NORDUnet BoD Service

OGF40 Oxford, January 15th 2014

Henrik Thostrup Jensen <htj@nordu.net>

Overview

ETA: TNC2014 (early May)

Working, not perfect and shiny

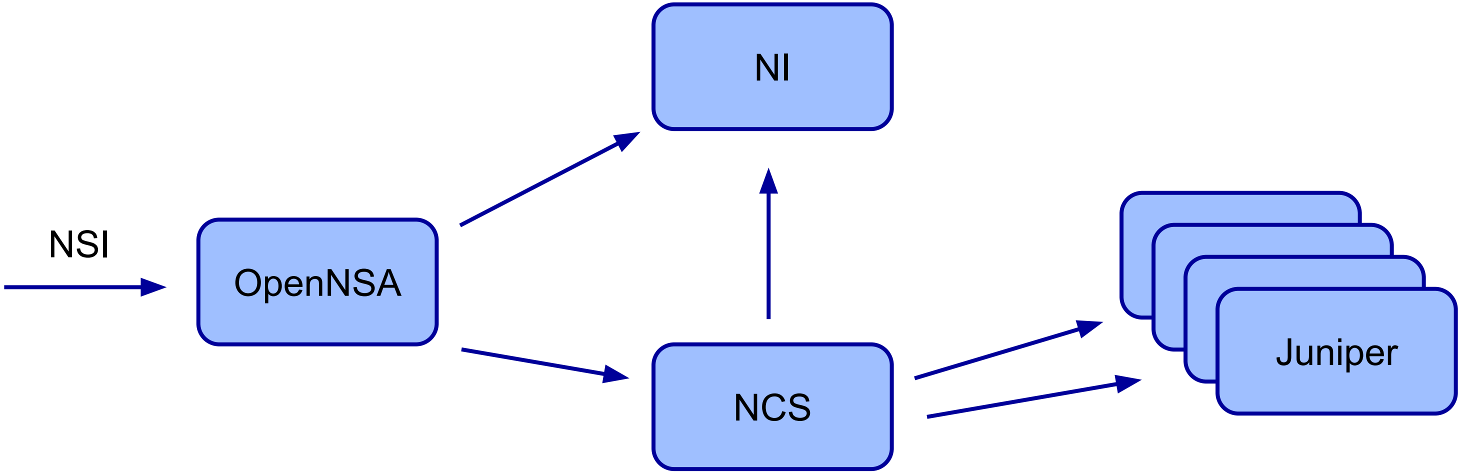
NSI2 CS r116

Stack: OpenNSA + NCS + NI + Juniper

Close cooperation with SURFnet

Some augmentations for the CS protocol

Technical Setup



NCS = Network Control System (commercial product)

NI = Network Inventory

Technical Setup

Demarcation points will be Ethernet + VLAN

MPLS tunnels internally

Initially no flow control / limiting / shaping

- Likely to be added later

Most of this will be over 100G pipes

NORDUnet-SURFnet Cooperation

Purpose: Create NSI production infrastructure

Involves a lot more than just the CS protocol

Group intentionally kept small, but open about what is happening

Close communication with AutoBAHN

Protocol augmentations

- AAI
- Connection Traces
- Topology based on announcing reachability

AAI + Connection Traces will be covered by Hans Thursday

Topology

Announce reachability, instead of global port knowledge

The model is always chain, control plane = data plane

An NSA ONLY talk to its peers

Exported topology complies with NML schema (for revised schema)

Pathfinding = 20 lines of straightforward Python code

```
<nsi:NSA id="urn:ogf:network:surfnet.nl:nsa">
  ...
  <gns:TopologyReachability>
    <nml:Topology id="urn:ogf:network:nordu.net.nl:topology" gns:cost=5>
    <nml:Topology id="urn:ogf:network:sunet.se:topology"    gns:cost=10>
    <nml:Topology id="urn:ogf:network:deic.dk:topology"     gns:cost=15>
  </gns:TopologyReachability>
  ...
</nsi:NSA>
```

OpenNSA

Used on several sites (6ish)

Open source (BSD), multiple backends, easily hackable

Updated to r116 in early january (queryNotification not done)

Interoperability test pending

Likely to support both NML and topology reachability

NORDUnet augmentation/integration will be kept optional / easily separable

```
git clone git://git.nordu.net/opennsa.git
```

Questions?

Bonus slide 1: AAI

Two request types

- Transit : Has at least one endpoint towards a customer
- Termination : Request user/group/token is checked against STP

SAML Example

```
<nsi:sessionSecurityAttr>
  <s:Attribute Name="user">
    <s:AttributeValue> htj@nordu.net </s:AttributeValue>
   </s:Attribute>
   <s:Attribute Name="group">
     <s:AttributeValue>nordu.net</s:AttributeValue>
     <s:AttributeValue>dev.nordu.net</s:AttributeValue>
   </s:Attribute>
</nsi:sessionSecurityAttr>
```

Bonus slide 2: Connection Traces

We need to see origin of request

CS protocol does not provide this functionality

Looks like this (goes in the NSI header)

```
<gns:ConnectionTrace>
  <gns:Connection>urn:ogf:network:aruba:2013:nsa:AR-Tfe07c58e3f</gns:Connection>
  <gns:Connection>urn:ogf:network:bonaire:2013:nsa:BO-s7780</gns:Connection>
  <gns:Connection>urn:ogf:network:curacao:2013:nsa:CU-1234</gns:Connection>
</gns:ConnectionTrace>
```

Can also be used for loop detection