

Service Termination Points (STPs) in NSI

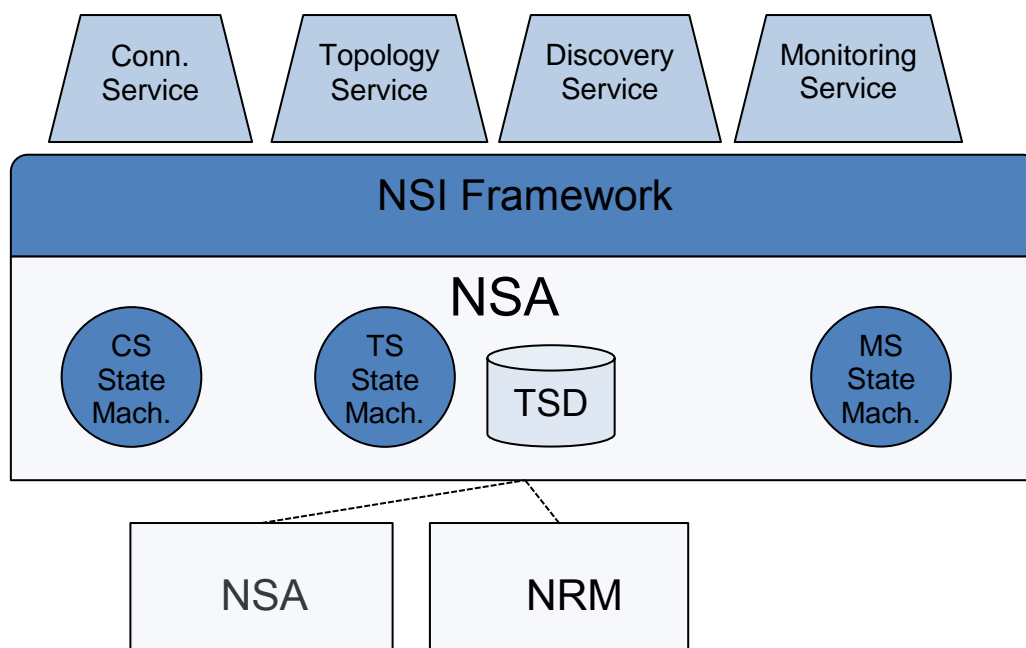
Scope

This document summarises the usage of STPs in NSI. Only the usage of STPs in the Connection Service is considered here, usage of STPs in the Topology Service and in the Monitoring Service is out of scope. The details of how STPs map to NML objects is described in the NSI Topology Service.

Agreed decisions relating to STPs

NSI framework

- The following set of services are supported:
 - Topology Service (TS)
 - Connection Service (CS)
 - Monitoring Service (MS)
 - Discovery Service (DS)
- The TS, CS and MS each have their own state-machine/s each of which is associated with an NSA. This is shown diagrammatically here:



Definitions, usage and description of STPs

- The NSI Topology is referenced in several places:
 - STP concepts are to be introduced in the NSI Framework
 - Use of STP for CS described are to be described in the CS protocol document
 - The Topology Service will describe how STPs map to NML and will define how to exchange topology

This document considers only the usage of STPS in the Connection Service

Definition of a Network

- A Network is a group of STPs that belong to a common topology description.

- A 1:1 relation exists between an NSA instance and an associated Network.

Definition of an STP

- A Network has a set of STPs which are announced using the topology service.
- STPs (Service Termination Points) are identifiers and are globally unique and persistent
- STPs point to an NML port described using the NML port syntax
- STPs are used in Connection requests to identify the endpoints of a Connection, and optionally as intermediate points on the path of the Connection to be used as path finding constraints.
- A confirmation of a connection request will return the two STPs that form the service end-points

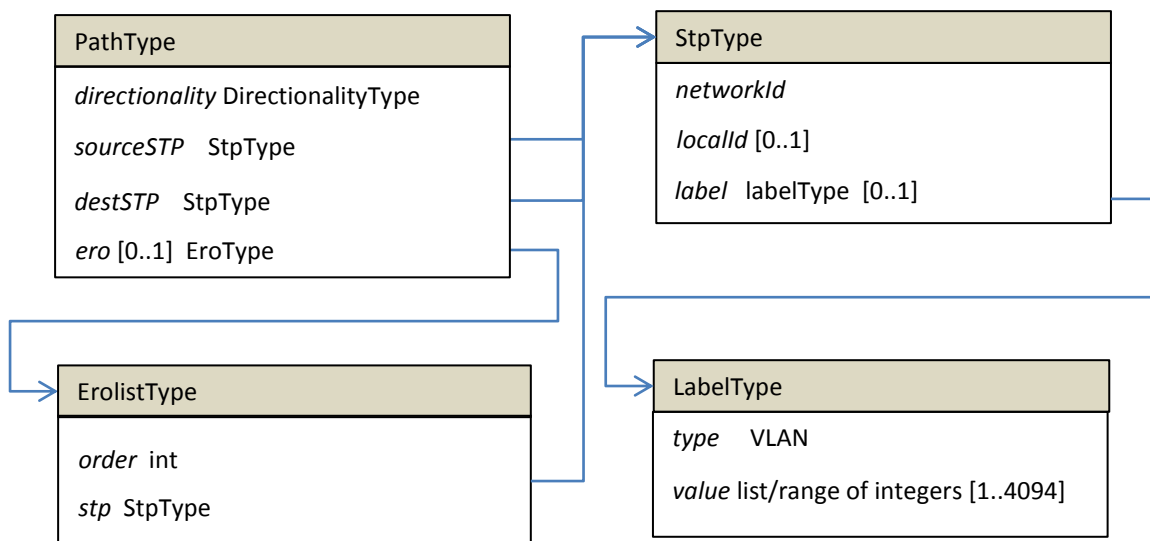
Note: these concepts around STPs and Networks remain as currently described in the NS Framework.

Definition of an SDP

- An SDP (Service Demarcation Point) is defined as a grouping of two peering STPs on the edge of different Networks.

Connection Requests in NSI v2.0

In v2.0 a connection request includes a <path> object. The 'Path' describes the endpoints and routing constraints of a requested connection.



Paths

- A path has a directionality attribute which indicates if the path is of type uni or bi directional.
- A path contains sourceSTP which is the STP of the beginning of the path
- A path contains destSTP which is the STP of the end of the path
- A path contains an ordered list of ERO's each of which is an STPs which is to be used as a routing constraints. These constraints must be used in the route of the path in the sequence provided in the ERO list. The path computation may return a path with additional STPs beyond those in the ERO list.

STPs

- An STP is constituted of networkId, localId and Label

NetworkId

- The networkId is a globally unique identifier that identifies the Network. Rather than forcing parsing of an STP to determine the Network, a separate Network object is defined to allow an intermediate NSA to forward the message to the target Network without needing to know about the STPs within that domain.
- The syntax of <networkId> is urn:ogf:network:<DNSname>:<date>:nsi:net:<NSInetwork>, where:
 - urn:ogf:network:<DNSname>:<date> conforms to GFD.191 and ensures that the STP is globally unique.
 - <DNSname> is a registered domain name.
 - <date> is a year in case the domain name is reused.
 - <NSInetwork> is the name of the dynamic service network.

localId

- A <localId> is an opaque string which is unique to the NSI network. The string must conform to URN characters.

Label

- A <Label> conforms to type LabelType

LabelType

- LabelType is a type-value pair describing an optional technology label.
- Type is a string which defines a technology specific attributes. Currently only 'VLAN' is supported. In this case VLAN is defined in the Service Definition to conform to <http://schemas.ogf.org/nml/2013/10/ethernet/vlan>. No other technology types are currently defined. In future other technology specific attributes may be added.
- For type 'VLAN', the value can be a list or range of integer values of 1 through 4094.

Candidate/Instance

- Connection **requests** are made up of 'candidate STPs' where a candidate STP describes a list or range of labels (VLANs). E.g 118-259 or 118, 342,4,259
- Connection **confirmations** will return an 'STP instance' i.e no label part.

Service Definitions and STPs

The Service Definition will state the framing of the service. For example if the service type is 802.1q, then the sourceSTP and destSTP LocalId part of an STP instance must point to a VLAN. In the case where a candidate STPs is requested, the Label must be of type VLAN and no other type will be accepted.

Example of Network/port/label mapping

Requested STP candidate		Returned STP instance	
VLAN range	3000-3600	VLAN	3450
Ethernet port	2-3-4	Ethernet port	2-3-4
Switch	sw1.lon.uk	Switch	sw1.lon.uk
PoP	London	PoP	London
NSA	62.40.112.34	NSA	62.40.112.34
NSI network	BoDservice	NSI network	BoDservice

DNSname geant.net

DNSname geant.net

STP type	Syntax
Ethernet port instance	<networkId> == urn:ogf:network:geant.net:2013:nsi:net:BoDservice <localId> == sw1.lon.uk:2-3-4
VLAN instance	<networkId> == urn:ogf:network:geant.net:2013:nsi:net:BoDservice <port> == sw1.lon.uk:2-3-4:3450
Candidate VLAN range	<networkId> == urn:ogf:network:geant.net:2013:nsi:net:BoDservice <port> == sw1.lon.uk:2-3-4 <label type> == VLAN <label value> == 3000-3600

Re-advertising STPs (network indirection)

It is legitimate for a Network to advertise a set of STPs some of which come from underlying providers. Eg NorthernLightDS advertises an STP as being part its own network when it is in fact originally assigned as SunetDS STP.

Option 1: the STP is advertised as a local STP and the SUNet local identifier is mapped to a new local identifier. Syntax for this case:

original STP:

```
<networkId>==urn:ogf:network:su.net:2007:nsi:net:SUNetDS  
<localId>==s01p03
```

Nordunet would advertise this:

```
<networkId>== urn:ogf:network:nordu.net:2007:nsi:net:NorthernLightDS  
<localId>==NL_s01p03
```

So there is a new local port identifier generated which NorthernLightDS can easily locally map to a SUNet port. (how mapping to new localId is done is not part of NSI protocol standard)

Option 2: If a NSA wishes to advertise that it can handle a 3rd party STP. In this case NorthernLightDS will simply advertise STP:

```
<networkId>==urn:ogf:network:su.net:2007:nsi:net:SUNetDS  
<localId>==s01p03
```

Path computation

- The NSI Connection Service supports only v2.0 flat pathfinding i.e layer adaptations are assumed to not be present.

Appendix 1: Examples

Example v1.1 NSI Connection request

```
<path>  
  <directionality>Bidirectional</directionality>  
  <sourceSTP>  
    <stpId>urn:ogf:network:stp:czechlight.ets:ps-80</stpId>  
  </sourceSTP>
```

```
<destSTP>
  <stpId>urn:ogf:network:stp:czechlight.ets:ams-80</stpId>
</destSTP>
</path>
```

Example v2.0 NSI Connection request

<!-- Example 1: bidirectional path request, untagged port instances, no ERO -->

```
<path>
  <!-- Two STPs - a bidirectional path -->
  <directionality>Bidirectional</directionality>
  <sourceSTP>
    <networkId>urn:ogf:network:cesnet.cz:2011:nsi:net:czechlight</networkId>
    <localId>intf3-2-1</localId>
  </sourceSTP>
  <destSTP>
    <networkId>urn:ogf:network:surfnet.nl:2001:nsi:net:netherlight</networkId>
    <localId>switchAport2-1</localId>
  </destSTP>
</path>
```

<!-- Example 2: bidirectional path request, tagged port instances, no ERO -->

```
<path>
  <!-- Two STPs - a bidirectional path -->
  <directionality>Bidirectional</directionality>
  <sourceSTP>
    <networkId>urn:ogf:network:nordu.net:2012:nsi:net:northernlight</networkId>
    <localId>intf3-2-1VLAN=2310</localId>
  </sourceSTP>
  <destSTP>
    <networkId>urn:ogf:network:sne.science.uva.nl:2012:net:science.uva.nl</networkId>
    <localId>switchAport2-1VLAN=2322</localId>
  </destSTP>
</path>
```

<!-- Example 3: bidirectional path request, Port candidate - use a VLAN within label range, no ERO -->

```
<path>
  <!-- Two STPs - a bidirectional path -->
  <directionality>Bidirectional</directionality>
  <sourceSTP>
    <networkId>urn:ogf:network:nordu.net:2012:nsi:net:northernlight</networkId>
    <localId>intf3-2-1</localId>
    <label>
      <type>VLAN</type>
      <value>1719-1834,2103-2106</value>
    </label>
  </sourceSTP>
  <destSTP>
    <networkId>urn:ogf:network:sne.science.uva.nl:2012:net:science.uva.nl</networkId>
    <localId>switchAport2-1</localId>
  </destSTP>
</path>
```

```

</sourceSTP>
<destSTP>
  <networkId>urn:ogf:network:sne.science.uva.nl:2012:net:science.uva.nl</networkId>
  <localId>switchAport2-1</localId>
  <label>
    <type>VLAN</type>
    <value>451,341,486</value>
  </label>
</destSTP>
</path>

```

<!-- Example 4: bidirectional path, tagged port instances, with ERO -->

```

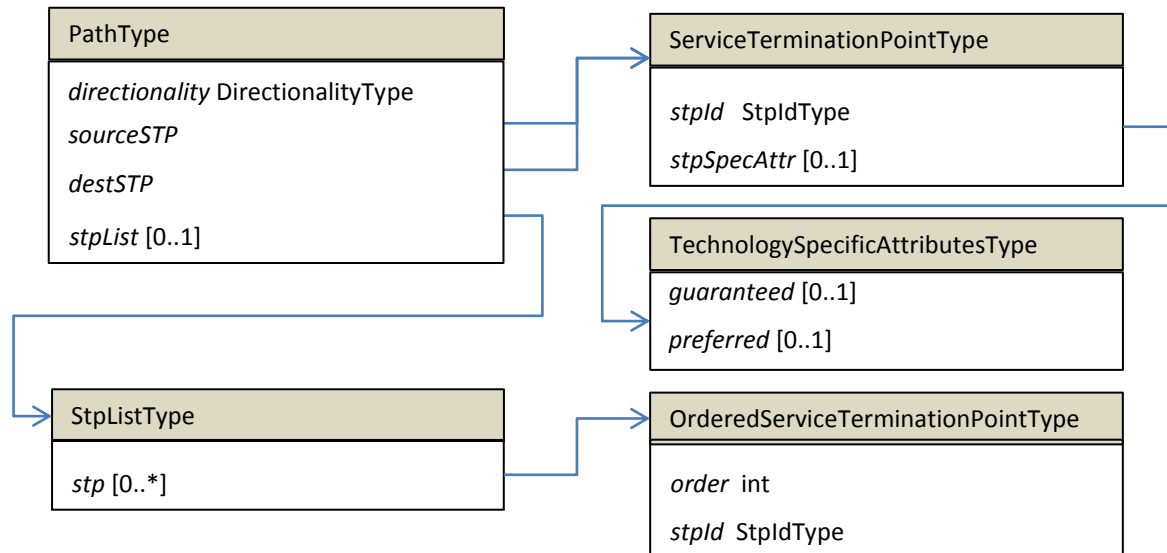
<path>
  <!-- Two STPs - a bidirectional path -->
  <directionality>Bidirectional</directionality>
  <sourceSTP>
    <networkId>urn:ogf:network:nordu.net:2012:nsi:net:northernlight</networkId>
    <localId>intf3-2-1VLAN=2310</localId>
    <label>
      <type>VLAN</type>
      <value>1719-1834,2103-2106</value>
    </label>
  </sourceSTP>
  <destSTP>
    <networkId>urn:ogf:network:sne.science.uva.nl:2012:nsi:net:science.uva.nl</networkId>
    <localId>switchAport2-1VLAN=2322</localId>
    <label>
      <type>VLAN</type>
      <value>451,341,486</value>
    </label>
  </destSTP>
  <eroList>
    <!-- first STP in ERO list, order = 1. -->
    <order>1</order>
    <stp>
      <networkId>urn:ogf:network:nordu.net:2012:nsi:net:northernlight</networkId>
      <localId>intf7-2-1:VLAN=2310</localId>
    </stp>
  </eroList>
  <eroList>
    <!-- second STP in ERO list, order = 2. -->
    <order>2</order>
    <stp>
      <networkId>urn:ogf:network:sne.science.uva.nl:2012:nsi:net:science.uva.nl</networkId>
      <localId>switchZport3-1</localId>
    </stp>
  </eroList>
</path>

```

Appendix 2: NSI v1.1

Connection Requests in NSI v1.1

In v1.1 a connection request includes a <path> object:



Connections but be identified as being either bidirectional or unidirectional.

Where <stpList> is optional ordered list of type <STP>.

The NSI v1.1 syntax for an STPID is: urn:ogf:network:stp:<networkId>:<localId>

Issues with this approach:

1. Not compliant with Freek's new GFD URN naming.
2. Handling of unidirectional STPs is clumsy for proposes of monitoring.
3. The routing of a Connection could be ambiguous as end STPs do not have an inherent direction. (ingress/egress interpretation problem)
4. The Network associated with an STP has to be found by parsing the URN.
5. Where an STP is re-advertised by a federating NSA, a new STP is created with the NetworkId of the Federating Network, this could be resolved by assigning a separate <Network> object.
6. Candidate VLANs ranges are not supported in connection requests.