NSI Topology/Service Model Suggestion

We currently have a number of topology suggestions. None of them can adequately describe complex transit policies. This document describes some of the issues of NML and GNS (The UvA suggestion is not a topology model), along with issues which we haven't considered yet. The document describes a new document for creating services and setting up paths. The model is not a classical topoogy model, but more of a service table for networks and links.

Overview

The main policy tool of NML is the switching service, which describes how ports in a node can be connected. This approach falls short when selective transit is provided, as this cannot be described as a switching service. It also means that it is not possible to enforce this policy locally by looking at the cross connect.

The GNS suggestion announces reachability to other networks, done at NSA level. Hans pointed out that it should really be per network. However there are cases where networks are connected over multiple links with different policies, meaning that reachability would have to be provided per link. This approach is then getting awfully close to that of BGP.

In BGP all reachability is explicit, as IP prefixes are announced (default route is the exception, but reachability is still explicit). Further, AS paths are announced, which makes it possible to build a network model. This enables faster recovery in case of link/router failures as it is possible to re-route traffic more efficiently. This is typically only done in backbone networks.

Unlike BGP, NSI does not have a mechanism for aggregating its address space (ports). Since Uppsala, we can infer the network from the port name, however there is no way indicating which ports are closer to a certain demarcation link, which makes optimal pathing difficult. However, it is possible to split a network into multiple smaller networks for this, but a metric is still needed. No good technical metric exists to announce distances, and due to link pricing and policies, a distance metric must be administrative (works well in BGP).

Network & Services

Given our decision to link STPs and network identifiers it becomes possible to do path finding to a network, instead to a specific port. This drastically reduces the amount of information a path finder must be aware of, and removes the problem of connecting to newly created STPs.

Path finding per network is somewhat difficult in NML as the problem of adaptation between technologies in order to get the right service requires intricate knowledge of the capabilities of the networks. IP solves this problem by being a single universal service that can live on top of different technologies. Since we want to be able to provide multiple service types through NSI, this will not work for us. We also been struggling quite a bit of tying together technology, adaptations and services in NML as the combination becomes complicated quite quickly.

Recently I've talked to several persons about the ANA infrastructure. The ANA infrastructure is interesting as the hardware comes from multiple vendors, and the infrastructure must provide different services (best effort circuits, circuits with guarantied bandwidth, and the same with protected circuits). Focusing on technological capabilities becomes quite messy in this scenario. Instead the important thing is to focus on what services can be provided, and let the NSI Agents handle the service to technology mapping. This is quite different from our previous approach.

In short: I think we need to move to a model that is service centric instead of focusing on the technology of the equipment.

Topology Model

I suggest a model that focuses on describing capability instead of the basic resources. It is based around two principles:

- Network Reachability
- Service Capability

The first allows us to express complex transit policies in simple and well proved way (BGP). The second allows us to abstract away technology specific models and focus on what can services can actually be provided. This model puts policy and services before technological capabilities.

By describing reachability, path finding becomes simple and predictable. This is major advantage over the graph search approach in NML which can produce extremely unfortunate effects such as cross-atlantic loops. Defining reachability will require manual work, however this will always be the case when describing policies.

Here is how the topology description could look like in XML:

```
<Network id="urn:ogf:network:aruba" version="123">
    <Name>Aruba</Name>
    <Link id="urn:ogf:network:aruba:bonaire-otn" demarcation="urn:ogf:network:bonaire:aruba-otn">
        <Name>Bonaire OTN Link</Name>
        <Service type="EVTS">
            <ReachableNetwork id="urn:ogf:network:bonaire" distance="1" />
            <ReachableNetwork id="urn:ogf:network:curacao" distance="2" />
        </Service>
        <Service type="OTN">
            <ReachableNetwork id="urn:ogf:network:bonaire" distance="1" />
        </Service>
    </Link>
    <Link id="urn:ogf:network:aruba:dominica-eth" demarcation="urn:ogf:network:dominica:aruba-eth">
        <Name>Dominica Ethernet Link</Name>
        <ServiceTransit type="EVTS" />
    </Link>
    <Service type="SwitchingService" />
</Network>
```

The topology description is really more of a transit table for services than an actual topology description (however we have never been describing topology, only networks and demarcations). Reachability is unidirectional and cannot be turned around, however the service provided might bidirectional. The description covers what the link can be used for from outside the network.

The above example describes a network (Aruba), with two links. First link is an OTN link towards the Bonaire network. This link provides two services: EVTS and OTN. The EVTS service can reach the bonaire and curacao network, where the OTN service can only reach bonaire. How the EVTS service is carried on the OTN link is up to the aruba and bonaire network. The details are encapsulated. The second link in the network is an ethernet link towards the dominica network. This link provides what is called service transit, for the EVTS service. This means that the dominica can use the link to connect to any EVTS endpoint in the NSI system. This is similar to the concept of a default route in BGP. Finally a switching service is listed in the network. This denotes the capability to create a service in the network.

There are several choices and tradeoffs in the design:

- Only describe demarcation links.
  This means significantly less data to distribute and that the data will be fairly stable. This also removes the need for complex distribution mechanisms.

- Each link lists supported services and reachable networks for each service type along with a cost.
  This makes path finding straightforward: Select the link that provides reachability to the desired network that has the lowest value.

  Cost is an administrative value. Lower is better (think hops).

  The service first structure was chosen as I suspect it will be the smallest, but it could as well be network first and services listed under that.

  If a network cannot bridge a service type from one link to another, reachability for that network is not announced on link. This encapsulates complicated technology restrictions in a simple way, that would otherwise have to be built-in to pathfinders.

- A link can provide service transit. This means that all networks supporting the service type should be reachable through that link. This would typically be used by a backbone network towards its customers.

  A customer may have other links that it could favor for some networks. Similar to how most networks have a transit provider, but will have private peering/links to other networks.

- Can describe network services

  Some networks may provide services that are hosted in the network. This could be a switching service which could be used to connect p2p links in order to build a multi-domain switched network (I'll refrain from debating if this is a good idea).

A list of potential network services:

- Switching service.

Will on creation return a number of STPs that can be used to connection P2P links

- Aggregation.

Aggregate the data from two or more unidirectional circuits into one.

- Ping service

Provides an STP with a pingable address.
The network may of course also supply these like today with pre-configured STP.

- PerfSonar

Would spawn a perfsonar instance and return an STP endpoint.

Label swapping capability is intentionally left out. We should stop engineering for old technology, and underspecified STPs goes a long way towards solving the issue if it should occur.

By describing services and the capabilities of the network in this way, it becomes easier to introduce new services and possible to describe policies that would otherwise be complicated or impossible to express with previous suggestions.

Pathfinding

As the networks explicitly describe reachability, an NSI Agent only needs to fetch the topology (service table) document from its peers to know which networks are reachable. This means that a document-distributing system is not necessary, as the connections between networks is largely static. The reachability document would only have to be updated at low intervals. For leaf networks, an NSA would simply be setup with another NSA as its transit provider. This NSA would then leave all pathfinding to be done by the backbone network. This situation is similar to how many organizations and smaller NRENs have a single transit provider, which does all their connectivity.

The scheme assumes a chain-like path setup, which is intentional. Setting up a circuit in chain mode allows each NSA to verify that the transit policies of the network is enforced as it can see the destination network, and check it against announced reachability. It is not possible to perform such a check for UPAs when a circuit is setup in tree mode, as only the crossconnect is know.

However chain mode will probably have to broken when crossing exchanges, as the NSI agents for such will typically be UPAs and not care about transit policies and link AUPs. Please refer to my previous email on secure crossing of exchanges for details on this. This means that we will end up with a hybrid pathfinding that does a tree with one branch being one layer deep each time an exchange is crossed.

As chain pushes the decision for how to connect downstream, it does not allow the same control over the circuit (unless using EROs). However delegating this decision is likely to lead to fewer circuit setup failures. Consider the following setup:

A = B = C

Three networks: A,B and C. A and B are connected via two links, and similar for B and C (the = sign is two links). If A wants to create a circuit to C, it will setup a link to B and ask B to setup a cross connect into C. If one of the circuits between B and C cannot support the capacity, B can automatically choose the link with enough capacity to C. Only B and C has the knowledge about the state of their link, and are hence better suited towards taking the decision of which one to use.

With explicit reachability and distances the process for setting up circuits becomes simpler and more predictable. The distance vector in particular makes sure that an optimal path is taken. This in contrast to NML that has no mechanism to prefer certain paths over others.

Handling Failure in Pathfinding

Handling failure circuit(service) setup failures and attempting alternate paths (re-pathing) with only reachability information can be a bit tricky. This section covers how this can be solved.

Currently we have the notion that aggregators should not try and fix failures, only return them to the UPA. If we relax this latter notion, it opens up the possibility for aggregators along the path to try alternate paths. However such attempts should be restricted to certain behavior, e.g., a backbone network can try alternative links towards a customer. Re-pathing should not use transit/peers towards customers or use transit to go to peers.

If the notion of aggregators not being able to re-pathing holds, it is possible for a UPA to perform re-pathing, by constructing an ERO that doesn't take the link without enough capacity. Constructing such an ERO is likely to require more information than what is available from the topology of the peers and transit provider for the UPA. To construct the ERO it would have to fetch the topology of the backbone networks to build up a model of the network to find an alternative path. The reason only backbone networks is relevant is that end organization are typically multi-homed into the same provider, which makes smaller network largely irrelevant for re-pathing. Since the number of backbone network is relatively small and the information in their topology description is largely static, fetching them directly is simple and convenient.

Performing re-pathing at the UPA is however suboptimal. Leaf networks will typically be very limited in their connectivity, where the service provider (backbone network) will have a much degree of connectivity. Hence it makes sense for the service provider to perform re-pathing. This is similar to how BGP re-routing is not done by smaller tier-2 and tier-3 networks. Re-pathing at smaller networks does not make sense as their connectivity is typically rather limited. However customers and providers can agree where and if re-pathing should be done (similar to how some IP customers get a full routing table, and others do not)