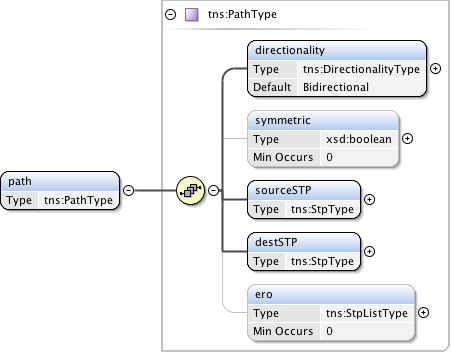
# Path and STP in NSI v2.0 Connection Service WSDL

**Path**

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



A <path> has a directionality attribute which indicates if the path is of type uni or bi directional. Unidirectional Connections transport data from sourceSTP to destSTP. By default the directionality is bidirectional.

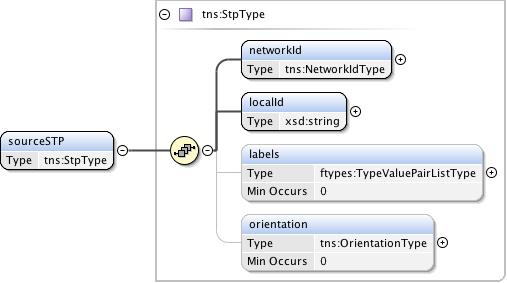
C:\Users\guy\AppData\Local\Microsoft\Windows\Temporary Internet Files\Content.Word\ogf_nsi_connection_types_v2_0_xsd_Element_directionality.png

Symmetric attribute is used to indicate if the go and return directions of a bi-directional path must use the same routing.

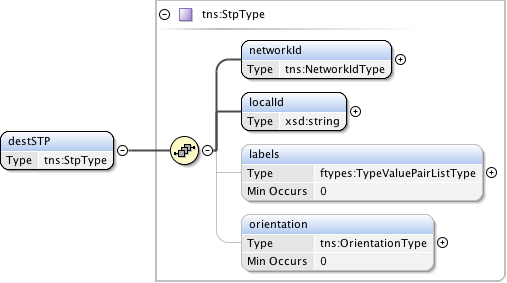
C:\Users\guy\AppData\Local\Microsoft\Windows\Temporary Internet Files\Content.Word\ogf_nsi_connection_types_v2_0_xsd_Element_symmetric.png

**STP**

A path contains sourceSTP which is the STP of the beginning of the path. This can be either a uni-directional or bi-directional.



A <path> contains destSTP which is the STP of the end of the path. This can be either a uni-directional or bi-directional.



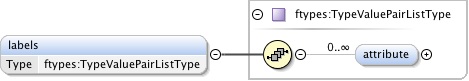
An STP is constituted of networkId, localId and Label

**NetworkId**

* The networkId is a globally unique identifier that identifies the Network.
* The syntax of <networkId> is urn:ogf:network:<DNSname>:<date>:<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> has syntax of <networkId> is urn:ogf:network:<DNSname>:<date>:<LocalId>, 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.
  + <LocalId> is the local part of STP



**Label**

* Label is a type-value pair describing an optional technology label.
* Where type is defined in the topology, e.g nmleth:vlan
* And were value is of type string (note: value is opaque in this message, however to be meaningful musts conform to type defined in relevant topology). Lists and ranges are allowed.

C:\Users\guy\AppData\Local\Microsoft\Windows\Temporary Internet Files\Content.Word\ogf_nsi_connection_types_v2_0_xsd_Element_orientation.png

**Orientation**

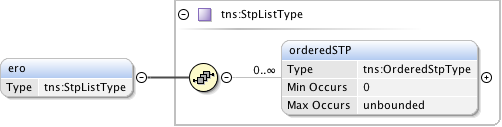
* Orientation attribute is used to assign a direction to a bidirectional STP to remove ambiguity when performing path computation. Orientation type is ingress or egress. Ingress/Egress is relative to the Network.

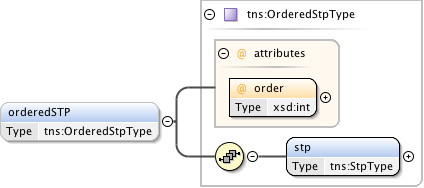
**Service Definitions and STPs**

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 reference STPs is requested, the Label must be of type nmleth:vlan (or equivalent) and no other type will be accepted.

The Service Definition may include an attribute to validate the STP label type for sourceSTP and destSTP.

A path may also contain an optional Explicit Route Object (ERO). The ERO is an ordered list of STPs which are to be used as constraints in path computation. These transit points must be incorporated into the path in the sequence provided in the ERO. The path computation may return a path with additional STPs beyond those in the ERO list.





|  |  |
| --- | --- |
| **NML object** | **STP type example** |
| NML:Ethernet port | <networkId> == urn:ogf:network:geant.net:2013:BoDservice  <localId> == urn:ogf:network:geant.net:2013:sw1.lon.uk:2-3-4 |
| NML:VLAN instance | <networkId> == urn:ogf:network:geant.net:2013:BoDservice  <localId > == urn:ogf:network:geant.net:2013:sw1.lon.uk:2-3-4:3450 |
| A range of candidate ML:VLAN | <networkId> == urn:ogf:network:geant.net:2013:BoDservice  <localId> == urn:ogf:network:geant.net:2013:sw1.lon.uk:2-3-4  <label type> == nmleth:vlan  <label value> == 3000-3600 |

Note: more examples needed to show opaque nature of localId and networkId.

## 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.

Note: do we allow multiple NSAs for each Network? This needs to be clarified…

**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:SUnetDS

<localId>== urn:ogf:network:nordu.net:2007:s01p03

Nordunet would advertise this:

<networkId>== urn:ogf:network:nordu.net:2007:NorthernLightDS

<localId>== urn:ogf:network:nordu.net:2007: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:SUnetDS

<localId>== urn:ogf:network:su.net:2007:s01p03

**Path computation**

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