Pathfinding with Explicit Topologies

Jeroen van der Ham

December 21, 2011

1 Introduction

This document will cover the NSI topology representation and pathfinding use-case. Section 2 describes the topology description. Section 3 shows how pathfinding is done in several scenarios. In section 4 we show the complexity of this approach and finally in section 5 we provide some ideas for multi-layer descriptions.

2 Topology Description

The *explicit* of the eplicit topologies refers to the labels being an explicit part of the domain topologies. In this approach, each of the domains can publish their own topology , using the standard NSI syntax, in a way they see fit, as long as they are referrable by their neighbors in some way. To support pathfinding, the domain topologies must contain at least the following:

Ports Each connection to another domain is tied to a *Port* object.

Label A *Label* is an object that contains both a *value* and the *type* of label that is used, for example *'VLAN=1872'*.

Labelsets On each of the *Port* objects, there is a *Labelset* to describe the (normally) available labels for that port. This does not have to be current.

Capabilities The domains should also announce whether they can switch or swap labels for each type they support.

It is also recommended that domains describe internal connectivity information between all their ports. This connectivity information can also be used to describe restrictions on labels. As we will see later, pathfinding complexity is greatly improved by having current availability information.

3 Pathfinding

In the startup phase of a pathfinder, it parses the locally provided topology. Then it recursively fetches and parses the domain descriptions of connected domains, until the pathfinder has discovered the global network. The pathfinder can then build up a global network of inter-domain connections as a simple graph. The intra-domain connectivity can be represented in this graph as well by expanding the domain from a single node into a larger network.

The applicable nodes in the domains have their switching and swapping capabilities annotated, and the Labelsets are also added to the Port objects.

3.1 Without VLAN Retagging

The pathfinder will have to do pathfinding in a two-step process. First using the regular graph to select a path through the network, and secondly selecting a label from all available labels along that path.

In this case the VLAN cannot be retagged, so the same label has to be used along the whole path. The intersection of the available labelset has to be determined. In case the intersection is empty (the same label is not available along that path), the next candidate path must be selected from the first step, and performing the same label selection procedure.

Assuming a path request from UvA's ps-80 and KRLight's ps-80, the pathfinder will find an available path from the topology, and issue a reservation message to each of the four domains. In case one of the VLANs is not available, this can either be learned directly from an up-to-date topology, or from a failed reservation. In that case no further reservation messages are sent because no other path is available. Depending on availability information, the worst-case scenario is that no messages are sent.

3.2 VLAN Retagging at Netherlight Only

As before the pathfinder finds a path through the simple graph. From the topology the pathfinder learns that there is a retagging capability at Netherlight. The label selection process can then be split into two sections in this case from the UvA to Netherlight, and from Netherlight to KRLight.

Again assuming a reservation from UvA ps-80 to KRlight ps-80, each of the four domains will be contacted to do a reservation. If this fails, the whole request will fail, since there is only one possible path still, the retagging at Netherlight does not change that fact.

3.3 All Networks have VLAN Retagging

As before the pathfinder finds a path through the simple graph. For each of the sections between VLAN retagging capabilities a label is selected, preferably the same label for as many sections as possible.

For a reservation from UvA ps-80 to KRlight ps-80 there are now more options. At each inter-domain connection there are now 4 options, for a total of 64 different paths through the network. The best-case scenario given wide availability is still 4 reservation messages.

If a VLAN (un)availability is not advertised, this will add to the number of reservation messages that is sent, since that is the only way to discover that information. An absolute worst-case situation would be where only one VLAN combination is available at each of the connections, none of the networks advertise their availability, and the pathfinder will always pick that one as the last one. In that case the maximum number of reservation messages is 64, since that is the maximum number of possible paths. In practice this number will be a lot closer to 4, especially if availability is advertised.

4 Complexity Analysis

With the approach above we are breaking the pathfinding into two steps. The first step is to find the shortest path in a regular graph, this scales logarithmically with the size of the graph. The complexity comes when combining the result with a label. Depending on the availability of the labels, a different path must be selected, combined possibly with retagging capabilities.

Pathfinding in a labelled graph is an NP complete problem. This means that finding a shortest labelled path in a labelled network may be very hard to do, but in practice so far it does not seem to be a problem. Many heuristic approaches have already been developed and we expect this trend to continue.

One thing to note is that this approach does require a change to the way the protocol is currently used. We would have to include a label into a request. Our suggestion is to do this in the technology-specific field that is already part of the message format. However, the behavior of the NSI agents will have to be updated to take this information into account.

5 Multi-Layer Descriptions

Currently networks are implemented on multiple (possible heterogeneous) layers. In practice, domains have converged on a single transport medium, Tagged Ethernet. In the future we expect that users may want to select different layers for transporting their data, for example alien wavelengths as has been shown in a connection between SURFnet and NORDUnet.

The topology description solution for explicit labels would support multilayer networks very easily. The current label is typed, adding more layers means adding different types for those layers, for example 'wavelength'. This does mean that these labeltypes should be standardised, and also the functions for being able to go from one label type to another.