VINCI: Virtual Intelligent Networks for Computing Infrastructures

An Integrated Network Services System to Control and Optimize Workflows in Distributed Systems

CHEP February 2006

Harvey Newman
and Iosif Legrand
California Institute of Technology

OUTLINE

- Introduction
- The MonALISA framework
  - ➔ Monitoring
  - ➔ Support for distributed services and Agents
- The VINCI architecture
- Main Services
  - ➔ End System Agent (LISA)
  - ➔ Discovery & AAA
  - ➔ Control of Optical planes
  - ➔ Interfaces with GMPLS, MLPS, SNMP …
- Prediction, Learning and Self Organization

The Need for Network Services

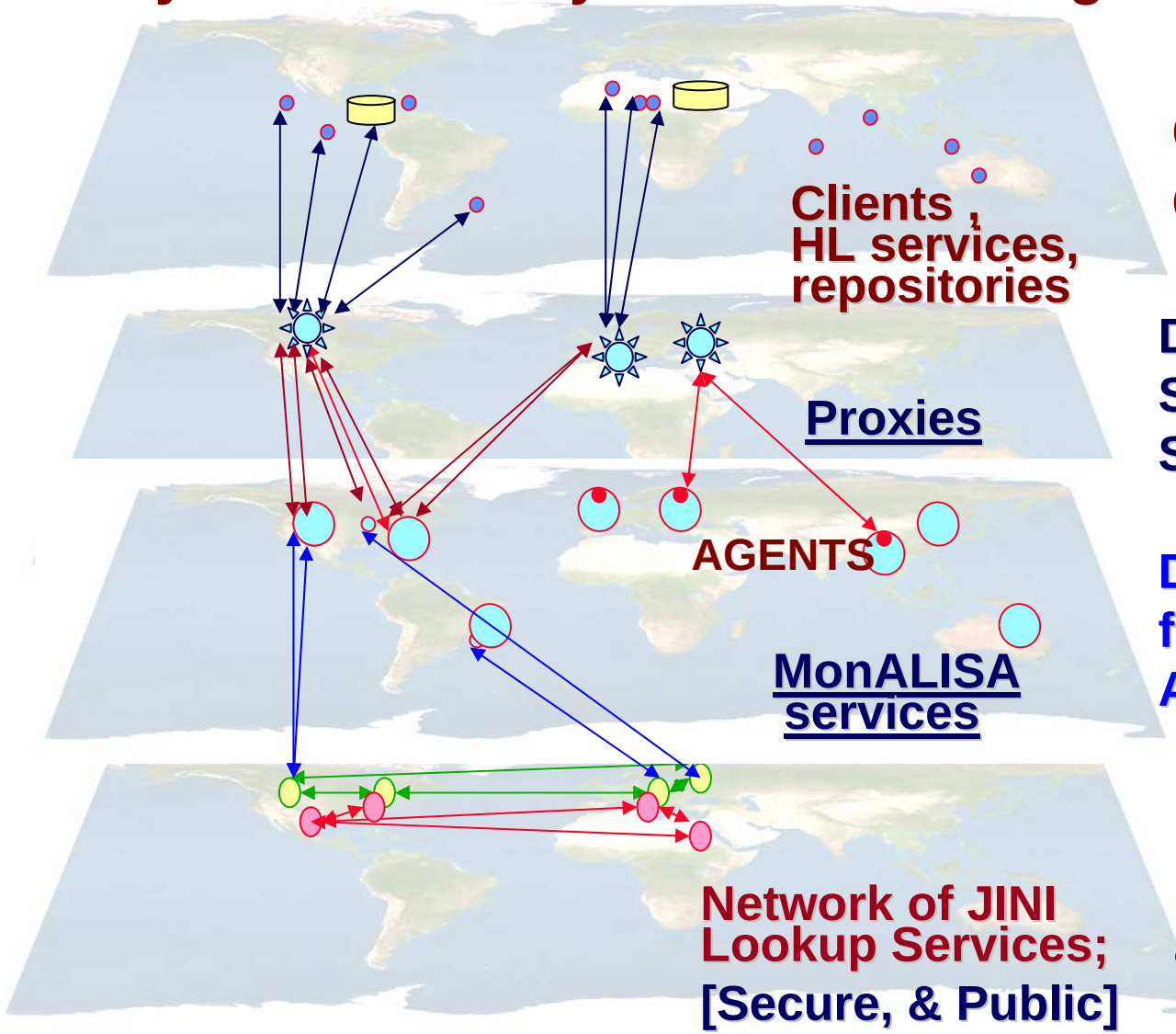
- The main objective of the VINCI project is to enable users' applications, at the LHC and in other fields of data-intensive science, to effectively use and coordinate network resources
- VINCI dynamically estimates and monitors the achievable performance along a set of candidate (shared or dedicated) network paths, and correlates these results with the CPU power and storage available at various sites, to generate optimized workflows for grid tasks
- This should significantly improve the overall performance and reduce the effective costs of global-scale grids
- The VINCI system is implemented as a dynamic set of collaborating Agents in the MonALISA framework, exploiting MonALISA's ability to access and analyze in-depth monitoring information from a large number of network links and grid sites in real-time

VINCI: A Multi-Agent System

- **VINCI and the underlying MonALISA framework use a system of autonomous agents to support a wide range of dynamic services**
- **Agents in the MonALISA servers self-organize and collaborate with each other to manage access to distributed resources, to make effective decisions in planning workflow, to respond to problems that affect multiple sites, or to carry out other globally-distributed tasks**
- **Agents running on end-users' desktops or clusters detect and adapt to their local environment so they can function properly. They locate and receive real-time information from a variety of MonALISA services, aggregate and present results to users, or feed information to higher level services**
- **Agents with built-in "intelligence" are required to engage in negotiations (for network resources, for example), and to make pro-active run-time decisions, while responding to changes in the environment**

MonALISA : An Agent-based System of Distributed Services

Fully Distributed System with no Single Point of Failure



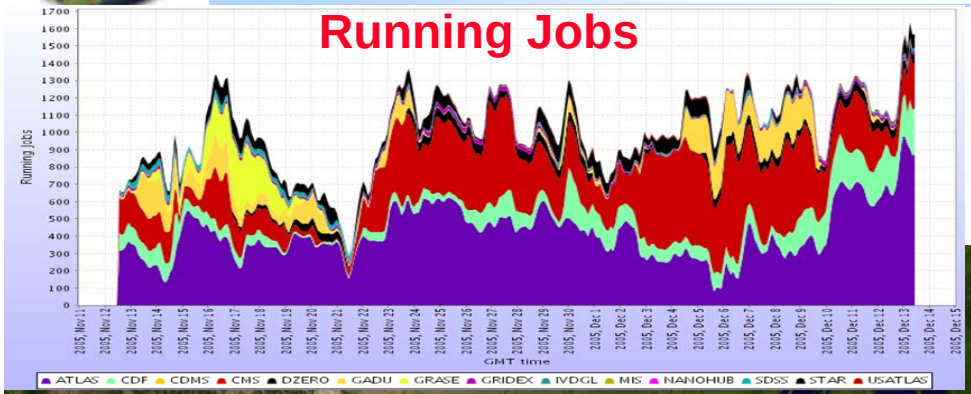
Global Services or Clients

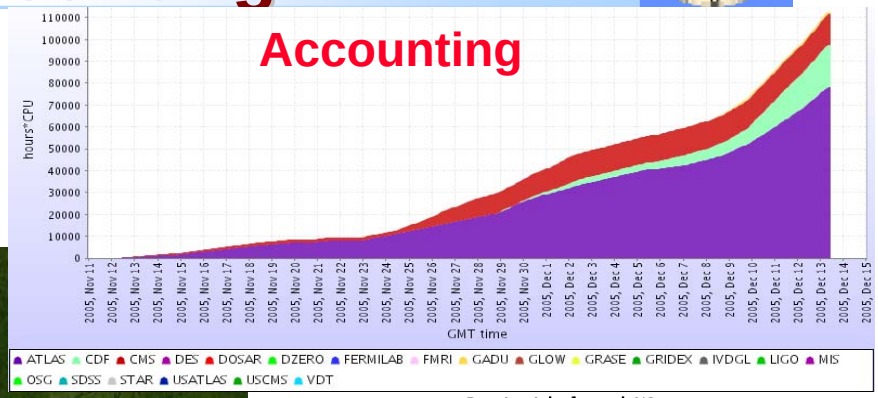
Dynamic load balancing
Scalability & Replication
Security AAA for Clients

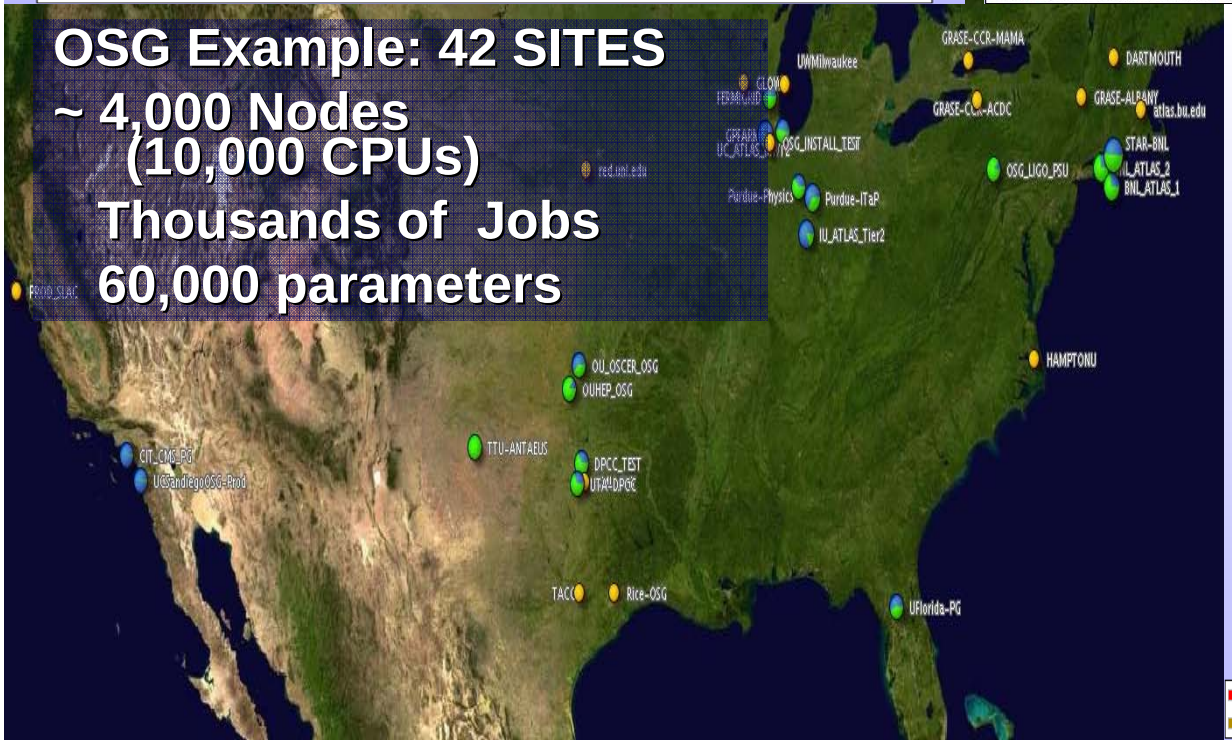
Distributed System for gathering and Analyzing Information.

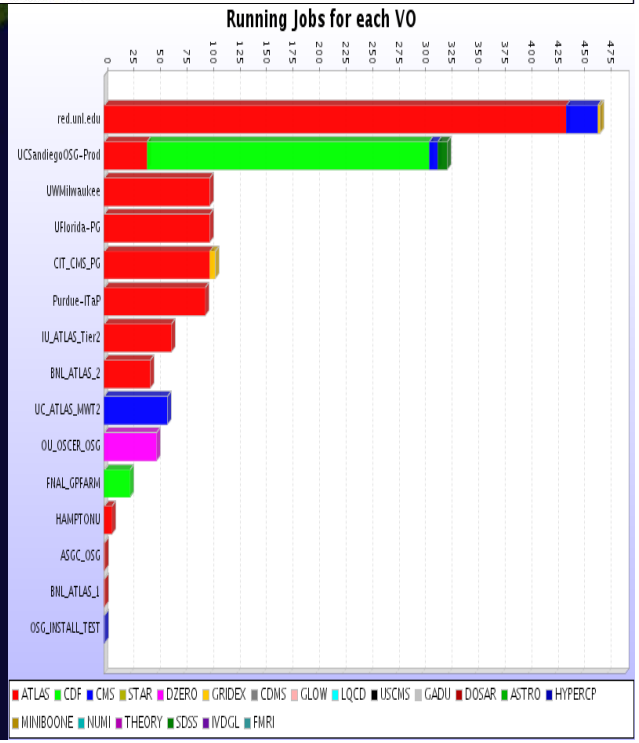
Distributed Dynamic Discovery- based on a lease Mechanism

Monitoring OSG: Resources, Jobs & Accounting

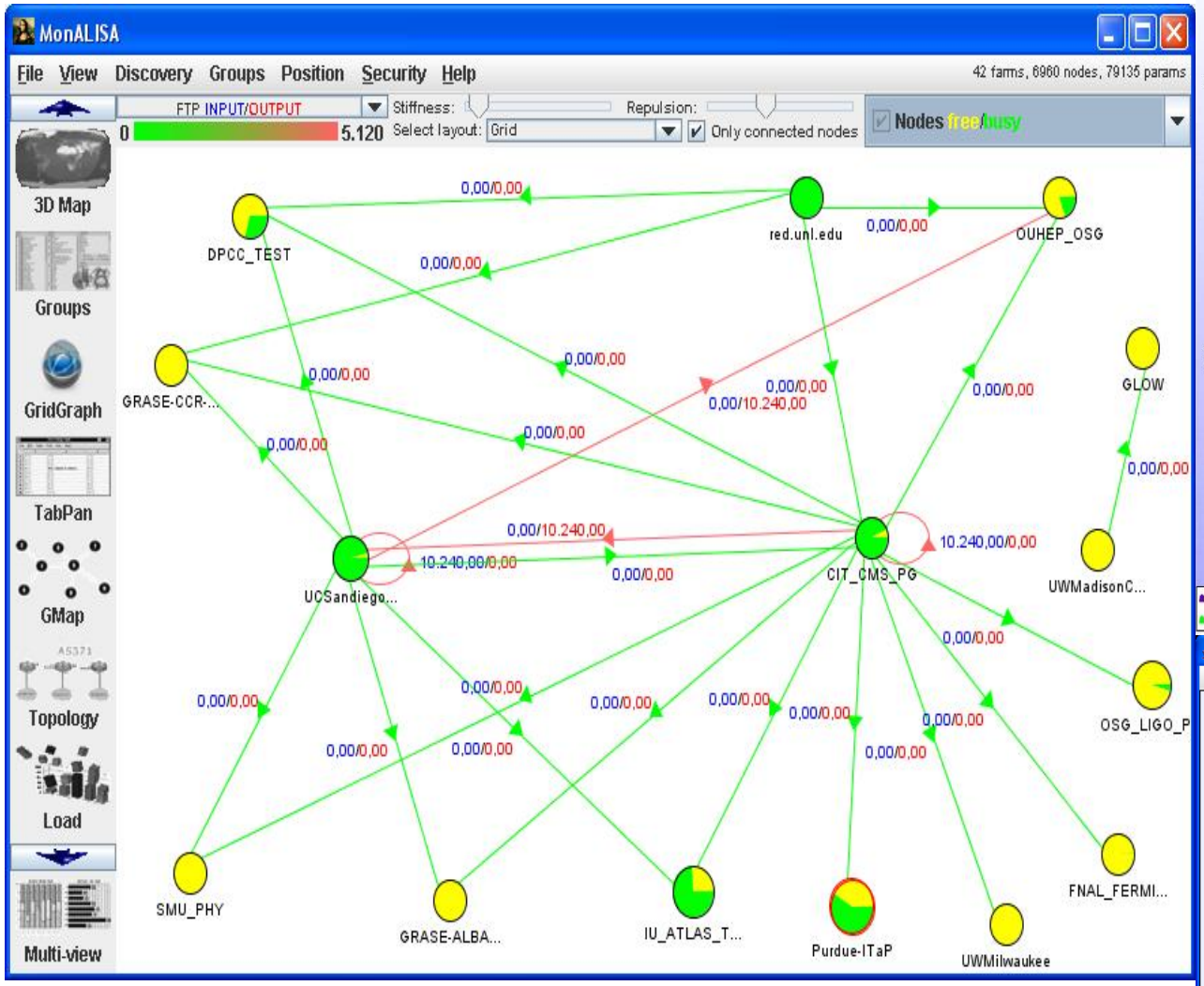


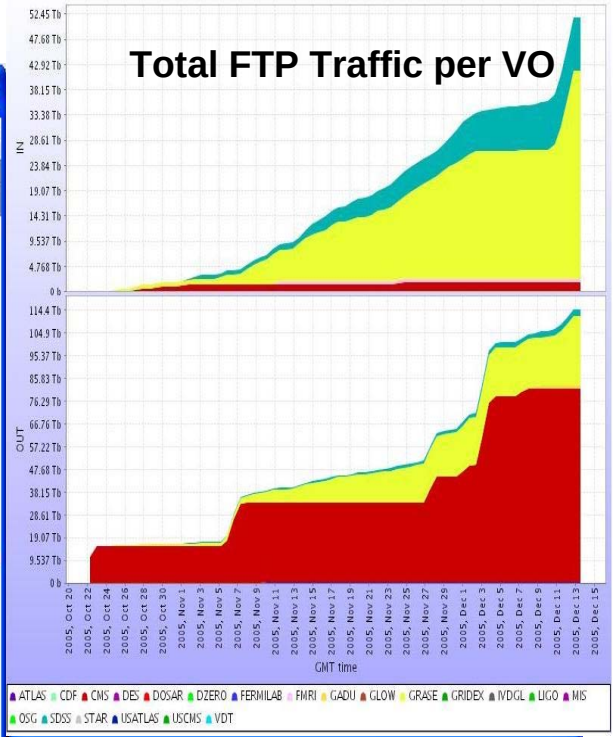


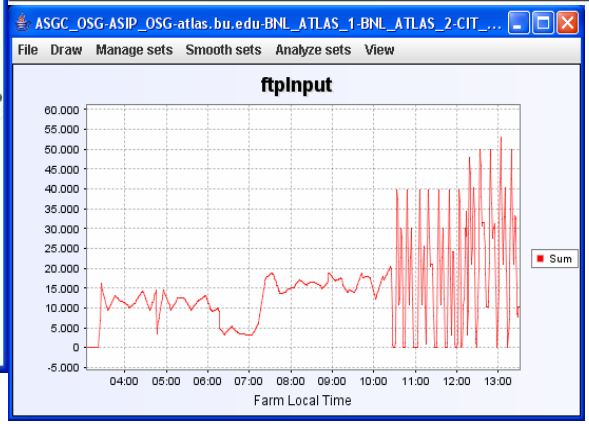




FTP Data Transfers Among Grid Sites

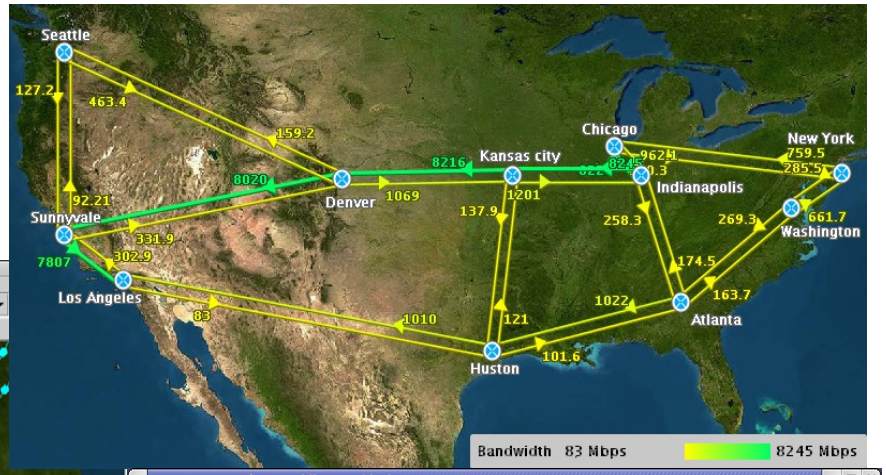


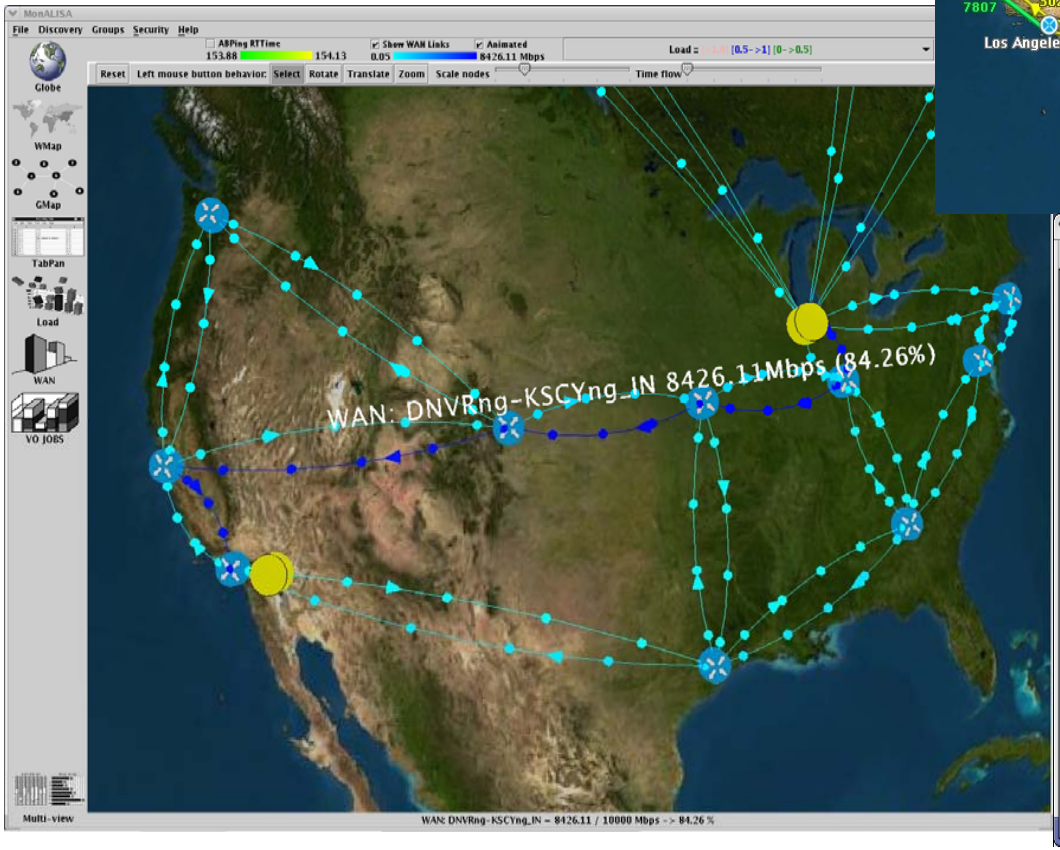


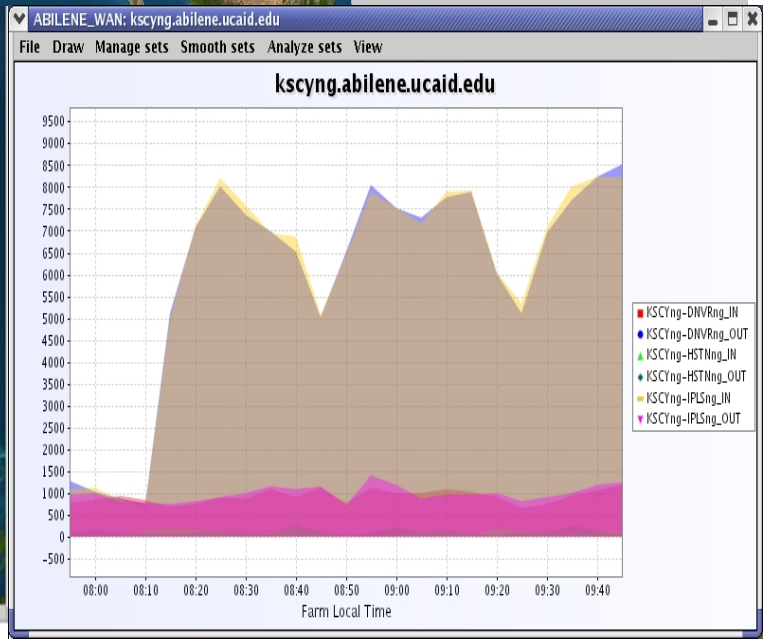


Monitoring the Abilene Backbone Network

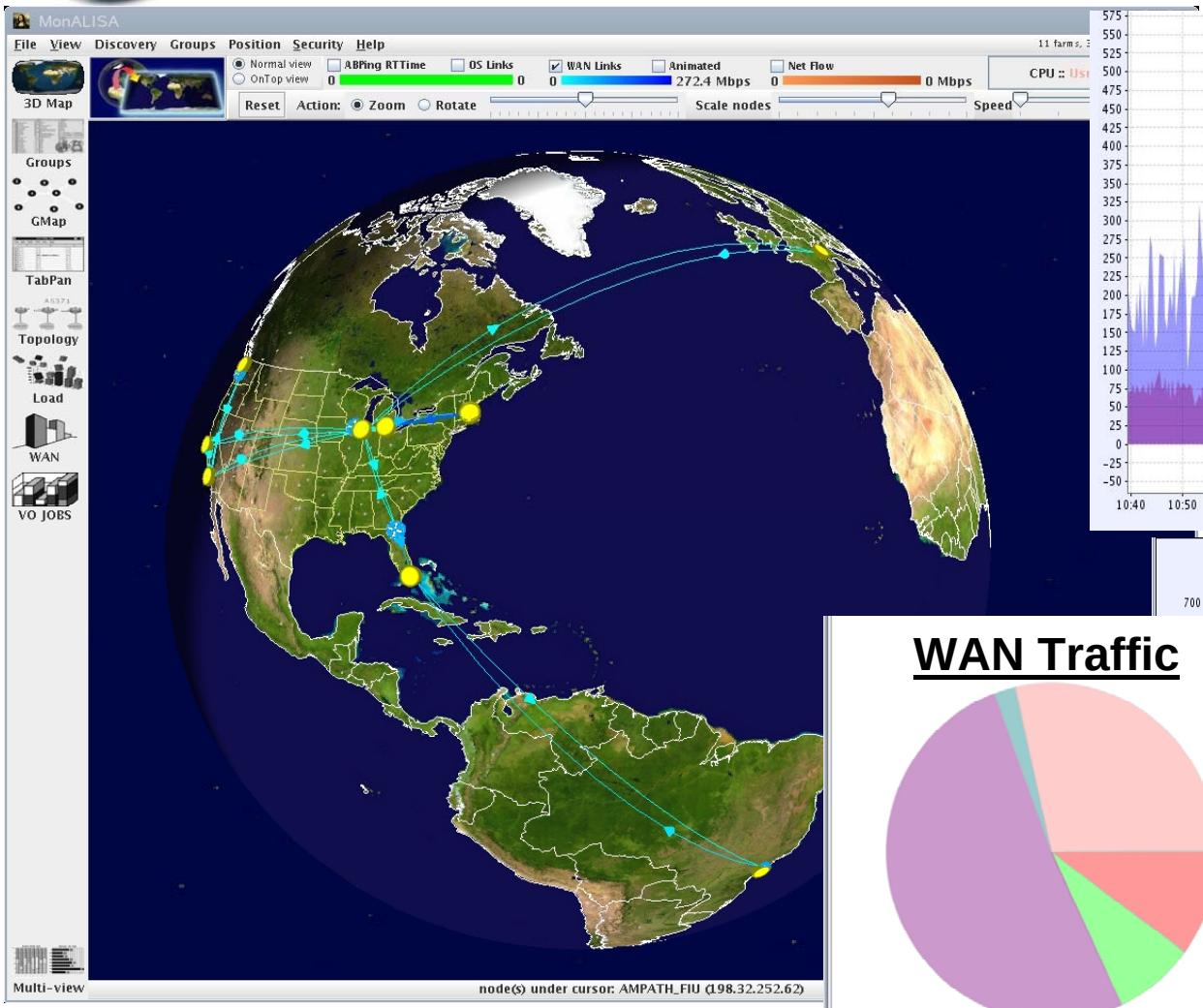
- Test for a Land Speed Record
- ~ 7 Gb/s in a single TCP stream from Geneva to Caltech

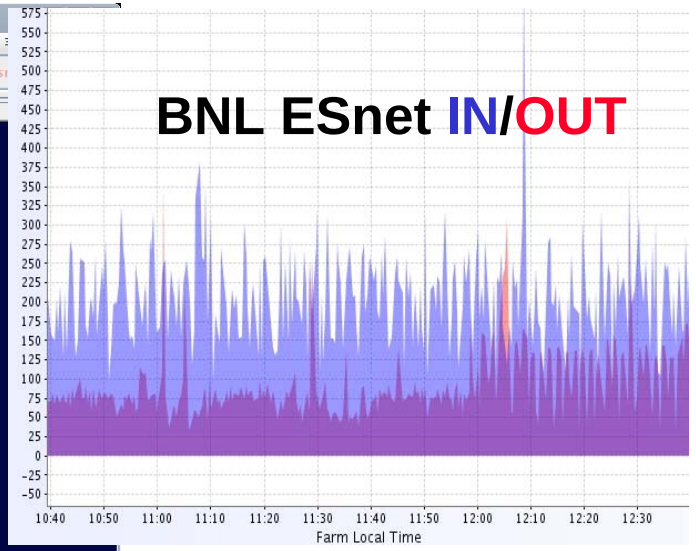


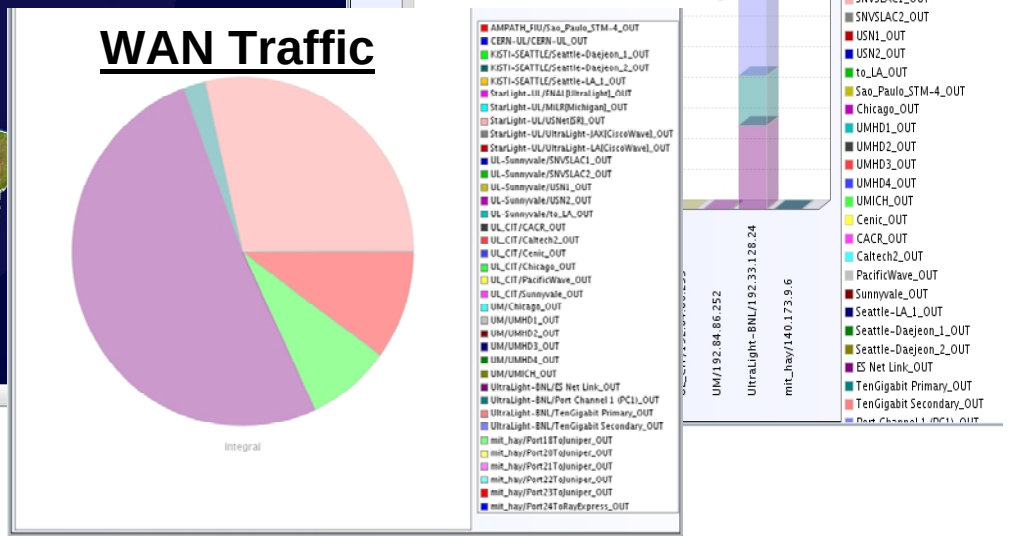




The UltraLight Network







Available Bandwidth Measurements

Embedded Pathload module.

MonALISA Repository
UltraLight

MonALISA Client
Click on the button below to start the Monalisa Client.
Client

- MonALISA Repository
- Global Views
- Node Info
- Available Bandwidt
  - Spider View
  - Matrix View
  - History
- Sites Status
- Site Info

close all

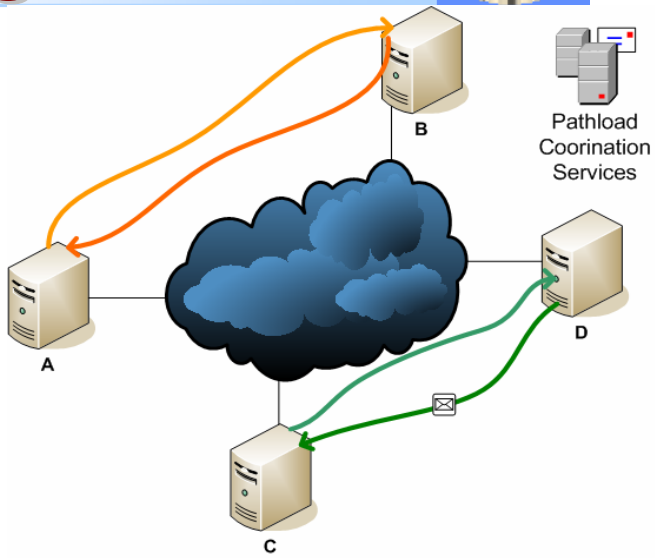
ABPing Configuration
Site Administration

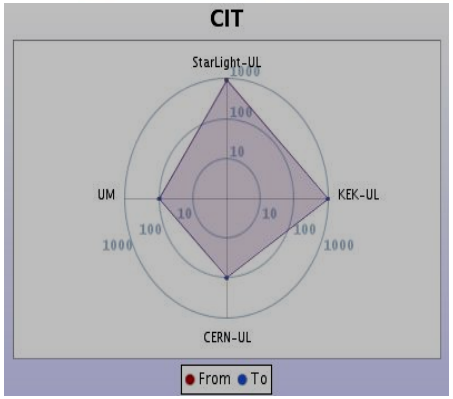
MonALISA
MONitoring Agents using a Large Integrated Services Architecture

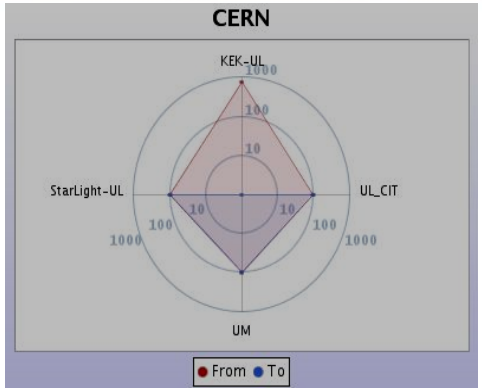
Available bandwidth measurements using pathload

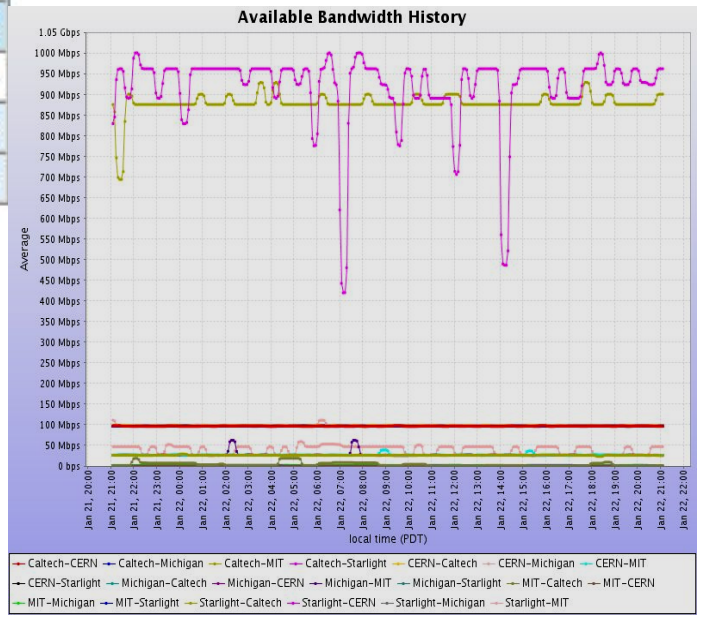
Available bandwidth between UL sites (average)

Site (from->to)	CERN-UL	KEK-UL	SPRACE-UL	UL_CIT	UM
CERN-UL	-	794.4 Mbps	97.37 Mbps	97.73 Mbps	97.85 Mbps
KEK-UL	750 Mbps	-	96.32 Mbps	993.2 Mbps	96.34 Mbps
StarLight-UL	97.4 Mbps	97.53 Mbps	-	875 Mbps	97.5 Mbps
UL_CIT	97.5 Mbps	993.2 Mbps	876 Mbps	-	96.63 Mbps
UM	97.48 Mbps	96.84 Mbps	96.55 Mbps	96.85 Mbps	-
Min	**97.4 Mbps**	**97.53 Mbps**	**96.32 Mbps**	**96.85 Mbps**	**96.34 Mbps**
Max	**750 Mbps**	**993.2 Mbps**	**876 Mbps**	**993.2 Mbps**	**97.85 Mbps**



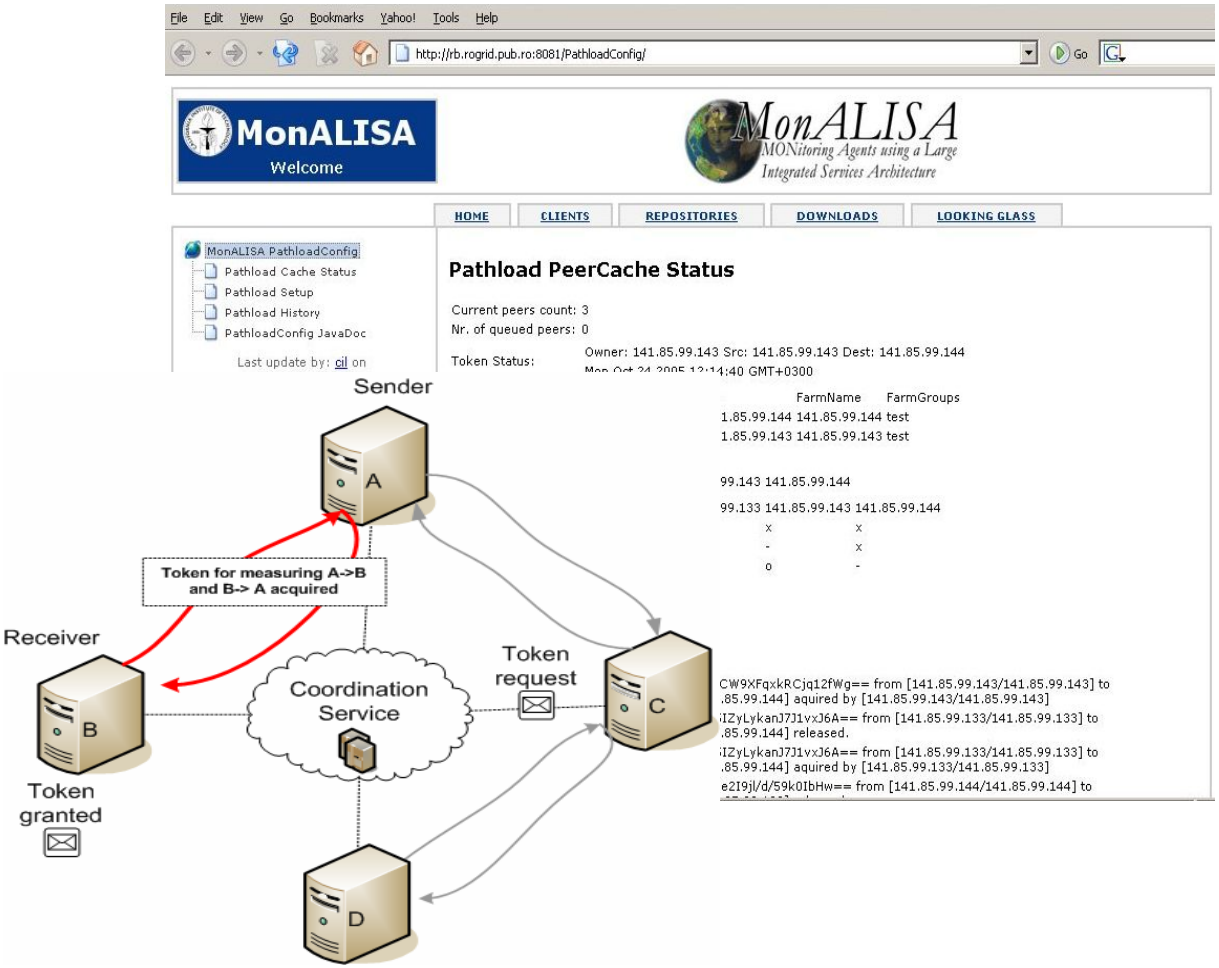




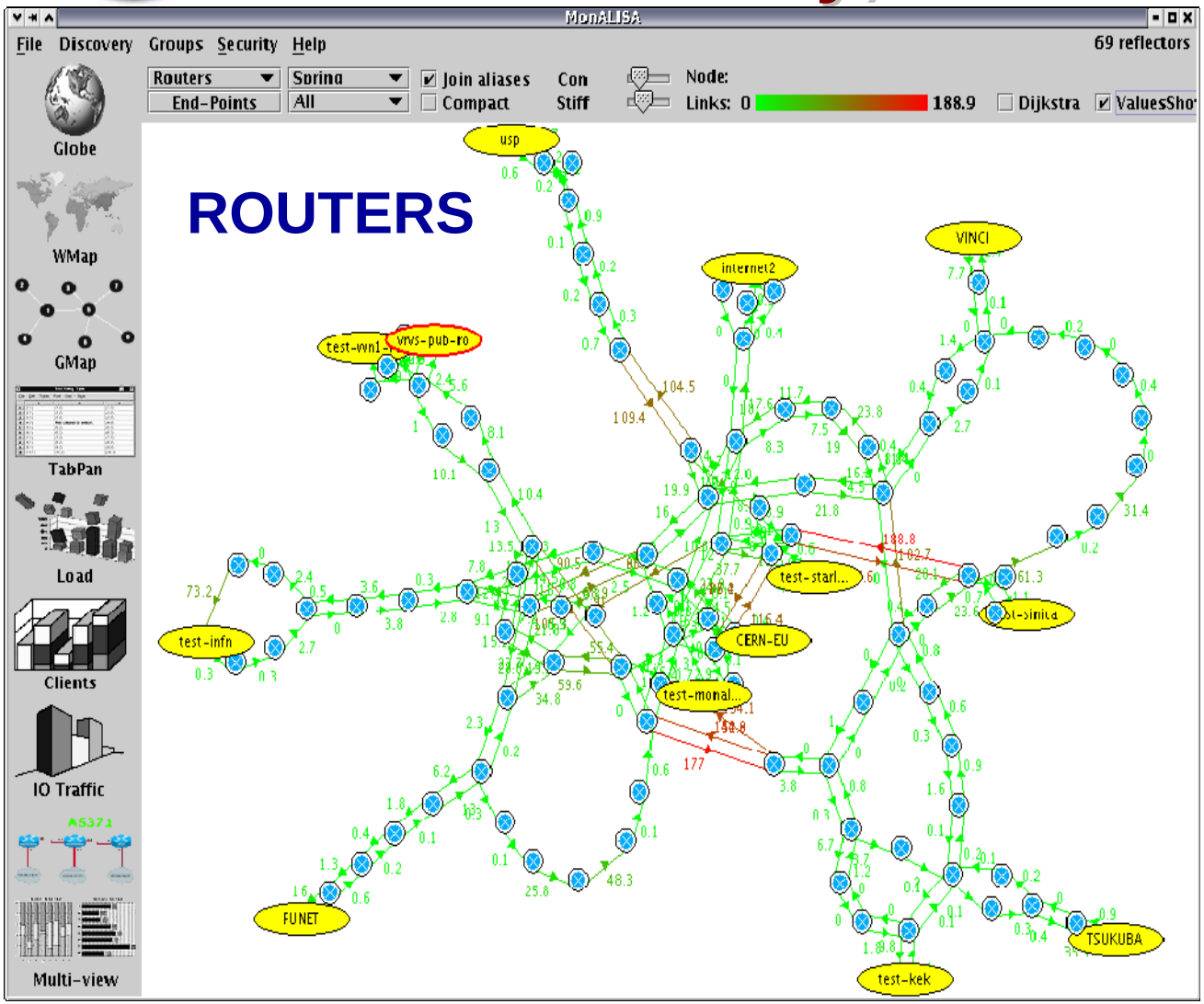


Coordination Service for Available Bandwidth Measurements

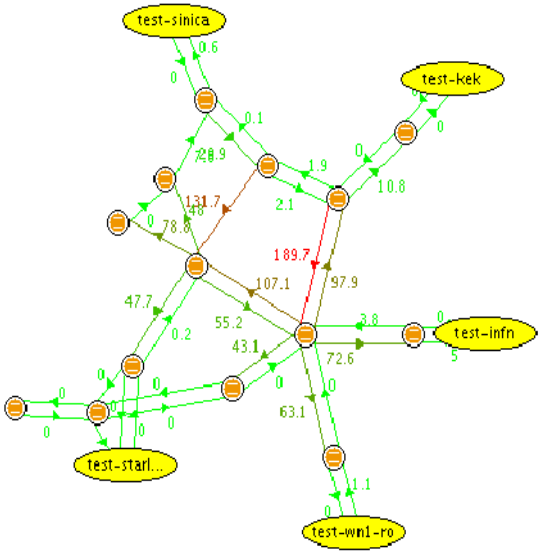
- Enforces measurement fairness
- Avoids multiple probes on shared network segments
- Dynamic configuration of measure-ment timing
- Logs events
- Provides service redundancy by using a master-slave model



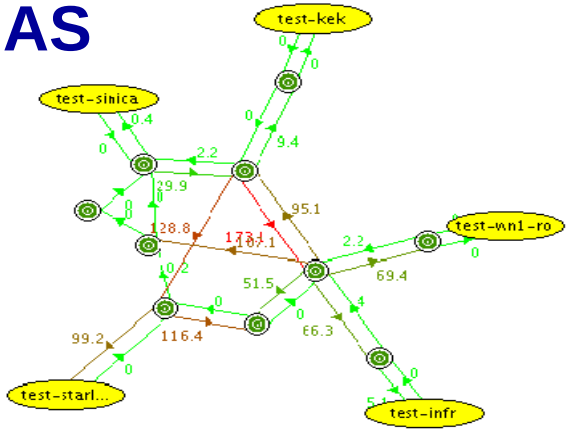
Monitoring Network Topology, Latency, Routers



NETWORKS

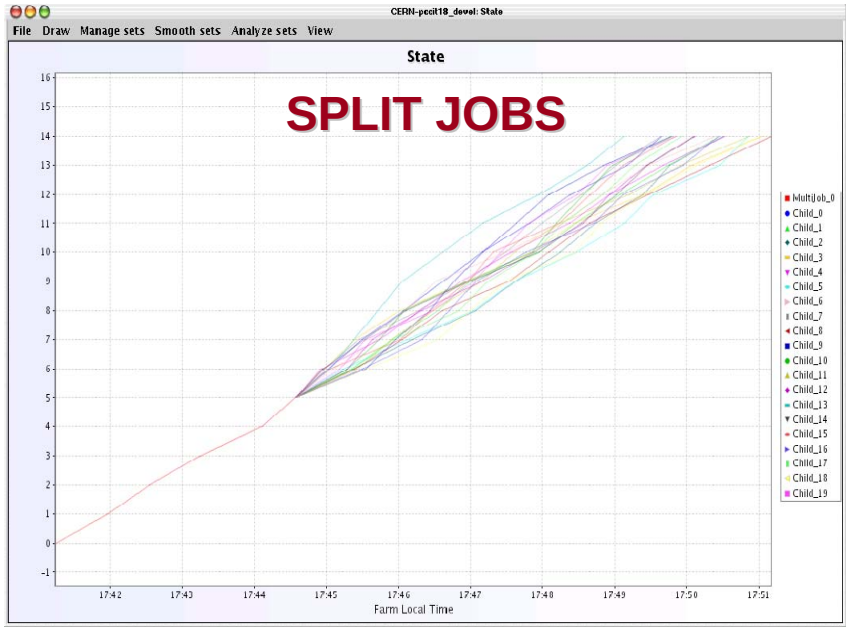


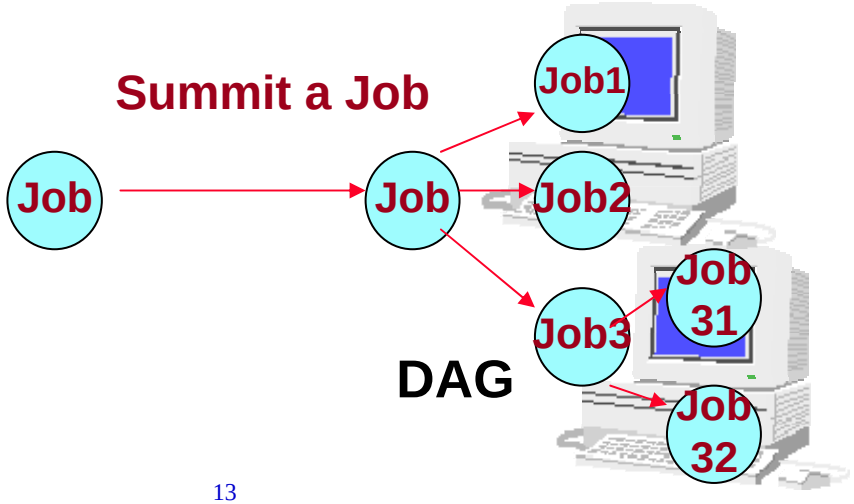
AS



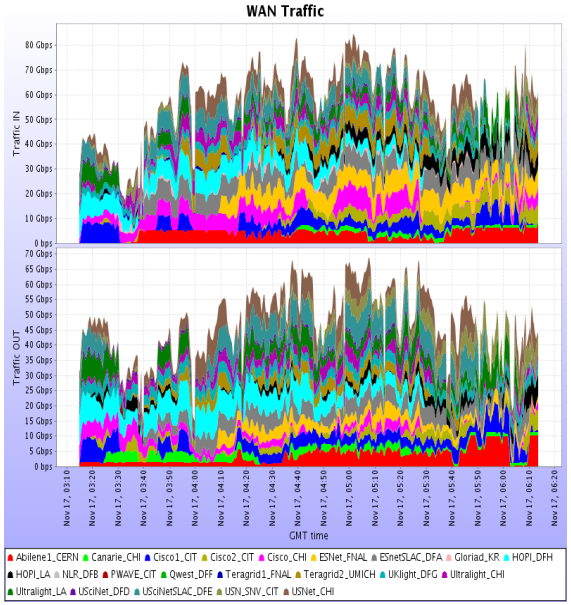
Real Time Topology Discovery & Display

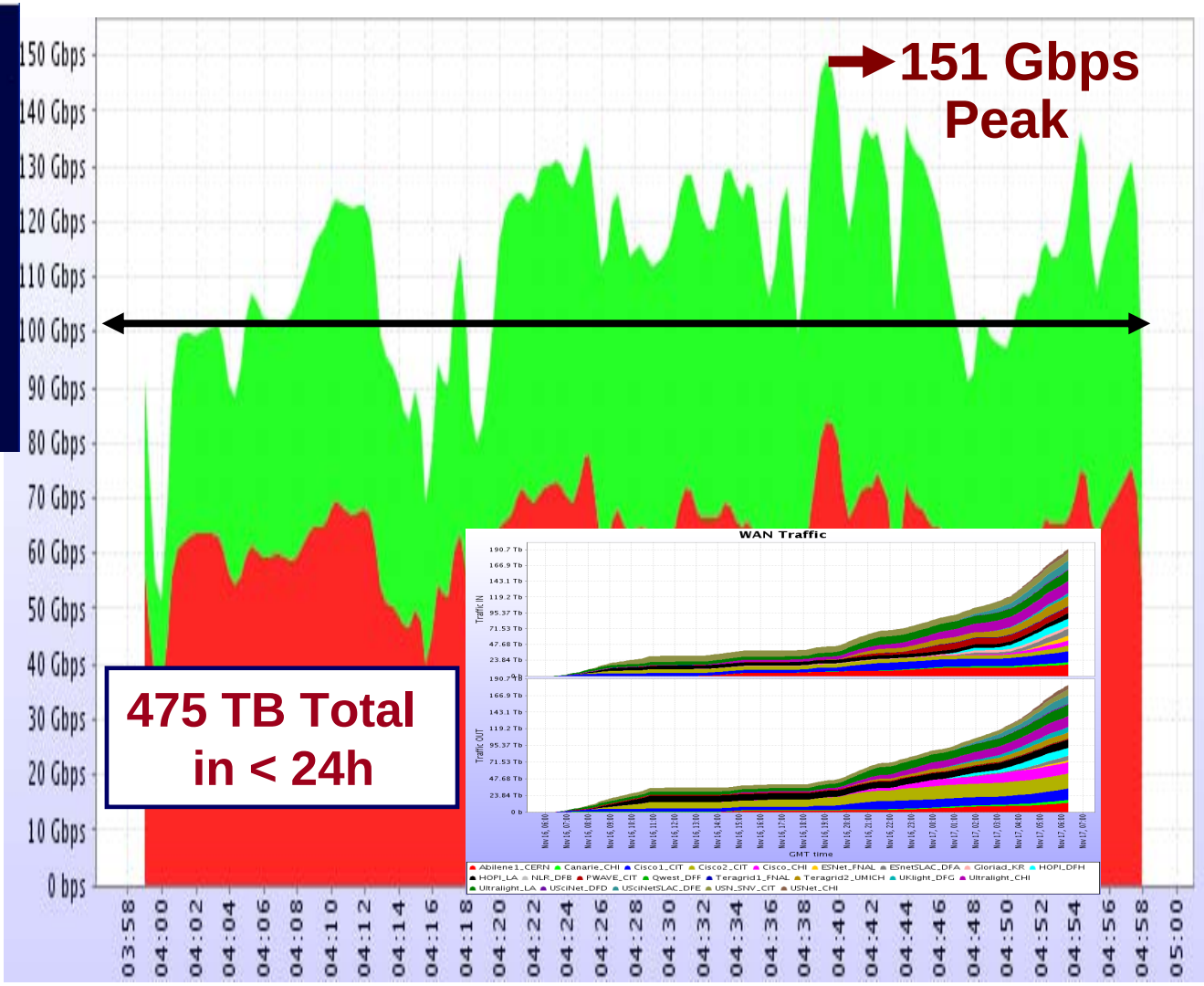
Monitoring the Execution of Jobs and their Time Evolution



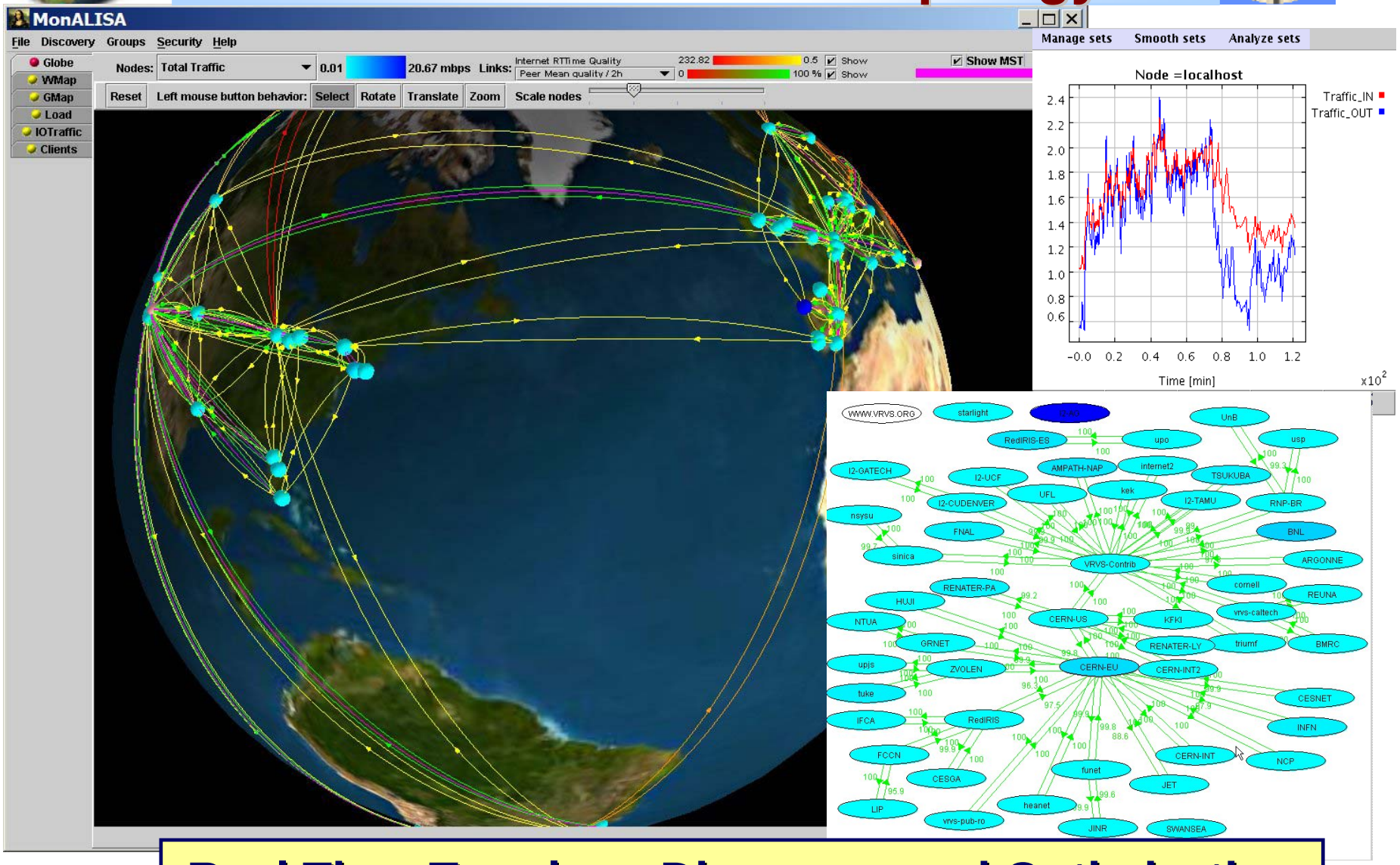


Bandwidth Challenge at SC2005



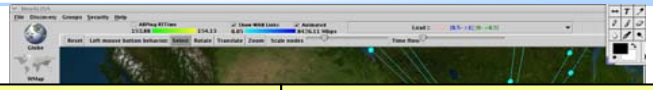


Monitoring VRVS Reflectors and Communication Topology



Real Time Topology Discovery and Optimization

Communities using MonALISA



Major Communities

- OSG
- CMS
- ALICE
- D0
- STAR
- VRVS
- LGC RUSSIA
- SE Europe GRID
- APAC Grid
- UNAM Grid (Mx)
- ABILENE

- ULTRALIGHT
- GLORIAD
- LHC Net
- RoEduNET

MonALISA Today

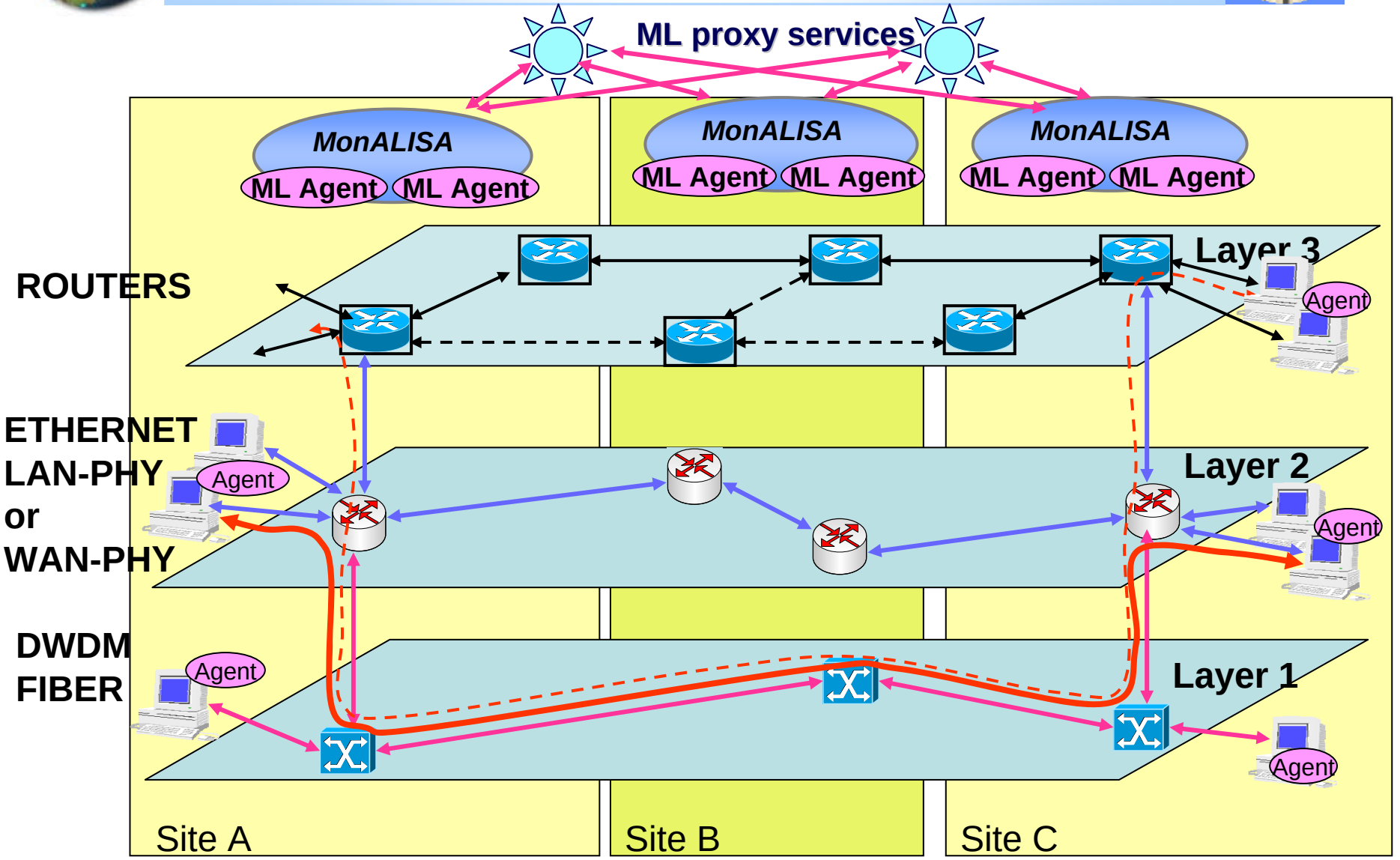
Running 24 X 7 at 250 Sites

- Collecting 250,000 parameters in near real-time
- Update rate of 25,000 parameter updates per second
- Monitoring
  - 12,000 computers
  - > 100 WAN Links
- Thousands of Grid jobs running concurrently

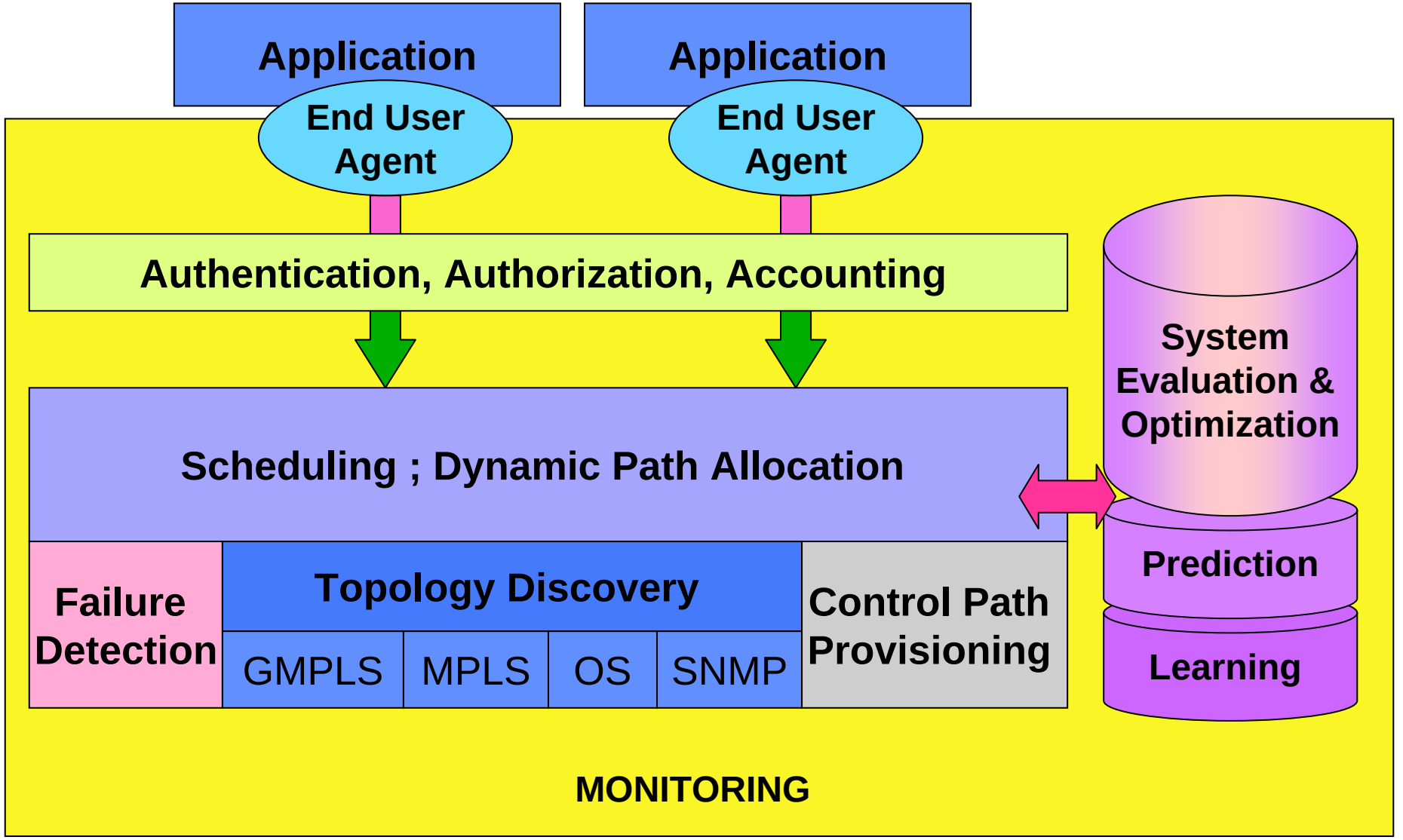
Demonstrated at:

- SC2003
- Telecom World 2003
- WSIS 2003
- SC 2004
- Internet2 2005
- TERENA 2005
- IGrid 2005
- SC 2005

The Functionality of the VINCI System



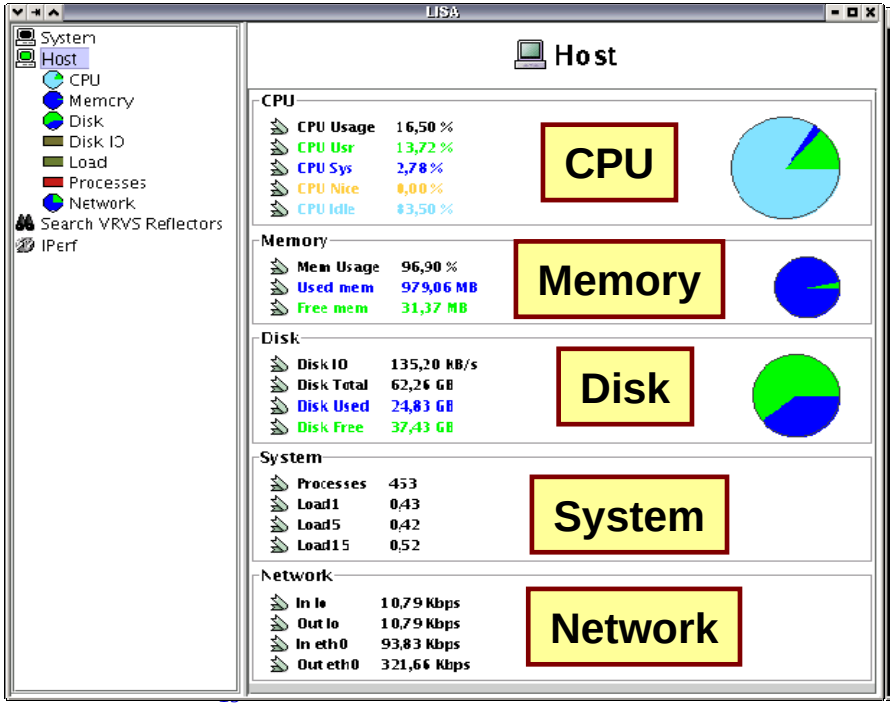
The Main VINCI Services

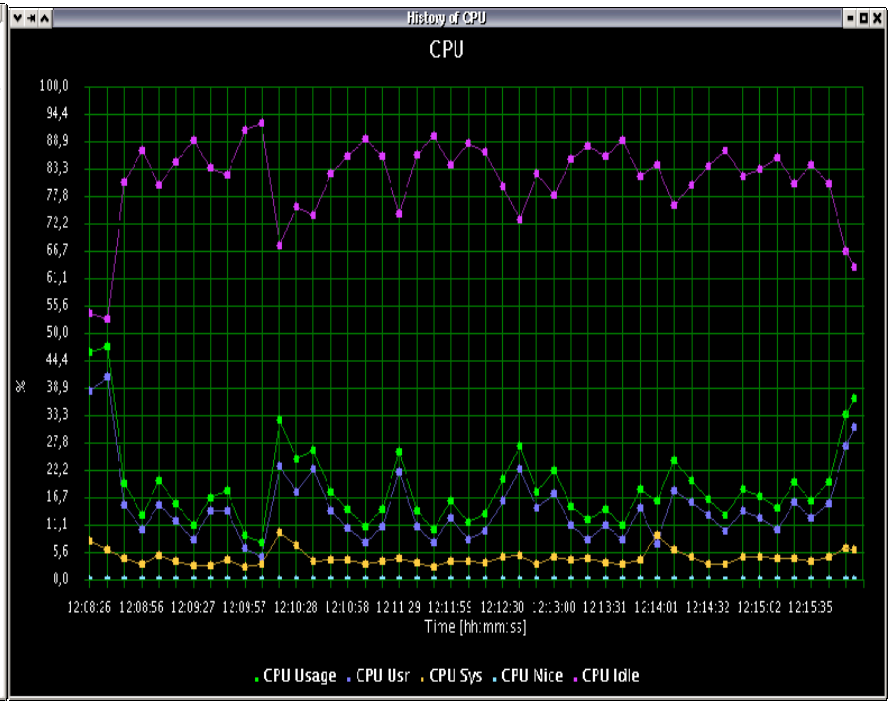


End User / Client Agent

LISA- Localhost Information Service Agent

- Authorization
- Service discovery
- Local detection of the hardware and software configuration
- Complete end-system monitoring: Per-process load, Disk Storage and I/O, per-port network throughputs, etc.
- End-to-end performance measurements
- **Acts as an active listener for all events related to the requests generated by its local applications**





Secure Service Discovery and AAA

Service Registration and Discovery

We use JINI Lookup Services to provide a reliable mechanism to dynamically register services, and their dynamic sets of attributes

Authentication, Authorization and Accounting for Users

We use external AAA services supported by different Virtual Organizations. Loadable plug-in interface modules to support different protocols and services will provide the necessary flexibility to work with different grids and networks

Topology Discovery Using Specialized Agents

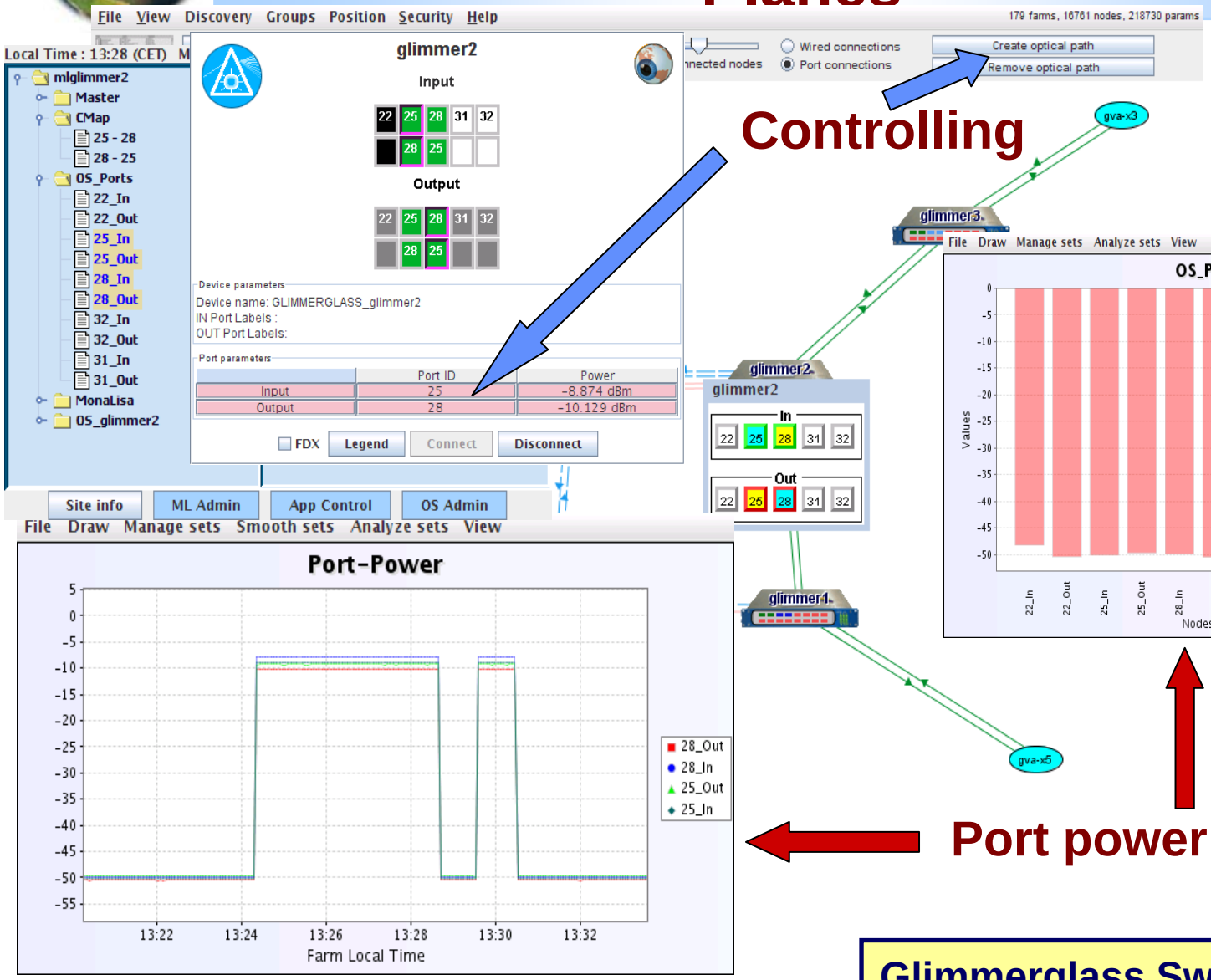
- Specialized agents are used to
  (1) discover the connection topology for each service
  (2) keep a dynamic map of how they are allocated & used, and
  (3) get information on the traffic on each segment.
- Agents running on multiple MonALISA services in parallel provide the basic information to the scheduling system
- These agents draw on information from MPLS/GMPLS /DRAGON/Optical Path agents, where the infrastructure provides this functionality

Targeted Capabilities for Topology Discovery & Path Selection

Examples of Capabilities:

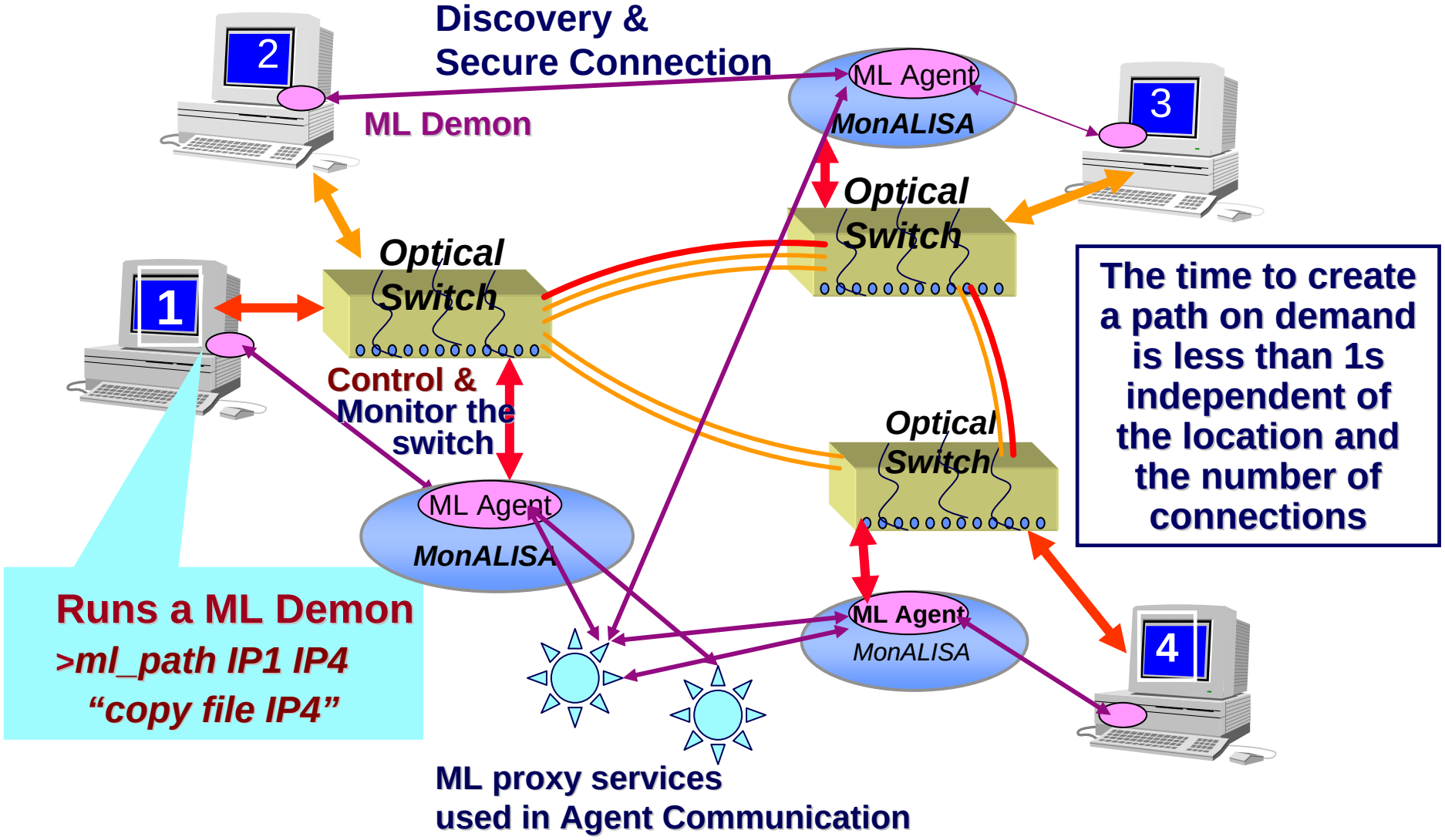
- Determine which path-options exist between two locations in the network
- List components in the path that are “manageable”
- Locate network resources and services which have agreements with a given VO
- Given two replicas of a data source, “discover” (in conjunction with monitoring) the estimated bandwidth and reliability, and hence the “estimated time of successful delivery” of each to a given destination.

Monitoring and Controlling Optical Planes

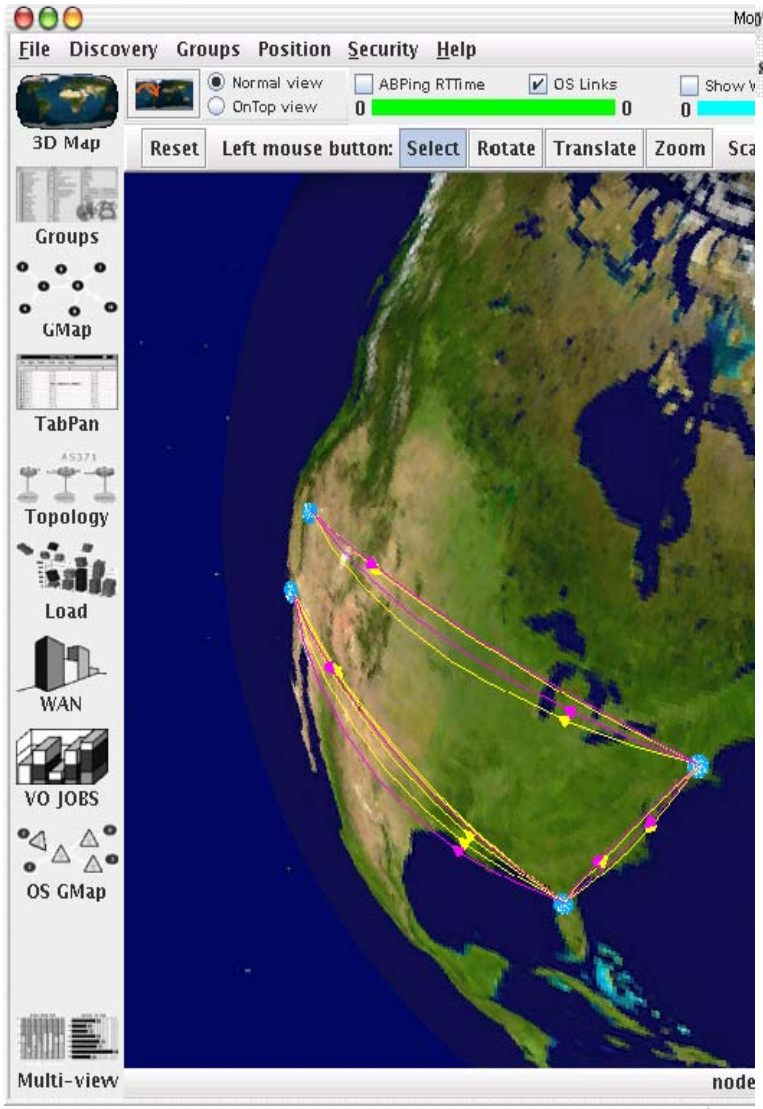


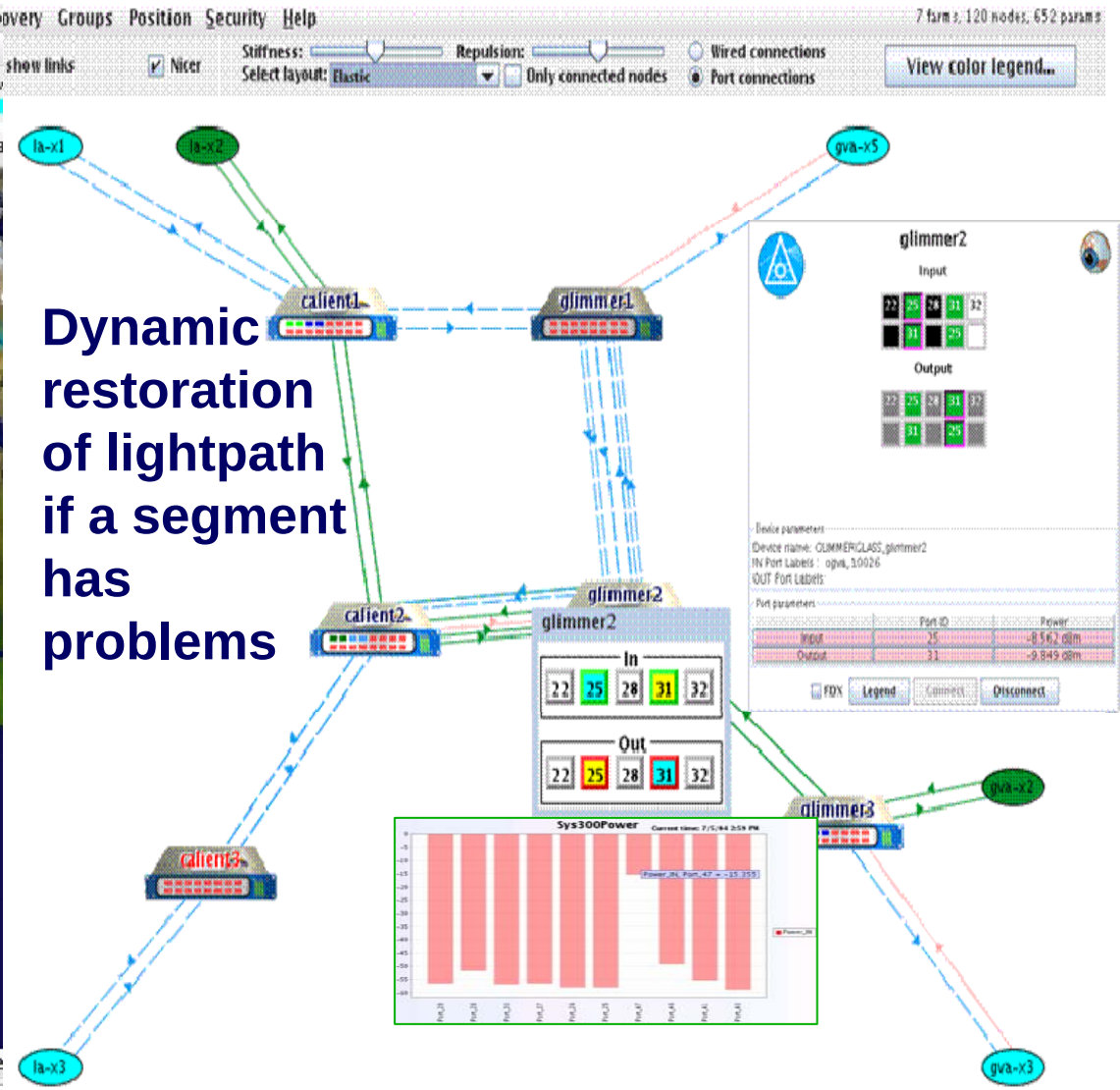
Glimmerglass Switch Example

Agents to Create on Demand an Optical Path or Tree



A Real-World Working Example: Agents Create an Optical Path on Demand





Dynamic restoration of lightpath if a segment has problems

The Workflow Scheduler

Scheduler is implemented as a set of collaborating agents

- It provides complete autonomy to each provider of resources, who can implement his own policy
- There is no single point of failure

"Market Model" Scheduling Scheme

- Each agent uses policy-based priority queues; it negotiates for an end-to-end connection using a set of cost functions
- A lease mechanism is implemented for each offer an agent accepts from its peers
- Two phase commit and periodic lease renewal are used for all agents; this allows a flexible response of the agents to task completion, as well as to application failure or network errors
- If network errors are detected, supervising agents cause all segments to be released along a path
- An alternative path then may be set up: rapidly enough to avoid a TCP timeout, so that the transfer can continue uninterrupted

Interfaces with Network Services
MPLS, GMPLS

We are developing agents capable of interacting with MPLS and GMPLS controllers, to provide in near real-time topology maps for other services and to generate connection requests. These agents will continuously monitor and supervise the connections they created.

- **The MPLS agents can be used together with the optical path agents to create an end-to-end network configuration.**
- **For networks where GMPLS is supported, the agents only need to interface with the head-end devices. The GMPLS standard protocols provides topology discovery (LMP), routing (OSPF) and provisioning (RSVP) and allow interoperability across domains. For example, special routing can be done with VINCI agents if not included in OSPF, but the topology discovery and provisioning can be done with GMPLS.**
- **We also provide agents capable of configuring routers or switches using SNMP or TL1 directly**
- **We will provide agents capable to interacting with other network services systems (for example those in the DRAGON project)**

Learning and Prediction

- Learning algorithms (e.g. Self Organizing Neural Networks) will be used to evaluate the traffic created by other applications, to identify major patterns, and dynamically setup effective connectivity maps
- It is very difficult if not impossible to assume that we could predict all possible events in a complex environment like a grid in advance
- Heuristic learning is thus the only practical approach, where agents can acquire the necessary information to describe their environments
- The multi-agent learning task includes two levels:
  - the local level of individual learning agents
  - the global level, exploiting inter-agent communication
- We need to ensure that each agent can learn to optimize its actions locally, while the global monitoring mechanism acts as a ‘driving force’ that causes the agents’ behavior to evolve collectively, based on the accumulated experience

Mumbai-Japan-US Links