Using the Network Description Language in Optical Networks

van der Ham, J.J.; Grosso, P.; van der Pol, R.; Toonk, A.; de Laat, C.T.A.M.

[Link to publication](#)

Citation for published version (APA):
van der Ham, J. J., Grosso, P., van der Pol, R., Toonk, A., & de Laat, C. T. A. M. (2007). Using the Network Description Language in Optical Networks. In *Integrated Network Management, 2007. IM '07. 10th IFIP/IEEE International Symposium on* (pp. 199-205). IEEE. http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?arnumber=4258536

Using the Network Description Language in Optical Networks

Jeroen van der Ham[*‡], Paola Grosso[*], Ronald van der Pol[†], Andree Toonk[†], Cees de Laat[*]

[*]System and Network Engineering research-group, Universiteit van Amsterdam
[†]SARA Computing and Networking Services, Amsterdam, the Netherlands
[‡]TNO Defense, Security and Safety, The Hague, the Netherlands
Email: vdham@science.uva.nl

Abstract— Current research networks allow end users to build their own application-specific connections (lightpaths) and Optical Private Networks (OPNs). This requires a clear communication between the requesting application and the network. The Network Description Language (NDL) is a vocabulary designed to describe optical networks based on the Resource Description Framework (RDF). These descriptions aid applications in querying the capabilities of the network and allow them to clearly express requests to the network. This article introduces NDL and shows its current applications in optical research networks.

I. INTRODUCTION

In recent years the design of research networks have been moving to a hybrid model[1], [2], [3]. Optical hybrid networks consist of a routed IP part and a circuit switched optical part (*lightpaths*). On these hybrid networks end-users can setup lightpaths through the network on demand. The lightpaths can be used to quickly move large amounts of data, or to get a guaranteed fixed quality of service regarding bandwidth, delay, or jitter. With these *lightpaths* users can dynamically build Optical Private Networks (OPNs) and an optimal OPN topology can be setup for each specific application.

Currently, most lightpaths are configured manually, a process which can take days or weeks. If the lightpath goes through multiple domains then the process can take even longer. The provisioning of a lightpath connection requires clear communication between all the parties involved.

Applications that require these kinds of services include the experiments in the new particle accelerator at CERN, the LHC[4], which will produce several Petabytes of physics data per year. This data is spread to several research institutes around the world. Another example is the eVLBI community[5], where experiments are conducted using several radio-telescopes in parallel. The data from these telescopes is sent to a correlator station, where the data is combined in real-time. This effectively creates a large interferometer that can be used to study radio objects in the sky. Consistent delay, jitter is very important in this case.

In this article we propose the Network Description Language (NDL), an ontology based on the Resource Description Framework (RDF). NDL is primarily intended for use within optical networks to describe the topology in a machine-readable format. NDL provides a common semantic to the application, the network and the service provider, so that the communication between them is unambiguous. It can be used to create inter-domain network graphs at various abstraction levels, to provide an information model for service discovery, and to facilitate lightpath provisioning.

One of the advantages of using NDL over other network data models, is that we can leverage already established semantic web tools while using an ontology specifically designed for the optical realm. This makes the application and deployment of NDL straightforward.

The rest of the article is organised as follows: section II describes hybrid networks and the concept of lightpaths. In section III we explain the difficulties when requesting lightpaths, both through single and multiple domains. In section IV we describe related work, both regarding network data models, and provisioning tools currently developed and used within research networks. The Network Description Language is described in section V. Current applications of NDL, results and future plans are described in section VI. The conclusion is given in section VII.

II. HYBRID NETWORKS

Hybrid networks are networks that offer both the traditional routed IP access to the Internet and circuit switched point-to-point connections. These connections are usually referred to as *lightpaths*. They are based on OSI layer 1 or layer 2 technology with well-known quality of service aspects.

In recent years National Research and Education Networks (NRENs) have been actively pursuing the vision of hybrid networks by acquiring their own fiber networks. In doing so, they have become their own transport providers instead of buying these services from carriers. Besides the economical advantage, this enables the NRENs to provide researchers with lightpaths on demand through the hybrid network.

Because NRENs own the fiber, the capacity of the network is determined by the hardware that is connected to the fibers. Typically they use *Dense Wavelength Division Multiplexing* (DWDM)[6] equipment, which allows parallel wavelengths through the same fiber. Currently it can accommodate close to 100 different wavelengths, each with a capacity of 10 Gigabit per second. Using *Time Division Multiplexing* (TDM, SONET)[7], each of these 10 Gbps links can be carved up into smaller capacity links. A *lightpath* can be a complete (10 Gbps) wavelength or pieces of it using TDM. Both

technologies offer the customer a dedicated data channel with known capacity, roundtrip time and jitter.

The following two sections describe examples of networks that provide lightpaths, SURFnet6 and GLIF. SURFnet6 is a single domain network, while GLIF is a co-operation of several networks, working at addressing the inter-domain lightpath management and provisioning issues. We are actively working with both organisations and we are conducting experiments with NDL in these networks.

A. SURFnet6

SURFnet6[8] is the new network of the Dutch NREN, SURFnet. It consists of a 6000 kilometre dark fiber network. The hybrid network is built using Nortel DWDM and TDM equipment, and Avici routers. The TDM equipment consists of Nortel OME6500 devices that are connected with 10 Gbps connections via the DMWM network to other OME6500 devices.

Customers, i.e. the universities, colleges and research institutes, typically connect via a 1 or 10 Gigabit per second connection. These TDM circuits are provisioned through the OME6500 network, and can either be used for lightpaths or for routed IP services.

B. GLIF

The GLIF, *Global Lambda Integrated Facility*[9], is an international virtual organisation of research networks, research consortia and institutions whose aim is to build a worldwide networking facility for scientific research. GLIF consists of a collection of optical exchange points, GOLEs (GLIF Open Lightpath Exchanges) and links between them. A global network is formed through lightpath connections to other GOLEs.

The global network currently consists of over a dozen GOLEs spread over North America, Europe, East Asia, and Australia, with numerous links across and between the continents.

In this facility, scientists can let their applications take full advantage of the reconfigurability of the infrastructure, for example for diverse end to end paths. If we think of the concept of *computing grid*, where resources are dynamically assigned to the applications depending on need and availability, then GLIF is in essence a *lambda grid*. In a lambda grid lightpaths are assigned to applications for the duration of time needed to perform the computing tasks, and engineered according to the availability of the underlying network resources.

III. Problem Statement

In order to integrate dynamic lightpath provisioning with the workflow management middleware used in the new e-Science projects it is mandatory to support automatic topology discovery and pathfinding for these connections spanning multiple administrative domains. Therefore, the different domains must be able to exchange the required information expressed in an interoperable format to be able to understand the complexity inherent with lightpath requests.

Currently, when a scientist has a need for a lightpath within one NREN (intra-domain), he will send an email to or phone the contact person at the NREN. During the conversation, the NREN-contact tries to translate the scientists desires to the parameters for the connection, e.g. the end-points, the bandwidth, delay, jitter, et cetera. The request is then relayed to the engineers, who provision the actual connection. This process typically takes at least a few days and sometimes even a few weeks.

If however, the scientist requires a connection that spans multiple domains, e.g. across the ocean, or across continents, then multiple parties are involved, who must all be contacted. The scientist must either do this himself, or ask the NREN to do this. The provisioning process now becomes more complicated:

1) Based on external references, and experience, the contact determines through which domains the path could go.
2) For each of the domains involved in the path:
   - Inquire to the possibility of a path through the domain.
   - Determine if the segment agrees with the requirements.
   - Reserve the segment.
3) If a problem is encountered, go back to step 1.
4) Configure the end hosts and start troubleshooting the connection.

Note that the path determination is far from trivial. There are a lot of parameters involved when configuring inter-domain lightpaths. These settings must be configured correctly on both sides of the connection, so they must clearly be communicated to all parties involved. See [10] for a discussion on the parameters involved in requesting a lightpath.

Currently, the whole process of acquiring a (working) lightpath across multiple domains often takes several weeks, a lot of emails and phone calls and extensive testing. It is clear that the whole process needs to be improved and automated, in order to scale for many domains.

In GLIF several parties realised that in order for lightpaths to scale, the provisioning process must be (partly) automated. Currently there are a few applications available to do provisioning and brokering, UCLP, DRAGON and DRAC.

UCLP, User Controlled Lightpaths[11], is one of the first applications that allowed end-users to configure lightpaths. It is a project funded by CANARIE, the Canadian research network. Currently they are implementing version 2, which will allow users to request lightpaths using web-services.

DRAGON, Dynamic Resource Allocation via GMPLS Optical Networks[12], is a framework that can provision connections on heterogeneous networks. It provides support for provisioning connections using GMPLS, even for networking devices that do not support it.

A third framework is DRAC, Dynamic Resource Allocation Controller from Nortel. It enables applications to request connections through the network. It is expected to be ready for deployment and testing on SURFnet6 soon.

The applications mentioned above are all still under development, although proof-of-concepts do exist. Currently they are all aimed at providing network provisioning capabilities for single domains, and only for multiple domains using the same application. At the moment these applications can not directly co-operate with each other in provisioning a connection. Within GLIF the goal would be to let all these applications work together in order to support inter-domain brokering and provisioning. To support this clear communication between the applications is an important issue. A key requirement is that all the applications involved have the same understanding of the GILF network resources and the network topology.

Once a lightpath is provisioned end to end, there still could be some problems with the connection. The troubleshooting process is very difficult[13], because of the low networking layer, which means that standard tools as `traceroute` or `ping` can not be used to isolate the problem. Every domain must check their segment of the lightpath and this process is further hampered by (often large) time-zone differences.

When troubleshooting a lightpath one needs to know about the details of the lightpath route, which network elements are involved, which interfaces, which time-slots or wavelengths.

IV. Related Work

In this section we discuss the related work in network descriptions. One of the first standards for describing information regarding the network is SMI, part of SNMP[14], the Simple Network Management Protocol[1]. It allows for simple retrieval and updating of diagnostic, performance and configuration information using the MIB, Management Information Base. The MIB is a tree of name value pairs, which contains standard and vendor specific branches. While most information can be stored in the standard branch, vendors often use their proprietary space. The model provided by SNMP is centred on the device itself and contains almost no information about the rest of the network.

A more complex information model is CIM[15], the Common Information Model, developed by the DMTF, the Distributed Management Task Force. CIM is an object-oriented model defined using the Unified Modelling Language. The model can capture information regarding computer systems, operating systems, networks and other diagnostic information. It is mainly implemented in consumer-oriented computing equipment, and operating systems, such as Windows and Solaris.

The Data Center Markup Language (DCML)[16] is an information model that uses RDF and OWL to describe data centers, including computing equipment, networks and other hardware. DCML is still in development, and has not been published yet, but appears to be aiming for the same level of complexity as the CIM model.

The current information models for networks and networking equipment are all aimed at diagnostic information and

information regarding the device itself. Our work with NDL on the other hand is aiming to provide a simple schema, that can be used to provide an overview of the network and the relation between the different devices. This kind of description can be used by applications that need only an overview of the network and not all diagnostic information. This is exactly what is required by automatic lightpath provisioning applications, which allow end-users to configure high-speed connections.

V. Network Description Language

In this section we provide an introduction of the Network Description Language, an RDF vocabulary for describing networks. First we explain the problems with describing networks, then we introduce the schema itself and we conclude the section with a short example.

Describing computer networks seems a simple problem at first sight. There are some machines connected to each other, and each machine can provide information about itself, its configuration and its knowledge of the network. The communication across the network has been more or less standardised with the TCP/IP stack. This stack also hides most networking details from the application. These abstractions are however not completely successful, some details always seep through the abstraction levels[17]. These leaky abstractions make it difficult to create a clear and consistent model for networks.

Another factor is that when users request connections, they do so with specific quality of service parameters in mind, especially in the case of optical connections. Depending on the experiment they are conducting, a user may want some combination of specific values for bandwidth, latency, packet-loss, or jitter. One of the main reasons for creating NDL is to support users and applications in describing a request. In the future this could become part of the Common Service Definition [10].

To avoid complexity we started out simple and only attempt to describe the physical network. To do this, we created the schema as shown in figure 1.

The schema consists of four classes, shown as ovals, and eight properties, shown as labelled arrows. Below we give a short description of the classes:

Location A place where devices are located.
Device Any kind of machine that is connected to the network.
Interface The connection between the device and the rest of the network.
Link An (abstracted) connection between two interfaces.

The eight properties are explained below:

name The hostname of a device or the name of an interface.
description A property to include additional human-readable information.
locatedAt The relation between a device and a location.
hasInterface Defines the relation between a device and an interface.
connectedTo Used to describe a physical connection between two interfaces or between a link and an interface.

---

[1]The name 'Simple Network Management *Protocol*' for the standard is deceptive. It is also the name of the whole architecture, which defines the protocol, but also contains a definition of an information model.
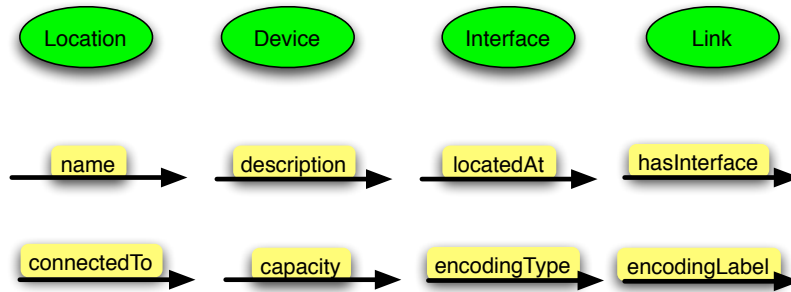
Fig. 1. Overview of the Network Description Language

capacity Defines the bandwidth capacity of an interface or link.

encodingType Defines what kind of encoding is used on an interface or link.

encodingLabel Provides further details about the encoding of the interface or link.

The above classes and properties allow a fine grained description of the network, including cables, capacity and transport type. On the other hand, they allow for more abstraction of the network, using the *Link* class. Connections between interfaces can either be defined directly using the *connectedTo* property, or by defining the intermediate *Link* object. The latter can be used to express information about the link itself, or for abstracting parts of the network that can not or will not be changed, for example if the line is leased for an extended period of time.

The values of the *capacity*, *encodingType*, and *transportType* properties can be defined using a separate namespace with the standard terminology from GMPLS[18] (the 'LSP encoding type' and 'General Protocol ID').

An example description is shown in figure 2, which describes a simple connection between two hosts. At the top in figure 2 is a sketch of the situation that is described using RDF below. Lines 4–6 define the location `Netherlight`, followed by a definition of the device `TDM 3` on lines 7–11. Line 8 specifies that `TDM 3` has an interface named `501/1`, which is defined on lines 12–17. Line 15 and 16 define that the interface has a capacity of $1.2 \cdot 10^9$ bytes per second (∼10 Gbps). A similar definition of the device `TDM4` and its interface `5/1` is given on lines 18–28.

Note that the connection between the two devices is defined twice, on lines 14 and 25. This duplicate definition is used to check that both devices are configured correctly and have the same information regarding their connection. More extended examples can be found on the NDL Homepage[19].

A. Resource Description Framework

As shown above, we have defined NDL using the Resource Description Framework. We have chosen this syntax above a simple XML or 'ad hoc' syntax for several reasons. The main reason is that RDF represents its data in a graph format. This fits well with our problem, we are essentially describing the network as an extended graph.

A second reason for choosing RDF over plain XML is extendibility. RDF applications are built with the assumption that they do not have to understand the complete description. They look for pieces of data they can understand and ignore the rest. On the one hand, this allows you to extend the ontology, while maintaining complete backward compatibility. On the other hand it also allows you to easily combine descriptions without adapting third party applications.

The available tools and parsers are another reason for choosing RDF. The semantic web community has worked very hard on these. As a result libraries and parsers are available for all major programming languages. There are specific visualisation tools and editors available to support development of descriptions.

Finally, RDF has been designed with support for multiple, possibly overlapping, ontologies. This makes it possible for multiple domains to start with different ontologies, yet maintain compatibility without changing their descriptions.

For more information about RDF, the Semantic Web and their developments, see [20], [21].

VI. APPLICATIONS OF NDL

NDL provides a powerful language to solve many of the operational issues that operators and users face in hybrid optical networks. It allows the automatic creation of network maps; it facilitates lightpath finding algorithms and the setup of associated reservation systems; it enhances the interoperability and the exchange of information between different administrative domains.

In this section we will describe in detail the above NDL capabilities, pointing to the specific applications we have worked on and stressing their use and applicability in the management of hybrid networks.

A. Network Graph Generation

One of the advantages of using NDL as the language for description of hybrid networks is the availability of semantic web tools for RDF, that can parse and consume the information in each NDL file. This means that extracting the information needed for network management, and in our specific case lightpath provisioning, is straightforward and simple.

Our first application of the language has been the generation of network maps. Given most lightpaths are still provisioned
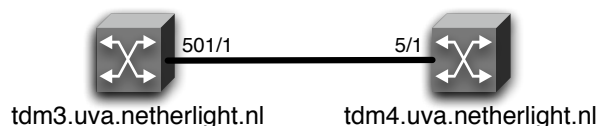
```
1  <?xml version="1.0" encoding="UTF-8"?>
   <rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
            xmlns:ndl="http://www.science.uva.nl/research/sne/ndl#">
       <ndl:Location rdf:about="#Netherlight">
5          <ndl:name>Netherlight Optical Exchange</ndl:name>
       </ndl:Location>
       <ndl:Device rdf:about="#tdm3.amsterdam1.netherlight.net">
           <ndl:name>tdm3.amsterdam1.netherlight.net</ndl:name>
           <ndl:locatedAt rdf:resource="#Netherlight"/>
10         <ndl:hasInterface rdf:resource="#tdm3.amsterdam1.netherlight.net:501/1"/>
       </ndl:Device>
       <ndl:Interface rdf:about="#tdm3.amsterdam1.netherlight.net:501/1">
           <ndl:name>tdm3.amsterdam1.netherlight.net:POS501/1</ndl:name>
           <ndl:connectedTo rdf:resource="#tdm4.amsterdam1.netherlight.net:5/1"/>
15         <ndl:capacity
              rdf:datatype="http://www.w3.org/2001/XMLSchema#float">1.2E+9</ndl:capacity>
       </ndl:Interface>
       <ndl:Device rdf:about="#tdm4.amsterdam1.netherlight.net">
           <ndl:name>tdm4.amsterdam1.netherlight.net</ndl:name>
20         <ndl:locatedAt rdf:resource="#amsterdam1.netherlight.net"/>
           <ndl:hasInterface rdf:resource="#tdm4.amsterdam1.netherlight.net:5/1"/>
       </ndl:Device>
       <ndl:Interface rdf:about="#tdm4.amsterdam1.netherlight.net:5/1">
           <ndl:name>tdm4.amsterdam1.netherlight.net:POS5/1</ndl:name>
25         <ndl:connectedTo rdf:resource="#tdm3.amsterdam1.netherlight.net:501/1"/>
           <ndl:capacity
              rdf:datatype="http://www.w3.org/2001/XMLSchema#float">1.2E+9</ndl:capacity>
       </ndl:Interface>
   </rdf:RDF>
```

Fig. 2.   Example of an NDL description

manually, at least when they involve crossing organisation boundaries, maps become the visual aid used by network engineers to setup the circuits. The information about the connection between domains needs to be up to date, accurate and consistent, because mistakes in lightpath provisioning can have impact on other lightpaths and in the case of hybrid networks also on the regular traffic.

There are certainly many ways to create a graphical overview of a network, and when working in a single network domain plenty of tools to choose from. But in multi-domain environments, as the ones we work with, we need to take into account that the information for each domain is not centrally maintained and there is a big potential for inconsistencies. The manual creation of these large-scale topology overviews requires a tedious conversion of *description styles* toward a consistent representation, and the verification of consistent information at the boundaries. After these steps one can then feed the information to the graphing tools.

NDL addresses all the issues above, and it provides a way to operators to consistently describe their network; the graphing tools we wrote allow to extract the distributed information about each domain, correlate it and verify its consistency in a quick and efficient way.

Starting from a network description in NDL format, we use a SPARQL query to get the connections between the devices and their names. Using a small script, this data is then converted to serve as input to GraphViz, an open source graph visualisation tool[22]. An example of such a graph is shown in figure 3. This the map of NetherLight[23], one of the network domains participating in the GLIF. The script used to generate the graph can be found at the NDL homepage [19].

Our current effort goes to further improve the visualisation of the network graphs starting from NDL files. GraphViz is an excellent way of automatically generating graphs, but changing the information can result in a completely different graph where the various elements are placed in different positions in the space, making successive version of a network map difficult to compare. Using the GPS co-ordinates information contained in NDL we can generate more consistent graphs. Using Google Maps[24] or other similar software to show networks on the global scale is also an option we are currently investigating.

B. Lightpath Planning in SURFnet6

A second application of NDL we worked on is the support to on demand provisioning of lightpaths, and backup lightpaths, within the SURFnet6 network. The idea is that NDL files hold information about the current topology, and that we can correlate this with additional dynamic information, to provide engineers with integrated management and planning tools.

NDL constitutes in this case the first step in a more complex system for lightpath provisioning.

This is how the system currently works. Part of the SURFnet6 hybrid network is formed by a collection of Nortel OME6500 TDM nodes. We obtain the topology information by using the neighbour knowledge from each of these devices. We

Fig. 3. A graph of NetherLight resources (extracted from NDL file)

gather the data by periodically sending and receiving discovery messages on the control plane. And we use NDL to describe the topology of the TDM layer in files.

Additionally, a MySQL database holds the cross-connect information for each OME6500 in the network. That is, the information on the current use of the lightpaths. This makes it possible to determine the amount of time-slots still available on each interface. Combining the NDL topology information and the database time-slot information together we can find a shortest path through the network that has enough free time-slots to accommodate a new user request. To find this path we use a constraint based shortest paths algorithm.

Currently these path calculations are for human operators who provision the lightpaths through SURFnet6. In the future we will expand the application so that it can be used automatically by provisioning tools such as the ones described in section IV. These can be invoked by the user directly allowing for provisioning without mediation of the network operator. The same application should in the future communicate with external lightpath provisioning applications to create true inter-domain lightpaths.

C. Lightpaths through GLIF

The last application of NDL we have been working on is the lightpath planning and provisioning in the GLIF, which was described in more detail in section II-B. GLIF constitutes an ideal environment to see NDL at work in a multi-domain and multi-administrator setup. The requirements are in this case the same as in the single domain case: that is, all parties involved need the same proper understanding of the network topology. Additionally, the information of each domain must be correlated to form a global information database.

NDL helps by describing the GLIF network in an abstract way. This means that only the information essential to the lightpath finding is provided and topology details inside each domain are hidden. To achieve this abstraction, each GOLE is described as a virtual device with several interfaces. These interfaces connect the GOLE to other GOLEs. To correlate the abstracted descriptions of each individual GOLE with each other we use the `seeAlso` property of RDF. Using these links, a linked web of descriptions is formed. This provides a global view of the network, where each domain maintains the description for their own GOLE.

Figure 4 shows an excerpt of the abstract description for the Netherlight GOLE. Line 1 describes which virtual device (GOLE) this is. Line 4 to 7 describe which interfaces the (virtual) device has. The descriptions are similar to the previous example. Interface `netherlight:if1` is described on line 10 to 16. line 11 describes its name and line 12 describes

where it is connected to. Finally, line 14 describes the capacity (total bandwidth) of the interface. Lines 18 to 20 provide a pointer for the description of the other side of the connection using the `seeAlso` property.

To find a path within the GLIF the first step is to read this NDL file. The description for the other GOLES is found by crawling the web using the links contained in the `seeAlso` properties. This way we can determine if a path exist and can be provisioned in the GLIF.

But there is more to be done: in such environment with different 'owners' and 'administrators', we need to take the authorisation policies of each domain into account. NDL can provide pointers to policy databases to the control plane that is determining the feasibility of the lightpath setup. The correlation of NDL to policy and authorisation models is part of future research, as well as the possible use of NDL in PNNI (Private Network-to-Network Interface) requests.

VII. Conclusion

In this article we have introduced hybrid networks and the related lightpath provisioning problems, both intra- and inter-domain. One of the main issues is the absence of an information model capable of describing the whole network, while also providing interoperability.

The ability to clearly describe resources and providing an interoperable description is one of the strong points of RDF. Based on RDF we have created the Network Description Language, a simple language that is capable of describing the physical, circuit-switched part of hybrid networks.

We have demonstrated several applications of NDL and have shown that it helps to reduce complexity and allows certain management and provisioning tasks to be automated. The classes and properties we created so far contain enough information to facilitate lightpath provisioning at the intra-domain level, to correlate information at the inter-domain level, and to create maps of both the local and global network.

So far we have intentionally kept the NDL schema simple, because we believe this facilitates the adoption from all the interested parties. Early adopters appreciate the relatively small work necessary to produce the NDL files especially when this is weighed against the results they allow to achieve.

In the future we plan to integrate this research in automatic provisioning tools. This will also allow us to test our model in a more practical situation and see if we must change or extend it further. We are also planning to extend NDL to be able to describe higher layers of the network, and to work on automatically creating abstractions of the network based on an NDL description.

```
1   <ndl:Device rdf:about="#netherlight">
        <ndl:name>Netherlight</ndl:name>
        <ndl:locatedAt rdf:resource="#NetherLight"/>
        <ndl:hasInterface rdf:resource="#netherlight:if1"/>
5       <ndl:hasInterface rdf:resource="#netherlight:if5"/>
        <ndl:hasInterface rdf:resource="#netherlight:if6"/>
        <ndl:hasInterface rdf:resource="#netherlight:if10"/>
    </ndl:Device>

10  <ndl:Interface rdf:about="#netherlight:if1">
        <ndl:name>if1</ndl:name>
        <ndl:connectedTo
         rdf:resource="http://networks.internet2.edu/manlan/manlan.rdf#manlan:if1"/>
        <ndl:capacity
15       rdf:datatype="http://www.w3.org/2001/XMLSchema#float">1.2E+9</ndl:capacity>
    </ndl:Interface>

    <ndl:Interface rdf:about="http://networks.internet2.edu/manlan/manlan.rdf#manlan:if1">
        <rdfs:seeAlso rdf:resource="http://networks.internet2.edu/manlan/manlan.rdf"/>
20  </ndl:Interface>
```

Fig. 4.   Part of the abstracted Netherlight NDL file

REFERENCES

[1] C. de Laat, E. Radius, S. Wallace, The rationale of the current optical networking initiatives, Future Generation Computer Systems 19 (6) (2003) 999–1008.

[2] L. Gommans, F. Dijkstra, C. de Laat, A. Taal, A. Wan, T. Lavian, I. Monga, F. Travostino, Applications drive secure lightpath creation across heterogeneous domains, Communications Magazine, IEEE 44 (3) (2006) 100–106.

[3] O. Martin, The ongoing evolution from packet based networks to hybrid networks in research & education networks, 2005.
URL http://www.jinr.ru/NEC/NEC-2005/proceeding2005/Martin.doc

[4] CERN, Large hadron collider.
URL http://lhc.web.cern.ch/lhc/

[5] European very large baseline interferometry network.
URL http://www.evlbi.org/

[6] International Telecommunication Union, Recommendation on DWDM (G.692).
URL http://www.itu.int/rec/T-REC-G.692/

[7] G. Bernstein, B. Rajagopalan, D. Saha, Optical Network Control – Architecture, Protocol and Standards, Addison-Wesley, 2004, Ch. 2, pp. 21–49.

[8] SURFnet, SURFnet6 lightpaths mark start of new Internet era (press release).
URL http://www.surfnet.nl/info/en/artikel_content.jsp?objectnumber=107197

[9] Global Lambda Integrated Facility (GLIF).
URL http://www.glif.is/

[10] J. Sobiesky, T. Lehman, Common service definition, Tech. rep., Mid-Atlantic Crossroads (MAX) (2005).
URL http://dragon.maxgigapop.net/twiki/bin/view/DRAGON/CommonServiceDefinition

[11] User-Controlled LightPaths (UCLP).
URL http://www.canarie.ca/canet4/uclp/

[12] T. Lehman, J. Sobieski, B. Jabbari, DRAGON: A framework for service provisioning in heterogeneous grid networks, IEEE Communications Magazine 44 (3).

[13] F. Dijkstra, Notes on fault isolation, Tech. rep., Universiteit van Amsterdam (2005).
URL http://staff.science.uva.nl/~fdijkstr/publications/fault_isolation.html

[14] J. Case, M. Fedor, M. Schoffstall, J. Davin, Simple Network Management Protocol (SNMP), RFC 1157 (Historic) (May 1990).
URL http://www.ietf.org/rfc/rfc1157.txt

[15] DMTF, Common Information Model (CIM).
URL http://www.dmtf.org/standards/cim/

[16] OASIS, Data Center Markup Language (DCML).
URL http://www.dcml.org/

[17] J. Spolsky, The law of leaky abstractions.
URL http://www.joelonsoftware.com/articles/LeakyAbstractions.html

[18] L. Berger, Generalized Multi-Protocol Label Switching (GMPLS) Signaling Functional Description, RFC 3471 (Proposed Standard) (Jan. 2003).
URL http://www.ietf.org/rfc/rfc3471.txt

[19] Network description language homepage.
URL http://www.science.uva.nl/research/sne/ndl/

[20] N. Shadbolt, T. Berners-Lee, W. Hall, The semantic web revisited, IEEE Intelligent Systems 21 (3).
URL http://dsonline.computer.org/portal/site/dsonline/menuitem.9ed3d9924aeb0dcd82ccc6716bbe36ec/index.jsp?&pName=dso_level1&path=dsonline/2006/07&file=x3sem.xml&xsl=article.xsl

[21] Resource Description Framework (RDF).
URL http://www.w3.org/RDF/

[22] Graphviz – graph visualization software.
URL http://www.graphviz.org/

[23] NetherLight.
URL http://www.netherlight.nl/

[24] Google, Google maps.
URL http://maps.google.com/

[25] F. Travostino, Using the semantic web to automate the operation of a hybrid internetwork, in: GridNets conference proceedings, 2005.