTN)", ITU-T Recommendation G.8201, April 2011.

[ITU.G826]
International Telecommunications Union, "End-to-end error performance parameters and objectives for international, constant bit-rate digital paths and connections", ITU-T Recommendation G.826, December 2002.

[ITU.G872]
International Telecommunications Union, "Architecture of optical transport networks", ITU-T Recommendation G.872, January 2017.

[ITU.G874]
International Telecommunications Union, "Management aspects of optical transport network elements", ITU-T Recommendation G.874, July 2010.

[ITU.G874.1]
International Telecommunications Union, "Optical transport network (OTN): Protocol-neutral management information model for the network element view", ITU-T Recommendation G.874.1, November 2016.

[ITU.G959.1]
International Telecommunications Union, "Optical transport network physical layer interfaces", ITU-T Recommendation G.959.1, November 2009.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/info/rfc2119>.

[RFC2578] McCloghrie, K., Ed., Perkins, D., Ed., and J. Schoenwaelder, Ed., "Structure of Management Information Version 2 (SMIv2)", STD 58, RFC 2578, DOI 10.17487/RFC2578, April 1999, <https://www.rfc-editor.org/info/rfc2578>.

[RFC2579] McCloghrie, K., Ed., Perkins, D., Ed., and J. Schoenwaelder, Ed., "Textual Conventions for SMIv2", STD 58, RFC 2579, DOI 10.17487/RFC2579, April 1999, <https://www.rfc-editor.org/info/rfc2579>.

[RFC2580] McCloghrie, K., Ed., Perkins, D., Ed., and J. Schoenwaelder, Ed., "Conformance Statements for SMIv2", STD 58, RFC 2580, DOI 10.17487/RFC2580, April 1999, <https://www.rfc-editor.org/info/rfc2580>.

[RFC2863] McCloghrie, K. and F. Kastenholz, "The Interfaces Group MIB", RFC 2863, DOI 10.17487/RFC2863, June 2000, <https://www.rfc-editor.org/info/rfc2863>.

[RFC3591] Lam, H-K., Stewart, M., and A. Huynh, "Definitions of Managed Objects for the Optical Interface Type", RFC 3591, DOI 10.17487/RFC3591, September 2003, <https://www.rfc-editor.org/info/rfc3591>.

[RFC6205] Otani, T., Ed. and D. Li, Ed., "Generalized Labels for Lambda-Switch-Capable (LSC) Label Switching Routers", RFC 6205, DOI 10.17487/RFC6205, March 2011, <https://www.rfc-editor.org/info/rfc6205>.

11.2. Informative References

[I-D.ietf-ccamp-dwdm-if-mng-ctrl-fwk] Kunze, R., Grammel, G., Beller, D., Galimberti, G., and J. Meuric, "A framework for Management and Control of DWDM optical interface parameters", draft-ietf-ccamp-dwdm-if-mng-ctrl-fwk-11 (work in progress), June 2018.

[RFC2629] Rose, M., "Writing I-Ds and RFCs using XML", RFC 2629, DOI 10.17487/RFC2629, June 1999, <https://www.rfc-editor.org/info/rfc2629>.

[RFC3410] Case, J., Mundy, R., Partain, D., and B. Stewart, "Introduction and Applicability Statements for Internet-Standard Management Framework", RFC 3410, DOI 10.17487/RFC3410, December 2002, <https://www.rfc-editor.org/info/rfc3410>.

[RFC4054] Strand, J., Ed. and A. Chiu, Ed., "Impairments and Other Constraints on Optical Layer Routing", RFC 4054, DOI 10.17487/RFC4054, May 2005, <https://www.rfc-editor.org/info/rfc4054>.

[RFC4181] Heard, C., Ed., "Guidelines for Authors and Reviewers of MIB Documents", BCP 111, RFC 4181, DOI 10.17487/RFC4181, September 2005, <https://www.rfc-editor.org/info/rfc4181>.

Appendix A. Change Log

This optional section should be removed before the internet draft is submitted to the IESG for publication as an RFC.

Note to RFC Editor: please remove this appendix before publication as an RFC.

Appendix B. Open Issues

Note to RFC Editor: please remove this appendix before publication as an RFC.

Authors' Addresses

Gabriele Galimberti (editor)
Cisco
Via Santa Maria Molgora, 48 c
20871 - Vimercate
Italy

Phone: +390392091462
Email: ggalimbe@cisco.com

Ruediger Kunze
Deutsche Telekom
Winterfeldtstr. 21-27
10781 Berlin
Germany

Phone: +491702275321
Email: RKunze@telekom.de

Dharini Hiremagalur (editor)
Juniper
1133 Innovation Way
Sunnyvale - 94089 California
USA

Email: dharinih@juniper.net

Gert Grammel (editor)
Juniper
Oskar-Schlemmer Str. 15
80807 Muenchen
Germany

Phone: +49 1725186386
Email: ggrammel@juniper.net