# Distribution of NSI Topology

Some form of Topology distribution is required in order for an inter-domain NSI network to function. In NSI 1.0 this process was performed out-of-band, mostly through e-mail. For NSI 2.0 we take the opportunity to define an NSI Topology Service for NSI topology exchange, which can support the NSI Connection Service.

## Transport and Service plane relationship

The NSI Connection Service is implemented on Network Service Agents (NSAs), which together form a network and tree-like structure. This graph represents how reservation requests would propagate through the network, but not necessarily reflects the transport-plane. One NSA may be an aggregation point for other NSAs, not visible from the outside.

The messaging between the NSAs will happen on the service plane, which is completely separate from the transport plane.

## Elements of a Topology Exchange Mechanism

There are three main elements of topology exchange:

### **Bootstrapping Topology Exchange**

To start the initial Network Service the NSAs must be able to find each other, in order to communicate details about the network. So some form of bootstrapping is required, with initial synchronization between domains on both the service plane and the transport plane, i.e. the NSAs of both domains must be able to contact each other, and the details of transport plane connections between them have to be synchronized as well.

Initiating a transport plane connection between two networks is not a frequent occurrence, and a longer process, involving out-of-band (for NSI) contact. Part of that process can be that the networks also communicate the access details for the NSAs, thus forming an NSA relationship.

### **Expanding the Topology Exchange**

Once the neighboring (on the control-plane) NSAs have exchanged details, they can also distribute details about the rest of the network, both the control plane details and connectivity, but also some transport details.

### **Update Mechanism**

The transport network is not static, and links are added or removed from time to time. An update mechanism is thus required to inform other NSAs about these kinds of events.

## Topology Exchange Implementations

The above mechanisms can be implemented in five different ways:

#### **Centralized Manual Distribution**

**An initial attempt at topology distribution in the Automated GOLE demonstration was through a central maintainer. This maintainer collected all topology information from the networks involved, gathered all the topology data and sent out a topology file through e-mail. The network maintianers would than download the attachment and insert it into their provisioning system. Updates to the topology were all handled through the central maintainer, distributed through e-mail.**

**While this system worked initially, it soon ran into scaling problems. This system also does not allow to have a good way of doing automatic updates or insertion.**

#### **Version Controlled Distribution**

**The Automated GOLE demonstration has transitioned into a different distribution mechanism using a Git source code repository, available on GitHub. This mechanism also still has a central maintainer, this also allows networks to manipulate their own topology information. The distribution mechanism is either directly from the GitHub website, or through the git version control system itself. The git system has the added advantage that it is a distributed version control system, so it is not required to download directly from GitHub.**

**Bootstrapping and updating all happens through the git system itself.**

#### PerfSonar Lookup Service

The PerfSonar monitoring system suite also contains a service for looking up information. This service uses a “home Lookup Service” (hLS) where metadata of information is registered, which is then uploaded into the “global Lookup Service” (gLS) (this can be cloud of services).

The retrieval of information happens by first querying the gLS, then the relevant hLS, followed by the service where the actual information is stored.

Since topology information would be stored locally, no update mechanism is necessary, except for location changes of the services itself. However, this method of storing and lookup does require full connectivity between all NSAs to provide and retrieve information, which may not be possible.

#### **HTTP Distribution**

**A common way of distributing information is using the HTTP protocol. The topology files would include links to topology description files, which would allow other domains to directly fetch the topology description.**

**However, as with the Lookup Service, this requires direct access between NSAs which may not be possible.**

#### **A Peer-to-Peer Distribution Protocol**

**Another method is to define a new protocol for NSI topology distribution. As explained above the protocol would only require a small set of primitives, and would work directly between peering NSAs.**

**Once an NSA comes up, it contacts its neighbors to request the topology information that they know about, and subscribes to future topology updates. These updates are propogated in a peer-to-peer fashion through the whole network.**

**The end-result would be that all the NSAs have a global view of the topology, with only local interaction.**

## **Summarizing Topology Distribution**

**Above we have described five different topology distribution mechanisms, from these five only the version controlled and peer-to-peer systems fit the requirements that the NSI has, others require too much manual operation, or require globally reachable NSAs. The peer-to-peer system would be the best solution with minimal interaction between systems, and no reliance on outside mechanisms. For practical reasons we believe the best solution right now is a version controlled distribution system, and in time evolve to a peer-to-peer distribution protocol.**