# NSI Service Table
# A Sortof-Topology Suggestion

Henrik Thostrup Jensen htj@nordu.net

NSI Phone Call Presentation 25th February 2015

# Introduction

- NSI = Network Service Interface
  - Service…
  - NSI support multiple service types
  - We have spend a lot of time modeling technology
    - Not so much on services
    - We actually haven't modeled topology either…
- This suggestion is about modeling services
  - And only a little bit on topology

# Services & NML

- EVTS = ethernet#vlan in NML?
- How to tell burst policies from an NML port?
  - And how to map this?
- The idea that path finders must map between services, topology and technology capabilities makes them very complicated
- Security and Policies left as an exercise…

# Service Table

- Inspiration from BGP
  - BGP is the result of a lot of real-world experience and routing research
  - Remember: IP is a service
  - Policy is expressed with reachability and exit discriminators
  - Connectivity with AS paths
  - BGP lesson: Try not to do clever things
- It is not really a topology
  - We have painted ourselves into a corner
  - We have to describe the topology; how else could it work…

# Service Table

- Deliberate choices
  - Only describe demarcation / links
  - Model transit and network roles in path finding
- Tradeoffs
  - List capabilities, not fabric
  - Let the NSI Agent do the service -> technology mapping
    - We already do this in the reserve request, but not in the pathfinding

# Service Table

## Example

Network, id="urn:ogf:network:aruba", version="123"

Name: Aruba

Link id="urn:ogf:network:aruba:topology:link_a", demarcation=…

Name: LinkA

Service type=EVTS

ReachableNetwork id="urn:ogf:network:bonaire:topology", distance=1

Link id="urn:ogf:network:aruba:topology:link_b", demarcation=…

ServiceTransit type=EVTS

Service type=…

# Service Table

- Links & Services
  - What services can be provided and where to
  - This is only relevant if the link connects to you
    - Otherwise polices can be applied
  - Reachability has to be engineered to match policies
  - Technology of the link does NOT matter
    - Service mapping must be agreed between networks
      - Like IP
    - This encapsulates adaptation
      - Like IP

# Service Table

- ServiceTransit
  - Similar to BGP default route
    - Meaning all traffic can be send via that link
  - Allows simple configuration for network with a single transit provider
  - Many networks only have a single transit provider
  - Still possible to combine with peerings / PNIs

# Distribution

1. Fetch service table from peers
2. Apply policies / rules and build table

   Typically:

   Announce customers to customers + peers

   Announce peers to customers
3. Publish service table
4. Repeat at some interval

# Path Finding

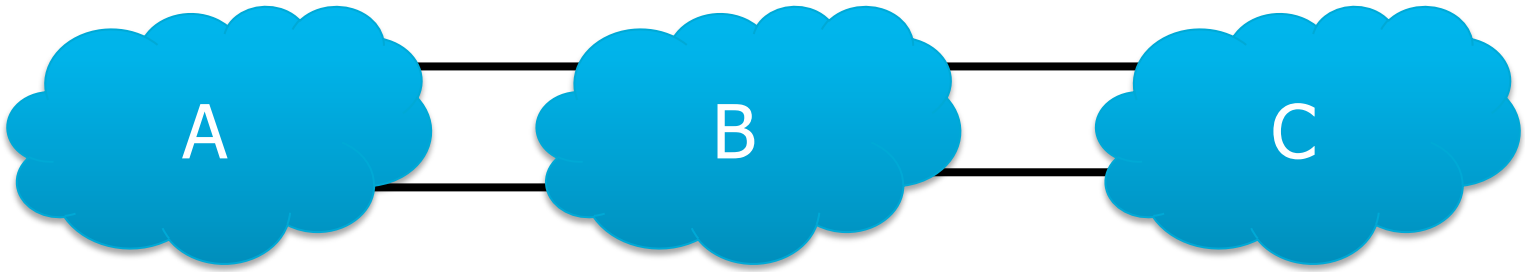
- Chain
  - But why!
- In most cases the networks to traverse to setup a circuit is trivial
  - Most cases look like this:

University – National NREN – Transit NREN – Transit NREN – National NREN – University

  - Notice the business relationships
    - No one wants to be your transit provider unless you give them money
  - The difficult part is to select the right link

# Path Finding

- Why chain
  - Most networks connect over multiple links




  - A & B know the reservations for A-B links
  - Having C choose which link to use between A-B is suboptimal
  - A/B can decide this the best

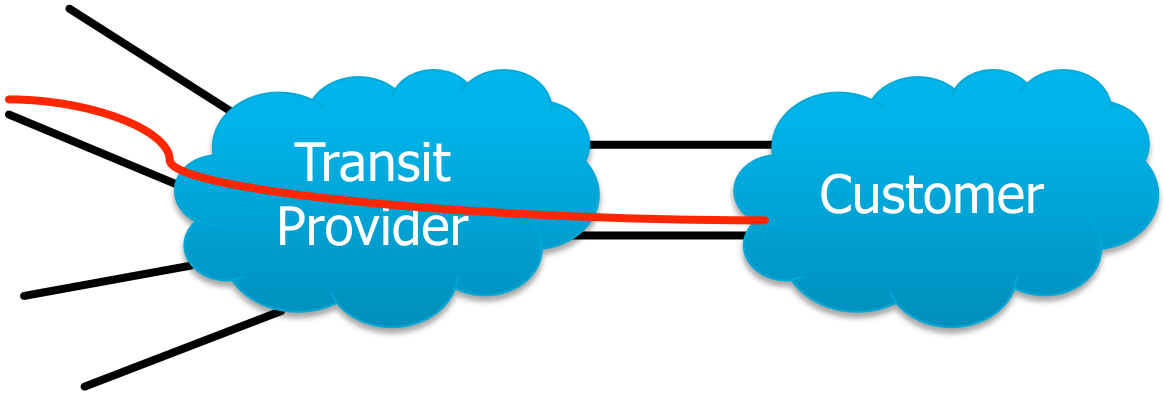
# Path Finding

- Two scenarios for transit providers
  1. Request from customer

     Allows transit provider to connect to the destination in the best way
  2. Request from peer/transit

     Allow the transit provider to verify that the customer agrees to the circuit. Avoids the situation where a peer sets up a circuit and the transit provider doesn't know if the customer has agreed to the link.

     The infrastructure of a transit provider is paid by the customers. Having outside parties reserve a circuit in the infrastructure without customer verification (as tree does) is highly problematic.



# Path Finding

- Chain is still compatible with EROs
  - Makes it possible to
  - Checking EROs with policy is easy

# Path Finding

- Resolving circuit allocation failure
- Service table lists links and their connectivity
  - Possible to build network model
  - Can do re-routing in case of failures
  - Mostly relevant for transit networks
  - Similar to AS paths in BGP can help with re-routing in case of failures
- End networks typically have low connectivity
  - Transit networks have high connectivity
  - In most cases, it makes sense to have transit provider try to re-route

# LHCONE Example

- Encode LCHONE VRF as a separate network

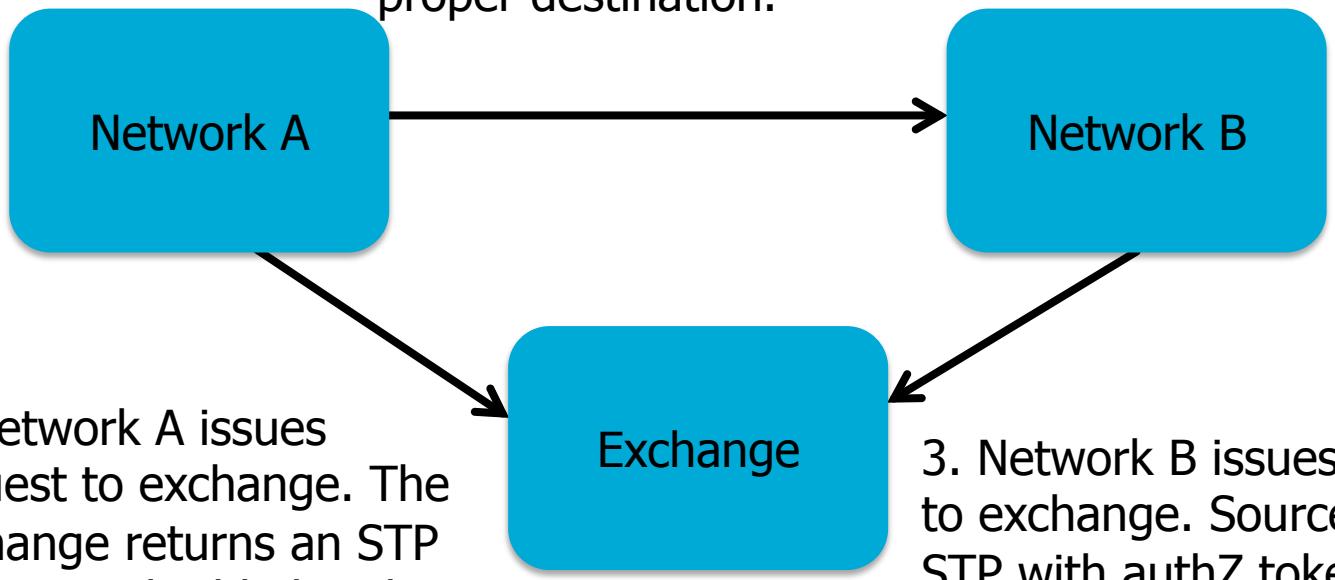
  urn:ogf:network:example.org:topology

  urn:ogf:network:example.org:lhcone
- Reachability can be defined on common or separate links
  - Some network run the VRF on separate infrastructure, some do along their general infrastructure
- This idea isn't fully baked
  - (but neither are the requirements AFAIK)
  - One problem is that ports cannot be in both

# Exchanges

- The NSI model assumes that networks demarcates on links
- On exchanges it demarcates in the switch fabric, as the networks owns/rents a port in the exchange
- A single reserve allocates resources across two networks
- Exchanges often don't care about policies, etc
  - Have to be applied by networks

# Exchanges


2. Network B issues request to B. Source is the STP with authZ token and proper destination.
Network A
Network B
Exchange
1. Network A issues request to exchange. The exchange returns an STP with an embedded authZ token
3. Network B issues request to exchange. Source is the STP with authZ token from A and destination is the port to network B

# Summary

- Problems that disappear
  - Topology distribution
  - Complex path finding
  - Proxy requests – makes revocation easy
- Solves
  - Transit policies & link AUPs
  - Adding new services is (more) straightforward
  - Exchanges are crossed in a way where other networks equipment are respected