StarPlane take-off
Cees de Laat
SURFnet
BSIK
EU
University of Amsterdam
DAS = Distributed ASCI Supercomputer

- Project DAS-1 started in 1997 by Andrew Tanenbaum
- To prove distributed clusters were as effective as super...
- 4-5 clusters connected via high speed links
  - DAS-1 -> 6 Mbit/s full mesh ATM
  - DAS-2 -> Gbit/s L3
  - DAS-3 -> StarPlane
- DAS-1 ran BSD, changed to Linux (Andrew... :-)  
- DAS-1 and 2 uniform architecture, not so in DAS-3 
- http://www.cs.vu.nl/das/
DAS 1 - 2 Cluster

- Head node
- Ethernet
- Local interconnect
- Fast interconnect
- 32 compute nodes

To local University and wide area interconnect
History - 2

SURFnet6 Architecture discussions 2001-2002

- photonic backbone
- (L2 and) L3 services
- NORTEL
- Static
- Summer 2004 K&C
- NWO-GLANCE
- StarPlane
- PHD-PD-SP
- Start 1-feb-06, Li Xu, Jan Philip Velders, Jason Maasen

ref: cdl-2002-01-18-UCL-opt.ppt
Common Photonic Layer (CPL) in SURFnet6

Subnetwork 1: Green

Subnetwork 2: Dark blue

Subnetwork 3: Red

Subnetwork 4: Blue Azur

Subnetwork 5: Grey

>6000 km
StarPlane
DWDM
backplane
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
DAS-3 Cluster

head node

To local University

To SURFnet

1 Gbit/s Ethernet

Local interconnect

32 compute nodes

Fast interconnect

10 Gbit/s Ethernet lanphy

To SURFnet

University

DAS-3 Cluster

32 compute nodes
StarPlane Goals

1. fast, application-driven allocation of the photonic network resources
2. application-specific composition of the protocol stack that controls the resources
3. low-level resource partitioning (and, hence, no interference)
4. high-level requests (whereby policies and inference are used to assist the user).

To achieve and validate these goals the project will deliver:

1. the implementation of the StarPlane management infrastructure
2. the implementation of an intelligent broker service to handle high-level requests
3. the modification of a set of real applications to exploit the functionality of such a management plane
4. a library of standard components (protocols, middleware) to support and build new applications
GRID-Colocation problem space

Extensively under research

New!
The StarPlane project addresses two concerns in optical networks:

1. The Basic StarPlane Management Infrastructure
   StarPlane allows applications to take advantage of the increased bandwidth and potential flexibility in optical networks by letting them create their own network topology in a simple way.

2. The Applications and Their Needs
   StarPlane will discover how this new freedom to manipulate the network will benefit the applications.

Staff members of the research team:
- Prof. dr. H.E. Bal
- Dr. ir. H. Bos
- Dr. ir. C.T.A.M. de Laat
- Prof. dr. P.M.A. Sloot
- VU: professor
- VU: assistant professor
- VU: associate professor
- UvA: professor
- UvA: associate professor

StarPlane will use the physical infrastructure provided by SURFnet 6 and the distributed supercomputer DAS-3. Hybrid optical networks such as SURFnet 6 allow network administrators to partition the network and to create multiple overlay networks, each with a different logical topology. The novelty of StarPlane is that it does give this flexibility directly to the applications by allowing them to choose the logical topology in real time, ultimately with subsecond switching times.

NOC: network operations center
Control Plane
NLight
DAS 3
TU Delft
Universiteit van Amsterdam (VL-e & Multimedia)
Key issue #1: how to describe such networks?
UvA/SARA LightHouse
A joint network research lab of the University of Amsterdam and SARA. Connects end resources to NetherLight. Proof of concept e.g. tier 0/1, webservices, GSP
"a universal medium for the exchange of data where data can be shared and processed by automated tools as well as by people’’

The Resource Description Framework (RDF) uses XML as an interchange syntax.

Data is described by triplets:

Subject → Predicate → Object
NDL - Network Description Language

A way to describe network resources using RDF. Parser can use the data to:
- generate network maps
- provide information to schedulers

```xml
<ndl:Device rdf:about="#Vangogh3">
  <ndl:name>Vangogh3</ndl:name>
  <rb:isOfType>ComputingElement</rb:isOfType>
  <ndl:locatedAt rdf:resource="#Lighthouse"/>
  <ndl:hasInterface rdf:resource="#Vangogh3:eth2"/>
</ndl:Device>
```
<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:resource="#tdm3.amsterdam1.netherlight.net:POS501/1"/>
    <ndl:hasInterface rdf:resource="#tdm4.amsterdam1.netherlight.net:5/1"/>
    <ndl:hasInterface rdf:resource="#tdm3.amsterdam1.netherlight.net:501/2"/>
    <ndl:hasInterface rdf:resource="#tdm3.amsterdam1.netherlight.net:POS501/2"/>
    <ndl:hasInterface rdf:resource="#tdm1.amsterdam1.netherlight.net:12/1"/>
</ndl:Device>
</rdf:RDF>
RDF -> Picture
Key issue #2: How to book resources on such networks?
Web services

Web services interfaces provide the API for the reservation framework:

<wsl:operation name="getResourceInformation">
<wsl:operation name="getResourceList">
<wsl:operation name="getTypeList">
<wsl:operation name="getResourcesOfType">
<wsl:operation name="reservePath">
<wsl:operation name="getPossiblePaths">
<wsl:operation name="isPathAvailable">
<wsl:operation name="confirmPathReservation">
<wsl:operation name="cancelPathReservation">
Resource Brokering: Your Ticket Into NetherLight

Lambda networking allows the creation of application specific light paths.

Lambda networking facilities empower users to request services and provision end-to-end light paths if and when they need it.

NetherLight, located in Amsterdam, The Netherlands, is one of such facilities.

The Amsterdam Lighthouse is a joint research laboratory of the UvA and SARA.

Resources in the LightHouse can be used by collaborators to prove the concepts of hybrid networks.

Lightpath setup components:

- Topology Information
  - We make use of semantic web techniques. The description of the network is contained in RDF files.

- Reservation System
  - We provide web services interfaces to the client for:
    - reservation handling.

- Management System
  - We provision the paths on the Lighthouse equipment.

Semantic Web
The Network Description Language, an RDF Schema, describes networks in a standard, interoperable way.

Web Services
A WSDL file describes the interfaces to the service available to clients. Clients can interact with the service directly or via a portal.

Our SC19 demonstration
We show the setup of dynamic connections between two computing nodes through the Lighthouse/ NetherLight Optical Exchange.
Risks

what have we today
what to avoid
Click the Start button to begin...
Three Easy Steps:

1. Click the START button
2. Insert money...
   - $0.25 per minute...
   - Example:
     - $1 = 4 minutes
     - $5 = 20 minutes
   - No change is provided!
3. Surf the web!
OUT OF ORDER
external -> traffic engineering
What do we need

• vlan’s
• trunking
• spanning tree modified?
• source routing modified
• Policy interfaces
• AAA interaction (EduRoam, Shibboleth)
Token Based Networking
Access Control, Resource Management and Path Selection in Optical Networks using Tokens


Virtual Machine Traffic Controller

Tokens will allow:
- Separation of (slow) authorization process and real time usage.
- Binding to many different types of attributes: user, time, resource, etc.
- Policy Decision to be abstracted from Policy Enforcement Point.
- Anonymous usage
- Resource Management

OXC
Amsterdam - Netherlight

Chicago - Starlight

Tokens performing Path Selection and Access Control at Optical Inter-Connection Points

Link Owner
Link Owner PDP

Token PDP
Token PEP

UvA Token Switch

UvA RFC 2933 Generic AAA server

Rule Based Engine
Application Specific Module
Policy Repository

Token marked IP packets will allow:
- Economic Link Owners to assign usage rights without routing changes.
- Recognition at Inter-Connection Points (Optical Exchanges). When authentic and valid, token marked traffic will use the Link Owners path.
- Implementations that support different business models
- Hardware (NPU based) recognition rate expected to be a 10 Gbit/s.

IP packets containing tokens
Amsterdam - Netherlight
Chicago - Starlight

Leon Comnans

UvA - Universiteit van Amsterdam
Fred Wan, Cees de Laat
GigaPort
Mihai Cristea

Nord GPO
Questions ?
(lesson learned)