Lambda-Grid developments

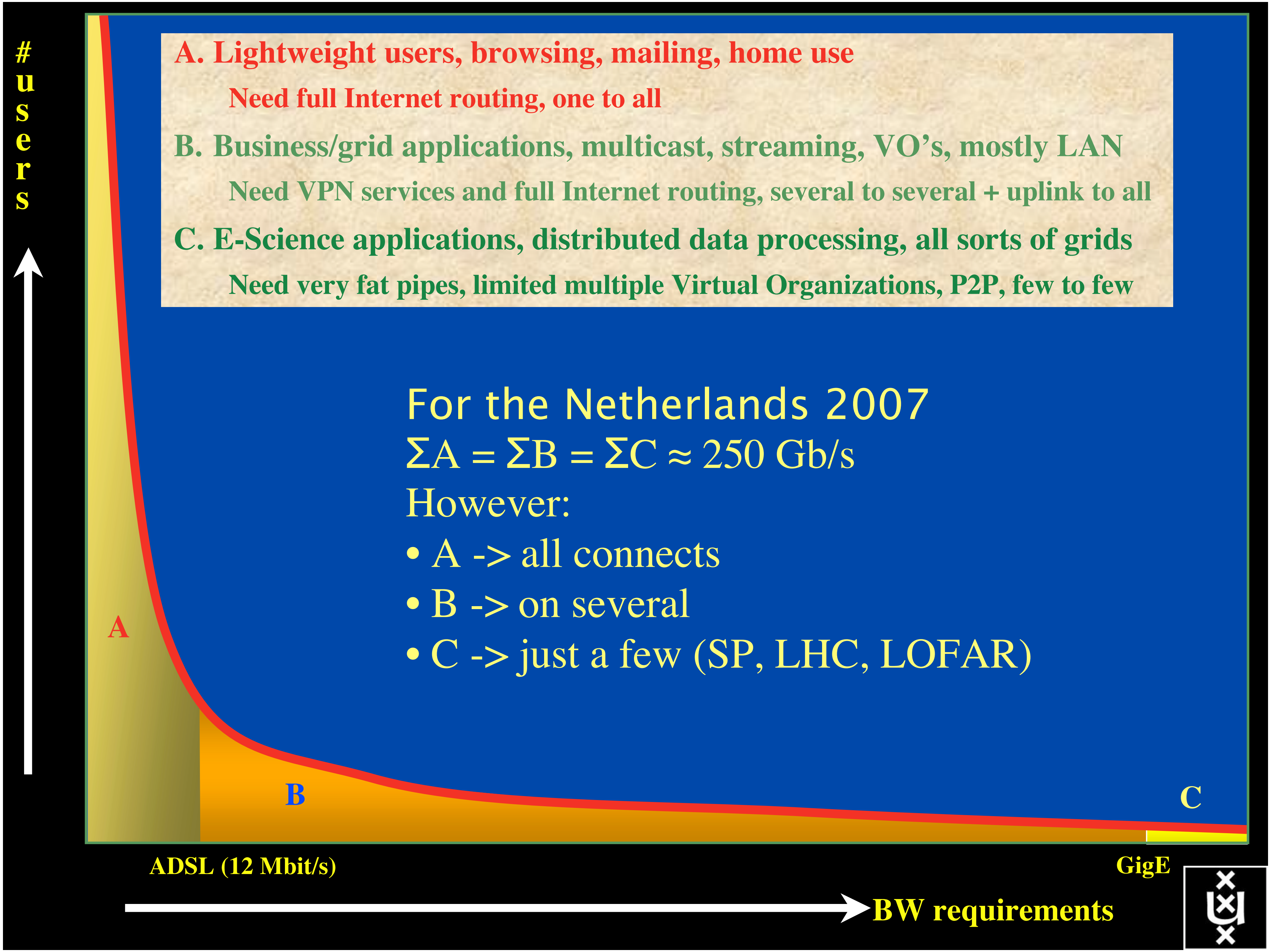
History - Present - Future

Cees de Laat

University of Amsterdam

SURFnet

Contents


A. Lightweight users, browsing, mailing, home use
Need full Internet routing, one to all
B. Business/grid applications, multicast, streaming, VO's, mostly LAN
Need VPN services and full Internet routing, several to several + uplink to all
C. E-Science applications, distributed data processing, all sorts of grids
Need very fat pipes, limited multiple Virtual Organizations, P2P, few to few
For the Netherlands 2007
ΣA = ΣB = ΣC ≈ 250 Gb/s
However:
• A -> all connects
• B -> on several
• C -> just a few (SP, LHC, LOFAR)
users
A
B
C
ADSL (12 Mbit/s)
GigE
BW requirements

Towards Hybrid Networking!

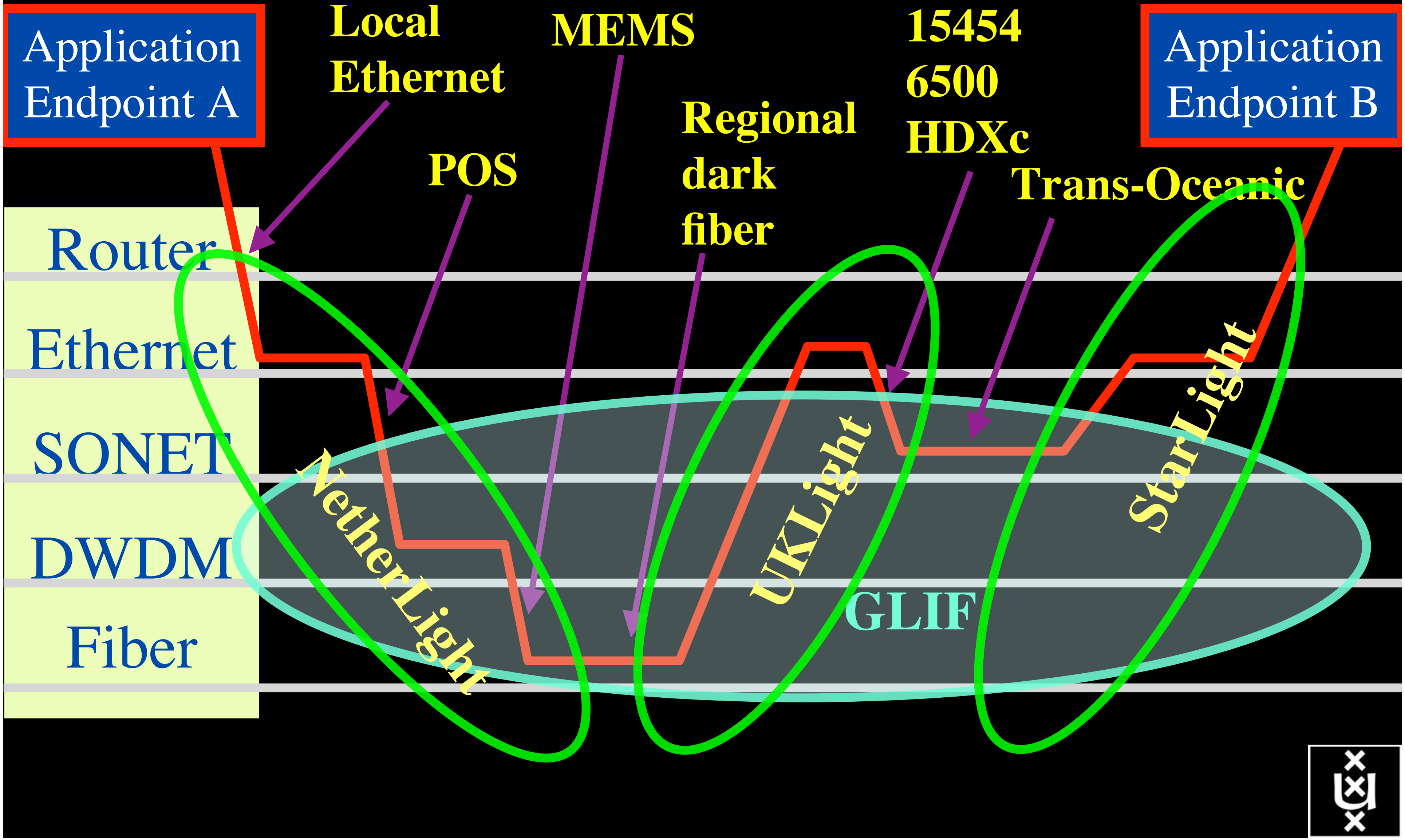
- Costs of photonic equipment 10% of switching 10 % of full routing
  - for same throughput!
  - Photonic vs Optical (optical used for SONET, etc, 10-50 k$/port)
  - DWDM lasers for long reach expensive, 10-50 k$
- Bottom line: look for a hybrid architecture which serves all classes in a cost effective way
  - map A -> L3 , B -> L2 , C -> L1 and L2
- Give each packet in the network the service it needs, but no more !

L1 ≈ 2-3 k$/port

L2 ≈ 5-8 k$/port

L3 ≈ 75+ k$/port

How low can you go?



In The Netherlands SURFnet connects between 180:

- universities;
- academic hospitals;
- most polytechnics;
- research centers.

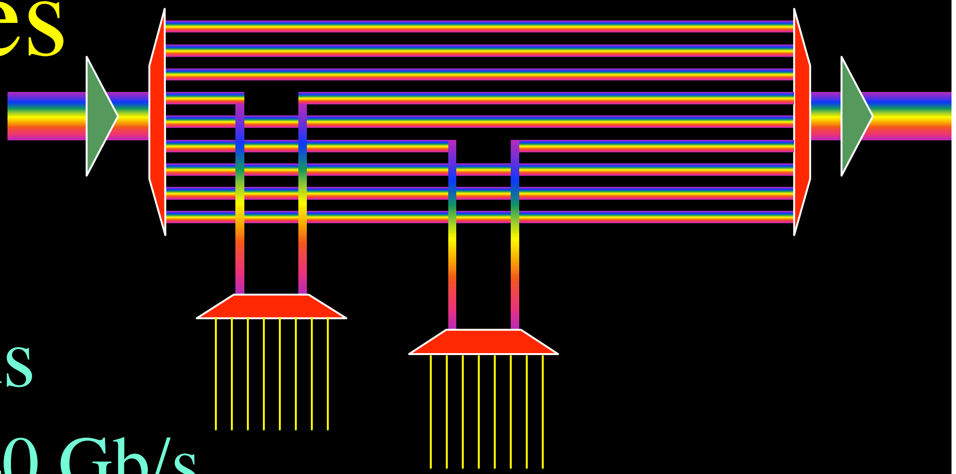
with an indirect ~750K user base

Red crosses = StarPlane

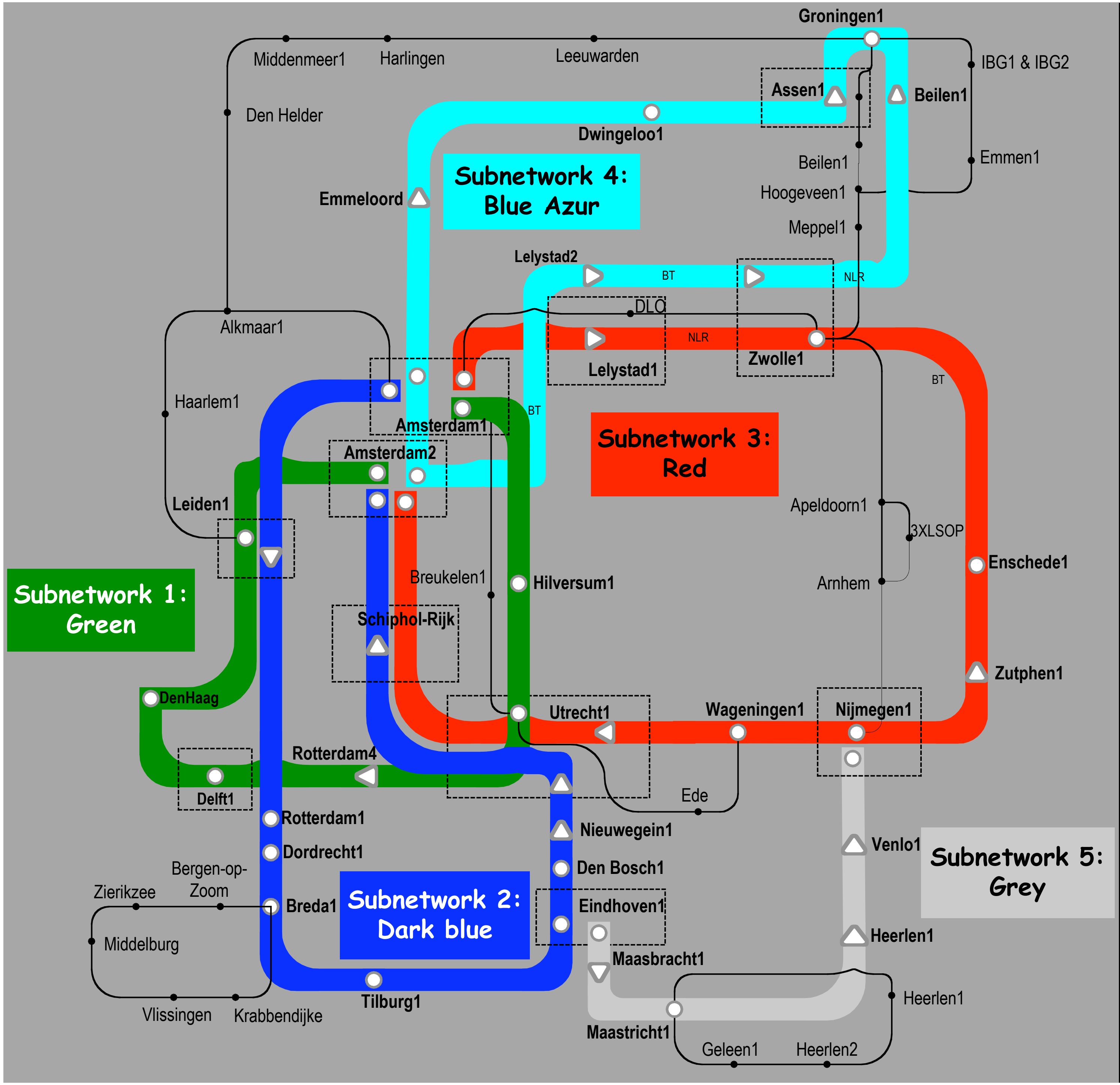
~ 6000 km

scale comparable to railway system

SURFnet 6 principles



- Based on dark fiber
- 4 DWDM rings of 9 bands
  - Each capable of 10, later 40 Gb/s
  - each 4 (100 GHz spacing) or 8 (50 GHz spacing) colors
- Universities each have 1 band to connect their Routers +LightPaths
- Connect with 1 or 10 Gb/s Ethernet LanPhy
- Routing in Amsterdam in 2 core POP's!
- International connectivity in Amsterdam
- Lambda service between ring POP's and to NetherLight



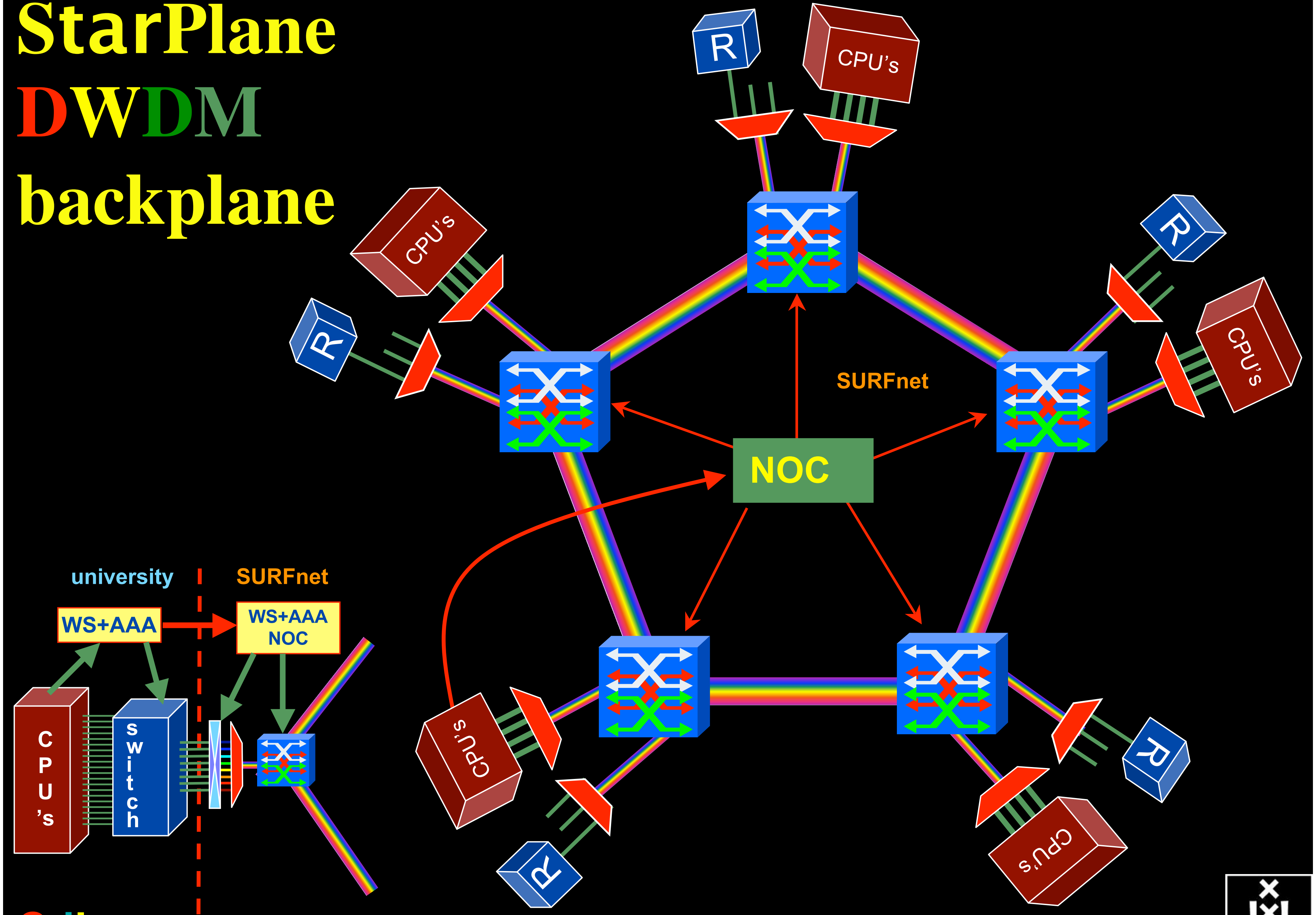
Common Photonic Layer (CPL) in SURFnet6

supports up to
72 Lambda's of
10 G each
40 G soon.

Contents

1. The need for hybrid networking
2. StarPlane; a grid controlled photonic network
3. Cross Domain Authorization using Tokens
4. RDF/Network Description Language
5. Tera-networking
6. Programmable networks

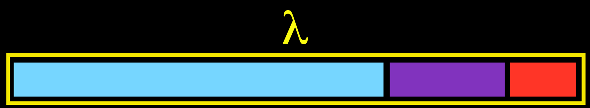
StarPlane DWDM backplane



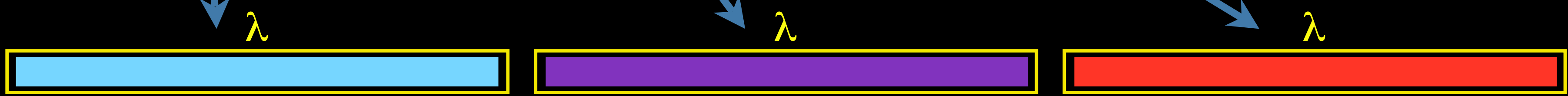
CdL

QOS in a non destructive way!

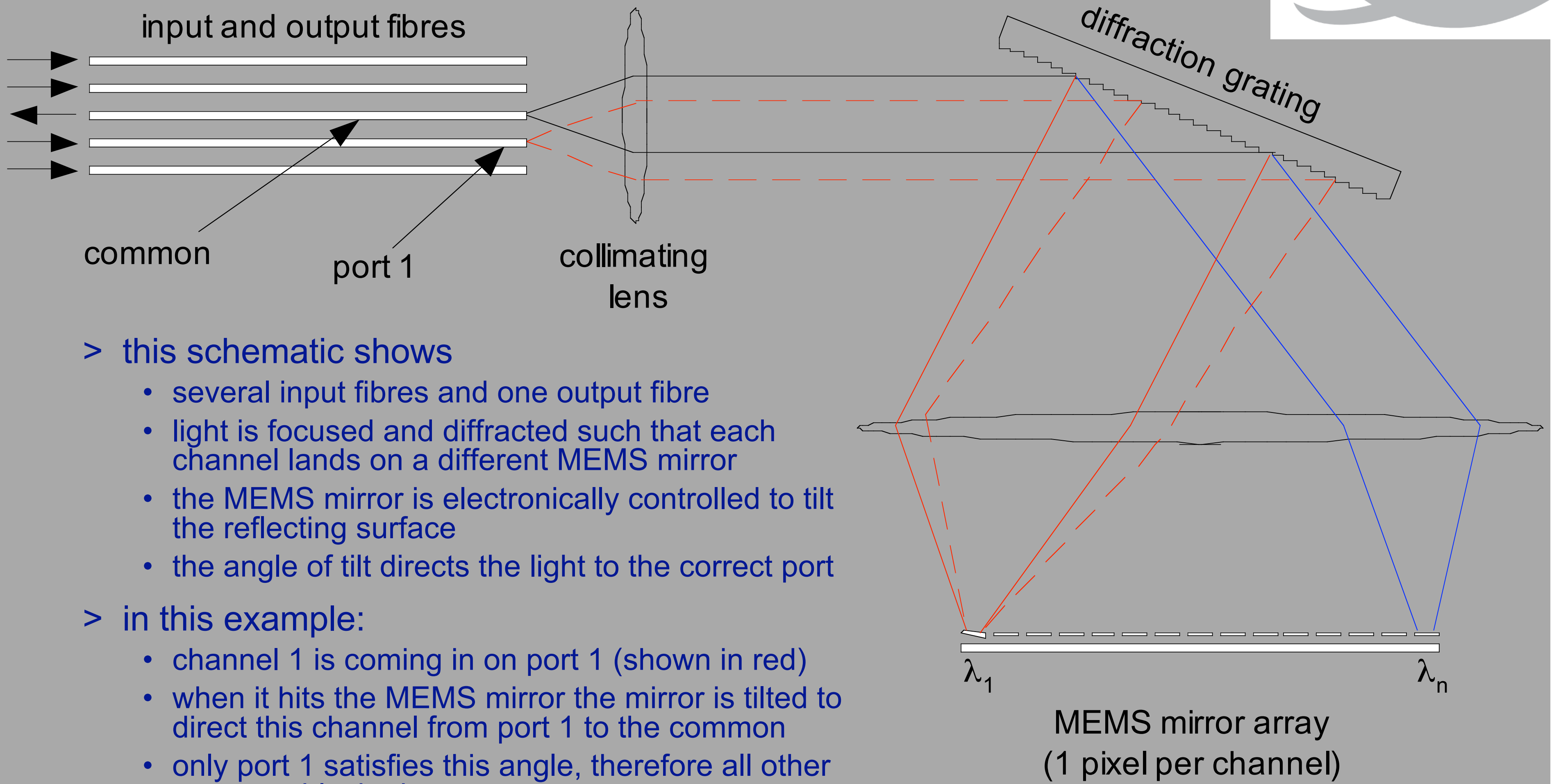
- Destructive QOS:
  - have a link or λ
  - set part of it aside for a lucky few under higher priority
  - rest gets less service



- Constructive QOS:
  - have a λ
  - add other λ's as needed on separate colors
  - move the lucky ones over there
  - rest gets also a bit happier!



Module Operation



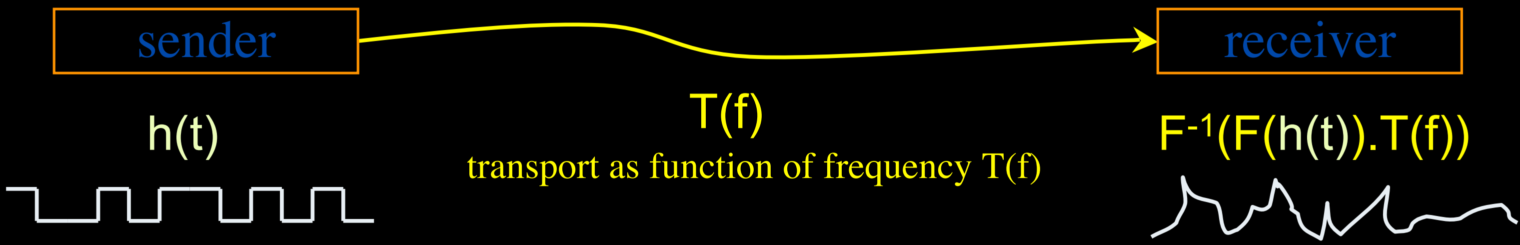
> this schematic shows
- several input fibres and one output fibre
- light is focused and diffracted such that each channel lands on a different MEMS mirror
- the MEMS mirror is electronically controlled to tilt the reflecting surface
- the angle of tilt directs the light to the correct port

> in this example:
- channel 1 is coming in on port 1 (shown in red)
- when it hits the MEMS mirror the mirror is tilted to direct this channel from port 1 to the common
- only port 1 satisfies this angle, therefore all other ports are blocked

ref Eric Bernier, NORTEL

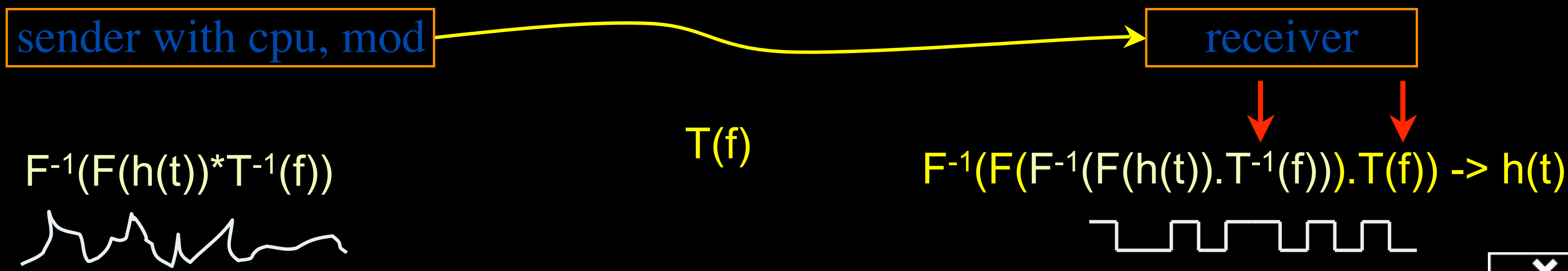
Dispersion compensating modem: eDCO from NORTEL

(Try to Google eDCO :-)



Solution in 5 easy steps for dummy's :

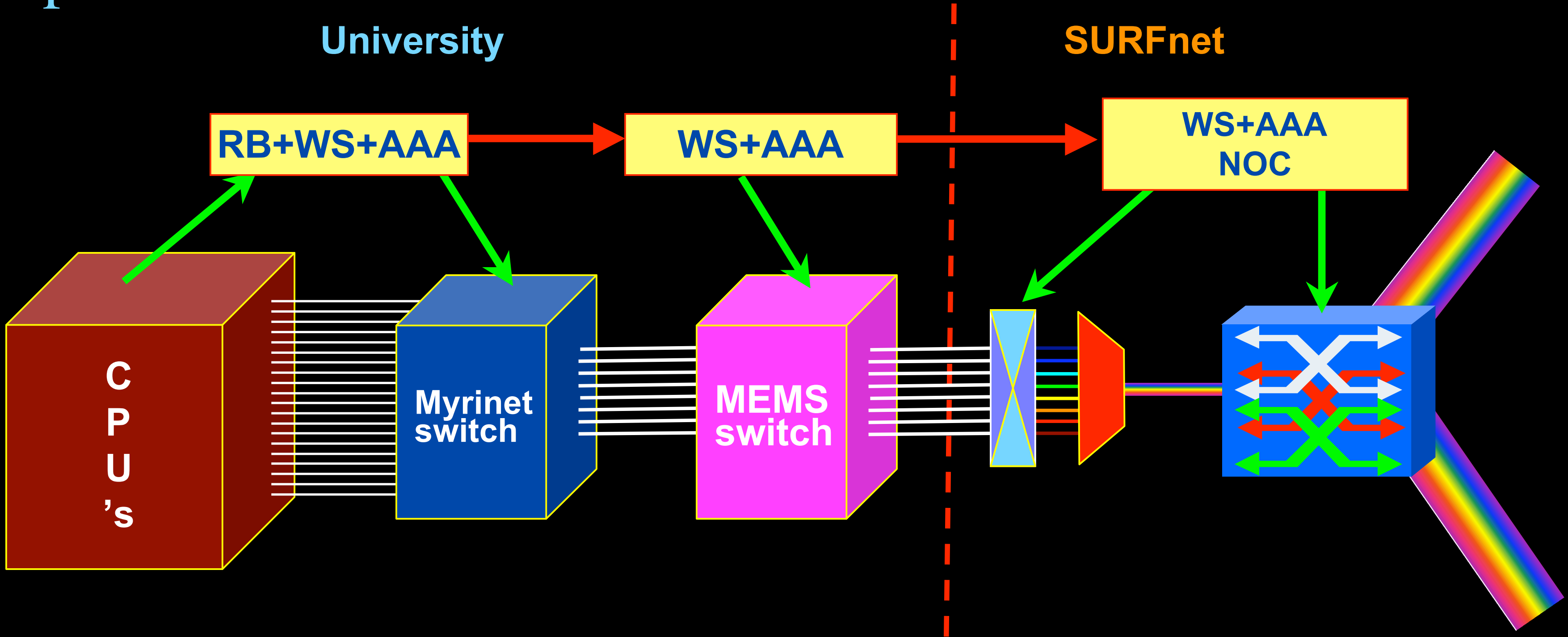
1. try to figure out T(f) by trial and error
2. invert T(f) -> $T^{-1}(f)$
3. computationally multiply $T^{-1}(f)$ with Fourier transform of bit pattern to send
4. inverse Fourier transform the result from frequency to time space
5. modulate laser with resulting $h'(t) = F^{-1}(F(h(t)).T^{-1}(f))$



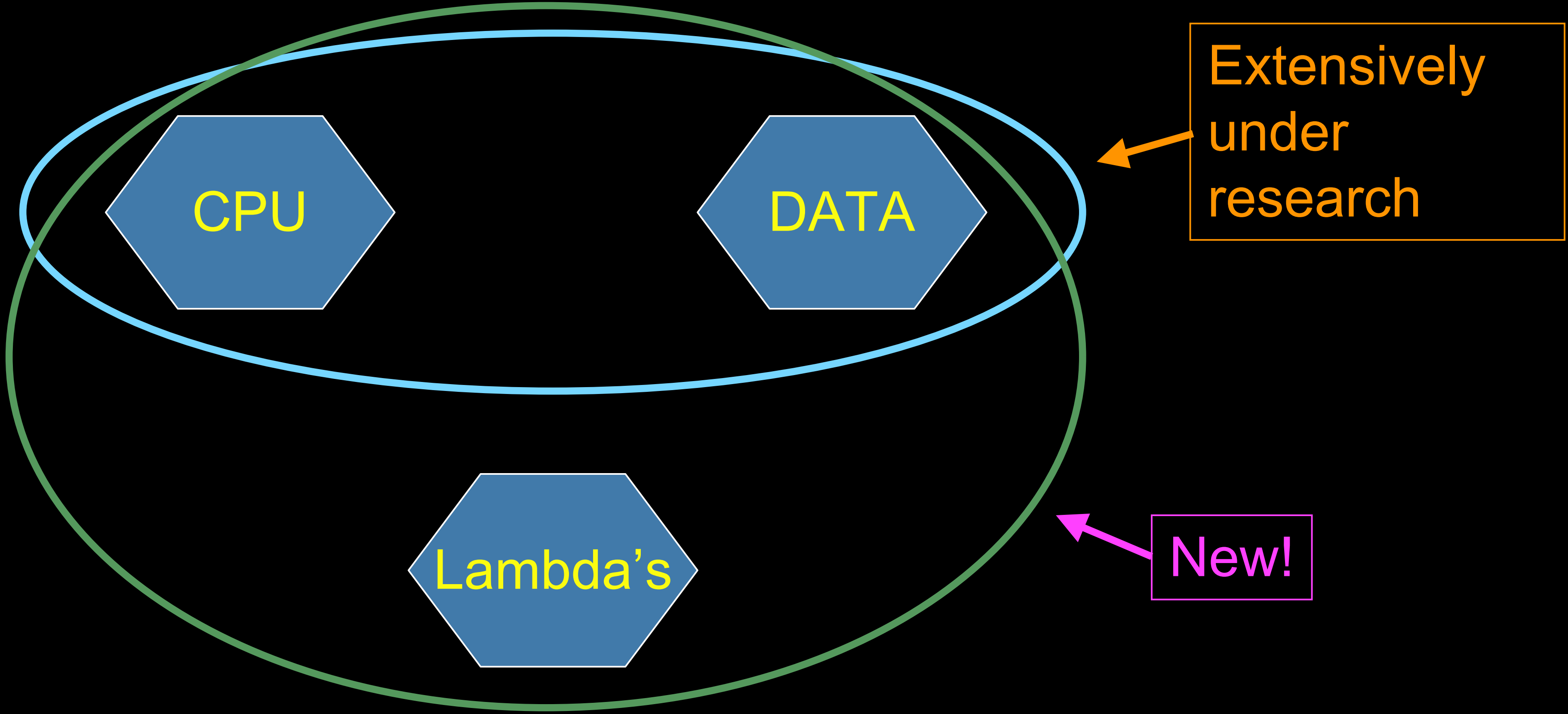
(ps. due to power ~ square E the signal to send **looks** like uncompensated received but is not)

The challenge for sub-second switching

- bringing up/down a λ takes minutes
  - this was fast in the era of old time signaling (phone/fax)
  - λ 2 λ influence (Amplifiers, non linear effects)
  - however minutes is historically grown, 5 nines, up for years
  - working with Nortel to get setup time significantly down
- plan B:

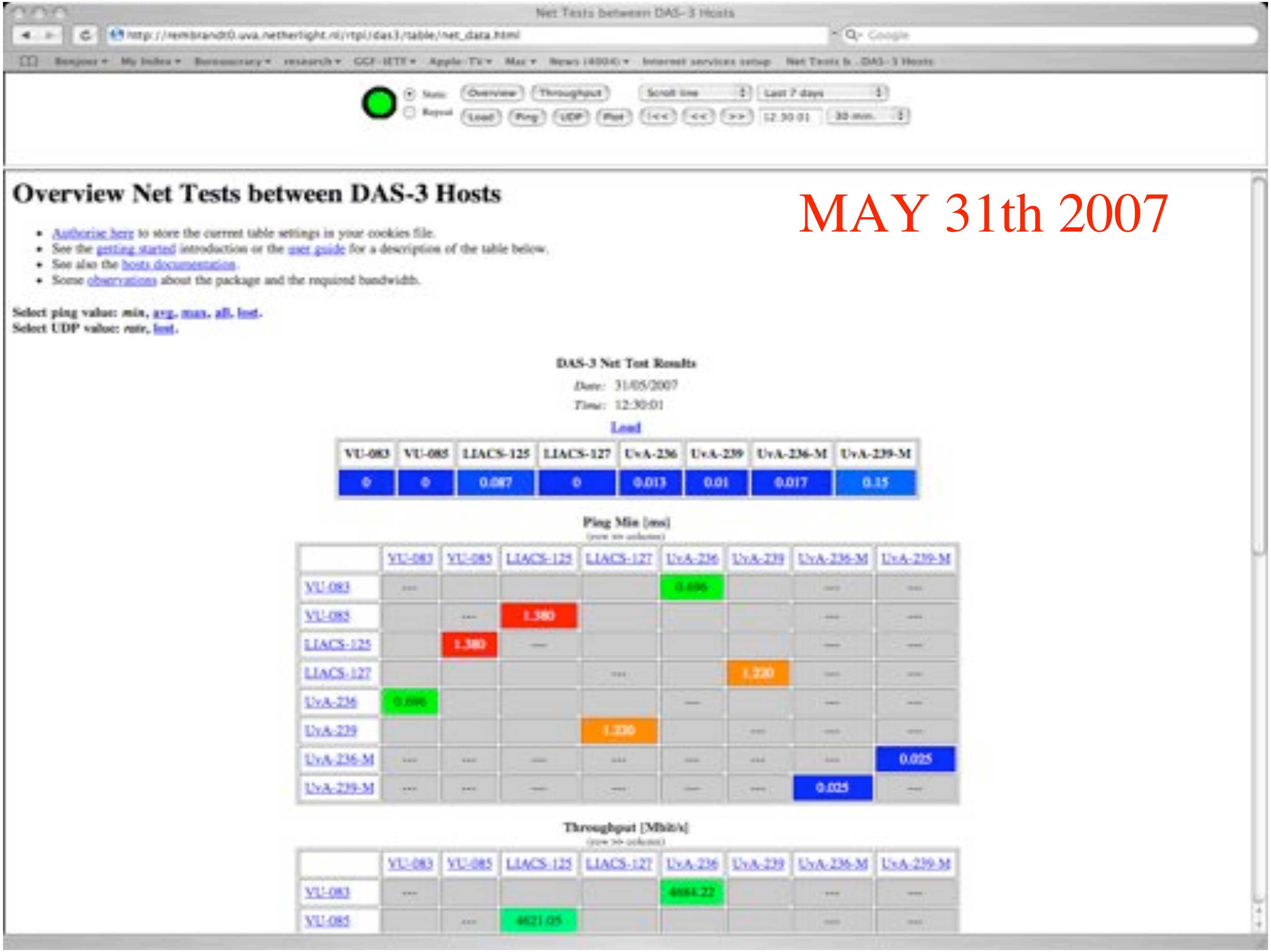


GRID Co-scheduling problem space



The StarPlane vision is to give flexibility directly to the applications by allowing them to choose the logical topology in real time, ultimately with sub-second lambda switching times on part of the SURFnet6 infrastructure.

MAY 31th 2007



Overview Net Tests between DAS-3 Hosts

- Authorise here to store the current table settings in your cookies file.
- See the getting started introduction or the user guide for a description of the table below.
- See also the hosts documentation.
- Some observations about the package and the required bandwidth.

Select ping value: ***min***, avg, max, all, lost.
Select UDP value: ***rate***, lost.

DAS-3 Net Test Results

Date: 31/05/2007

Time: 12:30:01

Load

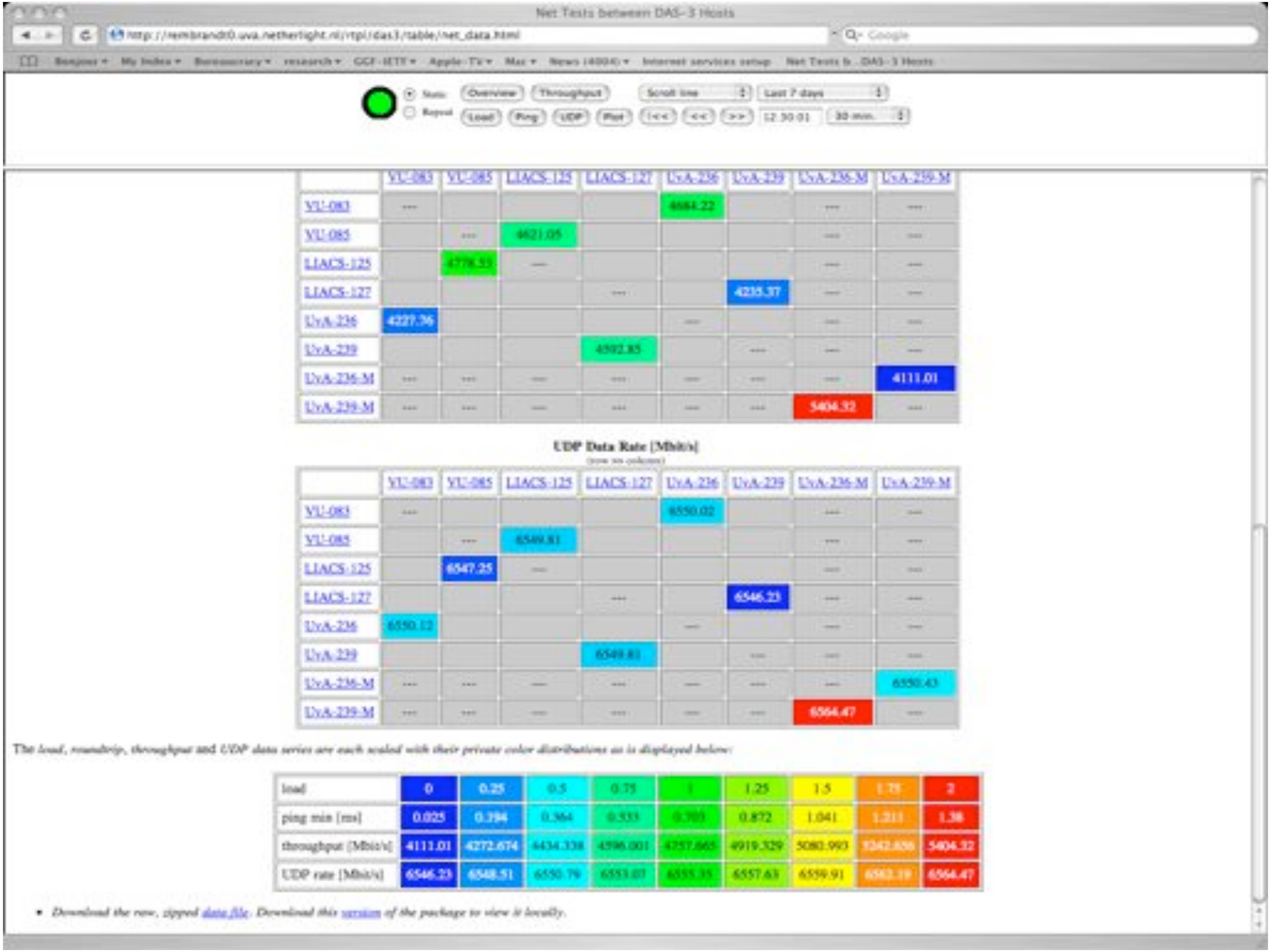
VU-083	VU-085	LIACS-125	LIACS-127	UvA-236	UvA-239	UvA-236-M	UvA-239-M
0	0	0.087	0	0.013	0.01	0.017	0.15

Ping Min [ms]
(row => column)

	VU-083	VU-085	LIACS-125	LIACS-127	UvA-236	UvA-239	UvA-236-M	UvA-239-M
VU-083	---				0.696		---	---
VU-085		---	1.380				---	---
LIACS-125		1.380	---				---	---
LIACS-127				---		1.220	---	---
UvA-236	0.696				---		---	---
UvA-239				1.220		---	---	---
UvA-236-M	---	---	---	---	---	---	---	0.025
UvA-239-M	---	---	---	---	---	---	0.025	---

Throughput [Mbit/s]
(row => column)

	VU-083	VU-085	LIACS-125	LIACS-127	UvA-236	UvA-239	UvA-236-M	UvA-239-M
VU-083	---				4684.22		---	---
VU-085		---	4621.05				---	---



Net Tests between DAS-3 Hosts

http://rembrandt0.uva.netherlight.nl/rtpl/das3/table/net_data.html

	VU-083	VU-085	LIACS-125	LIACS-127	UvA-236	UvA-239	UvA-236-M	UvA-239-M
VU-083	---				4684.22		---	---
VU-085		---	4621.05				---	---
LIACS-125		4778.53	---				---	---
LIACS-127				---		4235.37	---	---
UvA-236	4227.76				---		---	---
UvA-239				4592.85		---	---	---
UvA-236-M	---	---	---	---	---	---	---	4111.01
UvA-239-M	---	---	---	---	---	---	5404.32	---

UDP Data Rate [Mbit/s]

(row on column)

	VU-083	VU-085	LIACS-125	LIACS-127	UvA-236	UvA-239	UvA-236-M	UvA-239-M
VU-083	---				6550.02		---	---
VU-085		---	6549.81				---	---
LIACS-125		6547.25	---				---	---
LIACS-127				---		6546.23	---	---
UvA-236	6550.12				---		---	---
UvA-239				6549.81		---	---	---
UvA-236-M	---	---	---	---	---	---	---	6550.43
UvA-239-M	---	---	---	---	---	---	6564.47	---

The *load, roundtrip, throughput* and *UDP data series are each scaled with their private color distributions as is displayed below:*

load	0	0.25	0.5	0.75	1	1.25	1.5	1.75	2
ping min [ms]	0.025	0.194	0.364	0.533	0.703	0.872	1.041	1.211	1.38
throughput [Mbit/s]	4111.01	4272.674	4434.338	4596.001	4757.665	4919.329	5080.993	5242.656	5404.32
UDP rate [Mbit/s]	6546.23	6548.51	6550.79	6553.07	6555.35	6557.63	6559.91	6562.19	6564.47

- *Download the raw, zipped data file. Download this version of the package to view it locally.*

Net Tests between DAS-3 Hosts

http://rembrandt0.uva.netherlight.nl/rtpl/das3/table/net_data.html

Ping All [ms] from / to node125.das3.liacs.nl (LIACS-125)

Skipped tests: UvA-236-M, UvA-239-M

Date	*Time*	>> VU-083	<< VU-083	>> VU-085	<< VU-085	>> LIACS-127	<< LIACS-127	>> UvA-236	<< UvA-236	>> UvA-239	<< UvA-239
31/05/2007	12:30:01			1.380 / 1.382 / 1.410	1.380 / 1.383 / 1.420						
31/05/2007	12:00:01			1.380 / 1.383 / 1.410	1.380 / 1.384 / 1.450						
31/05/2007	11:30:01			1.380 / 1.383 / 1.410	1.380 / 1.382 / 1.390						
31/05/2007	11:00:02			1.380 / 1.382 / 1.410	1.380 / 1.382 / 1.400						
31/05/2007	10:30:01			1.380 / 1.383 / 1.390	1.380 / 1.382 / 1.390						
31/05/2007	10:00:01			1.380 / 1.382 / 1.410	1.380 / 1.383 / 1.410						
31/05/2007	09:30:01			1.380 / 1.384 / 1.410	1.380 / 1.382 / 1.400						
31/05/2007	09:00:01			1.380 / 1.382 / 1.410	1.380 / 1.383 / 1.400						
31/05/2007	08:30:02			1.380 / 1.383 / 1.410	1.380 / 1.382 / 1.400						
31/05/2007	08:00:01			1.380 / 1.383 / 1.410	1.380 / 1.383 / 1.410						
31/05/2007	07:30:02			1.380 / 1.382 / 1.390	1.380 / 1.381 / 1.390						
31/05/2007	07:00:01			1.380 / 1.382 / 1.410	1.380 / 1.383 / 1.400						
31/05/2007	06:30:01			1.380 / 1.383 / 1.410	1.380 / 1.382 / 1.390						
31/05/2007	06:00:01			1.380 / 1.382 / 1.410	1.380 / 1.382 / 1.420						
31/05/2007	05:30:01			1.380 / 1.382 / 1.400	1.380 / 1.382 / 1.410						
31/05/2007	05:00:01			1.380 / 1.382 / 1.410	1.380 / 1.382 / 1.390						
31/05/2007	04:30:01			1.380 / 1.381 / 1.390	1.380 / 1.381 / 1.390						
31/05/2007	04:00:01			1.380 / 1.382 / 1.410	1.380 / 1.384 / 1.410						
31/05/2007	03:30:02			1.380 / 1.384 / 1.410	1.380 / 1.382 / 1.400						
31/05/2007	03:00:02			1.380 / 1.382 / 1.410	1.380 / 1.382 / 1.400						
31/05/2007	02:30:01			1.380 / 1.382 / 1.400	1.380 / 1.382 / 1.400						
31/05/2007	02:00:01			1.380 / 1.383 / 1.410	1.380 / 1.384 / 1.410						
31/05/2007	01:30:01			1.380 / 1.382 / 1.410	1.380 / 1.382 / 1.390						
31/05/2007	01:00:01			1.380 / 1.382 / 1.410	1.380 / 1.383 / 1.400						

Very constant and predictable!

What makes StarPlane fly?

- Wavelength Selective Switches
  - for the “low cost” photonics
- Sandbox by confining StarPlane to one band
  - for experimenting on a production network
- Optimization of the controls to turn on/off a Lambda
  - direct access to part of the controls at the NOC
- electronic Dynamically Compensating Optics (eDCO)
  - to compensate for changing lengths of the path
- traffic engineering
  - to create the OPN topologies needed by the applications
- Open Source GMPLS
  - to facilitate policy enabled cross domain signaling

Power is a big issue

- UvA cluster uses (max) 30 kWh
- 1 kWh ~ 0.1 €
- per year -> 26 k€/y
- add cooling 50% -> 39 k€/y
- Emergency power system -> 50 k€/y
- per rack 10 kWh is now normal
- **YOU BURN ABOUT HALF THE CLUSTER OVER ITS LIFETIME!**

- Terminating a 10 Gb/s wave costs about 200 W
- Entire loaded fiber -> 16 kW
- Wavelength Selective Switch : few W!

Contents

1. The need for hybrid networking
2. StarPlane; a grid controlled photonic network
3. Cross Domain Authorization using Tokens
4. RDF/Network Description Language
5. Tera-networking
6. Programmable networks

Simple service access

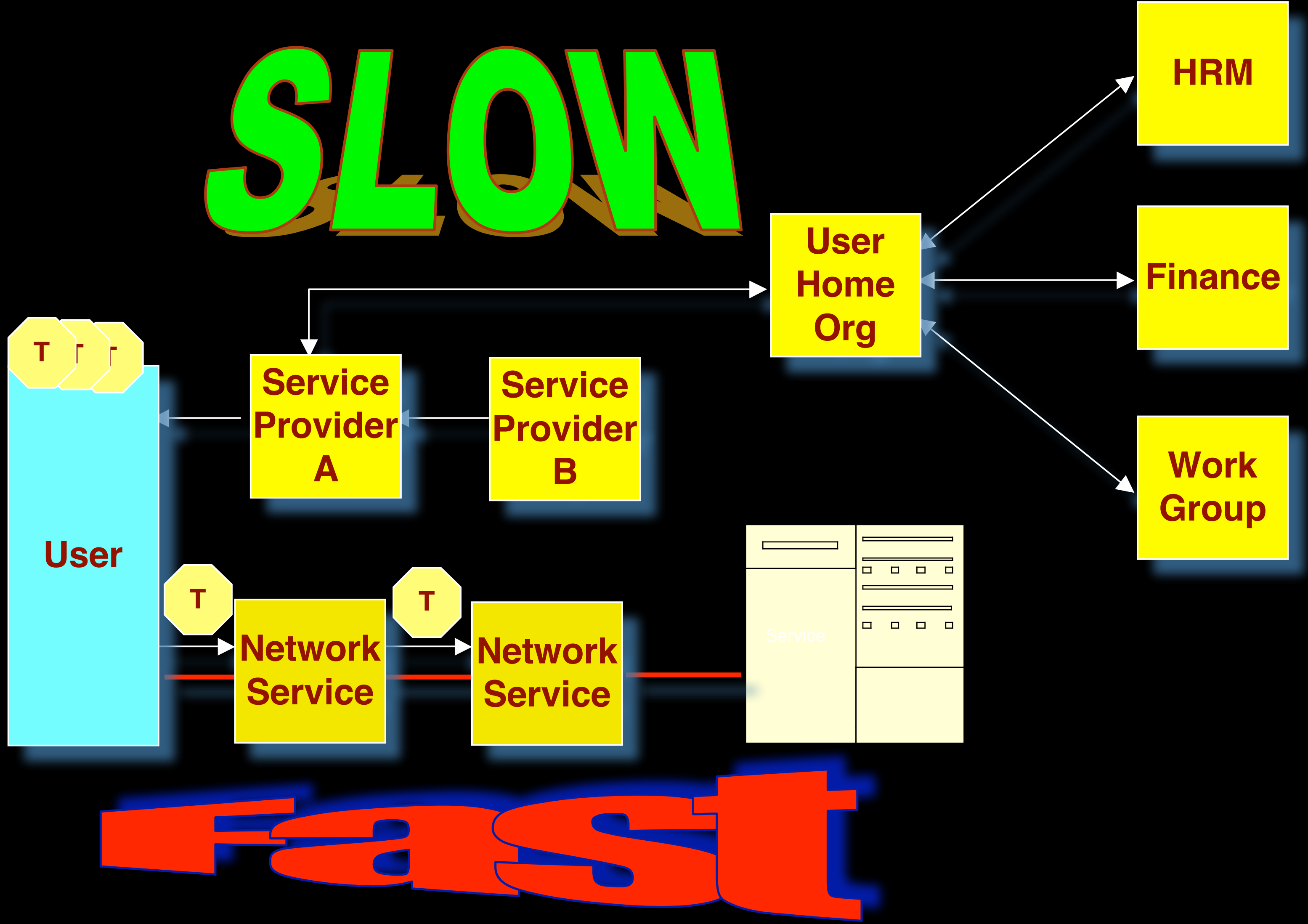
Pitlochry, Scotland - Summer 2005

Universiteit van Amsterdam

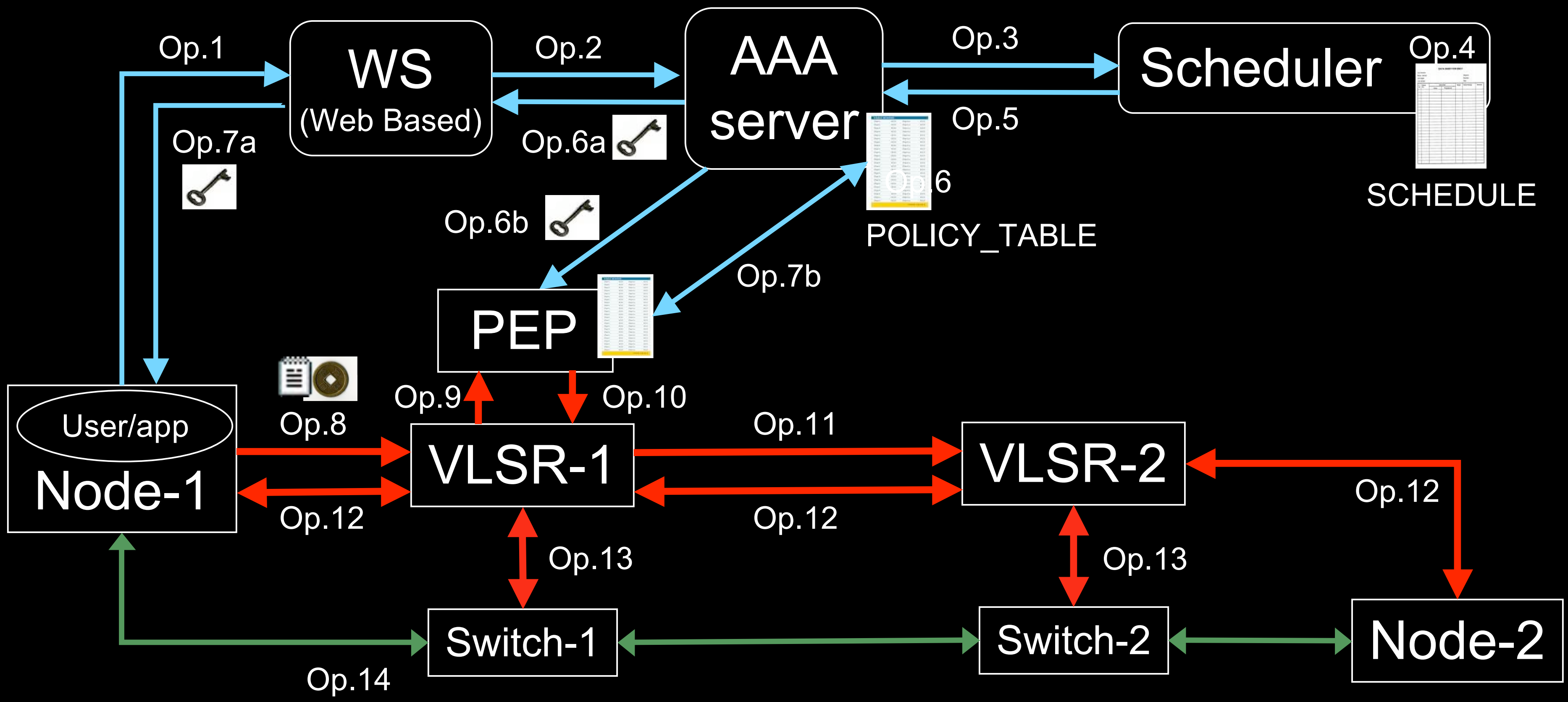
GigaPort

NextGRID

Use AAA concept to split (time consuming) service authorization process from service access using secure tokens in order to allow fast service access.



DRAGON GMPLS & TBN Demo, SC06 Tampa



1. User (on Node1) requests a path via web to the WS.
2. WS sends the XML requests to the AAA server.
3. AAA server calculates a hashed index number and submits a request to the Scheduler.
4. Scheduler checks the SCHEDULE and add new entry.
5. Scheduler confirms the reservation to the AAA.
6. AAA server updates the POLICY_TABLE.

6a. AAA server issues an encrypted key to the WS.

6b. AAA server passes the same key to the PEP.

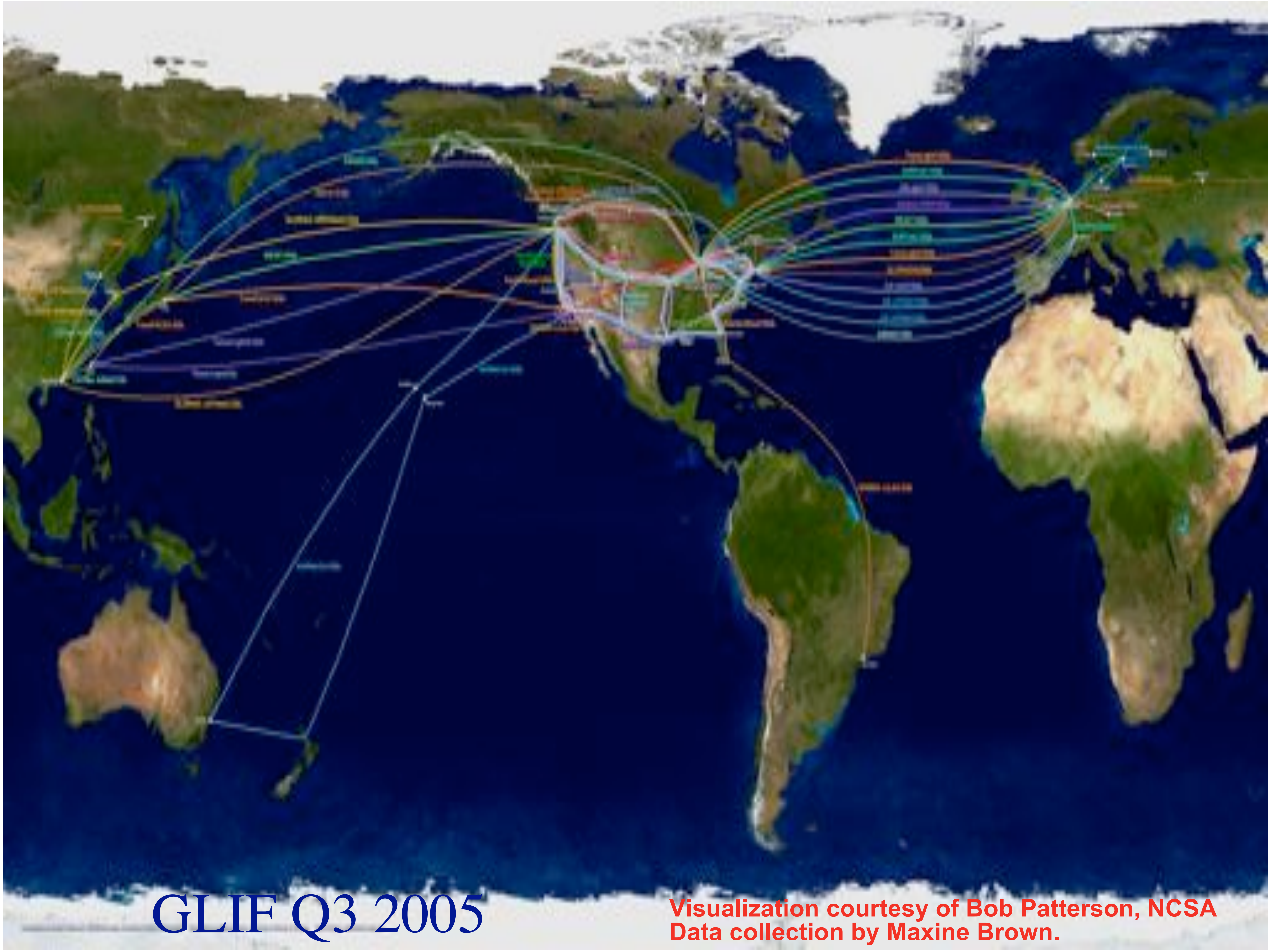
7a. WS passes the key to the user.

7b. AAA server interacts with PEP to update the local POLICY_TABLE on the PEP.

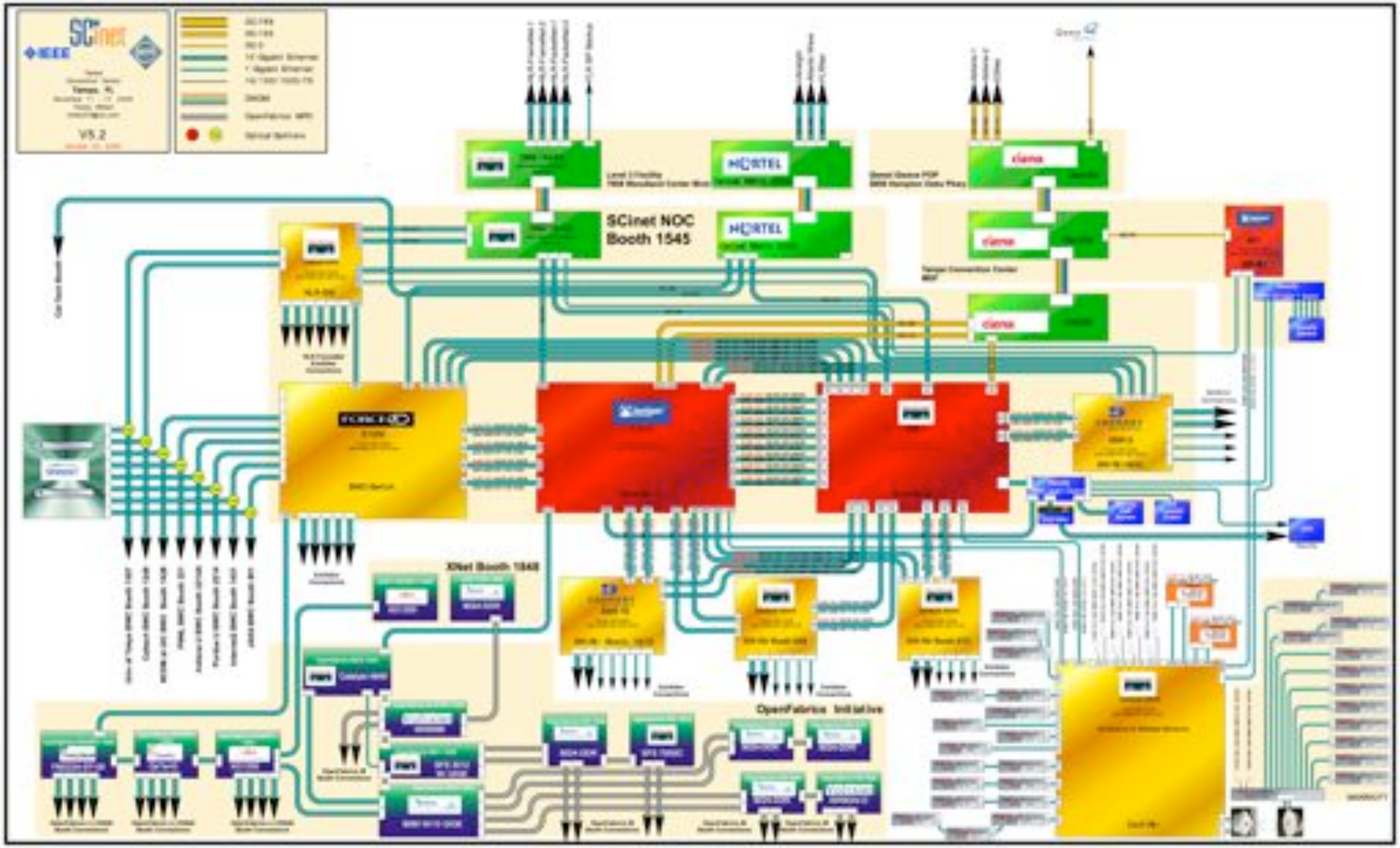
8. User constructs the RSVP message with extra Token data by using the key and sends to VLSR-1.
9. VLSR-1 queries PEP whether the Token in the RSVP message is valid.
10. PEP checks in the local POLICY_TABLE and return YES.
11. When VLSR-1 receives YES from PEP, it forwards the RSVP message.
12. All nodes process RSVP message(forwarding/response)
13. The Ethernet switches are configured
14. LSP is set up and traffic can flow

Contents

1. The need for hybrid networking
2. StarPlane; a grid controlled photonic network
3. Cross Domain Authorization using Tokens
4. RDF/Network Description Language
5. Tera-networking
6. Programmable networks

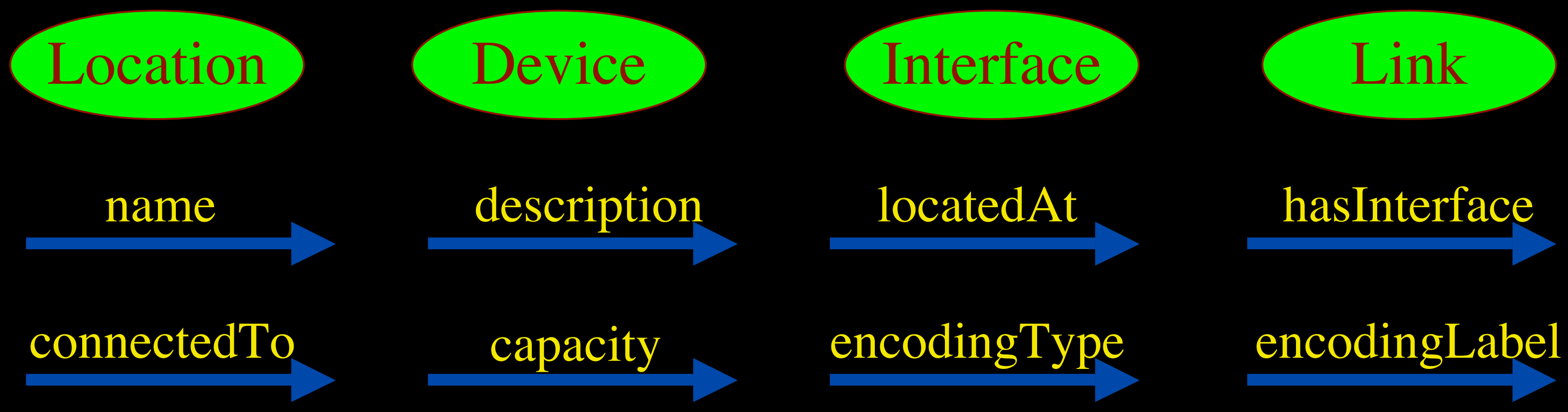

GLIF Q3 2005
Visualization courtesy of Bob Patterson, NCSA
Data collection by Maxine Brown.

Architecture SC06

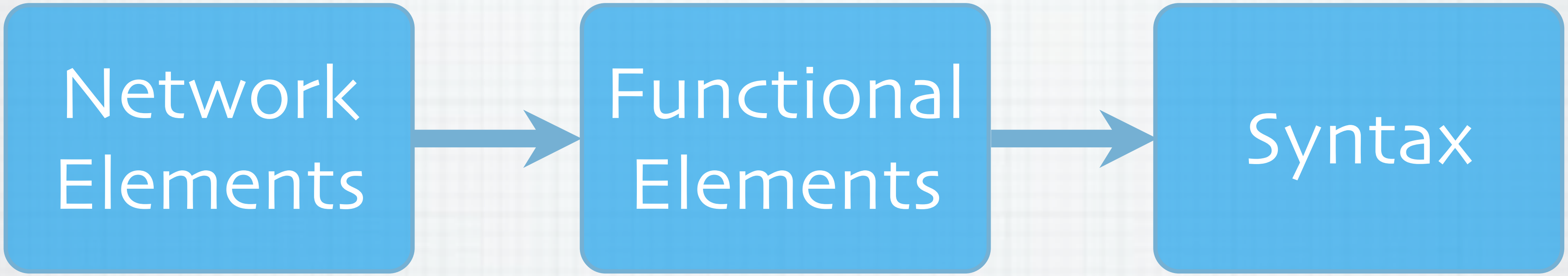


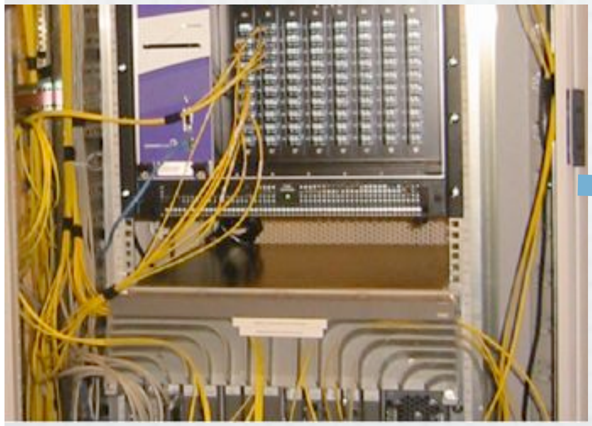
Network Description Language

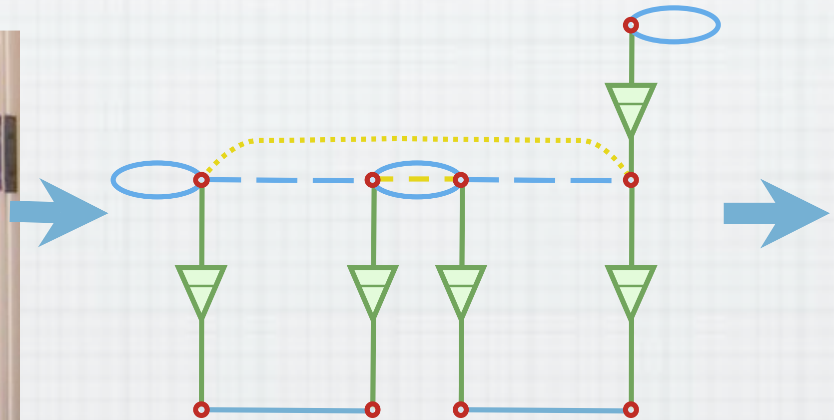
- From semantic Web / Resource Description Framework.
- The RDF uses XML as an interchange syntax.
- Data is described by triplets:

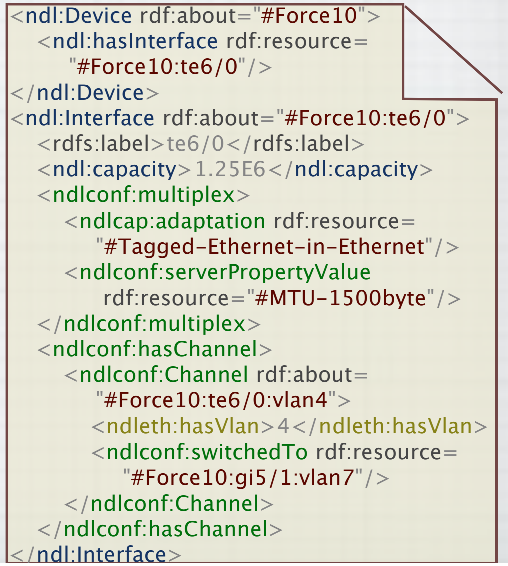


The Modelling Process









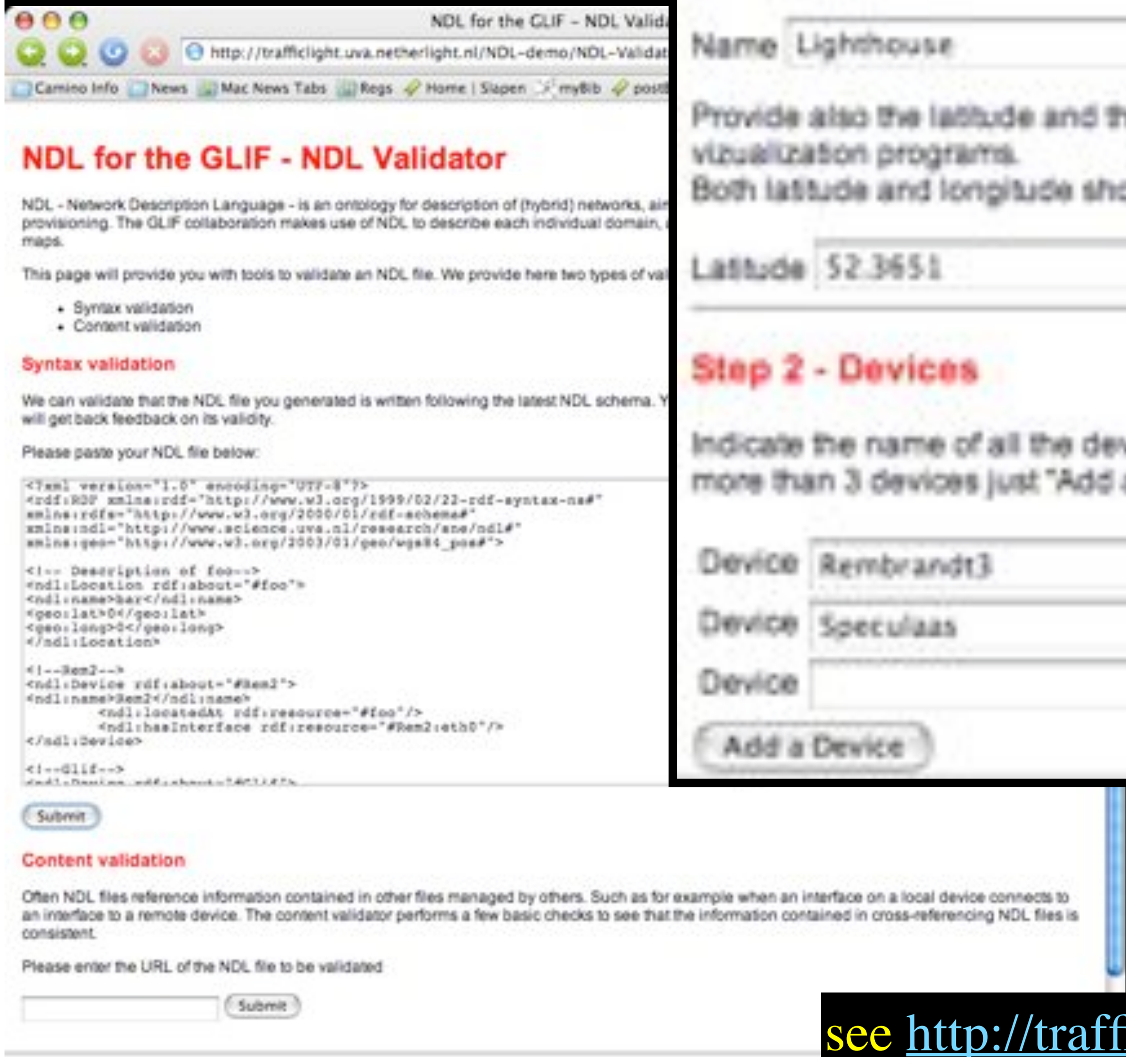
```
<ndl:Device rdf:about="#Force10">
   <ndl:hasInterface rdf:resource=
       "#Force10:te6/0"/>
</ndl:Device>
<ndl:Interface rdf:about="#Force10:te6/0">
   <rdfs:label>te6/0</rdfs:label>
   <ndl:capacity>1.25E6</ndl:capacity>
   <ndlconf:multiplex>
      <ndlcap:adaptation rdf:resource=
         "#Tagged-Ethernet-in-Ethernet"/>
      <ndlconf:serverPropertyValue
         rdf:resource="#MTU-1500byte"/>
   </ndlconf:multiplex>
   <ndlconf:hasChannel>
      <ndlconf:Channel rdf:about=
         "#Force10:te6/0:vlan4">
         <ndleth:hasVlan>4</ndleth:hasVlan>
         <ndlconf:switchedTo rdf:resource=
            "#Force10:gi5/1:vlan7"/>
      </ndlconf:Channel>
   </ndlconf:hasChannel>
</ndl:Interface>
```

NetherLight in RDF

```
<?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/air/ndl#">
<!-- Description of Netherlight -->
<ndl:Location rdf:about="#Netherlight">
      <ndl:name>Netherlight Optical Exchange</ndl:name>
</ndl:Location>
<!-- 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:501/1"/>
      <ndl:hasInterface rdf:resource="#tdm3.amsterdam1.netherlight.net:501/3"/>
      <ndl:hasInterface rdf:resource="#tdm3.amsterdam1.netherlight.net:501/4"/>
      <ndl:hasInterface rdf:resource="#tdm3.amsterdam1.netherlight.net:503/1"/>
      <ndl:hasInterface rdf:resour
      <ndl:hasInterface rdf:resour
      <ndl:hasInterface rdf:resour
      <ndl:hasInterface rdf:resour
      <ndl:hasInterface rdf:resour
      <ndl:hasInterface rdf:resour
      <ndl:hasInterface rdf:resour
      <ndl:hasInterface rdf:resour
```

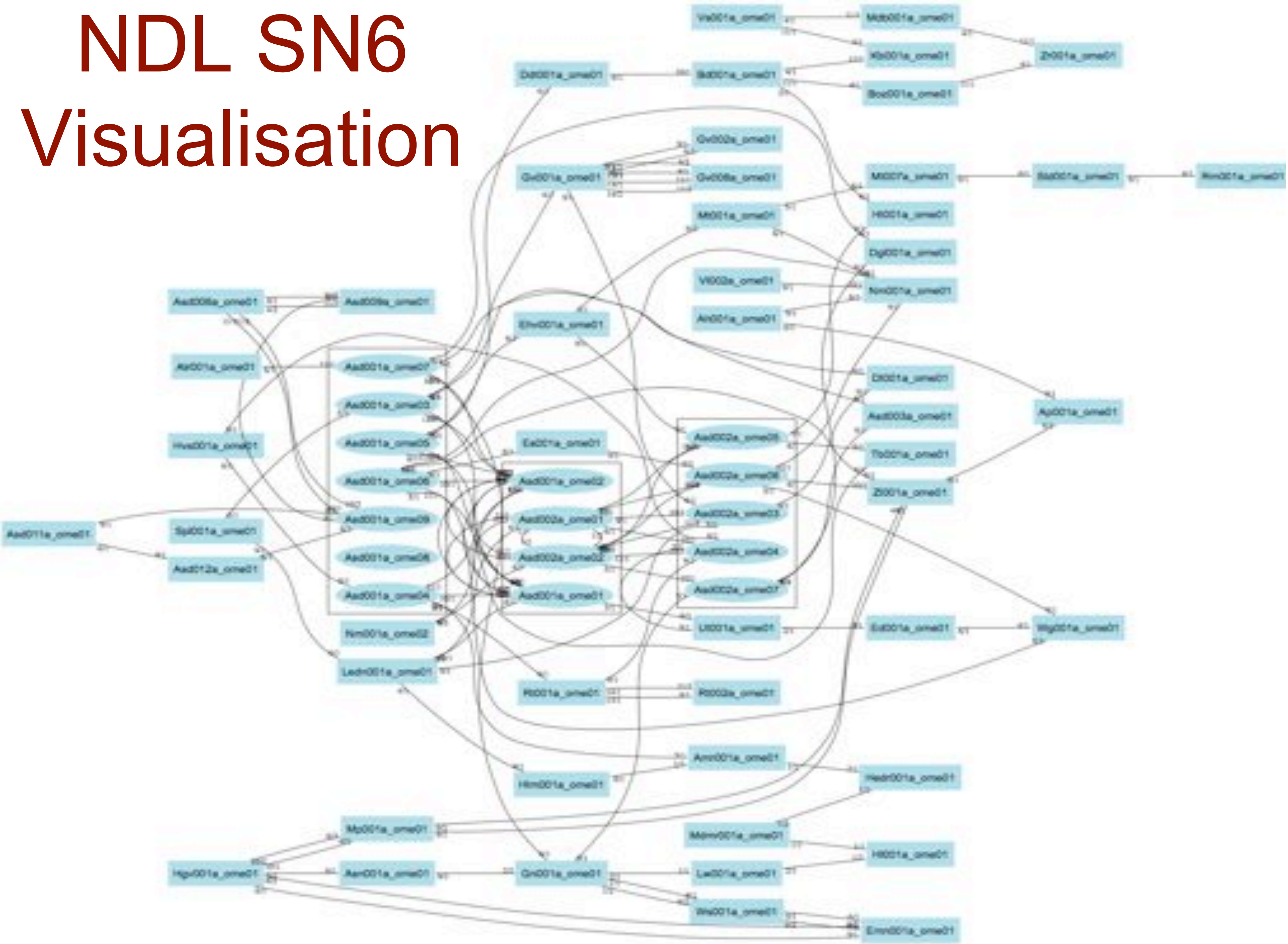
```
<!-- all the interfaces of TDM3.amsterdam1.netherlight.net -->

<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"/>
 </ndl:Interface>
<ndl:Interface rdf:about="#tdm3.amsterdam1.netherlight.net:501/2">
            <ndl:name>tdm3.amsterdam1.netherlight.net:POS501/2</ndl:name>
            <ndl:connectedTo rdf:resource="#tdm1.amsterdam1.netherlight.net:12/1"/>
 </ndl:Interface>
```

NDL Generator and Validator

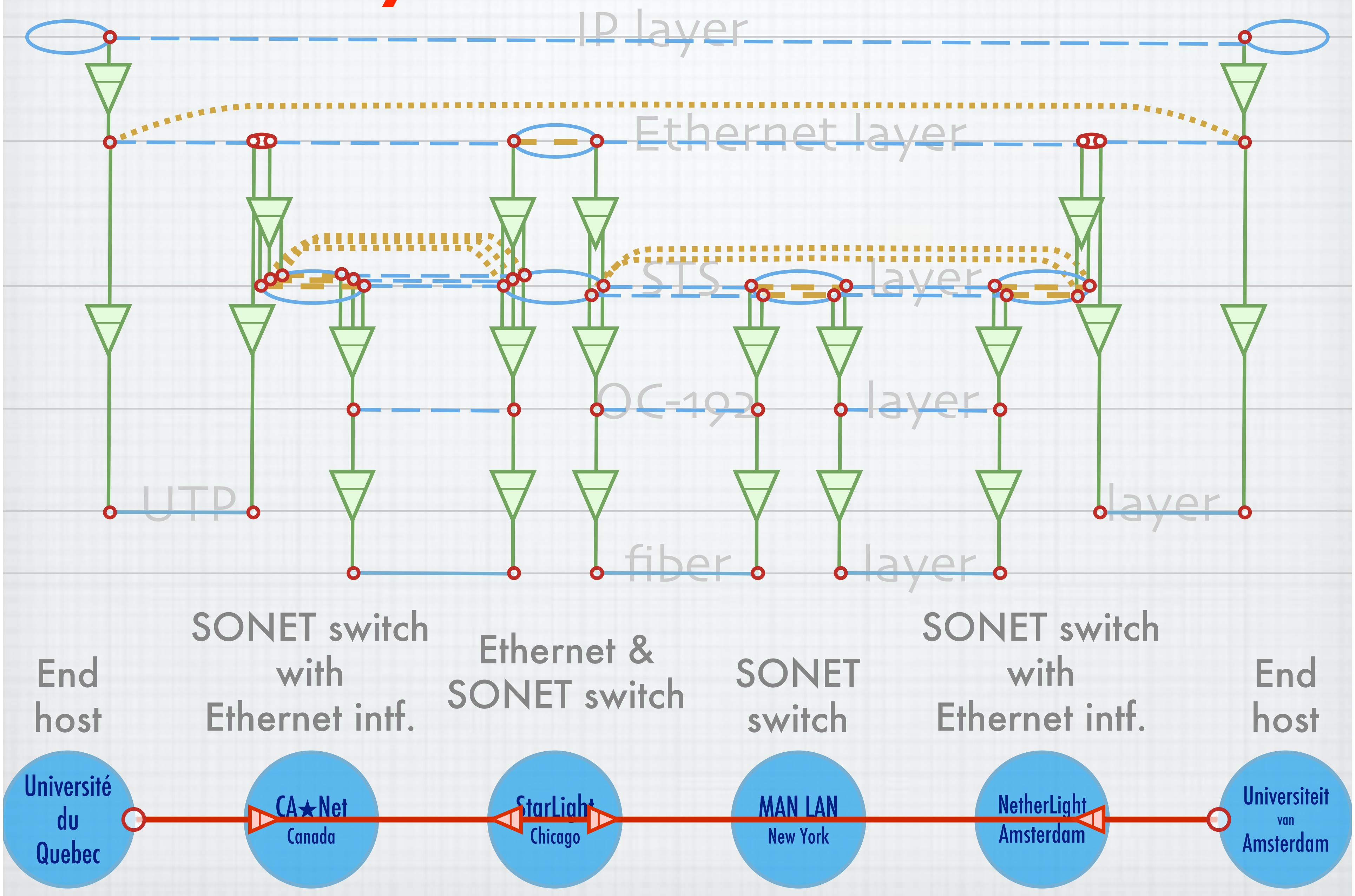


see http://trafficlight.uva.netherlight.nl/NDL-demo/

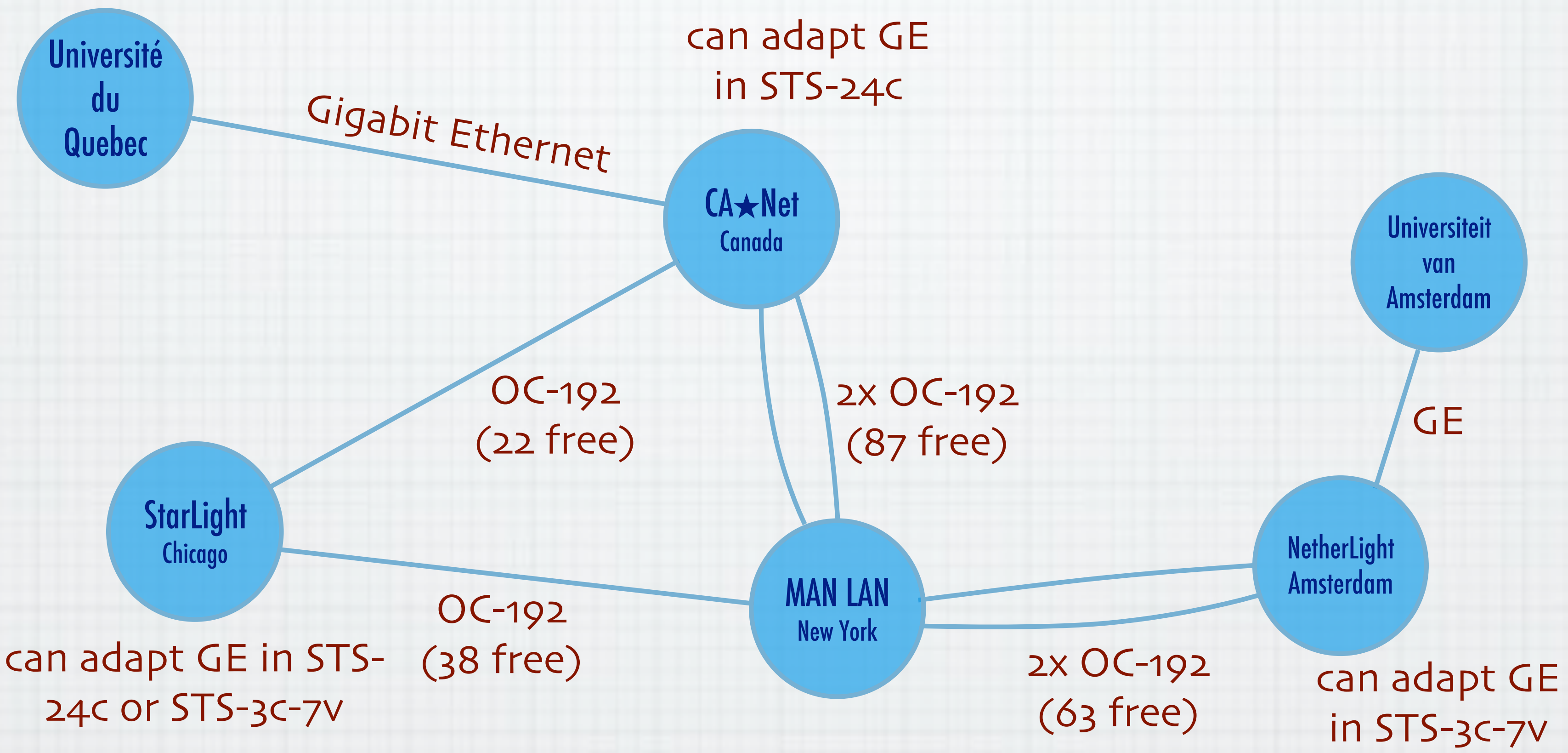
NDL SN6 Visualisation



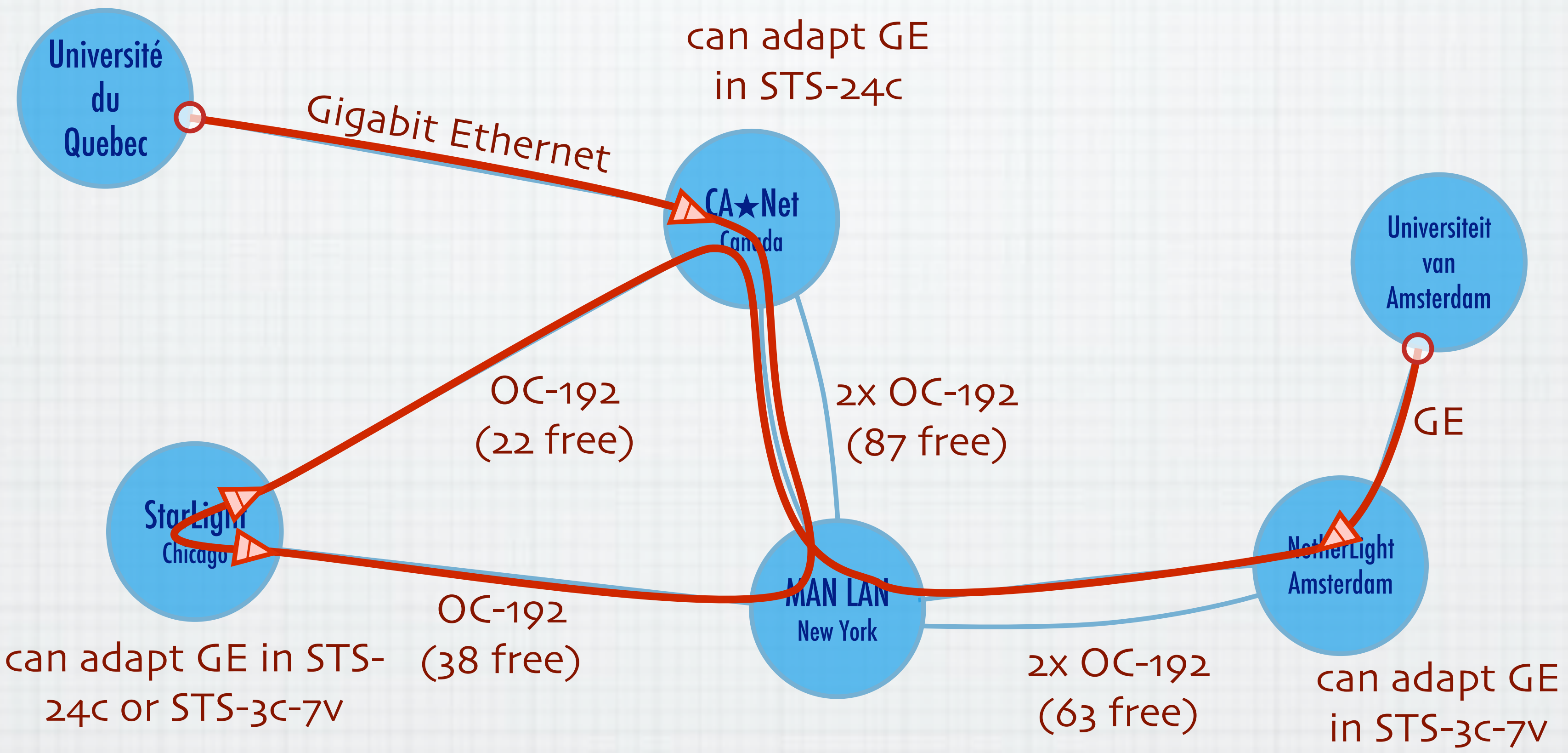
Multi-layer extensions to NDL



A weird example



The result :-)



Thanks to Freek Dijkstra & team

OGF NML-WG

Open Grid Forum - Network Markup Language workgroup

Chairs:

Paola Grosso – Universiteit van Amsterdam

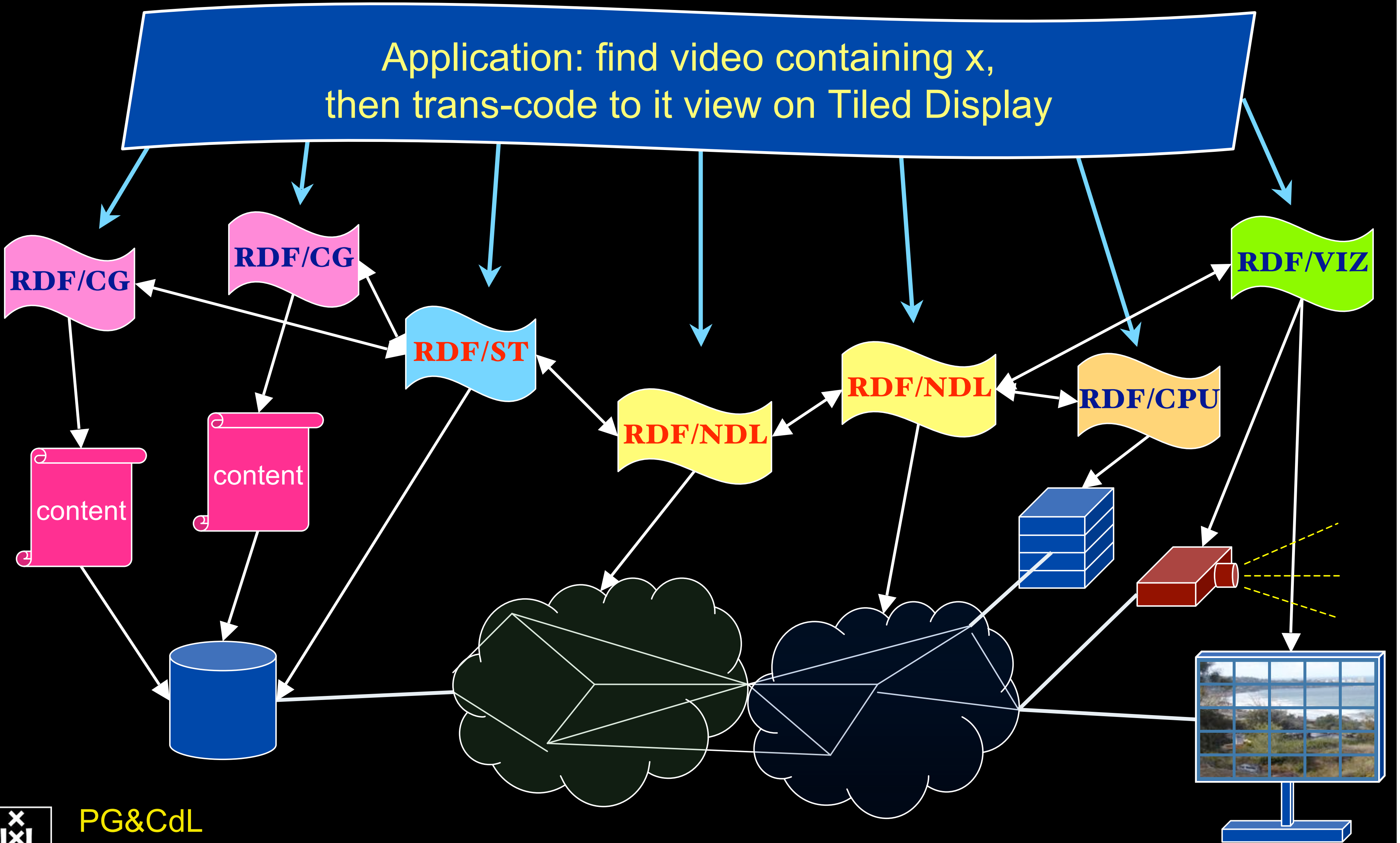
Martin Swany – University of Delaware

Purpose:

To describe network topologies, so that the outcome is a standardized network description ontology and schema, facilitating interoperability between different projects.

https://forge.gridforum.org/sf/projects/nml-wg

RDF describing Infrastructure



PG&CdL

Contents

1. The need for hybrid networking
2. StarPlane; a grid controlled photonic network
3. Cross Domain Authorization using Tokens
4. RDF/Network Description Language
5. Tera-networking
6. Programmable networks

TeraThinking

- What constitutes a Tb/s network?
- CALIT2 has 8000 Gigabit drops ?->? Terabit Lan?
- look at 80 core Intel processor
  - cut it in two, left and right communicate 8 TB/s
- think back to teraflop computing!
  - MPI makes it a teraflop machine
- massive parallel channels in hosts, NIC's
- TeraApps programming model supported by
  - TFlops -> MPI / Globus
  - TBytes -> OGSA/DAIS
  - TPixels -> SAGE
  - TSensors -> LOFAR, LHC, LOOKING, CineGrid, ...
  - Tbit/s -> ?

ref Larry Smarr & CdL

Need for discrete parallelism

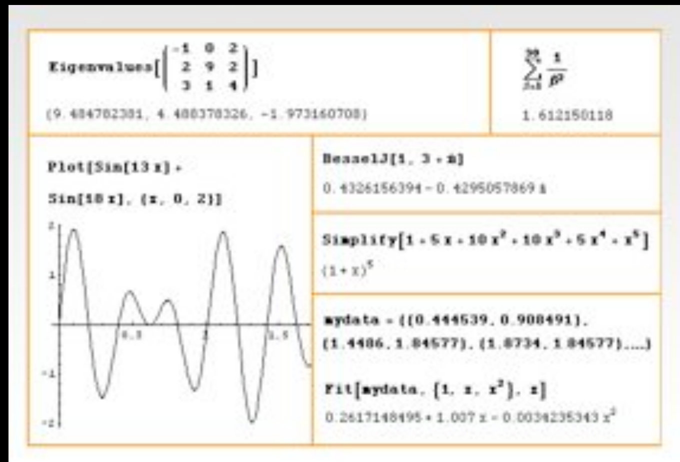
- it takes a core to receive 1 or 10 Gbit/s in a computer
- it takes one or two cores to deal with 10 Gbit/s storage
- same for Gigapixels
- same for 100's of Gflops
- Capacity of every part in a system seems of same scale
- look at 80 core Intel processor
  - cut it in two, left and right communicate 8 TB/s
- massive parallel channels in hosts, NIC's
- Therefore we need to go massively parallel allocating complete parts for the problem at hand!

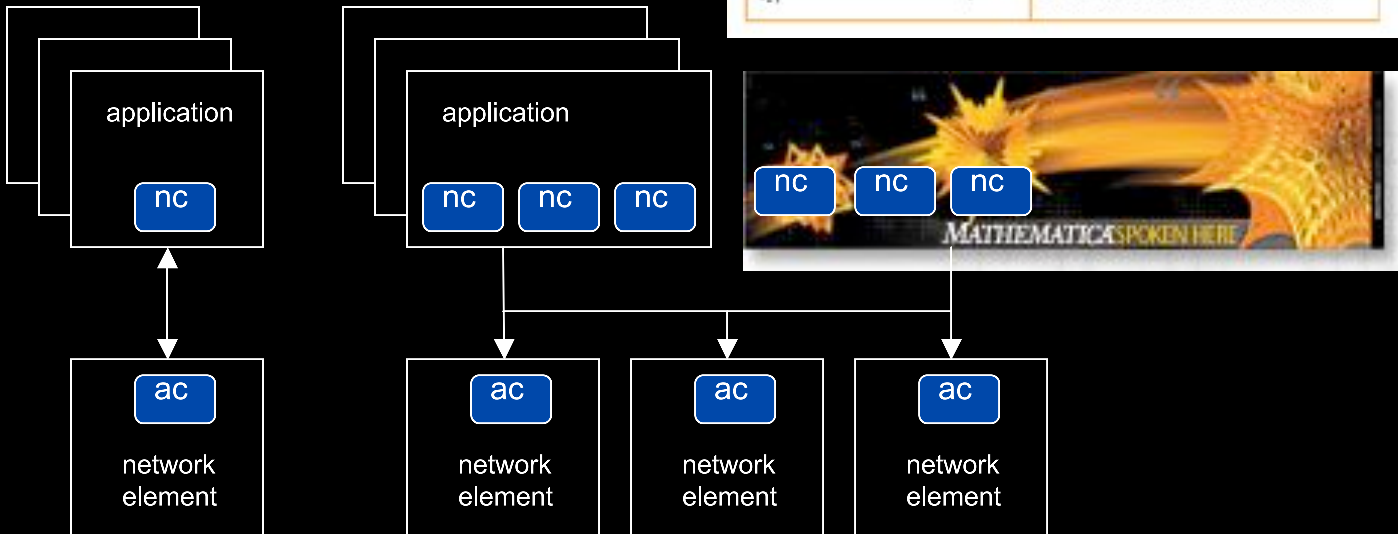
Contents

1. The need for hybrid networking
2. StarPlane; a grid controlled photonic network
3. Cross Domain Authorization using Tokens
4. RDF/Network Description Language
5. Tera-networking
6. Programmable networks

User Programmable Virtualized Networks allows the results of decades of computer science to handle the complexities of application specific networking.

- The network is virtualized as a collection of resources
- UPVNs enable network resources to be programmed as part of the application
- Mathematica, a powerful mathematical software system, can interact with real networks using UPVNs





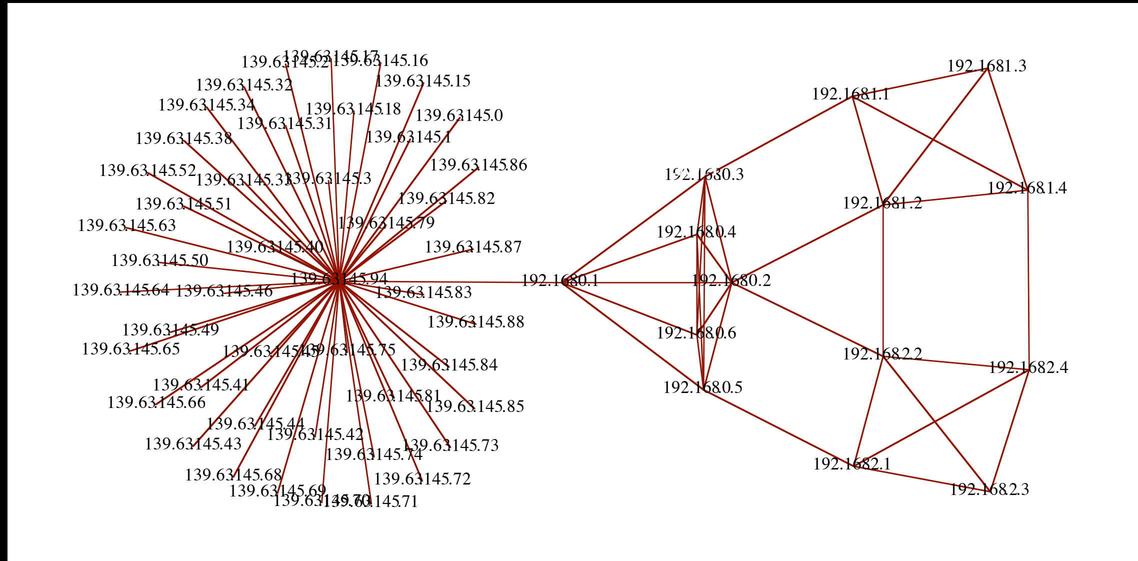
StarPlane

Mathematica enables advanced graph queries, visualizations and real-time network manipulations on UPVNs

Topology matters can be dealt with algorithmically
Results can be persisted using a transaction service built in UPVN

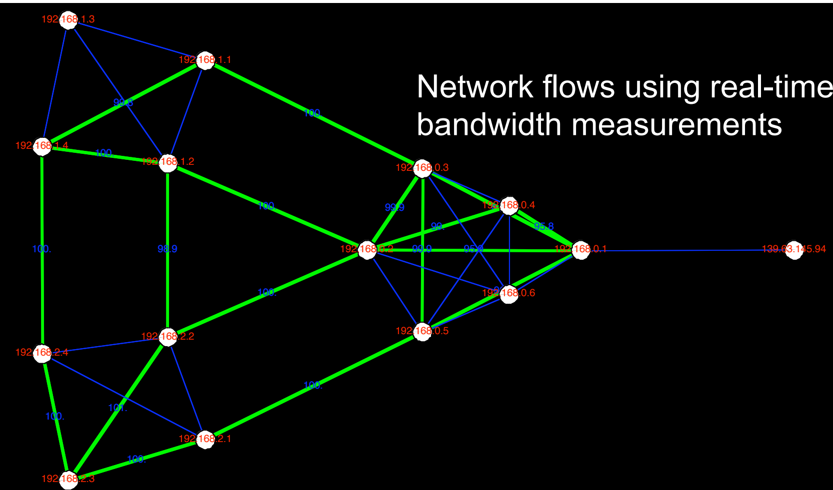
Initialization and BFS discovery of NEs

```
Needs["WebServices`"]
<<DiscreteMath`Combinatorica`
<<DiscreteMath`GraphPlot`
InitNetworkTopologyService["edge.ict.tno.nl"]

Available methods:
{DiscoverNetworkElements,GetLinkBandwidth,GetAllIpLinks,Remote,
NetworkTokenTransaction}

Global`upvnverbose = True;
AbsoluteTiming[nes = BFSDiscover["139.63.145.94"];][[1]]
AbsoluteTiming[result = BFSDiscoverLinks["139.63.145.94", nes];][[1]]

Getting neigbours of: 139.63.145.94
Internal links: {192.168.0.1, 139.63.145.94}
(...)
Getting neigbours of:192.168.2.3
Internal links: {192.168.2.3}
```



Transaction on shortest path with tokens

```
nodePath = ConvertIndicesToNodes[
                ShortestPath[ g,
                        Node2Index[nids,"192.168.3.4"],
                        Node2Index[nids,"139.63.77.49"]],
                        nids];
Print["Path: ", nodePath];
If[NetworkTokenTransaction[nodePath, "green"]==True,
        Print["Committed"], Print["Transaction failed"]];

Path:
{192.168.3.4,192.168.3.1,139.63.77.30,139.63.77.49}

Committed
```



Network flows using real-time bandwidth measurements

ref: Robert J. Meijer, Rudolf J. Strijkers, Leon Gommans, Cees de Laat, User Programmable Virtualiized Networks, accepted for publication to the IEEE e-Science 2006 conference Amsterdam.

Questions ?