Universiteit van Amsterdam

SNE

System and Network Engineering

Translating From DCN to NDL and Back Again

Jeroen van der Ham (vdham@uva.nl)

9th October 2009

Abstract

The topology descriptions used at Internet2 are provided in an XML format for use in the Dynamic Circuit Network suite. The topology descriptions developed by the University of Amsterdam is the Network Description Language.

In August and September 2009 Jeroen van der Ham worked at Internet2 on the translation of topology descriptions. This report describes some of the findings in creating this translation.

SNE technical report SNE-UVA-09-02

http://www.science.uva.nl/research/sne/reports/

1 Introduction

Topology descriptions are a necessary component of inter-domain provisioning in circuit oriented networks. In the past few years several different projects have created provisioning software, each with their own way of describing network topologies. In 2007 the Network Markup Language working group (NML)[1] was started in the Open Grid Forum (OGF) to build on all these efforts and create a standard for topology descriptions. Building a standard is a long and complicated process, especially with several different initiatives involved. The NML schema is progressing, but there are still some open issues that must be discussed.

In this report I examine the translation process from the topology descriptions used by the Dynamic Circuit Network software (DCN)[2, 3] created by the DICE collaboration, and the Network Description Language (NDL)[4] created by the University of Amsterdam.

The remainder of this report is structured as follows: in section 2 I describe the topology descriptions of IDC and NDL in more detail. In section 3 I explain how we translate from IDC and NDL and back again. Section 4 provides a summary and some conclusions.

2 Topology Descriptions

2.1 DCN Topology Descriptions

The topology descriptions used by the DCN have been created in the Network Measurements working group (NMWG)[5] of the OGF, called the NMWG Control Plane Schema[6]. This schema defines a model for describing networking topologies, initially aimed at describing network measurements. They have also defined an XML schema to describe network topologies in XML. An example is included in the appendix in listing 1.

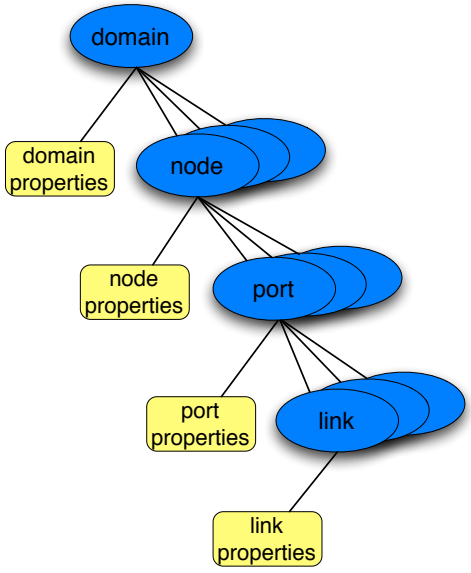

Figure 1: The abstract tree structure of the NMWG schema

The basic structure of the schema is shaped like a tree, see figure 1. A domain topology contains node elements, which contain port elements, which can contain one or more link elements. The link elements contain references to identifiers of other links, to describe the connection. The elements that make up the tree are described in more detail below:

domain contains a set of nodes that this topology is about. The domain element is also used to provide some details about this domain, such as the IDC identifier.

node each of the node elements provides a basic description of the node, together with a set of port elements.

port a port element describes the total capacity of that port, and contains link elements.

link a link element also describes the total capacity for that link, and also some information on the encoding and switching capability of the link. It also contains a `remoteLinkId` which refers to another link element, describing the other end of the link.

The encoding and switching properties of the link have been inspired by the way that GMPLS describes multi-layer topologies.

For a more extensive description of the NMWG Control Plane Schema see [6].

2.2 Network Description Language

The Network Description Language describes an ontology for describing network topologies in RDF. An example is included in the appendix in listing 2.

The basic structure of an NDL description follows the basic RDF structure, information is described in a graph with labelled edges. An overview of the current NDL schema is given in 2.

Figure 2: The main types and predicates of the NDL schema

Administrative Domain describes a collection of elements that fall under one operator.

Network Domain can be used to describe an aggregation of a set of devices without exposing the inner details.

Device describes a basic description of a network element, along with relations to a set of Interfaces.

SwitchingMatrix is the component of the device that performs the switching between the different interfaces that are connected to it.

Interface provide details about the interface, including its relation to switching matrices, connections to other interfaces, and adaptations.

Adaptation is an instance of an adaptation function, describing how data is translated between a client and a server layer. For example 10Gbase-R describes how data is translated between the Ethernet and wavelength layers.

Layer describes at which technology layer a network element operates.

For a more extensive description of NDL see [7, 4].

3 Translation Process

The translation process is defined from the NMWG format to NDL. The NMWG format is more strict than the NDL format, or at least the topologies as used by the IDC provisioning software are. This makes it easier to define the translation from NMWG to NDL, since the reverse is not always possible.

Fortunately the NMWG schema uses a globally unique identifier scheme that is compatible with the identifier scheme in NDL, so they can be used directly. However, the other way around is somewhat harder, see section 3.1.2.

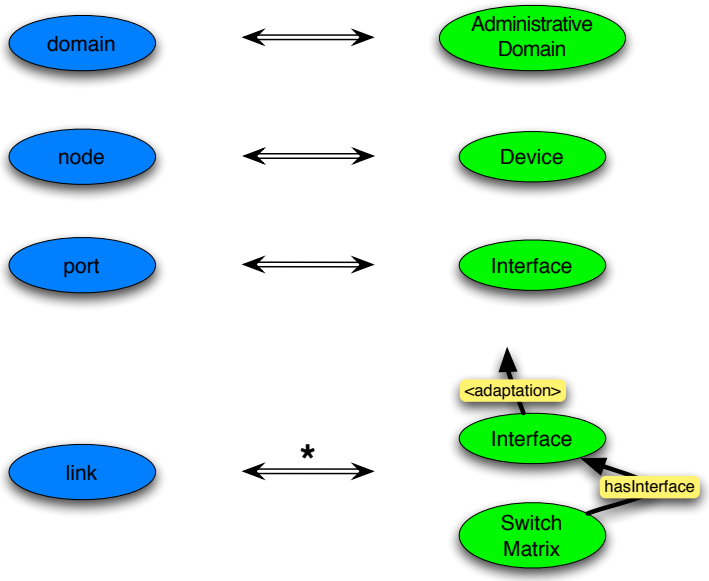

Figure 3: Mapping of objects between DCN (left) and NDL (right)

To describe the translation process, we follow the tree structure as described in the NMWG Schema. The translation is also shown graphically in figure 3. The first element to translate is the `domain`, this maps directly to an NDL `AdminDomain` object. The second element we encounter in the tree is a `node`, which again translates directly onto an NDL `Device` object. In the NMWG schema the relation between the two is implicit by inclusion, in NDL the relation has to be made explicitly using an `inAdminDomain` relation.

In NMWG a `node` contains one or more `port` elements. These map directly to NDL `Interface` objects, along with their capacity properties.

The `link` element is where the translation becomes somewhat difficult. The link element in NMWG also contains the encoding and switching capabilities of that link and port, while in NDL these are defined more explicit and in different places. The link end is translated to an `Interface` object in NDL. The switching capability of the link is translated to a `SwitchingMatrix` object that the new Interface is part of. The encoding of the link determines the layer of the Interface object from the containing port element. The combination of the switching capability and the encoding determine the `Adaptation Function` used between the NDL 'port' `Interface` and the 'link' `Interface`.

For example, if we have a `port` P with a `link` L, which is connected to link M, contained in port Q. Translated to NDL we then have an `Interface` P, connected to an Interface Q. Then we also have an `Adaptation` between L and P, and a similar one between M and Q.

Currently the IDC topologies only contain Ethernet connections, with VLANs provisioned over them. The availability of VLAN numbers is described in the NMWG format by the tag `vlanRangeAvailability` of a link element. This is translated in NDL to a label set on the link `Interface` object.

3.1 Translation Difficulties

While implementing the translation I encountered some difficulties that came up because DCN and NDL treat certain issues differently. These issues are ports with multiple links in DCN descriptions, the different identifier schemes and the way that multi-layer topologies are described. Another issue is with the way that domains are translated, which came to light only in the discussion of this report.

3.1.1 Multi-Link Ports

The NMWG schema as used by the DCN suite allows multiple links per port. This is used to describe cases where there is a single port that is connected to a third party (or parties), which passes the connection on to two (or more) other domains. The third party is left out of the topology, and instead a single port with multiple links is described, with different available VLAN ranges. The situation is described abstractly by figure 4.

It is not possible to translate this directly to an NDL description. A single port with multiple connections is not allowed. It is possible to translate the above situation to the NDL model by describing the node of the third party as a static component, as shown in figure 5. This description also maintains all of the constraints of the original topology description, the total bandwidth of both connections is still shared by the single port on the left, and the separate VLAN ranges are published on the interfaces towards domains A and B. The switching inside the virtual node is predetermined because it is defined not to be able to do VLAN retagging, and has disparate sets of VLAN ranges on the links going to domains A and B.

Both of the above descriptions describe the network topology as they see it, which description is preferred will depend on what the description is used for. The DCN description shows only the components that perform switching and provides as simple a view as possible for pathfinding. On

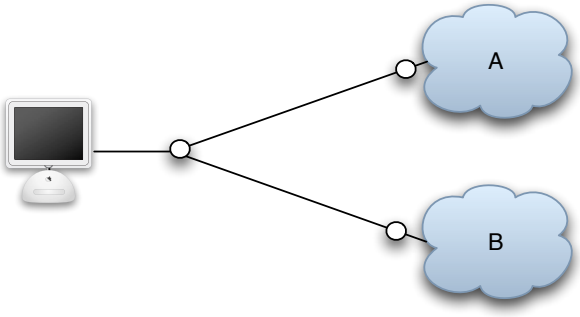

Figure 4: The DCN description of a port with multiple links

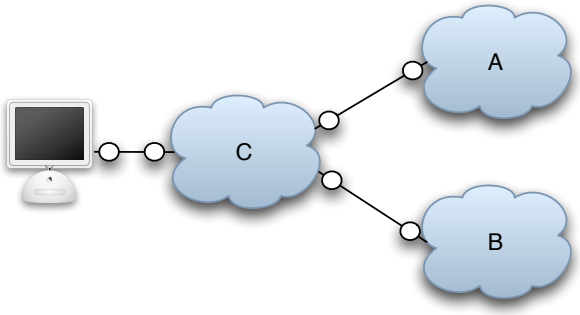

Figure 5: The NDL translation of the scenario as described in figure 4

the other hand, the NDL description provides a representation that follows the actual situation more closely, which can be an advantage for monitoring.

The general consensus in the NML group is also that a port should not have multiple links attached to it. The argument for this is that the NML model aims to be as generic as possible. A port with multiple links breaks with the general model that has been defined so far in NML.

Another argument is that the second description captures the fact that there is a switching decision taken by the third party. However, this decision is based on an agreement with the domain describing the connection, however this switching is not as dynamic as the rest of the network.

However, describing the third domain raises an important issue: who is allowed to describe network topologies? In this case the originating domain has an agreement with domain C to perform the switching. This domain does not want to participate in the topology exchange. It seems reasonable that the original domain describes the situation as in figure 5. In general it seems impossible to provide rules or even guidelines for situations like these.

Another issue came to light in discussing the above scenario. There are several different applications that want to have topology information. The most important that we are aware of now are:

Inter-Domain Pathfinding needs as highly abstracted view of the topology as possible, while still having the important details for pathfinding.

Provisioning must have a description of what is to be provisioned within the network.

Monitoring requires a detailed description of the important components of the (inter-domain) path so that it can be monitored and measurements can be performed.

In this case the provisioning and monitoring applications require a view as provided in figure 5, while the inter-domain pathfinding only requires the view as provided by figure 4. The goal of the NML-WG is to provide topology descriptions suitable for all three applications, so this issue should be addressed.

3.1.2 Identifiers

On the surface the identifiers in both schemes are very similar. DCN uses structured Uniform Resource Names (URNs), and NDL follows the RDF approach and uses Uniform Resource Identifiers (URIs). URIs are either Uniform Resource Locators (URLs) or URNs. However, the DCN and NDL models have a different approach to the way that they use and generate identifiers.

The identifiers used in DCN have a predefined structure, best explained by an example. The identifier `urn:ogf:network:domain=es.net:node=denv-ar1:port=ge-0/1/0:link=xe-0/2/0.1604` contains several elements, the first `urn:ogf:network:` is a prefix to specify that this is a network element. Following that are some type-value pairs which define the location where the object is defined (`es.net`), and the specific context of this identifier.

The identifier scheme in DCN requires the `domain=` part to be able to locate the object. The rest of the identifier is then left to be defined by the local authority. However, the DCN software currently also makes use of the rest of the structure of the identifier to make sense of the object. This can be easily changed to gathering that information from the structure of the XML document.

NDL uses the RDF approach to identifiers, meaning that the only requirement is that identifiers must be globally unique. A common shorthand in RDF is to use local identifiers starting with a pound sign (`#`). Any RDF software reading those identifiers will automatically prefix these with the prefix defined in the document, or the location of the file. However it should be noted that in RDF identifiers are only identifiers, there is no information embedded in them. The location of a topology description must be given separately using the standard `rdfs:seeAlso` predicate.

The two different approaches to identifiers brings up an important issue, location. Given an identifier, there must be some way to refer to where more information can be gathered about it. DCN defines this to be part of the identifier, while in NDL this is defined separately.

This issue somewhat resembles the 'identifier/locator split' debate currently going on in the IETF. However so far the identifiers for inter-domain provisioning are not used in routing, nor does this seem likely to happen.

Translation of identifiers is currently possible directly from DCN to NDL. The current version of the DCN software uses the structure of the identifier, so NDL identifiers must be transformed before they can be used. The defined DCN scheme requires that a reference to the location must be included, the local part of the identifier can then be used as is. However, given the way that the DCN software currently uses the identifiers, a more rigorous transformation must be made, including references to the context of the object (node, port, link). It is theoretically possible to perform this transformation in a way that also allows it to be translated back. However, since the syntax of URNs is more restrictive than that of URLs, an encoding must also be made, so that to a human reader it is not immediately obvious that the two identifiers are the same.

The different approaches to identifiers also allowed us to identify an underlying difference in the models for DCN and NDL. The domain part of the identifier in DCN is used to look up the topology description that describes the object associated with that identifier. In NDL this reference is provided separately using an explicit `seeAlso` or other reference to a topology description. This shows that an identifier scheme also has an impact on the way that look-ups

are performed. The DCN approach is not possible using NDL identifiers, however the RDF linking approach is possible using both identifier schemes, provided an identifier always comes with an explicit reference.

Another important issue regarding identifiers is authenticity, who is allowed to describe resources, and how can trust regarding descriptions be established. Currently neither identification scheme provides a way to cope with this issue. It is something that should be provided for in the NML schema.

3.1.3 Multi-Layer Descriptions

Currently the NML group is discussing how to describe multi-layer topologies. One of the reasons to work on these translations was to examine the differences in describing multi-layer topologies in DCN and NDL.

Unfortunately the topology descriptions that are currently implemented in the DCN network are not multi-layer. Almost everything in the currently available dynamic network is implemented on the Ethernet layer, and circuits are provisioned on the network by configuring VLANs. If another layer is described on a link, then the provisioning software concludes that the link is there and can be used without checking the label. In the end, the topology used in DCN is just a 'flattened' topology, where only restrictions on VLAN labels are checked.

It should be noted that it is currently possible to describe multiple layers in the topology descriptions for DCN, including label restrictions. The layer descriptions are given using the GMPLS terminology using the `encoding` and `switchingCapability` elements. These elements use the properties and their values as defined in RFC 3471[8].

In NDL adaptations between layers are defined explicitly as a capability on the node, or as an implemented function, together with the labels used. In GMPLS the adaptation between layers is described either not at all, or implicitly. An overview of the approaches is described below, for a more detailed overview, see [9].

GMPLS allows operators to configure Forwarding Adjacency LSPs (FA-LSPs)[10], these are links through a network that provide a virtual adjacency between two routers. These links are pre-provisioned, but allow other LSPs to be configured over them.

The idea of FA-LSPs has been further expanded by the introduction of a *Virtual TE-Link*[11]. This is basically the same as an FA-LSP, but without pre-provisioned resources. So an adjacency between two (or more) routers is defined, but the actual LSP is not in place yet. These again can be predetermined statically, or they can be dynamically found on the lower layer. A set of virtual TE-links defines a Virtual Network Topology.

Using FA-LSPs or a Virtual Network Topology it becomes possible to pre-define links on a lower layer, thereby defining an adjacency on a higher layer. This layering of LSPs is called *hierarchical* LSPs[10]. The operators pre-define the topologies on different layers, adaptations are not described, and no multi-layer pathfinding is necessary. However, this also means that networks may not be used as dynamically as they could be.

In GMPLS the concept of adaptations can also be described implicitly.[1] A node with an adaptation function is called a *hybrid node* in GMPLS. This node is described to have ports on different layers. The adaptation is then implicitly described by advertising two different ISCDs on both interfaces. So for example, a node capable of doing both (TDM,SONET/SDH) and (L2SC,Ethernet) would announce both pairs on both interfaces.

[1]Some terminology: GMPLS defines multi-region and multi-layer networking: A *region* is defined as a particular data plane technology, e.g. TDM. A *layer* is then a subset of that region described by a particular switching granularity, e.g. VC-4. Changing from one region or layer to another is done by an *adjustment* function in a node. In this document I will use the terms multi-layer which in GMPLS would mean multi-region and multi-layer, and I use adaptation where GMPLS uses adjustment.

The node may also have an internal limitation on the adaptation function, there is currently some discussion on defining an IACD, Interface Adjustment Capability Descriptor[12]. This would for example describe the bandwidth limitations of the internal adaptation function.

Currently, the DCN software tends to use the first approach, where adaptations are not described at all, and a virtual overlay network is described. The provisioning software will then use or provision the necessary connections on the lower layer. This is simple to translate to NDL, as this is almost a single layer description. The current schema for DCN also allows to describe the links using the second GMPLS approach, so that it implicitly describes adaptations. While they are allowed, the DCN software does not support them yet. It is also not completely clear yet how these can be translated to NDL. This is still an open issue.

3.1.4 Domains

In the translation process as described above I directly translate the DCN domain as an NDL AdminDomain. In the topology descriptions that are currently in use, it is treated that way. In later discussions it turned out that the original design of the schema also allows the domain element to directly contain ports or even links. When the domain element is used in that way, it acts as an NDL NetworkDomain. This shows that the translation is not always as straightforward as it may seem to be.

4 Conclusions

In this report I have described how we can translate from the DCN topology descriptions to NDL. There is not a complete match between the underlying models of both topology description language, but it was possible to define a reasonable translation between them.

In the process I have discovered some difficulties. The structure of the identifiers in DCN are important. This means that the identifiers from NDL cannot be translated directly, and have to be adapted to that structure.

Another difficulty is in some special cases where the DCN descriptions have multiple links per port. This is not possible to describe directly in NDL, and a more complicated translation has been defined.

The translation from NDL to IDC is currently only possible for a subset of NDL descriptions. The reason for this is that currently the DCN only supports topology descriptions for Ethernet and VLANs. It also requires information about the bandwidth availability of the link and the granularity with which reservations can be made. These properties are not always described in NDL descriptions.

At the moment it is also unclear what multi-layer descriptions would look like in the DCN topology descriptions. The descriptions are based on GMPLS, and two different ways of describing multi-layer networks are possible. The first basically abstracts the multi-layer away into a single layer using statically configured tunnels. As this basically creates a single layered network, this can be easily translated.

A second way of describing multi-layer networks describes the adaptation capability of nodes in the network implicitly. The translation of this to NDL is still an open issue, and a possible subject of future research.

The translation procedure, barring the limitations described above, has been successfully implemented in Python based on the `pynt` toolkit. It can take an XML DCN topology, and create an RDF NDL translation of it. A translation from NDL to DCN also works for a subset of NDL, all the interfaces of the topology must be on the Ethernet layer, and they should have

the capacity properties defined. The code and examples are available for download at http://staff.science.uva.nl/~vdham/projects/dcn-translation.html.

References

[1] *Network Markup Language Working Group (NML-WG).* http://forge.gridforum.org/sf/projects/nml-wg.

[2] *Dynamic Circuit Network Software Suite (DCN).* https://wiki.internet2.edu/confluence/display/DCNSS/.

[3] *Inter-Domain Controller (IDC).* http://www.controlplane.net.

[4] Jeroen van der Ham and Freek Dijkstra: *Network Description Language Homepage.* http://www.science.uva.nl/research/sne/ndl/.

[5] *Network Measurements Working Group (NM-WG).* http://forge.gridforum.org/sf/projects/nm-wg.

[6] Internet2: *DCN Software Suite v0.5.1: OSCARS Inter-Domain Controller (IDC) Installation Guide.* https://wiki.internet2.edu/confluence/download/attachments/19074/DCN-SUITE-OSCARS-INSTALL.pdf?version=11.

[7] Freek Dijkstra: *Modelling of Multi-Layer Transport Networks.* Ph.D. thesis, University of Amsterdam (2009).

[8] L. Berger: *Generalized Multi-Protocol Label Switching (GMPLS) Signaling Functional Description.* RFC 3471 (Proposed Standard) (January 2003). Updated by RFCs 4201, 4328, 4872, http://www.ietf.org/rfc/rfc3471.txt.

[9] E. Oki, T. Takeda, JL. Le Roux, and A. Farrel: *Framework for PCE-Based Inter-Layer MPLS and GMPLS Traffic Engineering.* RFC 5623 (Informational) (September 2009). http://www.ietf.org/rfc/rfc5623.txt.

[10] K. Kompella and Y. Rekhter: *Label Switched Paths (LSP) Hierarchy with Generalized Multi-Protocol Label Switching (GMPLS) Traffic Engineering (TE).* RFC 4206 (Proposed Standard) (October 2005). http://www.ietf.org/rfc/rfc4206.txt.

[11] K. Shiomoto, D. Papadimitriou, JL. Le Roux, M. Vigoureux, and D. Brungard: *Requirements for GMPLS-Based Multi-Region and Multi-Layer Networks (MRN/MLN).* RFC 5212 (Informational) (July 2008). http://www.ietf.org/rfc/rfc5212.txt.

[12] Dimitri Papadimitriou, Martin Vigoureux, Kohei Shiomoto, Deborah Brungard, and Jean-Louis Le Roux: *Generalized Multi-Protocol Label Switching (GMPLS) Protocol Extensions for Multi-Layer and Multi-Region Networks (MLN/MRN)* (August 2009). http://www.ietf.org/id/draft-ietf-ccamp-gmpls-mln-extensions-07.txt.

A Example Topology Descriptions

Figure 6 shows a network topology that we describe in both DCN and NDL below. The figure shows two nodes, but in the topology descriptions below we only describe the left node, raptor1. The description of the other node is very similar in both cases.

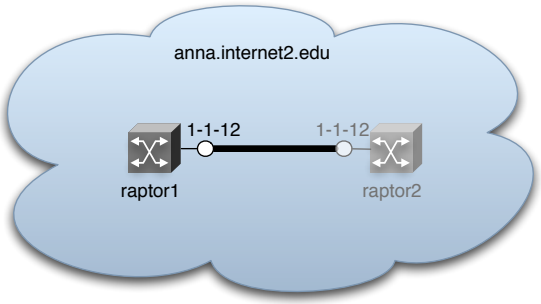

Figure 6: Diagram of the example topology described by both DCN and NDL

```
<topology xmlns="http://ogf.org/schema/network/topology/ctrlPlane/20080828/" id="rdf-generator-200909141713">
  <idcId>foobar</idcId>
  <domain id="urn:ogf:network:domain=anna.internet2.edu">
    <node id="urn:ogf:network:domain=anna.internet2.edu:node=raptor1">
      <port id="urn:ogf:network:domain=anna.internet2.edu:node=raptor1:port=1-1-12">
        <capacity>1000000000</capacity>
        <maximumReservableCapacity>1000000000</maximumReservableCapacity>
        <minimumReservableCapacity>1000</minimumReservableCapacity>
        <granularity>1000</granularity>
        <link id="urn:ogf:network:domain=anna.internet2.edu:node=raptor1:port=1-1-12:link=10.10.3.1">
          <remoteLinkId>urn:ogf:network:domain=anna.internet2.edu:node=raptor2:port=1-1-12</remoteLinkId>
          <trafficEngineeringMetric>5</trafficEngineeringMetric>
          <capacity>1000000000</capacity>
          <maximumReservableCapacity>1000000000</maximumReservableCapacity>
          <minimumReservableCapacity>1000</minimumReservableCapacity>
          <granularity>1000</granularity>
          <SwitchingCapabilityDescriptors>
            <switchingcapType>l2sc</switchingcapType>
            <encodingType>ethernet</encodingType>
            <switchingCapabilitySpecificInfo>
              <interfaceMTU>9000</interfaceMTU>
              <vlanRangeAvailability>0, 2-3373</vlanRangeAvailability>
            </switchingCapabilitySpecificInfo>
          </SwitchingCapabilityDescriptors>
        </link>
      </port>
    </node>
  </domain>
</topology>
```

Listing 1: DCN description of the network in figure 6.

```
<?xml version="1.0" encoding="UTF-8"?>
<rdf:RDF
  xmlns:ndl="http://www.science.uva.nl/research/sne/ndl#"
  xmlns:layer="http://www.science.uva.nl/research/sne/ndl/layer#"
  xmlns:nmwgt="http://ogf.org/schema/network/topology/ctrlPlane/20080828/"
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:capability="http://www.science.uva.nl/research/sne/ndl/capability#"
  xmlns:ethernet="http://www.science.uva.nl/research/sne/ndl/ethernet#"
  >

  <ndl:Device rdf:about="urn:ogf:network:domain=anna.internet2.edu:node=raptor1">
    <nmwgt:address>207.75.164.207</nmwgt:address>
    <ndl:hasInterface rdf:resource="urn:ogf:network:domain=anna.internet2.edu:node=raptor1:port
        =1-1-12"/>
    <ndl:hasInterface rdf:resource="urn:ogf:network:domain=anna.internet2.edu:node=raptor1:port
        =1-1-12:link=10.10.3.1"/>
    <capability:hasSwitchMatrix>
      <capability:SwitchMatrix rdf:about="urn:ogf:network:domain=anna.internet2.edu:node=
          raptor1_sm">
        <ndl:layer rdf:resource="http://www.science.uva.nl/research/sne/ndl/ethernet#
            EthernetNetworkElement" />
        <capability:hasSwitchingCapability rdf:resource="http://www.science.uva.nl/research/sne/ndl
            /ethernet#EthernetNetworkElement" />
        <ndl:hasInterface rdf:resource="urn:ogf:network:domain=anna.internet2.edu:node=raptor1:port
            =1-1-12:link=10.10.3.1"/>
      </capability:SwitchMatrix>
    </capability:hasSwitchMatrix>
  </ndl:Device>
  <ndl:Interface rdf:about="urn:ogf:network:domain=anna.internet2.edu:node=raptor1:port=1-1-12">
    <rdf:type rdf:resource="http://www.science.uva.nl/research/sne/ndl/ethernet#
        EthernetNetworkElement"/>
    <nmwgt:mtu>9000</nmwgt:mtu>
    <ndl:connectedTo rdf:resource="urn:ogf:network:domain=anna.internet2.edu:node=raptor2:port
        =1-1-12" />
    <ndl:capacity rdf:datatype="http://www.w3.org/2001/XMLSchema#float">1000000000</ndl:capacity>
    <ndl:layer rdf:resource="http://www.science.uva.nl/research/sne/ndl/ethernet#
        EthernetNetworkElement" />
    <ethernet:Tagged-Ethernet>
      <capability:PotentialMuxInterface rdf:about="urn:ogf:network:domain=anna.internet2.edu:node=
          raptor1:port=1-1-12:link=10.10.3.1">
        <rdf:type rdf:resource="http://www.science.uva.nl/research/sne/ndl#ConfigurableInterface"/>
        <rdf:type rdf:resource="http://www.science.uva.nl/research/sne/ndl/ethernet#
            EthernetNetworkElement"/>
        <nmwgt:vlanRangeAvailability>0, 2-3373</nmwgt:vlanRangeAvailability>
        <ndl:layer rdf:resource="http://www.science.uva.nl/research/sne/ndl/ethernet#
            EthernetNetworkElement" />
      </capability:PotentialMuxInterface>
    </ethernet:Tagged-Ethernet>
  </ndl:Interface>
</rdf:RDF>
```

Listing 2: NDL description of the network in figure 6.