Internet Engineering Task Force
Internet-Draft
Intended status: Standards Track
Expires: December 12, 2011

P. Peloso, Ed.
Alcatel-Lucent
G. Martinelli
Cisco
J. Meuric
France Telecom
C. Margaria
Nokia Siemens Networks

June 10, 2011

OSPF-TE Extensions for WSON-specific Network Element Constraints
draft-peloso-ccamp-wson-ospf-oeo-03

Abstract

The original content of this internet draft was to propose some extentions to OSPF encoding in the context of Wavelength Switched Optical Networks, especially for internal constraints of optical network elements. General description can be found in the framework document.

This update of the document still aims at specifying the detailled structure of OSPF LSAs for WSONs. Nevertheless, the proposed LSA layout slightly differs from the current content of the information model and encodings drafts. As a result, the following sections hilglight the differences between both approaches and summarize why the authors think these CCAMP's drafts would benefit from an update according to the proposed description.

Status of this Memo

Copyright Notice

Table of Contents

1. Introduction

The original content of this internet draft was to propose some extentions to OSPF encoding in the context of Wavelength Switched Optical Networks, especially for internal constraints of optical network elements. General description can be found in the framework document [RFC6163].

This update of the document still aims at specifying the detailled structure of OSPF LSAs for WSONs. Nevertheless, the proposed LSA layout slightly differs from the current content of the information model [I-D.ietf-ccamp-rwa-info] and encodings [I-D.ietf-ccamp-rwa-wson-encode] drafts. As a result, the following sections highlight the differences between both approaches and summarize why the authors think these CCAMP’s drafts would benefit from an update according to the proposed description.

More specifically, the sections below follow the scope of current documents, namely information model, encodings and OSPF-TE extensions. Building the latter allowed to identify some improvements which are described in the two former parts. In both, the line has been drawn between the optical information that can be specified by using generic protocol extensions and the one requiring some WSON-specific objects, as agreed by the working group.

1.1. Requirements Language

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 [RFC2119].

2. Information Model

This section provides a model of information needed by the routing and wavelength assignment (RWA) process in wavelength switched optical networks (WSONs). The purpose of the information described in this model is to facilitate constrained optical path computation in WSONs. This model takes into account compatibility constraints between WSON signal attributes and network elements but does not include constraints due to optical impairments.

It reports every WSON information model modification compared to [I-D.ietf-ccamp-rwa-info]. Alike this document, this section is organized in order to describe RWA specificities related to nodes and related to links. The nodes specificities can once again be splitted between generic and WSON specific needs of description, while the links specificities can all fit inside the generic needs.

In the following, the reduced Backus-Naur form (RBNF) syntax of [RFC5511] is used to aid in defining the RWA information model.

2.1. Node Information (General)

The node information described here contains the relatively static information related to a WSON node. This includes connectivity constraints amongst ports and wavelengths since WSON switches can exhibit asymmetric switching properties. These connectivity matrices are included with the node information while the switched and fixed port wavelength constraints are included with the link information. Formally,

```
<Node_Information> ::= <Node_ID> [<ConnectivityMatrix>...]
    [<Port_Label_Restrictions>...][<SharedRiskNodeGroup>]
```

Where the Node_ID would be an appropriate identifier for the node within the WSON RWA context. In order to describe complex node structures, multiple connectivity matrices may be used to describe all the constraints.

2.1.1. Connectivity Matrix

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 this matrix does not represent any particular internal blocking behavior but indicates which ingress ports and wavelengths could possibly be connected to a particular output port.

The connectivity matrix is a conceptual M by N matrix representing the potential switched or fixed connectivity, where M represents the number of ingress ports and N the number of egress ports. This is a "conceptual" matrix since the matrix tends to exhibit structure that allows for very compact representations that are useful for both transmission and path computation.

Note that the connectivity matrix information element can be useful in any technology context where asymmetric switches are utilized.

```
<ConnectivityMatrix> ::= <MatrixID> <ConnectType> <Matrix>
```

Where

MatrixID   is a unique identifier for the matrix.

ConnectType can be either 0 or 1 depending upon whether the connectivity is either fixed or potentially switched.

Matrix represents the fixed or switched connectivity in that Matrix(i, j) = 0 or 1 depending on whether ingress port i can connect to egress port j for one or more wavelengths. More explicitly implemented as,
<Matrix> ::= (<IngressLinkSet> <EgressLinkSet>)...
which is a list of pairs of sets of links identifiers, where the purpose of each pair is to define an association between ingress links and egress links, and for the sake of compactness, this is done with sets of links.

Hence, the RBNF can also be described as:

```
<ConnectivityMatrix> ::= <MatrixID> <ConnectType>
    (<IngressLinkSet> <EgressLinkSet>)...
```

DELTA:
No technical change, explicit detail of the content of <Matrix>

2.1.2. Port Label Restrictions

The port label restriction (PortLabelRestriction) represents the label restrictions associated either to links or to connectivity matrices. They can either model:

1. The restrictions coming from various equipments composing the internal structure of a node (such as mux and demuxes). These restrictions tell us which label may or may not be used between ports,

2. The restrictions coming from the internal structure of the link (such as amplifier which may have a limited amplification spectrum),

and as such are relatively static.

This plays an important role in fully characterizing a blocking switching device (e.g. a blocking WSON ROXC or ROADM), hence the port label restrictions are directly associated to a a given connectivity matrix.

```
<PortLabelRestriction> ::= <MatrixID> <RestrictionType>
    [<RestrictionParameters>]
```

DELTA:
No change.

2.1.3. Shared Risk Node Group

SRNG (Shared risk group for nodes) is defined after the concept of a shared risk link group ([RFC4202]) transposed to the grouping of nodes. A set of nodes may constitute a 'shared risk node group' (SRNG) if they share a resource whose failure may affect all nodes in the set. (This is explained in [G.7715]. Typical risk groupings for nodes can include those nodes in the same building, within the same city, or geographic region).

A node may belong to multiple SRNGs. Thus the SRNG Information describes a list of SRNGs that the node belongs to. An SRNG is identified by a 32 bit number that is unique within an IGP domain. The SRNG Information is an unordered list of SRNGs that the node belongs to.

If an LSR is required to have multiple diversely routed LSPs to another LSR, the path computation should attempt to route the paths so that they do not have any links nor any nodes in common, and such that the path SRNGs and SRLGs are disjoint.

The SRNG Information may start with a configured value, in which case it does not change over time, unless reconfigured.

The SRNG Information is optional and if a Link State Advertisement doesn't carry the SRNG Information, then it means that SRNG of that nodes is unknown.

DELTA:
No technical change.

2.2. Node Information (WSON specific)

As presented in [RFC6163] a WSON node may contain electro-optical subsystems such as regenerators, wavelength converters or entire switching subsystems. The model present here can be used in characterizing the accessibility and availability of limited resources such as regenerators or wavelength converters as well as WSON signal attribute constraints of electro-optical subsystems. As such this information element is fairly specific to WSON technologies.

2.2.1. Label Restrictions

This section is a preamble presenting the Label Restriction entity, which is refered many times later in this document.

Wavelength constraint are used in different part of the information model, either as static constraints (in the resource pool as RPWvlConstraints, and the resource block IngressWaveConstraint and EgressWaveConstraint) or representing dynamic properties of a given element (SharedAccessWvls in resource pool). In the GMPLS context Wavelengths are represented by Labels.

The wavelength constraints used in this document, although having different semantic, refer to the same notion of list of wavelength. Those constraints apply in addition to either the incoming part of a device (or group of device), the outgoing part or both if the constraint is the same, which is for instance not unusual for static wavelength constraint.

To support this we define in this section a LABEL_RESTRICTIONS encoding that carry a label set information and for which direction this label restriction is valid. The directions considered is upstream, downstream or both. The label set information is the one defined in [I-D.ietf-ccamp-rwa-info] as AvailableLabel.

This encoding is reused in different TLV or sub-TLV for different semantic but do not require to define a TLV per direction.

DELTA:

- Define a generic information for label restrictions

- Reuse generic label set and provide a compact representation

2.2.2. Resource Pools, Resource Blocks and Resource Description Containers

A WSON node may include regenerators or wavelength converters arranged in shared pools. As presented in [RFC6163] this can include OEO based 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 any OEO device / wavelength converter to convert from a given ingress wavelength on a particular ingress port to a desired egress wavelength on a particular egress port.

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

For modeling purposes and encoding efficiency identical processing resources such as regenerators or wavelength converters with identical limitations, and processing and accessibility constraints are grouped into "blocks". Such blocks can consist of a single resource, though grouping resources into blocks leads to more efficient encodings. Then, these resource blocks are gathered once more into resource pool, for which the blocks share the same accessibility constraints.

Definitions:

- Resource Block: A group of resources sharing both the same processing properties and the same accessibility constraints. Each Resource Block can contain a different number of ressources, but all the resources constituting the block are identical devices.

- Resource Pool: A group of resources sharing the same accessibility constraints, hence for the sake of efficient encoding a Resource Pool becomes a group of Resource Blocks sharing the same accessibility constraints. Each Resource Pool can contain a different number of blocks, each of different size, but all the devices in the pool are subject to the same accessibility constraints regarding the way these are linked to ingress and egress links of the WSON node containing the pool. One of the inherent reason for that being their being multiplexed on a given piece of equipment (like an Optical Amplifier, a splitter, a Wavelength Selective Switch port, a length of fiber...), which has some inherent implication on the related information model.

The following picture represents the model of WSON nodes with the help of Resource Blocks and Resource Pools entities.

```
                               +-----------+
        I1  +-----------+      |   RP #1   |      +-----------+ E1
       ---->|           |      | +-------+ |      |           +---->
        I2  |           |      | | Rb #1 | |      |           | E2
       ---->|           |      | +-------+ |      |           +---->
            |           |      | +-------+ |      |           |
            | Resource  +------+ |       | +------+ Resource  |
            | Ingress   |      | | Rb #2 | |      | Egress    |
            | Connectiv |      | |       | |      | Connectiv |
            | Matrix    |      | +-------+ |      | Matrix    |
            |           |      +-----------+      |           |
            |           |      +-----------+      |           |
        I3  |           |      |   RP #2   |      |           | E3
       ---->|           |      | +-------+ |      |           +---->
        I4  |           |      | | Rb #3 | |      |           | E4
       ---->|           +------+ +-------+ +------+           +---->
        IN  |           |      | +-------+ |      |           | EM
       ---->|           |      | | Rb #P | |      |           +---->
            |           |      | +-------+ |      |           |
            +-----------+      +-----------+      +-----------+
```

Figure 1

This figure shows a Resource Ingress Connectivity Matrix and another one of the egress, the model from [I-D.ietf-ccamp-rwa-info] gathers both these connectivity matrix inside a Resource Pool Accessibility item, which would lead to the following definition of a Resource Pool.

```
<ResourcePool> ::= <ResourcePoolID> [<ResourceDescription>]...
    [<ResourcePoolAccessibility>] [<ResourcePoolWvlConstraints>]...
    [<SharedAccessWvls>]... [<ResourceBlockState>]...
```

- ResourcePoolID is used to identify the pool,

- ResourceDescriptions are used to define the features of each type of resources held inside the pool,

- ResourcePoolAccessibility is meant to define the spatial connectivity constraints between the pool and the incoming and outgoing links of the node,

- ResourcePoolWvlConstraints may be used to define the structural (static) spectral constraints of accessibility of the pool,

- SharedAccessWvls should be used to provide the dynamic spectral availability coming from the usage of wavelengths by activated resources inside the pool,

- ResourceBlockStates are used to provide the dynamic availability of resources inside the pool.

Actually as stated in Section 2.2.3, it is more efficient to use the node’s own connectivity matrix to embed this kind of information with the one of the incoming and outgoing links of the nodes, hence the model simplifies itself into:

```
<ResourcePool> ::= <ResourcePoolID> [<ResourceDescription>]...
    [<ResourcePoolWvlConstraints>]... [<SharedAccessWvls>]...
    [<ResourceBlockState>]...
```

As this document means to have one ResourcePool entity per physical pool of resources inside the node, it can be observed that when a node contains multiple pools of resources, these ones are likely to share type of resources, hence their modeled respresentations are holding the same ResourceDescription entities. In order to avoid unnecessary information flooding, this document offers the opportunity to extract from the ResourcePool all these ResourceDescriptions and gather them inside a dedicated entity, that is named Resource Description Container.

which provides the alternative model of ResourcePool (which is consistent with the previous one, as the ResourceDescriptions were already optional:

```
<ResourcePool> ::= <ResourcePoolID> [<SharedAccessWvls>]...
    [<ResourcePoolWvlConstraints>]... [<ResourceBlockState>]...
```

and Resource Description Container, which is a list of Resource Descriptions:

```
<ResourceDescriptionContainer> ::= <ResourceDescription>...
```

DELTA:

- Introduced definition of Resource Pool.

- Introduced definition of Resource Pool ID.

- Introduced definition of Resource Description Container.

- Changed accordingly Figure 1 and 2 from [I-D.ietf-ccamp-rwa-info].

- Changed the RBNF from [I-D.ietf-ccamp-rwa-info].

- Changed the Resource Block Info into Resource Description (small semantic change, due to minor internal changes.

- Adapted some pieces of models which were related to Resource Block, to the Resource Pool level, namely: RPWvlConstraints

2.2.3. Resource Pool Accessibility

Every device inside a Resource Pool shares the same accessibility constraints, hence the accessibility is a property related to the pool. In order to depict the accessibility of a given pool, two pieces of information needs to be described:

- Which ingress links of the node can be connected to the entry of the Resource Pool,

- Which egress links of the node can be connected to the exit of the Resource Pool.

Following remarks can be made concerning these accessibility information:

- These information share the same nature as the one of the Connectivity Matrix (see Section 2.1.1),

- These information are relatively static, changing only when the switching fabric of the node is changing (either failure or upgrade),

- When a given node contains multiple Resource Pools, it is not unlikely that some of them share list of either ingress or egress links of the nodes to which they can be connected; hence it can be more efficient to gather the accessibility information related to every Resource Pool inside a single entity, instead of having a specific entity for each pool.

Hence, the accessibility information of every Resource Pool are embedded together inside the node own’s Connectivity Matrix. The solution used to do that consists in using both Local Link Identifiers and Resource Pool Identifiers inside the Link Sets of the Connectivity Matrix. To keep unchanged the definition of the Link

Set, 32 bits unnumbered IDs for the Resource Pool are needed (see Section 2.2.4). Thanks to this in the context of a node, the Connectivity Matrix is then providing associations between:

- On one side a set composed of a mix of: (1) ingress link(s) and (2) exit(s) of resource pool(s),

- On the other side a set composed of a mix of: (1) egress link(s) and (2) entry(ies) of resource pool(s).

Then the RBNF for the Connectivity Matrix (see Section 2.1.1) becomes,

```
<ConnectivityMatrix> ::= <MatrixID> <ConnectType>
    (<IngressSetOfMixedLink&Pool> <EgressSetOfMixedLink&PoolSet>)...
```

The Resource Pool Accessibility information are optional, if not defined, Resource Pool is meant to have no accessibility constraints: from every node ingress port it's possible to reach the pool and the pool egress can reach every egress port of the node.

DELTA:
This section could be compared to the Resource Block Accessibility constraint, and this is a major change that is proposed here.

2.2.4. Resource Pool ID

Alike a GMPLS unnumbered link benefits from the definition of Link Local and Link Remote Identifiers defined in [RFC4202] a resource pool benefits from a Resource Pool ID. For each Resource Pool, WSON node assigns one identifier to each side of the pool. This identifier is a non-zero 32-bit number that is unique within the scope of the WSON node that assigns it, hence the Resource Pool ID is composed by a couple of unique numbers.

Consider a (resource) pool inside WSON node A. WSON node A chooses two distincts identifiers for the pool (one for the ingress side and one for the egress side). Considering these identifiers being unique inside the scope of the WSON node A, implies that: no other (resource) pool inside WSON node A may be assigned the value corresponding to any of these two identifiers, neither any (unnumbered) link between WSON node A and any other node may be assigned a link local identifier (from the WSON node A perspective) value corresponding to any of these two identifiers.

Support for resource pools in routing includes carrying information about the identifiers of these pools. Specifically, when an LSR advertises a resource pool, the advertisement carries both the ingress and the egress identifiers of the link.

<RPoolID> ::= <RESOURCE_INGRESS_ID> <RESOURCE_EGRESS_ID>

2.2.5. Resource Block State

The Resource Block State keep track of the current usage of a resource block within a resource pool.

The state indicate for the resource the number of available resources and optionnaly the total number (or maximum number) of resources. decoupling ResourceDescription from the the ResourceBlock configuration and allowing a better aggregation of the ResourceDescription. The state available in info model is the following:

Resource Block State definition

```
<ResourceBlockState> ::= <ResourceBlockID> [<CountMaxResources>]
    <CountAvailableResources>
```

DELTA:
This definition fo the Resource Block State allow to separate the total number of resources from the resource description (differing in this from [I-D.ietf-ccamp-rwa-info]). This enable a sharing of the resource description between all the pools, while the other solution requires that each pool holds the same number of devices to share the same ResourceBlockDescription (see Section 2.2.6).

2.2.6. Resource Description

The resource block information contains the pieces of information needed to fully identify the resource block static and dynamic information. The static information consist of the characteristics that do not depend on the LSPs using the resource block. In particular the wavelength constraints are the one of the OEO and are independent of the LSPs. the static information is described by a ResourceDescription, which can be valid for several resource blocks, then referenced by their ResourceBlockID.

The ResourceBlockID identifies a resource block, it is a node wide stable and unique identifier (inside the node context). The ResourceBlockID is defined in the ResourceBlockState TLV held in the Resource Pool TLV and used in the Resource Description TLV.

```
<ResourceDescription> := <ResourceBlockID>... <InputConstraints>
    <ProcessingCapabilities> <OutputConstraints>
```

with,

```
<InputConstraints> ::= [<IngressWaveConstraint>] [<modulation-list>]
    [<fec-list>] [<rate-range-list>] [<client-signal-list>]

<ProcessingCapabilities> ::= <RegenerationCapabilities>
    [<FaultPerfMon>] [<VendorSpecific>]

<OutputConstraints> ::= [<EgressWaveConstraint>] [<modulation-list>]
    [<fec-list>]
```

IngressWaveConstraint and EgressWaveConstraint are described in Section 2.2.7. The modulation-list and fec-list represent the list of modulation formats and FEC encoding available within the resource block. This information MAY be present in the advertisement, the absence of this information means that potentially all Modulation and FEC are accepted and possible cranckback may occur.

DELTA:

- Split between static (can be in a separate LSA or in the resource pool) and dynamic information.

- The maximum number of resource is in the state to allow better summarization of the resourceDescription

- The static information is describing the properties, the ResourceDescription is more explicit than resourceInfo in this context

- Changed the RBNF from [I-D.ietf-ccamp-rwa-info], make use of generic label restriction for the wavelength restrictions.

2.2.7. Resource Pool Wavelength Constraints

This field defines any constraint at wavelength level within a resource pool, and is meaningful only when a subset of wavelengths could be configurable within the Pool. This information is static since it depends on specific physical resources within the pools and changes only if there is a node reconfiguration (OEO pools added or removed from an optical node, change in the mux or demuxing devices). As there is an ingress side and an egress side of a pool, this item needs to modelize the wavelength usage on each side.

This field takes the format of a Label_Restrictions Section 2.2.1. At most two instances of this item can be needed: one for each sides (incoming / outgoing) of the pool.

The field is optional, when this field is not present it means there are no specific wavelength constraints imposed by pool. As an example this field is equivalent to the Maximum Bandwidth field defined within [RFC3630]. As the Maximum Bandwith represents the true link capacity, the RESOURCE_POOL_WAVELENGTH_CONSTRAINTS represent the set of wavelengths that can possibly be configured on the pool.
Note that the usable set of wavelengths could be limited by other constraints: e.g. currently in-use wavelength (see Section 2.2.8) or due to OEO device constraint on compliant wavelengths (see Wavelength Constraints in Section 2.2.6).

DELTA:
Only wavelength constrain. While physical constraints are grouped in another set.

2.2.8. Shared Access Available Wavelengths

The SHARED_ACCESS_AVAILABLE_WAVELENGTHS represents wavelength usage in a Resource Pool hence it is related with the Resource Pool dynamic state.

If a wavelength is in use within a pool, the same wavelength cannot be reused in the same pool however the pool will be available for a different wavelength depending on free resource blocks (Resource Pool defintion as in Section 2.2.2). As there is an ingress side and egress side of a pool, this item needs to modelize the wavelength usage on each side. Hence, this representation automatically considers the case of wavelength conversion happening inside the pool.

This field takes the format of a Label_Restrictions Section 2.2.1. At most two instances of this item can be needed: one for each sides (incoming / outgoing) of the pool.

N.B.: Hence, SHARED_ACCESS_AVAILABLE_WAVELENGTHS has the same format as RESOURCE_POOL_WAVELENGTH_CONSTRAINTS defined in Section 2.2.7.

DELTA:
Only wavelength constraint. While physical constraints are grouped

in another set.

2.3. Link Information (General)

Idem preceding drafts

```
<LinkInfo> ::=  <LinkID> [<AdministrativeGroup>] [<InterfaceCapDesc>]
    [<Protection>] [<SRLG>...] [<TrafficEngineeringMetric>]
    [<PortLabelRestriction>...] [<AvailableWavelengths>]
    [<SharedBackupWavelengths>]
```

3. Encoding

3.1. Node related generic encodings

In this section we propose modification to [I-D.ietf-ccamp-general-constraint-encode].

3.1.1. Connectivity Matrix

The Connectivity Matrix Section 2.1.1 represents how ingress ports are connected to egress ports for network elements. The switch and fixed connectivity matrices can be compactly represented in terms of a minimal list of ingress and egress port set pairs that have mutual connectivity (see section 2.5 of [I-D.ietf-ccamp-general-constraint-encode]).

TLV encoding of this list of link set pairs is:

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|           Type                |             Length            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Connectivity  |   MatrixID    |              Reserved         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                         Link Set A #1                         |
:                               :                               :
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                         Link Set B #1                         |
:                               :                               :
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                        Additional Link set pairs as needed    |
:                      to specify connectivity                  :
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Where

Connectivity is the device type.

0: the device is fixed.

1: the device is switched (e.g., ROADM/ROXC)

MatrixID represents the ID of the connectivity matrix and is an 8 bit integer. The value of 0xFF is reserved for use with port wavelength constraints and should not be used to identify a connectivity matrix.

LinkSet Link Set A #1 and Link Set B #1 together represent a pair of link sets. As stated in Section 2.2.3, both Link Set A and Link Set B MAY contain Resource Pool IDs. There are two permitted combinations for the link set field parameter "dir" for Link Set A and B pairs:

* Link Set A dir=ingress, Link Set B dir=egress
  The meaning of the pair of link sets A and B in this case is that any signal that ingresses a link in set A can be potentially switched out of an egress link in set B.

* Link Set A dir=bidirectional, Link Set B dir=bidirectional
  The meaning of the pair of link sets A and B in this case is that any signal that ingresses on the links in set A can potentially egress on a link in set B, and any ingress signal on the links in set B can potentially egress on a link in set A.

Link Set field encoding is defined in section 2.1 of [I-D.ietf-ccamp-general-constraint-encode].

See [I-D.ietf-ccamp-general-constraint-encode] and its appendixes for examples of both types of encodings.

DELTA:
No change.

3.1.2. Port Label Restrictions

Port Label Restriction tells us what labels may or may not be used between ports of a node or onto a given link.

The port label restriction of Section 2.1.2 can be encoded as a sub-TLV as follows. More than one of these sub-TLVs may be needed to fully specify a complex matrix connectivity label constraint or a link related restrictions. When more than one of these sub-TLVs are present the resulting restriction is the intersection of the restrictions expressed in each sub-TLV.

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|           Type                |           Length              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|   MatrixID    |RestrictionType|      Reserved/Parameter       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     Additional Restriction Parameters per RestrictionType     |
:                                                               :
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Where

MatrixID is a reference to a unique identifier of a connectivity matrix.

RestrictionType can take the following values and meanings:

0: SIMPLE_LABEL (Simple label selective restriction)

1: CHANNEL_COUNT (Channel count restriction)

2: LABEL_RANGE (Label range device with a movable center label and width)

3: SIMPLE_LABEL & CHANNEL_COUNT (Combination of SIMPLE_LABEL and CHANNEL_COUNT restriction. The accompanying label set and channel count indicate labels permitted on the port and the maximum number of channels that can be simultaneously used on the port)

4: LINK_LABEL_EXCLUSIVITY (A label may be used at most once amongst a set of specified ports)

For description of the additional Restriction Parameters per RestrictionType, please refer to: section 2.6 of [I-D.ietf-ccamp-general-constraint-encode]

DELTA:
No change.

3.1.3. Shared Risk Node Group

This sub-TLV carries the Shared Risk Node Group information (see Section 2.1.3).

Its length is the length of the list in octets. The value is an unordered list of 32 bit numbers that are the SRNGs that the node belongs to. The format of the value field is as shown below:

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|           Type                |             Length            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                  Shared Risk Node Group Value                 |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                         ............                          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                  Shared Risk Node Group Value                 |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

The SRNG sub-TLV may occur at most once within the Node Attribute TLV.

DELTA:
No technical change.

3.2. Node related WSON specific encodings

This section refer to [I-D.ietf-ccamp-rwa-wson-encode]

3.2.1. Label Restrictions

Relatively to section 2.2 of [I-D.ietf-ccamp-general-constraint-encode] the LABEL_SET field is here slightly modified in order to define a Label Restrictions field.

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Action|U|D|    Num Labels     |           Length              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                          Base Label                           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|        Additional fields as necessary per action              |
:                                                               :
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Although it make sense only using the actions 0-Inclusive List, 2-Inclusive Range or 4-Bitmap. The U bit indicate a label set restriction valid at the upstream direction/incoming side of a resource pool/resource block. The D bit indicate a label set restriction valid at the downstrean/outgoing side of a resource pool/ resource block. At least one of U or D bit MUST be set, both U and D bit MAY be set.

DELTA:
The Num Labels field become 10 bits and this leave room for 1024 labels represented by this encoding. This encoding will be reused in specific TLVs, in case more than 1024 labels are needed multiple fields within TLVs can be used.

3.2.2. Id Set Field

With the introduction of resource description describing properties for a group of resource block we need to efficiently represent a set of IDs. To do so we introduce an IDSet field which has the same encoding as the LinkSet field defined in [I-D.ietf-ccamp-general-constraint-encode] but with a more generic description.

ID Set Field

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|    Action     |Dir|  Format   |          Length               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                          Identifier 1                         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
:                               :                               :
:                               :                               :
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                          Identifier N                         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

The Action, Dir have the same encoding as in [I-D.ietf-ccamp-general-constraint-encode]. The Format field indicates the format and length of the Identifier:

0 -- 32 Bit unnumbered identifier

1 -- IPv4 identifier

2 -- IPv6 identifier

This field is used later to define a set of resource blocks (e.g. to list the resource blocks sharing the same resource description).

3.2.3. Resource Pool Accessibility

The Resource Pool Accessibility needs no encoding of its own. As explained in the Section 2.2.3 this piece of information is merged inside the Connectivity Matrix object, which encoding is defined in Section 3.1.1, which is actually not impacted by this solution.

Nota: The Link Sets held inside the Connectivity Matrix are composed of LINK_LOCAL_IDENTIFIERS (32 bits identifiers), and the solution to describe the Resource Pool Accessibility consists in using either RESOURCE_INGRESS_ID or RESOURCE_EGRESS_ID (also 32 bits identifiers) which are by definition different from the LINK_LOCAL_IDENTIFIERS (see Section 2.2.4).

DELTA: A major change here as the content of this field are moved inside Connectivity Matrix.

3.2.4. Resource Block State

This TLV indicate the state of a resource block as defined in Section 2.2.5. It defines the ResourceBlockId, and provides the number of free resources and maximum in this resource block. The ResourceBlockID field is a 32 bit node-wide identifier,

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|           Type                |             Length            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                        ResourceBlockID                        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                    CountAvailableResources                    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                      CountMaxResources                        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

The information of the maximum number of resource is optional, this is encoded with a value of 0 in the CountMaxResource field, or with a Length value set to 8 instead of 12.

DELTA:
This is an adaptation of the resource pool status that fits the new definition of resource description.

3.2.5. Resource Description

Resource Description sub-TLVs represent the information described in Section 2.2.6.

The resource description TLV encoding follow the definition from Section 2.2.6 with a list of sub-sub TLV.

Resource Description TLV

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|           Type                |             Length            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                    ResourceBlockID Set Field                  |
:                                                               :
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                         Sub-TLVs (opt)                        |
:                                                               :
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

The ResourceBlockID Set Field is encoded using the IDSet field encoding using the ResourceBlockID as identifier with format 0.

The Sub-Sub TLVs are defined as follow, the order does not matter. Each of the Sub-Sub-TLV defined in this document MAY be repeated more than once, on receipt all Sub-Sub-TLV MUST be taken into account. The resulting information is the union of all the element of the Sub-Sub-TLVs (all Sub-Sub-TLVs of this document describe lists). For example an implementation may choose to indicate that in total 4 label can be used as 4 Label constraint Sub-Sub-tlv, each of them with 1 label.

Info model	Type	Encoding
IngressWaveConstraint	Label Constraints	Label restriction, see Section 3.2.1.
Input modulation-list	Modulation List	A list of Modulation Format Fields, described in [I-D.ietf-ccamp-rwa-wson-encode] section 4.2.1.
Input fec-list	FEC List	A list of FEC type, described in [I-D.ietf-ccamp-rwa-wson-encode] section 4.3.1.
Input rate-range-list	Rate Range	A list of rate range field, described in [I-D.ietf-ccamp-rwa-wson-encode] section 4.4.1.
Input client-signal-list	Client Signal List	A list of GPids, described in [I-D.ietf-ccamp-rwa-wson-encode] section 4.5.

ProcessingCapabilities	Processing Capabilities	A list of Processing Capabilities Fields, except processing cap "Number of Resources", described in [I-D.ietf-ccamp-rwa-wson-encode] section 4.6.1.
EgressWaveConstraint	Label Constraints	Label restriction, see Section 3.2.1.
Output modulation-list	Modulation List	see Input modulation-list
Output fec-list	FEC List	see Input fec-list

Resource description Sub-Sub-TLVs and relation to info model

The Label Constraints Sub-Sub-TLV is used for IngressWaveConstraint and EgressWaveConstraint as the Label Restriction field carries the U and D bit to allow to distinguish a label restriction valid for incoming, outgoing or both.

The Modulation List Sub-Sub-TLV is similarly used for the input and output modulation list. The Sub-Sub-TLV contains a list of Modulation format field, which indicate if they are valid for the input (I bit set to 1) or for the output (I bit cleared). The list of Modulation format field MUST contain at least one ingress FEC modulation format. If no Egress modulation format is present in the list it is implied that no modulation format conversion is impossible, the egress modulation list is the same as the ingress modulation list and modulation format is not performed.

The FEC list Sub-Sub-TLV is also representing both Input and Output FEC list. The Sub-Sub-TLV is defined as a list of FEC Fields, conceptually being Sub-Sub-Sub-TLVs indicating via the I bit if they are valid for ingress or egress. At least one ingress FEC MUST be present in the list, if no egress modulation format is present in the list it is implied that the egress FEC list is the same as the ingress FEC list. In such case FEC format conversion MAY be performed.

The Processing Capabilities Sub-Sub-TLV is the same as in [I-D.ietf-ccamp-rwa-wson-encode] section 4.6.1. except for the maximum number of resource which is represented in the ResourceBlockState. The FEC and Modulation format conversion capabilities are expressed via the Modulation and FEC list by not including any egress modulation/fec in the respective lists.

Bit-Rate Range and Client Signal lists are unchanged from [I-D.ietf-ccamp-rwa-wson-encode]

DELTA:

- use a common TLV for the label restriction

- use a common TLV for the FEC list

- use a common TLV for the Modulation format list

- re-use indirectly (via ID Set) the general encoding LinkSet for RBlockId set

- More explicit statement on FEC and Modulation format conversion capabilities

3.2.6. Resource Pool Wavelength Constraints

This TLV is used to describe static wavelength constraint, it follows the encoding of Label_Restrictions field Section 3.2.1

RESOURCE_POOL_WAVELENGTH_CONSTRAINTS TLV

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|           Type                |            Length             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                 Label_Restrictions field(s)                   |
:                                                               :
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

The Label_Restrictions field might be repeased several times depending on the U and D bit flags. In case multiple fields with the same U and D bits set to 1, the final resulting constrain will be the intersection of all Label_Restrictions. If multiple TLVs are present the resulting constraint is the intersection of all the TLV.

Example below:

- No RESOURCE_POOL_WAVELENGTH_CONSTRAINTS TLV meaning that these type of constraints are not described.

- A TLV present with one Label_Restrictions field with both the U or D bits MUST be set to 1. Which means the same constrains apply to both sides of the pool.

- A TLV present with three Label_Restrictions field presents, one field with U=1 so applicable upstream. The two other fields with D=1 so applicable downstream

------
DELTA: Small delta, just using the add-on bits to provide a direction/side semantic.

3.2.7. Shared Access Available Wavelengths

This TLV is used to describe dynamic wavelength availability, it follows the encoding of Label_Restrictions field. Section 3.2.1

SHARED_ACCESS_AVAILABLE_WAVELENGTH TLV

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|           Type                |             Length            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                 Label_Restrictions field(s)                   |
:                                                               :
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

The same rules and usage defined in Section 3.2.6 apply here.

3.2.8. Resource Pool

The RESOURCE_POOL TLV contains the preceding TLVs.

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|           Type                |             Length            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                     RESOURCE_INGRESS_ID                       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                     RESOURCE_EGRESS_ID                        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                  Sub-TLVs as needed (Opt)                     |
:                                                               :
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

List of possible Sub-TLVs:

Name	Static/Dynamic
Resource Description	Static
Resource Block State	Dynamic
Shared Access Available Wavelength	Dynamic
Resource Pool Wavelength Constraints	Static

DELTA:
Similar to Resource Pool inside [I-D.ietf-ccamp-rwa-wson-encode] with a different internal layout that allows for multiple instances.

3.2.9. Resource Description Container

The RESOURCE_DESCRIPTION_CONTAINER is a list of RESOURCE_DESCRIPTION. This one MAY be used to extract the static content of the previous TLV, in order to hold all this content inside this purposely defined static TLV. Then each one can be in separatly flooded entities (e.g. in separated LSAs see Section 4.1.

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|           Type                |           Length              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                      RESOURCE_DESCRIPTION                     |
:                                                               :
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
:                                                               :
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                      RESOURCE_DESCRIPTION                     |
:                                                               :
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

DELTA:
New item.

3.3. Link related encodings

This section does not differ from the equivalent in [I-D.ietf-ccamp-general-constraint-encode]

4. OSPF-TE Extensions

This section handles OSPF-TE extensions.

It starts with introducing the top view of the extensions provided by this draft. Then a sub-section dedicated for each top level TLV details the extensions relevant for this top level TLV.

4.1. Introduction

This introduction provides the layout of the preceding information model (Section 2) and encodings (Section 3) into top-level TLVs of opaque LSAs.

[RFC3630] introduces Link top level TLV (type 2). This document extends its content with the encodings depicted in Section 3.3. These extensions offer the capability to advertise restrictions on the list of available labels.
N.B.: This capability is specifically useful when these labels have a network wide semantic like suggested in a WSON context.

[RFC5786] introduces Node Attribute top level TLV (type 5). This document extends its content with the encodings depicted in Section 3.1. These extensions offer the capability to advertise restrictions on the switching capabilities of the node.
N.B.: This TLV is unique for a given node and contains static information only, hence no more than one LSA per node is expected to host such a TLV.

This document introduces a new top level TLV named RESOURCE_POOL (type value to be defined), which encodings are depicted in Section 3.2. RESOURCE_POOL TLV offers the capability to advertise one or multiple pools of OEO devices held in a given node. This object can carry resource descriptions, the available resources inside the pool(s) and the availability of wavelengths to reach the pool (refer to pool definition inside Section 2.2.2).
N.B.: A LSA can contain more than one RESOURCE_POOL top level TLV (allowing one LSA to advertise the description of all the pools of the originating node). Alternatively, a node can originate more than one LSA containing each RESOURCE_POOL top level TLVs (allowing each LSA to advertise an individual pool). In that case all the RESOURCE_POOL originated by the same node MUST have different RESOURCE_POOL_ID. As most of the information contained inside a RESOURCE_POOL are dynamic, an implementer may well choose to define one LSA per pool of resources in order to reduce the quantity of information flooded upon change in resource usage.

This document introduces another new top level TLV named

RESOURCE_DESCRIPTION_CONTAINER (type value to be defined), which encoding is depicted in Section 3.2.9.
RESOURCE_DESCRIPTION_CONTAINER TLV offers the capability to advertise a list of RESOURCE_DESCRIPTION valid in the scope of the originating node. A given node MUST NOT originate more than one LSA containing RESOURCE_DESCRIPTION_CONTAINER TLV. An LSA containing a RESOURCE_DESCRIPTION_CONTAINER TLV MUST NOT contain any additional top level TLV.
N.B.: This TLV is designed to be unique in the scope of the originating node and to gather all the resource descriptions relevant in this scope. This "optional" TLV is provided for the implementer who wants to mutualize static information of multiple (or even single) LSAs containing RESOURCE_POOL TLVs originated by the same node. In that case, the node’s LSAs containing RESOURCE_POOL TLV(s) are referring to the content of the node’s LSA containing the RESOURCE_DESCRIPTION_CONTAINER TLV. The content of this LSA is static.

Summarizing Table

Top-TLV Type	Name	Instances	Static/Dynamic
2	Link	1 per fiber	Mix
5	Node Attribute	1 per Node	Static
TBD	Resource Pool	1 per Pool	Dynamic
TBD	Resource Desc Cont	1 per Node	Static

DELTA:

- Renamed the Node Optical Property tlv into Resource Pool TLV

- Allow multiple instance of Resource Pool TLV

- Introduced an optional new TLV named Resource Description

4.2. Link top level TLV (Generic)

This section refer to [I-D.ietf-ccamp-gmpls-general-constraints-ospf-te].

The following new sub-TLVs are added to the Link top level TLV (type 2).

```
Sub-TLV Type  Length  Name
     TBD     variable Port Label Restrictions
     TBD     variable Available Wavelengths
     TBD     variable Shared Backup Wavelengths
```

In Link TLV, all the sub-TLV listed above are optional.

4.3. Node Attribute top level TLV (Generic)

This section refer to [I-D.ietf-ccamp-gmpls-general-constraints-ospf-te].

The following new sub-TLVs are added to the Node Attribute top level TLV (type 5).

```
Sub-TLV Type  Length  Name
     TBD     variable Connectivity Matrix
     TBD     variable Port Label Restrictions
     TBD     variable Shared Risk Node Group
```

In Node Attribute, all the sub-TLV listed above are optional. None of them contain sub-TLV.

4.4. Resource Pool top level TLV (WSON-specific)

This section refer to [I-D.ietf-ccamp-wson-signal-compatibility-ospf]

The following sub-TLVs are created for the Resource Pool top level TLV.

```
Sub-TLV Type  Length  Name
     TBD     variable Resource Description
     TBD     variable Resource Block State
     TBD     variable Shared Access Available Wavelength
     TBD     variable Resource Pool Wavelength Constraints
```

In Resource Pool, all the sub-TLV listed above are optional. The only one of them containing sub-TLV is the Resource Description.

4.4.1. Resource Description sub-TLV

The following sub-TLVs are created for the Resource Pool top level TLV.

Sub-TLV Type	Length	Name
TBD	variable	Modulation List
TBD	variable	FEC List
TBD	variable	Rate Range List
TBD	variable	Client Signal List
TBD	variable	Processing Capabilities
TBD	variable	Label Constraints

In Resource Description, all the sub-TLV listed above are optional.

4.5. Resource Description Container top level TLV (WSON-specific)

This section refer to [I-D.ietf-ccamp-wson-signal-compatibility-ospf]

The following sub-TLVs are created for the Resource Description Container top level TLV.

Sub-TLV Type	Length	Name
TBD	variable	Resource Description

5. Solution(s) Evaluation

Within this section we try evaluate the amount of information that needs to be exchanged through routing advertisements. For this evaluation we consider some minimum optical node reference design to make a OEO extension future proof.

This sections starts with summarizing the LSAs needed to depict a node with both the solution depicted by this document and the solution depicted by [I-D.ietf-ccamp-rwa-info]. Afterwards, the hypothesis concerning the node features that will serve as a basis for the solution evaluation will be presented, before the actual results of the solutions evaluations (both the one of this document and the one of [I-D.ietf-ccamp-rwa-info]).

5.1. RBNFs Comparison

In this section we try compare the how TLVs are composed according two this draft proposal versus existing WSON solutions. The goal here is to provide the all reference for and easy understanding where two solutions are different. Numbers will be provided in the next section.

The evaluation will be done on the Resource Pool top-level TLV since the Node address and Link TLV are considered equivalent.

WSON Drafts. According to

[I-D.ietf-ccamp-wson-signal-compatibility-ospf] in section 2 defines the Optical Node Property TLV which collect the WSON specific information. This TLV is composed of the following:

```
 <ResourcePool> ::= [<ResourceBlockInformation>]...
     [<ResourceBlockAccessibility>]... [<ResourceBlockWvlConstraint>]...
     [<ResourceBlockPoolState>...] [<SharedAccessWvls>...]
```

a) Resource Block Information. Defined as : ([<ResourceSet>] <InputConstraints> <ProcessingCapabilities> <OutputConstraints>). A resource block information defines here the number of devices inside the block.

b) Resource Block Accessibility. Defined as (<PoolIngressMatrix> <PoolEgressMatrix>) which is expanded in tuples like (<INGRESS_LINK_SET><ResourceSet>)* and (<EGRESS_LINK_SET><ResourceSet>)*. Note that INGRESS/ EGRESS_LINK_SET is a name defined here for the link set field used in the [I-D.ietf-ccamp-rwa-info] document.

c) Resource Block Wavelength Constraints. Defined as <IngressWaveConstraints><EgressWaveConstraints>. This is expanded in <ResourceSet>INPUT_WAVELENGHT_SET OUTPUT_WAVELENGTH_SET, for the static constraints of resource blocks.

d) Shared Access Wavelengths. Defined as <IngressWaveConstraints><EgressWaveConstraints>. This is expanded in <ResourceSet>INPUT_WAVELENGHT_SET OUTPUT_WAVELENGTH_SET, for the shared fibers between blocks.

e) Resource Block Pool State. <ResourceSet> <USAGE_STATE_BITMAP>

In current proposal there are two types of TLV.

First the Resource Pool TLV (with an instance per pool) is composed of the following information:

```
<ResourcePool> ::= <ResourcePoolID> [<ResourceDescription>]...
    [<ResourcePoolWvlConstraints>]... [<SharedAccessWvls>]...
    [<ResourceBlockState>]...
```

a) Resource Description. Which is defined as: (<RBlockID>...) <InputConstraints> <ProcessingCapabilities> <OutputConstraints>. This is equivalent to the item a) above without the number of devices inside the resource block, which allow this definition to be usable by any block. The number of available resource of a given type inside the pool being specified by the Resource Block

State below. When a Resource Description Container TLV is defined by a Node, the Resource Pool TLV of this same node SHOULD NOT contain any Resource Description sub-TLV.

b) Resource Block State. Where RBlockState is defined as <RBlockID> [<NumResources>] <NumberOfAvailableResources>. This field efficienty report how many of a given resource type is available inside the pool or not.

c) Shared Access Available Wavelength. This is composed of a Label Restriction field and SHOULD used to depict the dynamic constraints of the pool.

d) Resource Pool Wavelength Constraints. This is composed of a Label Restriction field and MAY be used to depict the static constraints of the pool.

Second the Resource Descriptor Container TLV (with a single instance per node) is used to gather all the Resource Descriptions of a given node, as these are static information composed of the following information:

<ResourceDescriptionContainer> ::= <ResourceDescription>...

a) Resource Description. Which is defined as: (<RBlockID>...) <InputConstraints> <ProcessingCapabilities> <OutputConstraints>. This is equivalent to the item a) above.

5.2. Depiction of the considered cases for evaluation

For the sake of the comparison we have considered the following parameters and values characterizing the optical node design:

- o Node Degree Connectivity: 4, 8 and 16.

- o WDM capacity: 100 wavelengths.

- o Switching capacity. Defines the total node switching capability and is calculated as Node Degree Connectivity x 100 wavelengths.

- o Regeneration Capability. We assume a value of 5% of the total switching capacity.

- o Add/Drop Capability. We assume a typical value of 25% of the switching capability. So in the average up to 30 wavelengths per incoming fiber can be added/dropped within the optical node.

- o Resource pool setup and capabilities. A physical resource pool contains a mix of Add/Drop and Regeneration capabilities. This has the effect of increasing the number of resource pool advertized. Resource pool can be fully flexible (connected to any port), partial (only to some port) or Fixed (can only be connected to one direction). This parameter influences the complexity of the connectivity matrix.

- o Number of Regenerator types. For a given node the number of OEO capabilities is limited, it is typically decided by the type of electrical equipment and optical modules (emitting laser and optical receiver).

- o Blocking Ratio. The Spatial/Spectral blocking ratio indicates how much port-based/wavelength based blocking a node is experiencing.

For example considering the typical design it results in the following static layout:

- o 3 OEO pools each having 3 Resource Block inside.

- o Connectivity Matrix: (8+30+30) 64x64 if considering one connectivity matrix. Ingress=64x3, Egress=3x64 (considering the OEO access with a multiple-wavelength link).

The following types of nodes and node designs were considered in this evaluation:

Node Types and designs

Node Type	Nodal Degree	Pool Type	Blocking
Small(S), Flexible	4	Partial	None
Small(S), Fixed(port)	4	Fixed	Port
Small(S), Fixed(label)	4	Partial	Lambda
Middle(M), Flexible	8	Flexible	None
Large(L), Flexible	16	Flexible	None

For the small nodes, 5 different type of regenerators are considered, for the Middle and Large ones 10 different type of regenerators are considered. Based on those designs we derived the following important figures:

- o Number of resourcePool : depends on the pool type and connectivity, which depend on the port blocking and number of Add/Drop and Regenerator capacity.

- o Number of resourceBlock. There is two numbers to be considered here : the number of resourceBlock for a given resource pool (this document) and total number of resourceBlock ([I-D.ietf-ccamp-rwa-info]). In this document the number of resource block within a resource pool is, worst case, the number of possible regenerator types, whereas in [I-D.ietf-ccamp-rwa-info] the number of resource block depends on the number of OEO types and on the connectivity.

o Number of connectivity matrix/number of pairs/link per pairs. The number of sub-matrix increase depending on the port blocking ratio, the number of pair in one connectivity matrix depends on the wavelength restrictions. Those two criteria do not depend on which information model is considered. The number of link per set is increased by the number of resource pool in this draft.

Those numbers for each node are shown in the following table:

Details of information elements per node

Node Type	Resource Pools	Resource Blocks	Matrix/Pair/Links
S, Flexible	1	5 (30)	1/1/10 (1/1/1)
S, Fixed(port)	4	5 (60)	4/4/4 (4/4/1)
S, Fixed(label)	4	5 (30)	4/1/10 (4/1/1)
M, Flexible	1	10 (30)	1/1/11 (1/1/1)
L, Flexible	1	10 (50)	1/1/21 (1/1/1)

Nota: Values for [I-D.ietf-ccamp-rwa-wson-encode] are between brackets

5.3. Comparing evaluation of the solutions

Based on those key information model elements both the tables "LSA size" indicate the size of the LSAs in this document and in [I-D.ietf-ccamp-rwa-wson-encode]. Number of flooded LSAs of a given type are indicated between brackets (when bigger than 1).

Solution of this document - Average size (and number) of LSAs per node type (unit: bytes)

Node Type	Node Attr LSA	Resource Pool LSA	Resource Desc LSA
S, Flexible	117	120 (6)	524
S, Fixed(port)	692	120 (12)	644
S, Fixed(label)	620	120 (6)	524
M, Flexible	127	120 (3)	904
L, Flexible	209	120 (5)	984

Solution of [I-D.ietf-ccamp-rwa-wson-encode] - Average size (and number) of LSAs per node type (unit: bytes)

Node Type	Node Attr LSA	Optical Node LSA
S, Flexible	49	2801
S, Fixed(port)	340	2980
S, Fixed(label)	132	4118
M, Flexible	52	2980
L, Flexible	54	2809

The Resource Description Container LSA contains several resource description TLVs. This LSA is smaller than the corresponding in [I-D.ietf-ccamp-rwa-wson-encode] mainly because the resource description do not depend on the port/lambda connectivity and number of device per block, thus allowing a better sharing of the information depicting the oeo capabilities.

The following summarizing table indicates the size of the sum of all LSA and the average size per update. In this document all the dynamic part is in the resource pool, allowing a more efficient updating behavior. The evaluation for [I-D.ietf-ccamp-rwa-wson-encode] are best case/worst case; the best case being an update of the RBState TLV and SharedAccessPool TLV only, which requires a multi-instance implementation of OSPF.

Summarizing Table (unit:bytes)

Node Type	Total LSA size	Total number of LSA	Avg size of an update
S, Flexible	1361 (2850)	8 (2)	120 (616/2801)
S, Fixed(port)	2776 (5411)	14 (2)	120 (1192/2980)
S, Fixed(label)	1864 (2941)	8 (2)	120 (616/4118)
M, Flexible	1391 (3032)	5 (2)	120 (448/2980)
L, Flexible	1793 (4172)	7 (2)	120 (720/2809)

Nota: Values for [I-D.ietf-ccamp-rwa-wson-encode] are between brackets

The node design considered are typical case, a worst case can be a node with high nodal degree, with lots of port and wavelength constraints. With considering a nodal degree of 8, resulting in 28 resource pool and 140 resource blocks, the total size is 9816 (11820) with 30 (2) LSAs.

6. Acknowledgements

This template was derived from an initial version written by Pekka

Savola and contributed by him to the xml2rfc project.

This document shares common material with the documents quoted, which seems fair as the target of this version is to highlight differences.

7. Contributors

Daniele Ceccarelli
Ericsson
Via A. Negrone 1/A
Genova - Sestri Ponente
Italy

Email: daniele.ceccarelli@ericsson.com

8. IANA Considerations

This memo requires many requests to IANA, which will be completed in a latter version.

9. Security Considerations

All drafts are required to have a security considerations section. See RFC 3552 [RFC3552] for a guide.

10. References

10.1. Normative References

[I-D.ietf-ccamp-general-constraint-encode]
Bernstein, G., "General Network Element Constraint Encoding for GMPLS Controlled Networks", draft-ietf-ccamp-general-constraint-encode-04 (work in progress), December 2010.

[I-D.ietf-ccamp-gmpls-general-constraints-ospf-te]
Zhang, F., Lee, Y., Han, J., Bernstein, G., Xu, Y., Zhang, G., Li, D., Chen, M., and Y. Ye, "OSPF-TE Extensions for General Network Element Constraints", draft-ietf-ccamp-gmpls-general-constraints-ospf-te-00 (work in progress), March 2011.

[I-D.ietf-ccamp-rwa-wson-encode]
Bernstein, G., Lee, Y., Li, D., Imajuku, W., and J. Han,

"Routing and Wavelength Assignment Information Encoding for Wavelength Switched Optical Networks", draft-ietf-ccamp-rwa-wson-encode-11 (work in progress), March 2011.

[I-D.ietf-ccamp-wson-signal-compatibility-ospf]
Lee, Y. and G. Bernstein, "OSPF Enhancement for Signal and Network Element Compatibility for Wavelength Switched Optical Networks", draft-ietf-ccamp-wson-signal-compatibility-ospf-04 (work in progress), March 2011.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC2629] Rose, M., "Writing I-Ds and RFCs using XML", RFC 2629, June 1999.

[RFC3552] Rescorla, E. and B. Korver, "Guidelines for Writing RFC Text on Security Considerations", BCP 72, RFC 3552, July 2003.

[RFC3630] Katz, D., Kompella, K., and D. Yeung, "Traffic Engineering (TE) Extensions to OSPF Version 2", RFC 3630, September 2003.

[RFC4202] Kompella, K. and Y. Rekhter, "Routing Extensions in Support of Generalized Multi-Protocol Label Switching (GMPLS)", RFC 4202, October 2005.

[RFC5511] Farrel, A., "Routing Backus-Naur Form (RBNF): A Syntax Used to Form Encoding Rules in Various Routing Protocol Specifications", RFC 5511, April 2009.

[RFC5786] Aggarwal, R. and K. Kompella, "Advertising a Router’s Local Addresses in OSPF Traffic Engineering (TE) Extensions", RFC 5786, March 2010.

10.2. Informative References

[I-D.ietf-ccamp-rwa-info]
Bernstein, G., Lee, Y., Li, D., and W. Imajuku, "Routing and Wavelength Assignment Information Model for Wavelength Switched Optical Networks", draft-ietf-ccamp-rwa-info-11 (work in progress), March 2011.

[I-D.narten-iana-considerations-rfc2434bis]
Narten, T. and H. Alvestrand, "Guidelines for Writing an

IANA Considerations Section in RFCs", draft-narten-iana-considerations-rfc2434bis-09 (work in progress), March 2008.

[RFC6163] Lee, Y., Bernstein, G., and W. Imajuku, "Framework for GMPLS and Path Computation Element (PCE) Control of Wavelength Switched Optical Networks (WSONs)", RFC 6163, April 2011.

Appendix A. Additional Stuff

This becomes an Appendix.

Authors' Addresses

Pierre Peloso (editor)
Alcatel-Lucent
R.te de Villejust
Nozay, 91620
France

Phone: +33 130 702 662
Email: pierre.peloso@alcatel-lucent.com

Giovanni Martinelli
Cisco
Monza, 20900
Italy

Phone: +39 039 209 2044
Email: giomarti@cisco.com

Julien Meuric
France Telecom
2, av. Pierre Marzin
Lannion, 22307
France

Phone: +33 296 052 828
Email: julien.meuric@orange-ftgroup.com

Cyril Margaria
Nokia Siemens Networks
St-Martin str. 76
Munchen, 81541
Germany

Phone: +49-89-5159-16934
Email: cyril.margaria@nsn.com