Network Working Group

Y. Lee
D. Dhody
Huawei

Internet Draft
Intended status: Standard Track
Expires: June 2015

March 5, 2015

A Yang Data Model for WSON Optical Networks

draft-lee-ccamp-wson-yang-00

Abstract

This document provides a YANG data model for the routing and wavelength assignment (RWA) process in wavelength switched optical networks (WSONs).

Status of this Memo

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

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

The list of current Internet-Drafts can be accessed at http://www.ietf.org/ietf/1id-abstracts.txt

The list of Internet-Draft Shadow Directories can be accessed at http://www.ietf.org/shadow.html

This Internet-Draft will expire on June 5, 2015.

Copyright Notice

Table of Contents

1. Introduction.....2
2. Routing and Wavelength Assignment Informational Model.....3
2.1. Connectivity Matrix Model.....3
2.2. Resource Pool Model.....4
2.3. Port Label Restriction Model.....8
2.4. Wavelength Availability on Links.....9
3. YANG Model.....10
4. WSON-RWA YANG Model.....11
5. Security Considerations.....17
6. IANA Considerations.....17
7. Acknowledgments.....17
8. References.....18
8.1. Normative References.....18
8.2. Informative References.....18
9. Contributors.....18
Authors' Addresses.....18

1. Introduction

This document provides a YANG data model for the routing and wavelength assignment (RWA) process in wavelength switched optical networks (WSONs). The YANG model described in this document is a WSON technology-specific Yang model based on the information model developed in [RFC7446] and the two encoding drafts [WSON-Encode] and [Gen-Encode] that developed protocol independent encodings based on [RFC7446].

What is not in scope of this document is both impairment-aware WSON and flex-grid.

2. Routing and Wavelength Assignment Informational Model

The relevant information model in this document comprises

- Connectivity Matrix Model (Section 2.1)
- Resource Pool Model (Section 2.2)
- Port Wavelength Restriction (Section 2.3)
- Wavelength Availability on Links (Section 2.4)

[Editor's Note: This version covers the corresponding YANG data model for the first two sections (Sections 2.1 and 2.2) and leaves the YANG model for Sections 2.3 and 2.4 in the later version.]

Sections 2.1 - 2.4 rehashes key information models from [RFC7446] to facilitate the development of the YANG model (Section 3).

2.1. Connectivity Matrix Model

The connectivity matrix (ConnectivityMatrix) represents either the potential connectivity matrix for asymmetric switches (e.g. ROADMs and such) or fixed connectivity for an asymmetric device such as a multiplexer.

Note that multiple connectivity matrices are allowed and the Node_ID would be an appropriate identifier for the node to point the Connectivity matrix within the WSON RWA context.

<Node_Information> ::= <Node_ID> [<ConnectivityMatrix>...]

<ConnectivityMatrix> ::= <MatrixID>

<ConnType>

<Matrix>

Where

<MatrixID> is a unique identifier for the matrix.

<ConnType> can be either 0 or 1 depending upon whether the connectivity is either fixed or switched.

<Matrix> represents the fixed or switched connectivity in that Matrix(i, j) = 0 or 1 depending on whether input port i can connect to output port j for one or more wavelengths.

2.2. Resource Pool Model

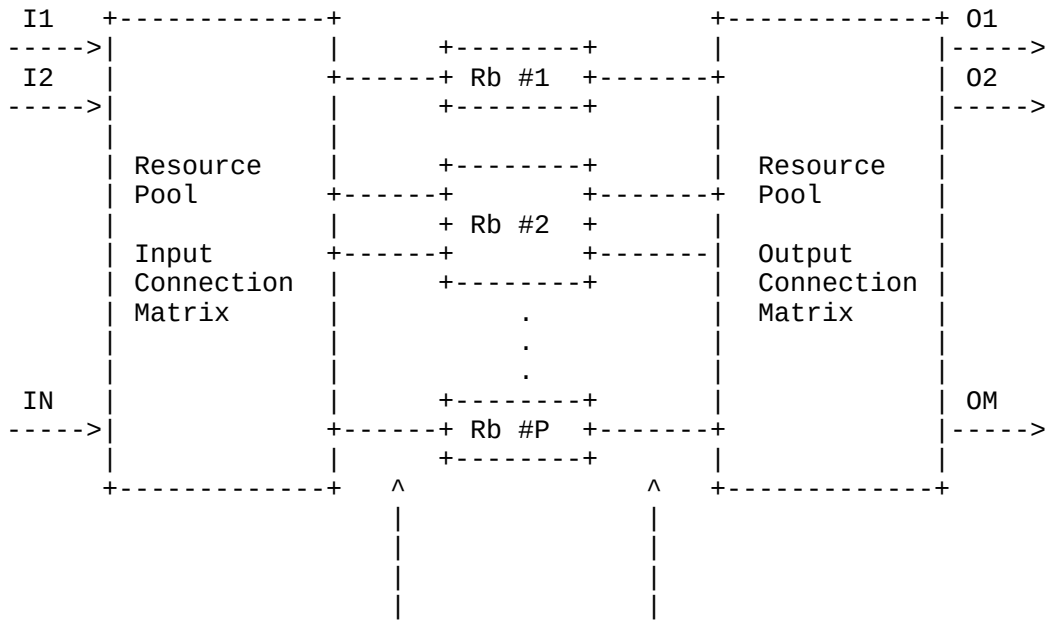
A WSON node may include regenerators or wavelength converters arranged in a shared pool. As discussed in [RFC6163] this can include Optical-Electronic-Optical (OEO) based Wavelength Division Multiplexing (WDM) switches as well. There are a number of different approaches used in the design of WDM switches containing regenerator or converter pools. However, from the point of view of path computation the following need to be known:

1. The nodes that support regeneration or wavelength conversion.

2. The accessibility and availability of a wavelength converter to convert from a given input wavelength on a particular input port to a desired output wavelength on a particular output port.

3. Limitations on the types of signals that can be converted and the conversions that can be performed.

The following Figures show resource pool architecture of WSON.



Note: Rb is a Resource Block.

Figure 1 Schematic diagram of resource pool model.

Since resources tend to be packaged together in blocks of similar devices, e.g., on line cards or other types of modules, the fundamental unit of identifiable resource in this document is the "resource block". A resource block may contain one or more resources. A resource is the smallest identifiable unit of processing allocation. One can group together resources into blocks if they have similar characteristics relevant to the optical system being modeled, e.g., processing properties, accessibility, etc.

This leads to the following formal high level model:

<Node_Information> ::= <Node_ID>

[<ConnectivityMatrix>...]

[<ResourcePool>]

Where

<ResourcePool> ::= <ResourceBlockInfo>...

[<ResourceAccessibility>...]

[<ResourceWaveConstraints>...]

[<RBPoolState>]

<ResourceAccessibility> ::= <PoolInputMatrix>

<PoolOutputMatrix>

<ResourceWaveConstraints> ::= <InputWaveConstraints>

<OutputOutputWaveConstraints>

<RBSharedAccessWaveAvailability> ::= [<InAvailableWavelengths>]

[<OutAvailableWavelengths>]

<RBPoolState> ::= <ResourceBlockID>

<NumResourcesInUse>

[<RBSharedAccessWaveAvailability>]

[<RBPoolState>]

<ResourceBlockInfo> ::= <ResourceBlockSet>

[<InputConstraints>]

[<ProcessingCapabilities>]

[<OutputConstraints>]

Where <ResourceBlockSet> is a list of resource block identifiers with the same characteristics. If this set is missing the constraints are applied to the entire network element.

<InputConstraints> ::= <SharedInput>

[<OpticalInterfaceClassList>]

[<ClientSignalList>]

<ProcessingCapabilities> ::= [<NumResources>]

[<RegenerationCapabilities>]

[<FaultPerfMon>]

[<VendorSpecific>]

<OutputConstraints> := <SharedOutput>

[<OpticalInterfaceClassList>]

[<ClientSignalList>]

<OpticalInterfaceClassList> ::= <OpticalInterfaceClass> ...

<ClientSignalList>::=[<G-PID>]...

1. Number of Resources within the block
2. Regeneration capability
3. Fault and performance monitoring
4. Vendor Specific capability

Note that the code points for Fault and performance monitoring and vendor specific capability are subject to further study.

2.3. Port Label Restriction Model

```
<LinkInfo> ::=  <LinkID>

                [<AdministrativeGroup>]

                [<InterfaceCapDesc>]

                [<Protection>]

                [<SRLG>...]

                [<TrafficEngineeringMetric>]

                [<PortLabelRestriction>...]
```

Note that these additional link characteristics only applies to line side ports of WDM system or add/drop ports pertaining to Resource Pool (e.g., Regenerator or Wavelength Converter Pool). The advertisement of input/output tributary ports is not intended here.

```
<PortLabelRestriction> ::= <MatrixID>

                           <Restriction parameters list>

<Restriction parameters list> ::=

                           <LabelSet> ...
```

Where

MatrixID is the ID of the corresponding connectivity matrix.

LabelSet is a conceptual set of labels (wavelengths).

MaxNumChannels is the maximum number of channels that can be simultaneously used (relative to either a port or a matrix).

LinkSet is a conceptual set of ports.

2.4. Wavelength Availability on Links

In the previously presented information model there are a limited number of information elements that are dynamic, i.e., subject to change with subsequent establishment and teardown of connections. Depending on the protocol used to convey this overall information model it may be possible to send this dynamic information separate from the relatively larger amount of static information needed to characterize WSON's and their network elements.

<DynamicLinkInfo> ::= <LinkID>

<AvailableLabels>

[<SharedBackupLabels>]

AvailableLabels is a set of labels (wavelengths) currently available on the link. Given this information and the port wavelength restrictions one can also determine which wavelengths are currently in use. This parameter could potential be used with other technologies that GMPLS currently covers or may cover in the future.

SharedBackupLabels is a set of labels (wavelengths) currently used for shared backup protection on the link. An example usage of this information in a WSON setting is given in [Shared]. This parameter could potential be used with other technologies that GMPLS currently covers or may cover in the future.

3. YANG Model (Tree Structure)

[Editor's Note: This version did not make use of augmentation of other modules. The augmentation of other modules will be considered once other modules being developed have been well established and can be used as a basis of this module.]

```
module: wson-topology
   +--rw wson-topology
      +--rw wson-topology* [wson-topology-id]
         +--rw wson-topology-id    wson-topology-id
         +--rw name?               string
         +--rw wson-node* [wson-node-id]
            +--rw wson-node-id                  wson-node-id
            +--rw wson-interface* [wson-interface-id]
            |  +--rw wson-interface-id    linkset-format
            |  +--rw wavelength-available-bitmap*   uint32
            +--rw connectivity-matrix* [matrix-id]
            |  +--rw matrix-id           uint8
            |  +--rw device-type?        devicetype
            |  +--rw dir?                directionality
            |  +--rw format              linkset-format
            |  +--rw matrix-interface* [in-port-id]
            |     +--rw in-port-id     wson-interface-ref
            |     +--rw out-port-id    wson-interface-ref
            +--rw resource-pool* [resource-pool-id]
               +--rw resource-pool-id    uint32
               +--rw pool-state          boolean
               +--rw matrix-interface* [in-port-id]
                  +--rw in-port-id     wson-interface-ref
                  +--rw out-port-id    wson-interface-ref
```

4. WSON-RWA YANG Model

<CODE BEGINS>

```
module   wson-topology {
   namespace "urn:ietf:params:xml:ns:yang:wson-topology";

   prefix wson;

   import ietf-inet-types {
      prefix inet;
   }

   organization
        "IETF CCAMP Working Group";

   contact
        "Editor:   Young Lee  <leeyoung@huawei.com>";

   description
        "This module contains a collection of YANG definitions for
        RWA WSON.

        Copyright (c) 2015 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 2015-03-05 {
        description
            "Initial revision.";
```

```
          reference
              "RFC XXX: A Yang Data Model for WSON Optical Networks ";
      }

      typedef wson-topology-id {
         type inet:uri;
         description
            "The WSON Topology ID";

      }

      typedef wson-node-id {
         type inet:ip-address;
         description
            "The WSON Node ID";
      }

      typedef devicetype {
         type enumeration {
            enum adm {
               value 1;

               description
                  "Device is ADM";
            }
            enum roadm {
               value 2;
            description
                  "Device is ROAMD/OXC";
            }
         }
         description
            "device type: fixed (ADM) or switched (ROADM/OXC)";
      }

      typedef directionality {
         type enumeration {
            enum bidir {
               value 0;
               description
                  "bi-directional";
```

```
            }
            enum input {
               value 1;

               description
                  "input direction";
            }
            enum output {
               value 2;
               description
                  "output direction";
            }
         }
         description
            "The directionality of link set";
      }

      typedef linkset-format {
         type enumeration {
            enum link-local-identifier{
               value 1;
               description
                  "";
            }
            enum local-interface-ipv4{
               value 2;
               description
                  "";
            }
            enum local-interface-ipv6{
               value 3;
               description
                  "";
            }
         }
         description
            "linkset type; link local/ipv4/ipv6";
      }

      typedef wson-interface-ref {
         type leafref {
```

```
            path "/wson-topology/wson-topology/wson-node/" +
                 "wson-interface/wson-interface-id";
         }
         description
         "This type is used by data models that need to
         reference WSON interface.";
      }

      container wson-topology {
         description
            "TBD";
         list wson-topology {
            key "wson-topology-id";
            description
               "The WSON Topology";
            leaf wson-topology-id {
               type wson-topology-id;
               description
                  "The WSON Topology Identifier";
            }
            leaf name {
               type string;
               description
                  "TBD";
            }

            list wson-node {
               key "wson-node-id";
               description
                  "The WSON node";
               leaf wson-node-id {
                  type wson-node-id;
                  description
                     "The WSON Node ID";
               }

               list wson-interface {
                  key "wson-interface-id";
                  description
                     "The list of WSON Interface";
                  leaf wson-interface-id {
```

```
                     type linkset-format;
                     description
                        "TBD";
                  }

                  leaf-list wavelength-available-bitmap {
                     type uint32;
                     description
                       "The list of available channels, correponding
                         to the bitmap in the info model.";
                  }
               }

               list connectivity-matrix {
                  key "matrix-id";
                  description
                     "connectivity-matrix of WSON node";
                  reference
                     "based on draft-ietf-ccamp-general-constraint-
   encode";
                  leaf matrix-id {
                     type uint8;
                     description
                        "matrix identifier";
                  }

                  leaf device-type {
                  type devicetype;
                  description
                        "device type: fixed (ADM) or switched
                        (ROADM/OXC)";
                  }
                  leaf dir {
                     type directionality;
                     description
                        "bi-directionality or input or output
                        of link set";
                  }
                  leaf format {
                     type linkset-format;
                     description
```

```
                        "format of identifier";
                    }
                    list matrix-interface {
                       key "in-port-id";

                       description
                          "matrix-interface describes input-ports
                          and out-ports around a connectivity
                          matrix";

                       leaf in-port-id {
                          type wson-interface-ref;
                          description
                             "The reference to in-port";
                       }

                       leaf out-port-id {
                          type wson-interface-ref;
                          description
                             "The reference to out-port";
                       }
                    }
                 }
                 list resource-pool {
                    key "resource-pool-id";
                    description
                       "The resource pool list";

                    leaf resource-pool-id {
                       type uint32;
                       description
                          "The resource pool ID";
                    }
                    leaf pool-state {
                       type boolean;
                       description
                          "TRUE is state UP; FALSE is state down";
                    }
                    list matrix-interface {
                       key "in-port-id";
                       description
```

```
                        "pool is described as matrix-interface
                        with input-ports and output-ports
                        around the pool";
                     leaf in-port-id {
                        type wson-interface-ref;
                        description
                           "The reference to in-interface";

                     }
                     leaf out-port-id {
                        type wson-interface-ref;
                        description
                           "The reference to out-interface";
                     }
                  }
               }
            }
         }
      }
   }
   <CODE ENDS>
```

5. Security Considerations

TDB

6. IANA Considerations

TDB

7. Acknowledgments

This document was prepared using 2-Word-v2.0.template.dot.

8. References

8.1. Normative References

[RFC7446] Y. Lee, G. Bernstein, D. Li, W. Imajuku, "Routing and Wavelength Assignment Information Model for Wavelength Switched Optical Networks", RFC 7446, Feburary 2015.

[Gen-Encode] G. Bernstein, Y. Lee, D. Li, W. Imajuku, "General Network Element Constraint Encoding for GMPLS Controlled Networks", work in progress: draft-ietf-ccamp-general-constraint-encode.

[WSON-Encode] G. Bernstein, Y. Lee, D. Li, W. Imajuku, "Routing and Wavelength Assignment Information Encoding for Wavelength Switched Optical Networks", work in progress: draft-ietf-ccamp-rwa-wson-encode.

8.2. Informative References

9. Contributors

Xian Zhang
Huawei Technologies
Email: zhang.xian@huawei.com

Authors' Addresses

Young Lee (ed.)
Huawei Technologies
5340 Legacy Drive, Building 3
Plano, TX 75023
USA

Phone: (469) 277-5838
Email: leeyoung@huawei.com

Dhruv Dhody
Huawei Technologies India Pvt. Ltd,

Near EPIP Industrial Area, Kundalahalli Village, Whitefield,
Bangalore - 560 037 [H1-2A-245]

Email: dhruv.dhody@huawei.com