



# Network description language (NDL) applications

May 23 - 2006

RoN Spring 2006 Progress Meeting, Utrecht

Andree Toonk

SARA - High Performance Networking  
Amsterdam, The Netherlands

- What is NDL?
- Example of NDL
- How to create a NDL file
- How to use the NDL information
- Example applications

# Resource Description Format (RDF)

- Resource Description Format (RDF) is a Semantic Web technique.
- NDL is based on RDF and used to describe networks

Can be used to:

1. *Provide an overview of resources*
2. *Make path discovery easier*

This is currently being used in the NOC

# Example network description

- Netherlight network description in RDF format:

```
<!-- Description of TDM3.amsterdam1.netherlight.net -->
<ndl:Device rdf:about="#tdm3.amsterdam1.netherlight.net">
  <ndl:name>tdm3.amsterdam1.netherlight.net</ndl:name>
  <ndl:locatedAt rdf:resource="#amsterdam1.netherlight.net"/>
  <ndl:hasInterface:rdf:resource="#tdm3.amsterdam1.netherlight.net:503/3"/>
</ndl:Device>
```

# Example network description

- Netherlight network description in RDF format:

```

<!-- Description of TDM3.amsterdam1.netherlight.net -->
<ndl:Device rdf:about="#tdm3.amsterdam1.netherlight.net">
  <ndl:name>tdm3.amsterdam1.netherlight.net</ndl:name>
  <ndl:locatedAt rdf:resource="#amsterdam1.netherlight.net"/>
  <ndl:hasInterface:rdf:resource="#tdm3.amsterdam1.netherlight.net:503/3"/>
</ndl:Device>

<!-- Description of interfaces 503/3 TDM3.amsterdam1.netherlight.net -->
<ndl:Interface rdf:about="#tdm3.amsterdam1.netherlight.net:503/3">
  <ndl:name>tdm3.amsterdam1.netherlight.net:POS503/3</ndl:name>
  <ndl:connectedTo rdf:resource="#tdm1.geneva1.netherlight.net:5/1"/>
  <ndl:capacity rdf:resource="#OC192"/>
  <ndl:encapsulation rdf:resource="#SONET"/>
</ndl:Interface>

```

# Step 1: Create a NDL File

- Create one manually
- Create with the help of java script NDL generator
- In the SURFnet6 SDH / OME6500 network this is done automatically:
  - Manual create a list of all Network elements
  - Script (based on TL1 Toolkit) to get all the cards / ports in the network elements
  - Script (based on TL1 Toolkit) to get neighbor NE's of SDH cards
  - Parse all the information and create the NDL file

Currently this is a file of 3756 lines

will probably grow to 4500 lines in the next 3 months

Conclusion: we don't want to do this manually in SURFnet6

# Use NDL information

- Result: Now we have a NDL File with information about NE's and ports
- Next: How to use this?
  
- Developed 2 Perl libraries:
  - NDL.pm
  - Capacity.pm
  
  - Creates the objects \$device and \$interface
  
- Can be used to get the information from the NDL files and use this information in Perl programs.



# Use NDL information

Some example functions:

NDL.pm:

```
$ndl->read_NDL();  
$ndl->get_devices();  
$ndl->get_interfaces();  
$ndl->get_neighbour($if);  
$ndl->get_device_for_interface($if);  
$ndl->get_capacity($if);  
$ndl->get_encapsulation($if);  
$ndl->get_slot($if);  
$ndl->get_port($if);  
$ndl->build_graph()
```

Capacity.pm:

```
$ndl->get_free_gigE();  
$ndl->get_nr_free_timeslots($if);  
$ndl->get_vc4s($if);
```



- Create an overview / list of devices and interfaces
- Visualization of the network
- Path finding

# Overview of devices

Create an overview of devices in HTML format  
is easy by using the Perl libraries

Let's look at an example:

[Overview of NE information](#)

[Backbone usage report](#)



Interface information for Asd001a\_ome02

Interface	Card Type	Neighbour	Capacity (STS Timeslots)	number of free STS timeslots
01a_ome02:1/1	SONET	Asd001a_ome01:1/1	48	3
01a_ome02:1/2	SONET	Asd001a_ome01:1/2	48	6
01a_ome02:10/1	SONET	Asd002a_ome02:10/1	192	69
01a_ome02:11/1	SONET	Nm001a_ome01:4/1	192	192
01a_ome02:12/1	SONET	Asd001a_ome05:11/1	192	192
01a_ome02:13/1	SONET	Asd001a_ome04:11/1	192	171
01a_ome02:14/1	SONET	Asd001a_ome03:11/1	192	171
01a_ome02:4/1	SONET	Nm001a_ome02:6/1	192	0
01a_ome02:5/1	SONET	Asd001a_ome06:10/1	192	81
01a_ome02:6/1	SONET	Asd001a_ome07:10/1	192	102

01a_ome01:6/1 - Ap001a_ome01:9/1	OC192 - free timeslots 6	97%
01a_ome01:9/1 - Nm001a_ome01:6/1	OC192 - free timeslots 24	88%
001a_ome01:1/1 - Hedr001a_ome01:2/1	OC48 - free timeslots 0	100%
001a_ome01:1/2 -	OC48 - free timeslots 6	88%
001a_ome01:5/1 - Asd001a_ome04:5/1	OC192 - free timeslots 102	47%
01a_ome01:6/1 - Zl001a_ome01:9/1	OC192 - free timeslots 9	95%
01a_ome01:9/1 - Ah001a_ome01:6/1	OC192 - free timeslots 6	97%
001a_ome01:1/1 - Asd001a_ome02:1/1	OC48 - free timeslots 3	94%
001a_ome01:1/2 - Asd001a_ome02:1/2	OC48 - free timeslots 6	88%
001a_ome01:10/1 - Asd002a_ome01:10/1	OC192 - free timeslots 129	33%
001a_ome01:13/1 - Asd001a_ome06:11/1	OC192 - free timeslots 192	0%
001a_ome01:3/1 - Ut001a_ome01:6/1	OC192 - free timeslots 0	100%
001a_ome01:4/1 - Ledn001a_ome01:5/1	OC192 - free timeslots 129	33%
001a_ome01:5/1 - Asd001a_ome03:10/1	OC192 - free timeslots 0	100%
001a_ome01:6/1 - Asd001a_ome04:10/1	OC192 - free timeslots 84	56%
001a_ome01:9/1 - Asd001a_ome05:10/1	OC192 - free timeslots 150	22%
001a_ome02:1/1 - Asd001a_ome01:1/1	OC48 - free timeslots 3	94%
001a_ome02:1/2 - Asd001a_ome01:1/2	OC48 - free timeslots 6	88%
001a_ome02:10/1 - Asd002a_ome02:10/1	OC192 - free timeslots 69	64%
001a_ome02:11/1 - Nm001a_ome01:4/1	OC192 - free timeslots 192	0%
001a_ome02:12/1 - Asd001a_ome05:11/1	OC192 - free timeslots 192	0%
001a_ome02:13/1 - Asd001a_ome04:11/1	OC192 - free timeslots 171	11%
001a_ome02:14/1 - Asd001a_ome03:11/1	OC192 - free timeslots 171	11%
001a_ome02:4/1 - Nm001a_ome02:6/1	OC192 - free timeslots 0	100%
001a_ome02:5/1 - Asd001a_ome06:10/1	OC192 - free timeslots 81	58%
001a_ome02:6/1 - Asd001a_ome07:10/1	OC192 - free timeslots 102	47%
001a_ome03:10/1 - Asd001a_ome01:5/1	OC192 - free timeslots 0	100%
001a_ome03:11/1 - Asd001a_ome02:14/1	OC192 - free timeslots 171	11%
001a_ome03:5/1 - Spl001a_ome01:6/1	OC192 - free timeslots 0	100%
001a_ome03:6/1 - Gv001a_ome01:6/1	OC192 - free timeslots 108	44%
001a_ome03:9/1 - Dt001a_ome01:6/1	OC192 - free timeslots 171	11%
001a_ome04:10/1 - Asd001a_ome01:6/1	OC192 - free timeslots 84	56%
001a_ome04:11/1 - Asd001a_ome02:13/1	OC192 - free timeslots 171	11%
001a_ome04:5/1 - Amr001a_ome01:5/1	OC192 - free timeslots 102	47%
001a_ome04:6/1 - Hvs001a_ome01:6/1	OC192 - free timeslots 150	22%
001a_ome04:9/1 - Rt001a_ome01:6/1	OC192 - free timeslots 87	55%
001a_ome05:10/1 - Asd001a_ome01:9/1	OC192 - free timeslots 150	22%
001a_ome05:11/1 - Asd001a_ome02:12/1	OC192 - free timeslots 192	0%

# Visualization of the network

- Provide an overview of resources by visualization
- Graphviz - Graph Visualization Software
  - uses .dot file as input
- Relatively easy to generate a .dot input file from an RDF file
- Python script to generate a .dot file from an RDF file\*
- Then generate a png,jpg,gif and lots more..
  - `dot -o netherlight.png -Tpng netherlight.dot`

Let's see the result.... ☺

\* Author: Jeroen van der Ham, University of Amsterdam



# Provide an overview of resources

Visualization of Netherlight network elements.  
(Netherlight elements only)

tdm4.amsterdam1.netherlight.net

5/1

501/1

tdm1.amsterdam1.netherlight.net

12/1

6/1

501/2

505/1

tdm3.amsterdam1.netherlight.net

504/4

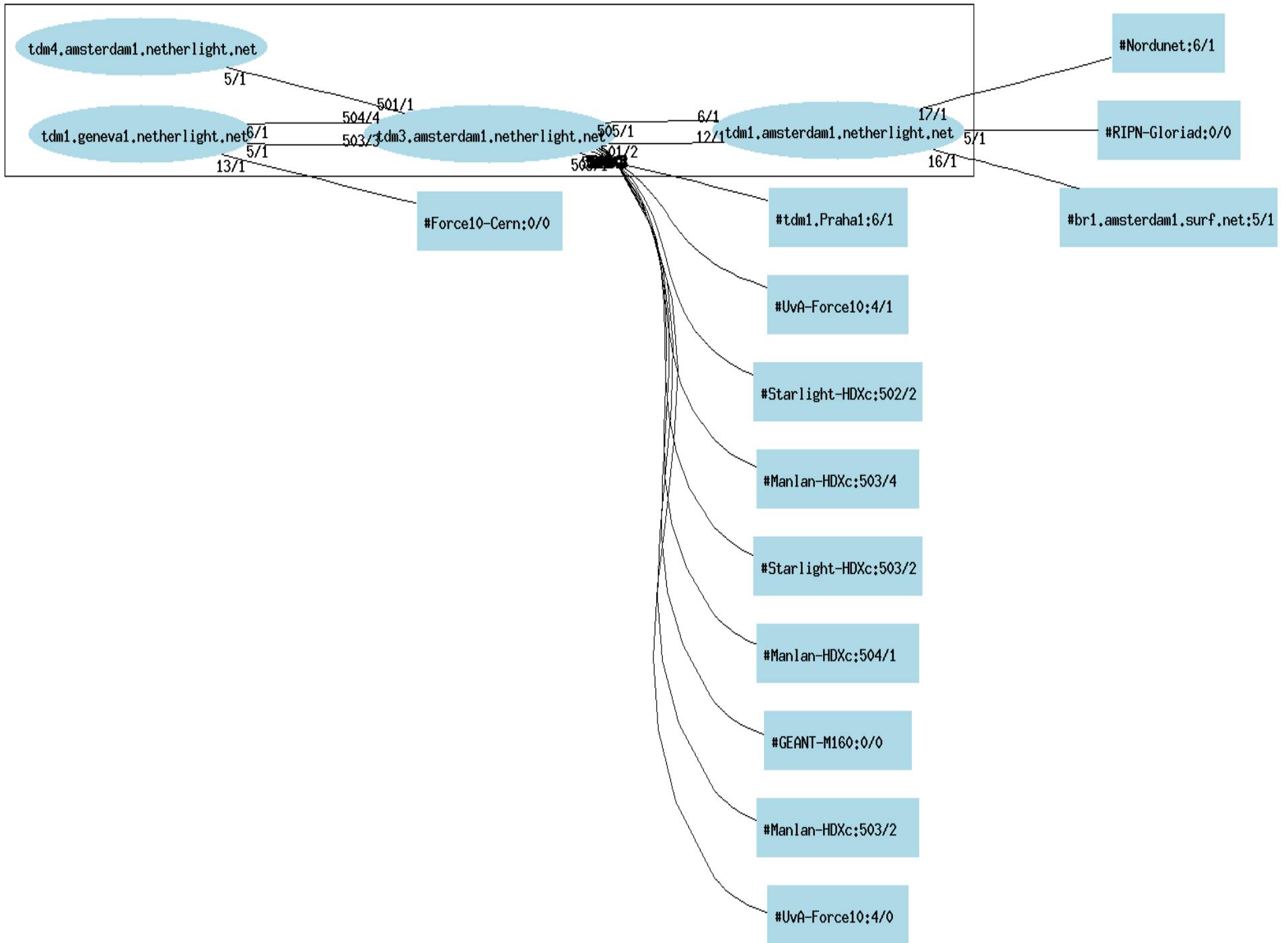
503/3

6/1

5/1

tdm1.geneva1.netherlight.net

Visualization of Netherlight network elements including edge devices on next slide

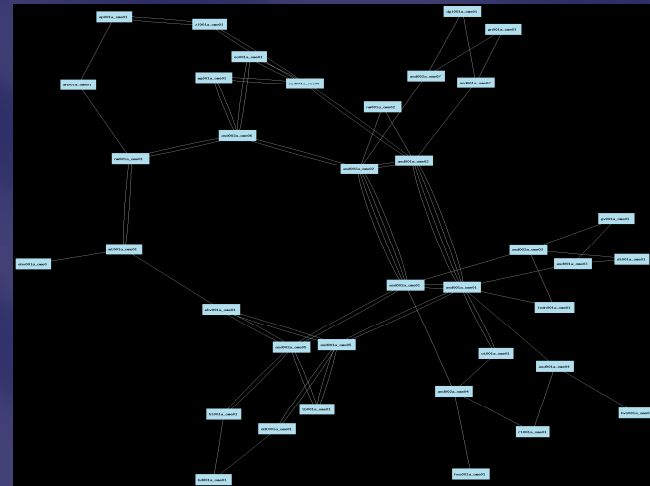






- Provisioning a lightpath through a 'big' network can be time consuming
  - 1) finding a (protected) path across the network
  - 2) enough time slots available?

Might sound easy but can be difficult →



- Tool written by SARA's network research group to automate this
- Uses an RDF network description as topology description
- Uses a MySQL database that has knowledge of all current circuits in the network

- With the RDF information we build a graph representing the network.
  - SPARQL query that gets all connected interfaces
  - check if these interfaces have enough free timeslots available (MySQL)
  - add the two interfaces as vertices to the graph
  - add the link between them as an edge of the graph
  - add edges to the graph for all internal connections between all the interfaces of one network element
  
- **Now a graph representing the network has been build.**
- Use the Dijkstra algorithm to find the shortest path
- returns all the information to create the cross connects on the network elements,
  - ▶ Includes: network elements, slots, ports, timeslots (MySQL) all the information to create the cross connects on the network elements,

The NOC uses this to:

- find path through the SURFnet6 network
- Application to make reservations

Demo:

[Path provisioning tool](#)

Next step:

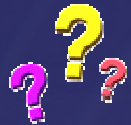
- Support to configure Xconnects on the network elements
  - Extend PCE (Path Computation Element) daemon
  - Describe other parts of the network (CPL)
  - How to integrate different types of networks in NDL
- 
- NDL in GLIF

# URLs + Questions

- SARA's network research group:  
Information about TL1 toolkit & NDL tools.

<http://nrg.sara.nl/>

Email: nrg@sara.nl



- Network Description Language:

<http://www.science.uva.nl/research/sne/ndl/>

That's all Folks!