Use of Explicit Routing Object in NSI

Status of This Document

Grid Forum Document (GFD), Recommendation (R).

This document should be read in conjunction with OGF GFD.212 Network Service Interface Connection Service, v2.0.

Copyright Notice

Copyright © Open Grid Forum (2008-2015). All Rights Reserved.

Trademark

OGSA is a registered trademark and service mark of the Open Grid Forum.

Abstract

This document outlines the use of the Explicit Routing Object (ERO) parameter in the NSI reservation request for the point-to-point service element. The ERO parameter allows the requester to guide the reservation request through specific STP or networks within the global topology. Both advertised edge STP and internal hidden STP are supported based on these guidelines.

Contents

Contents 1

1 Introduction 2

2 Notational Conventions 2

3 Explicit Routing Object 2

4 Explicit Routing Object in practice 3

4.1 The P2PS element 3

4.2 Ordering of ERO elements 5

4.3 Support of internal STP 6

4.4 Underspecified STP 7

4.5 Avoiding unnecessary loops 8

4.6 Error Handling 9

5 Glossary 10

6 Contributors 11

7 Intellectual Property Statement 11

8 Disclaimer 11

9 Full Copyright Notice 11

10 References 11

# Introduction

The Network Services Interface provides an API that allows applications to monitor, control, interrogate, and support network resources that are made available by the provider of the network. The NSI Connection Service deals specifically with the request and management of network Connections on transport networks. NSI is inherently agnostic to the technology used in the transport plane. This technology agnostic approach is built into the NSI topology representation and is supported through the use of Service Definitions.

A Connection Service can be requested by any application that has implemented an NSI CS Requester Agent (RA). Similarly, any network provider who has implemented an NSI Provider Agent (PA) can service the request.

Each service is managed by an exchange of NSI messages between agents. These messages operate using a set of service primitives. Service primitives are the set of instructions that allow the requester to set up and manage a service. Each service request will result in the allocation of a service identifier for the new service instance.

This document describes how the ERO reservation parameter within the point-to-point service element is used to guide a reservation’s service path within the network. This document should be read in conjunction with GFD-R.212 Network Service Interface Connection Service version 2.0 [GFD.212], Open Grid forum GFD-I.213, Network Services Framework v2.0 [GFD.213] and OGF GFD-I.217 NSI Signaling and Path Finding [GFD.217].

# Notational Conventions

The keywords “MUST”, “MUST NOT”, “REQUIRED”, “SHALL”, “SHALL NOT”, “SHOULD”, “SHOULD NOT”, “RECOMMENDED”, “MAY”, and “OPTIONAL” are to be interpreted as described in [RFC 2119]. Words defined in the glossary are capitalized (e.g. Connection). NSI protocol messages and their attributes are written in camel case and italics (e.g. *reserveConfirmed*)

# Explicit Routing Object

As defined in [GFD.212], section 3.2, a Connection request can optionally include an Explicit Routing Object (ERO) element. An ERO is an ordered list of STPs that describe the route that should be taken by the Connection. The inter-network pathfinder will use STPs listed in an ERO element as constraints during the path finding process. The Connection will include all of the STPs in the ERO in the sequence in which they are listed. However an ERO is not ‘strict’ in the sense that a Connection is allowed to transit intermediate STPs between the STPs listed in the ERO.

Figure 1 shows an example of a Connection. This Connection conforms to any of the following ERO: (STP b, STP d, STP f), or (STP c, STP e, STP g). Note that as the ingress and egress STPs of a Connection are defined in dedicated fields of the Connection request, they MUST NOT be included in the ERO. Also notice that STP at either end of an SDP can be used to uniquely identify the SDP to transit. Both STP on a single SDP are not required in the ERO, and in fact, only a single one should be specified.

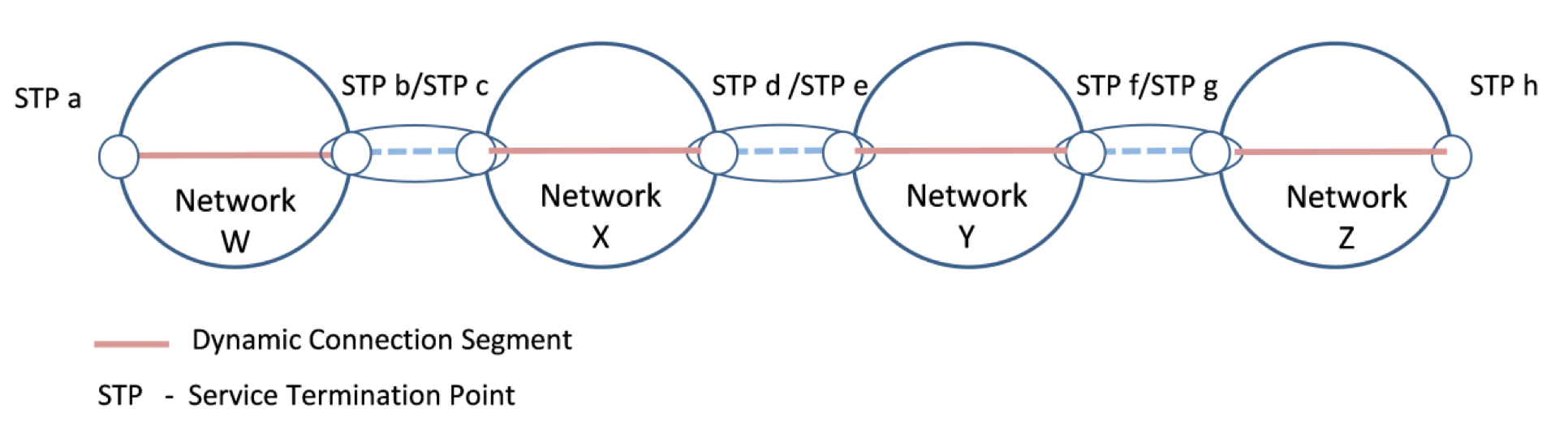


Figure – Example of an ERO.

The NSI CS does not require NSI messages to be forwarded through the same sequence of NSAs/Networks that the Connection transits, and as a consequence, both tree and chain type architectures are supported.

# Explicit Routing Object in practice

[GFD.212] provides a high level definition of the ERO, including the XSD schema used to communicate the ERO within the point-to-point service element, however, it did not discuss some of the more practical aspects of specifying the ERO and interpretation rules for pathfinders. The following sections will discuss these aspects in more detail.

## The P2PS element

When specifying an ERO within the *p2ps* element, the source and destination STP identifiers are contained within the *sourceSTP* and *destSTP* respectively. They are not repeated in the *ero* element even though they are considered bookends to the explicit path. For example, a *p2ps* element specifying a connection from:

source STP urn:ogf:network:kddilabs.jp:2013:topology:bi-ps?vlan=1782

to destination STP urn:ogf:network:uvalight.net:2013:topology:ps?vlan=1782

via intermediate STP urn:ogf:network:icair.org:2013:topology:netherlight?vlan=1782

Would appear as follows in the NSI reservation request:

<p2ps>  
 <capacity>100</capacity>  
 <directionality>Bidirectional</directionality>  
 <symmetricPath>true</symmetricPath>  
 <sourceSTP>urn:ogf:network:kddilabs.jp:2013:topology:bi-ps?vlan=1782</sourceSTP>  
 <destSTP>urn:ogf:network:uvalight.net:2013:topology:ps?vlan=1782</destSTP>  
 <ero>  
 <orderedSTP order="1">  
 <stp>urn:ogf:network:icair.org:2013:topology:netherlight?vlan=1782</stp>  
 </orderedSTP>  
 </ero>  
</p2ps>

This would be classified as a loose ERO (not strict) since it does not specify a full hop-by-hop path from source to destination. Figure 2 below visualizes the result of this request as a possible path on the Automated GOLE topology.

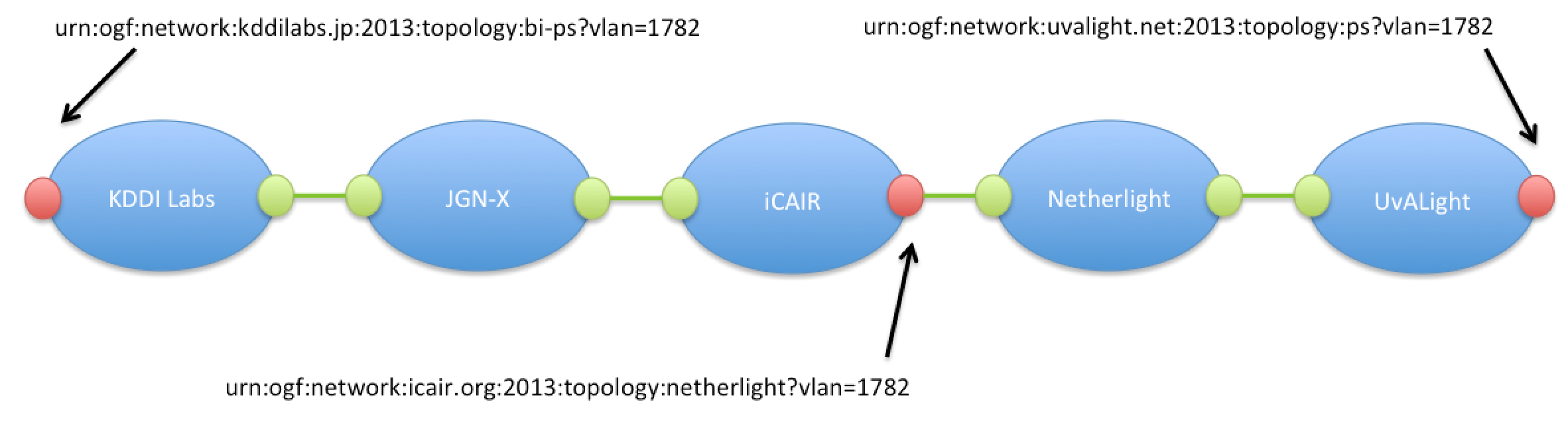


Figure – Loose ERO request with single STP.

A pathfinder could theoretically compute the following detailed path segments based on the reservation request and current Automated GOLE topology:

<p2ps>  
 <capacity>100</capacity>  
 <directionality>Bidirectional</directionality>  
 <symmetricPath>true</symmetricPath>  
 <sourceSTP>urn:ogf:network:kddilabs.jp:2013:topology:bi-ps?vlan=1782</sourceSTP>  
 <destSTP>urn:ogf:network:kddilabs.jp:2013:topology:bi-kddilabs-jgn-x?vlan=1782</destSTP>  
</p2ps>

<p2ps>  
 <capacity>100</capacity>  
 <directionality>Bidirectional</directionality>  
 <symmetricPath>true</symmetricPath>  
 <sourceSTP>urn:ogf:network:jgn-x.jp:2013:topology:bi-jgn-x-kddilabs?vlan=1782</sourceSTP>  
 <destSTP>urn:ogf:network:jgn-x.jp:2013:topology:bi-jgn-x-startap?vlan=1782</destSTP>  
</p2ps>

<p2ps>  
 <capacity>100</capacity>  
 <directionality>Bidirectional</directionality>  
 <symmetricPath>true</symmetricPath>  
 <sourceSTP>urn:ogf:network:icair.org:2013:topology:jgn-x?vlan=1782</sourceSTP>  
 <destSTP>urn:ogf:network:icair.org:2013:topology:netherlight?vlan=1782</destSTP>  
</p2ps>

<p2ps>  
 <capacity>100</capacity>  
 <directionality>Bidirectional</directionality>  
 <symmetricPath>true</symmetricPath>  
 <sourceSTP>urn:ogf:network:netherlight.net:2013:production7:starlight-1?vlan=1782</sourceSTP>  
 <destSTP>urn:ogf:network:netherlight.net:2013:production7:uva-3?vlan=1782</destSTP>  
</p2ps>

<p2ps>  
 <capacity>100</capacity>  
 <directionality>Bidirectional</directionality>  
 <symmetricPath>true</symmetricPath>

<sourceSTP>urn:ogf:network:uvalight.net:2013:topology:netherlight?vlan=1782</sourceSTP>  
 <destSTP>urn:ogf:network:uvalight.net:2013:topology:ps?vlan=1782</destSTP>  
</p2ps>

Notice that there is no *ero* element in the resulting connection segments. This is due to the fact an edge STP was specified in the original *ero* that was resolved to a *destSTP* parameter in a connection segment, and therefore, need not be repeated.

A strict ERO specifies an STP on each SDP from source to destination without any gaps. STP on both ends of a single SDP is not required as a single STP uniquely identifies the pair via the SDP relationship. The following figure shows an example set of STP on the previous path that would be required to consider it a strict ERO.

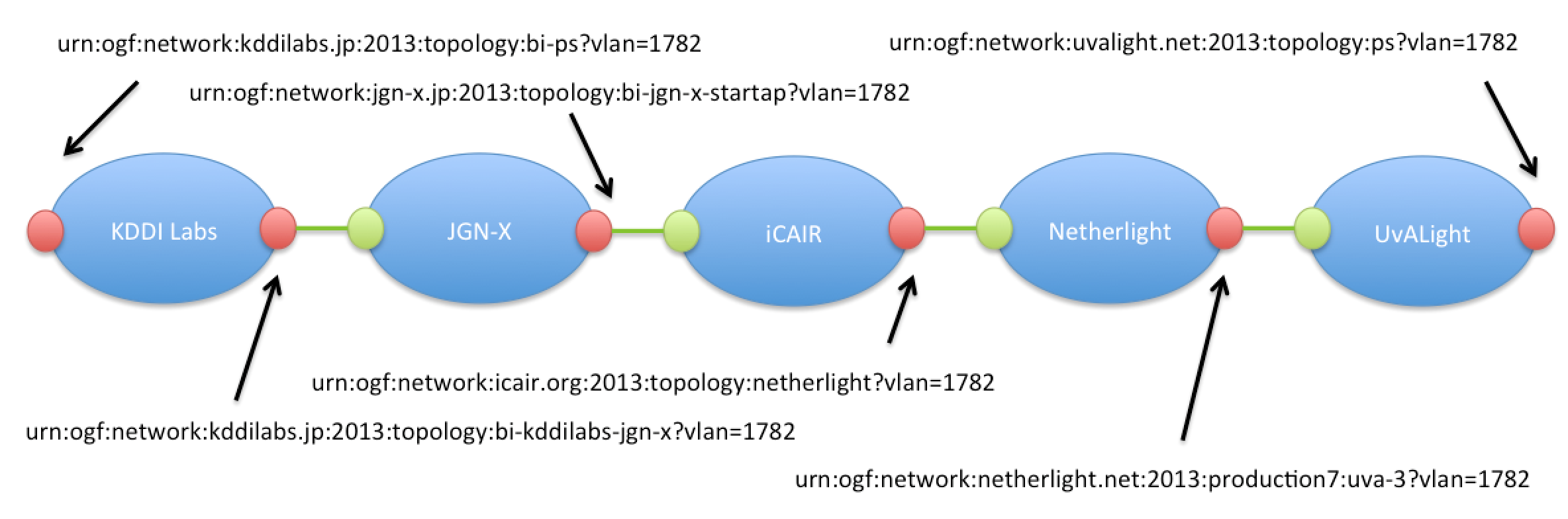


Figure – Strict ERO request.

The following *p2ps* element contains an ERO representing the path described in Figure 3:

<p2ps>  
 <capacity>100</capacity>  
 <directionality>Bidirectional</directionality>  
 <symmetricPath>true</symmetricPath>  
 <sourceSTP>urn:ogf:network:kddilabs.jp:2013:topology:bi-ps?vlan=1782</sourceSTP>  
 <destSTP>urn:ogf:network:uvalight.net:2013:topology:ps?vlan=1782</destSTP>  
 <ero>  
 <orderedSTP order="1">  
 <stp>urn:ogf:network:kddilabs.jp:2013:topology:bi-kddilabs-jgn-x?vlan=1782</stp>  
 </orderedSTP>  
 <orderedSTP order="2">  
 <stp>urn:ogf:network:jgn-x.jp:2013:topology:bi-jgn-x-startap?vlan=1782</stp>  
 </orderedSTP>  
 <orderedSTP order="3">  
 <stp>urn:ogf:network:icair.org:2013:topology:netherlight?vlan=1782</stp>  
 </orderedSTP>  
 <orderedSTP order="4">  
 <stp>urn:ogf:network:netherlight.net:2013:production7:uva-3?vlan=1782</stp>  
 </orderedSTP>  
 </ero>  
</p2ps>

Programmatically there is no way for a requester agent to specify whether an ERO is strict or loose, however, we define additional rules in section 4.5 that help a pathfinder enforce a strict ERO based on the STP specified in a reservation’s *ero* element.

## Ordering of ERO elements

As described in [GFD.212] the *ero* element contains an ordered list of STP identifiers, however, it does not define how the *order* attribute of the *orderedSTP* element is populated. It is recommended that the *order* attribute be populated with sequentially increasing integers starting from 1. The following example illustrates this rule:

<ero>  
 <orderedSTP order="1">  
 <stp>urn:ogf:network:kddilabs.jp:2013:topology:bi-kddilabs-jgn-x?vlan=1782</stp>  
 </orderedSTP>  
 <orderedSTP order="2">  
 <stp>urn:ogf:network:jgn-x.jp:2013:topology:bi-jgn-x-startap?vlan=1782</stp>  
 </orderedSTP>  
 <orderedSTP order="3">  
 <stp>urn:ogf:network:icair.org:2013:topology:netherlight?vlan=1782</stp>  
 </orderedSTP>  
 <orderedSTP order="4">  
 <stp>urn:ogf:network:netherlight.net:2013:production7:uva-3?vlan=1782</stp>  
 </orderedSTP>  
</ero>

## Support for internal STP

NSI pathfinders must support ERO specifying routing constraints using internal STP. An internal STP is defined as an STP within a network that is not an edge STP within the NSI topology (neither a client nor an inter-domain STP). These STPs are typically not described in the NSI topology so they cannot be resolved by the pathfinder, and therefore, cannot be utilized to make inter-domain routing decisions. Internal STP specified in the initial reservation request must be passed to the corresponding uPA within the *ero* element of their resolved connection segment even though these internal STPs could not be use by a pathfinder for path computation. The uPA can then utilize these internal STPs in local routing decisions.

Internal STPs specified in an *ero* element MUST follow the standard STP format. This will allow a pathfinder to determine the network containing the internal STP.

In the following example we see a reservation request for a connection on single network with the *ero* element containing two internal STP identifiers. This can be sent as-is to the uPA associated with the urn:ogf:network:kddilabs.jp:2013:topology for processing.

<p2ps>  
 <capacity>100</capacity>  
 <directionality>Bidirectional</directionality>  
 <symmetricPath>true</symmetricPath>  
 <sourceSTP>urn:ogf:network:kddilabs.jp:2013:topology:bi-ps?vlan=1782</sourceSTP>  
 <destSTP>urn:ogf:network:kddilabs.jp:2013:topology:bi-kddilabs-jgn-x?vlan=1782</destSTP>  
 <ero>  
 <orderedSTP order="1">  
 <stp>urn:ogf:network:kddilabs.jp:2013:topology:internalA</stp>  
 </orderedSTP>  
 <orderedSTP order="2">  
 <stp>urn:ogf:network:kddilabs.jp:2013:topology:internalB</stp>  
 </orderedSTP>  
 </ero>  
</p2ps>

In this previous example the internal STPs are bound by both a source and destination STP within the same network. It is recommended when specifying internal topology elements that you bound them by two valid edge STPs for that network. This will guide the pathfinder to make proper routing decisions without having context of the internal STPs, otherwise, an edge STP might be chosen that is suboptimal for the specified internal STP.

In the following example we see the same two internal STPs in a connection request bounded by only one advertised edge STP. This unintentionally (or intentionally) implies that the pathfinder is free to choose any egress STPs independent of the internal STP specified in the request.

<p2ps>  
 <capacity>100</capacity>  
 <directionality>Bidirectional</directionality>  
 <symmetricPath>true</symmetricPath>  
 <sourceSTP>urn:ogf:network:kddilabs.jp:2013:topology:bi-ps?vlan=1782</sourceSTP>  
 <destSTP>urn:ogf:network:uvalight.net:2013:topology:ps?vlan=1782</destSTP>  
 <ero>  
 <orderedSTP order="1">  
 <stp>urn:ogf:network:kddilabs.jp:2013:topology:internalA</stp>  
 </orderedSTP>  
 <orderedSTP order="2">  
 <stp>urn:ogf:network:kddilabs.jp:2013:topology:internalB</stp>  
 </orderedSTP>  
 </ero>  
</p2ps>

In this example we see a completely unbound internal STP. This has the effect of forcing the pathfinder to choose a path that includes the network urn:ogf:network:netherlight.net:2013:production7, but with no context the pathfinder may select two edge STPs that are not optimal for the internal STP. This option is supported but it is recommended that an intermediate network have bounded edge STP to give an optimal path for the specified internal STP.

<p2ps>  
 <capacity>100</capacity>  
 <directionality>Bidirectional</directionality>  
 <symmetricPath>true</symmetricPath>  
 <sourceSTP>urn:ogf:network:kddilabs.jp:2013:topology:bi-ps?vlan=1782</sourceSTP>  
 <destSTP>urn:ogf:network:uvalight.net:2013:topology:ps?vlan=1782</destSTP>  
 <ero>  
 <orderedSTP order="1">  
 <stp>urn:ogf:network:netherlight.net:2013:production7:internalA</stp>  
 </orderedSTP>  
 </ero>  
<p2ps>

## Underspecified STP

Underspecified STP can also be used in the *ero* element to guide a request through the network when the exact label utilized is not important. The example below contains underspecified STPs on the source and destination STP for a connection using a label in the range 1780-1790, as well as the single STP specified in the *ero* element using the same label range.

<p2ps>  
 <capacity>100</capacity>  
 <directionality>Bidirectional</directionality>  
 <symmetricPath>true</symmetricPath>  
 <sourceSTP>urn:ogf:network:kddilabs.jp:2013:topology:bi-ps?vlan=1780-1790</sourceSTP>  
 <destSTP>urn:ogf:network:uvalight.net:2013:topology:ps?vlan=1780-1790</destSTP>  
 <ero>  
 <orderedSTP order="1">  
 <stp>urn:ogf:network:icair.org:2013:topology:netherlight?vlan=1780-1790</stp>  
 </orderedSTP>  
 </ero>  
</p2ps>

If source routing and TREE signaling were used to compute a path for this previous example, the intermediate STP would need to be resolved to a specific label instance. All intermediate segments must have a specific label, while the original source and destination STP can remain underspecified. This is to ensure that the circuit is not disjoint (i.e. requests for different VLANs on the same peering between two domains), but “stitched” correctly across all the intermediate domains. Below is an example of resolved segments.

<p2ps>  
 <capacity>100</capacity>  
 <directionality>Bidirectional</directionality>  
 <symmetricPath>true</symmetricPath>  
 <sourceSTP>urn:ogf:network:kddilabs.jp:2013:topology:bi-ps?vlan=1780-1790</sourceSTP>  
 <destSTP>urn:ogf:network:kddilabs.jp:2013:topology:bi-kddilabs-jgn-x?vlan=1787</destSTP>  
</p2ps>  
  
<p2ps>  
 <capacity>100</capacity>  
 <directionality>Bidirectional</directionality>  
 <symmetricPath>true</symmetricPath>  
 <sourceSTP>urn:ogf:network:jgn-x.jp:2013:topology:bi-jgn-x-kddilabs?vlan=1787</sourceSTP>  
 <destSTP>urn:ogf:network:jgn-x.jp:2013:topology:bi-jgn-x-startap?vlan=1787</destSTP>  
</p2ps>  
  
<p2ps>  
 <capacity>100</capacity>  
 <directionality>Bidirectional</directionality>  
 <symmetricPath>true</symmetricPath>  
 <sourceSTP>urn:ogf:network:icair.org:2013:topology:jgn-x?vlan=1787</sourceSTP>  
 <destSTP>urn:ogf:network:icair.org:2013:topology:netherlight?vlan=1787</destSTP>  
</p2ps>  
  
<p2ps>  
 <capacity>100</capacity>  
 <directionality>Bidirectional</directionality>  
 <symmetricPath>true</symmetricPath>  
 <sourceSTP>urn:ogf:network:netherlight.net:2013:production7:starlight-1?vlan=1787</sourceSTP>  
 <destSTP>urn:ogf:network:netherlight.net:2013:production7:uva-3?vlan=1784</destSTP>  
</p2ps>  
  
<p2ps>  
 <capacity>100</capacity>  
 <directionality>Bidirectional</directionality>  
 <symmetricPath>true</symmetricPath>  
 <sourceSTP>urn:ogf:network:uvalight.net:2013:topology:netherlight?vlan=1784</sourceSTP>  
 <destSTP>urn:ogf:network:uvalight.net:2013:topology:ps?vlan=1780-1790</destSTP>  
</p2ps>

In this example, an intermediate STP is specified in the *ero* element, however, it only contains the network portion of the STP identifier. This is acceptable for an ERO and is interpreted as a request to include the specified network in the connection path. The pathfinder is free to choose any edge STP on the specified network that can help satisfy the request.

<p2ps>  
 <capacity>100</capacity>  
 <directionality>Bidirectional</directionality>  
 <symmetricPath>true</symmetricPath>  
 <sourceSTP>urn:ogf:network:kddilabs.jp:2013:topology:bi-ps?vlan=1782</sourceSTP>  
 <destSTP>urn:ogf:network:uvalight.net:2013:topology:ps?vlan=1782</destSTP>  
 <ero>  
 <orderedSTP order="1">  
 <stp>urn:ogf:network:icair.org:2013:topology</stp>  
 </orderedSTP>  
 </ero>  
</p2ps>

## Avoiding unnecessary loops

Due to the existence of networks that do not support label swapping it is possible that a request for a connection between two edge STPs with different labels on the same network may result in pathfinders trying to route the connection out of the network to swap labels, and then return to the initial network to complete the connection. In NSI topology terminology, the requested STPs are not within the same Service Domain (SD), and therefore, cannot be directly connected by the network. Figure 4 shows this loop case for a request on the existing Automated GOLE topology. Based on the NM topology description for the iCAIR domain (where label swapping is not supported), a pathfinder determines that the source and destination STP of the reservation are not within the same iCAIR Service Domain, but can connect them by routing the path through the Netherlight domain. Netherlight would then perform label swapping on the connection; thereby interconnect the two iCAIR Service Domains with the differing labels. Pathfinders should avoid these unnecessary loops, excluding any external connections as options when the two STP cannot be directly connected.

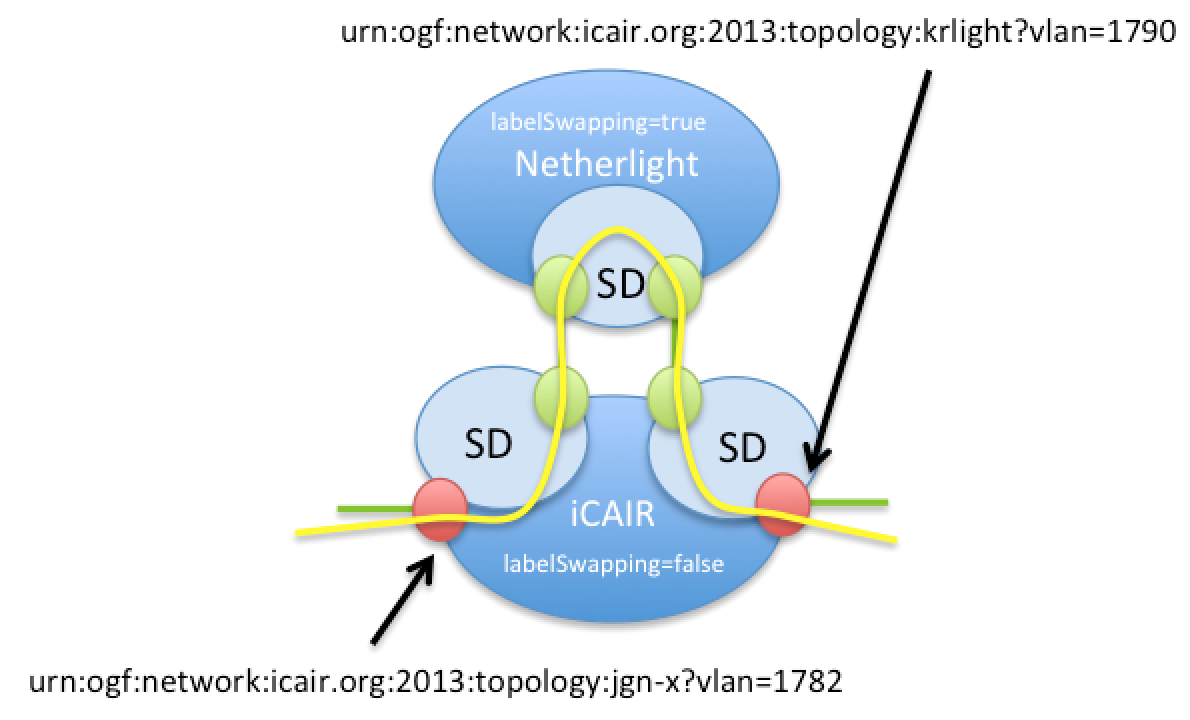


Figure – Unnecessary connection loops.

## Error Handling

If the pathfinder cannot satisfy an ERO then the reservation request fails and a *serviceException* is returned identifying the components of the ERO that caused the failure.

|  |  |  |  |
| --- | --- | --- | --- |
| Text | errorId | variables | Notes |
| UNKNOWN\_NETWORK | 00405 | stp | If the networkId of an STP specified in the ERO is not known. |
| NO\_PATH\_FOUND | 00403 |  | The general error for a case when a path cannot be found. If a more specific error is known then that error should be returned. |
| VLANID\_INTERCHANGE\_NOT\_SUPPORTED | 00703 | stp | If the ERO is requesting label swapping between two STP within a network that does not support it. |
| STP\_UNAVALABLE | 00704 | stp | If an STP specified in the ERO is not available for the specified reservation criteria. |
| UNIDIRECTIONAL\_STP\_IN\_BIDIRECTIONAL\_REQUEST | 00706 | stp | If a unidirectional STP was specified in a bidirectional reservation request. |
| BIDIRECTIONAL\_STP\_IN\_UNIDIRECTIONAL\_REQUEST | 00707 | stp | If a bidirectional STP was specified in a unidirectional reservation request. |
| INVALID\_ERO\_FORMAT | 00708 |  | Format of ERO is invalid. This could be caused by an invalid ordering or other structural issues. |
| INVALID\_ERO\_MEMBER | 00709 | stp | Invalid ERO stp member detected. This could be caused by a null member, an intermediate STP not associated with an SDP, etc. |

Table – *serviceException* error values.

As an example, a requester agent issues a reservation request to the ESnet Aggregator NSA identified by *nsaId* urn:ogf:network:es.net:2013:nsa:nsi-aggr-west. The *ero* element contains an intermediate edge *stp* element that is resolvable within the NSI topology, but is not associated with an inter-domain SDP. The Aggregator NSA should detect this error during the pathfinding phase and generate a *reserveFailed* response with the following *serviceException* element:

<serviceException>  
 <nsaId>urn:ogf:network:es.net:2013:nsa:nsi-aggr-west</nsaId>  
 <connectionId>urn:uuid:92d54ff8-dec2-4be8-ae9e-3c0244f2c82b</connectionId>  
 <serviceType>http://services.ogf.org/nsi/2013/12/descriptions/EVTS.A-GOLE</serviceType>  
 <errorId>00709</errorId>  
 <text>INVALID\_ERO\_MEMBER: Invalid ERO member detected (urn:ogf:network:grnet.gr:2013:topology:CLIENT\_port\_16).</text>  
 <variables>  
 <variable namespace="http://schemas.ogf.org/nsi/2013/12/services/point2point#p2ps" type="stp">  
 <value>urn:ogf:network:grnet.gr:2013:topology:CLIENT\_port\_16</value>  
 </variable>  
 </variables>  
</serviceException>

# Glossary

|  |  |
| --- | --- |
| Aggregator NSA (AG) | The Aggregator NSA is a Provider Agent that acts as both a requester and provider NSA. It can service requests from other NSA, perform path finding, and distribute segment requests to child NSA for processing. |
| Connection Service (CS) | The NSI Connection Service is a service that allows an RA to request and manage a Connection from a PA. See [OGF NSI-CS]. |
| Explicit Routing Object (ERO) | An Explicit Routing Object (ERO) is a parameter in a Connection request. It is an ordered list of STP constraints to be used by the inter-Network pathfinder. |
| Network Service Agent (NSA) | The Network Service Agent is a concrete piece of software that sends and receives NSI Messages. The NSA includes a set of capabilities that allow Network Services to be delivered. |
| Network Service Interface (NSI) | The NSI is the interface between RAs and PAs. The NSI defines a set of interactions or transactions between these NSAs to realize a Network Service. |
| Requester/Provider Agent (RA/PA) | An NSA acts in one of two possible roles relative to a particular instance of an NSI. When an NSA requests a service, it is called a Requester Agent (RA). When an NSA realizes a service, it is called a Provider Agent (PA). A particular NSA may act in different roles at different interfaces. |
| Service Domain (SD) | A Service Domain is an NSI network construct that defines the Service connectivity such that any STP within a service domain can be connected to any other STP within the same Service Domain. |
| XML Schema Definition (XSD) | XSD is a schema language for XML. See [W3C XSD] |
| eXtensible Markup Language (XML) | XML is a markup language that defines a set of rules for encoding documents in a format that is both human-readable and machine-readable. |

# Contributors

John H. MacAuley, ESnet

# Intellectual Property Statement

The OGF takes no position regarding the validity or scope of any intellectual property or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; neither does it represent that it has made any effort to identify any such rights. Copies of claims of rights made available for publication and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementers or users of this specification can be obtained from the OGF Secretariat.

The OGF invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights, which may cover technology that may be required to practice this recommendation. Please address the information to the OGF Executive Director.

# Disclaimer

This document and the information contained herein is provided on an “As Is” basis and the OGF disclaims all warranties, express or implied, including but not limited to any warranty that the use of the information herein will not infringe any rights or any implied warranties of merchantability or fitness for a particular purpose.

# Full Copyright Notice

Copyright (C) Open Grid Forum (2012-2015). Some Rights Reserved.

This document and translations of it may be copied and furnished to others, and derivative works that comment on or otherwise explain it or assist in its implementation may be prepared, copied, published and distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice and this paragraph are included as references to the derived portions on all such copies and derivative works. The published OGF document from which such works are derived, however, may not be modified in any way, such as by removing the copyright notice or references to the OGF or other organizations, except as needed for the purpose of developing new or updated OGF documents in conformance with the procedures defined in the OGF Document Process, or as required to translate it into languages other than English. OGF, with the approval of its board, may remove this restriction for inclusion of OGF document content for the purpose of producing standards in cooperation with other international standards bodies.

The limited permissions granted above are perpetual and will not be revoked by the OGF or its successors or assignees.

# References

[GFD.212] OGF GFD-I.212, Network Service Interface Connection Service, v2.0.

[GFD.213] OGF GFD-I.213, Network Services Framework v2.0.

[GFD.213] OGF GFD-I.217 NSI Signaling and Path Finding.