**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.

StpType

StpListType

*order* int

*stp* StpType

PathType

*ero* [0..1] StpListType

*sourceSTP* StpType

*destSTP* StpType

*localId* [0..1]

*networkId*

*label* labelType[0..1]

LabelType

*type* VLAN

*value* list/range of integers [1..4094]

*directionality* DirectionalityType

Figure : Option 1: if directionality is ‘unidirectional’, then STPs point to a nml:port, if the directionality type is ‘bidirectional’, then STPs points to a nml:bidirectionalPort

StpType

StpListType

*order* int

*stp* StpType

PathType

*ero* [0..n] StpListType

*ZendSTP* StpType

*srcLocalId*

*NetworkId*

*sinkLabel* labelType[0..1]

LabelType

*type* nmlType

*value*  string

*AendSTP* StpType

*sinkLocalId*

*srcLabel* labelType[0..1]

Figure : Option 2: STPs are a grouping of 2 unidirectional tuples <network><localId><label>, each tuple points to an nml:port. Were unidirectional connections are requested, the only one of the src/sink tuples are used.

StpType

StpListType

*order* int

*stp* StpType

PathType

*ero* [0..n] StpListType

*ZendSrcSTP* StpType

*ZendSinkSTP* StpType

*localId*

*networkId*

*label* labelType[0..1]

LabelType

*type* nmlType

*value*  string

*AendSinkSTP* StpType

*AendSrcSTP* StpType

Figure : Option 3: STP are always unidirectional and point to an nml:port. In this case unidirectional connections are identified by using either srcSTP or sinkSTP at each end as suitable.