Lambda Station: Alternate Network Path Forwarding for Production SciDAC Applications

Fermi National Accelerator Laboratory
Andrey Bobyshev, Matt Crawford, Phil DeMar,
Vyto Grigaliunas, Maxim Grigoriev, Alexander Moibenko,
Don Petravick

California Institute of Technology
Harvey Newman, Conrad Steenberg, Michael Thomas

CHEP2007
Victoria, BC, Canada
September 2-7

Outline of the talk

- some terms
- goals and building blocks of the project
- software architecture
- Java API, middleware
- production **SRM** environment
- Lambda Station (*λS*) service in production **SRM** environment
- problems and challenges, plans

Basic terms

- **Lambda Station (λS)** – a host with special software to control traffic path across LAN and WAN on-demand of applications

- **PBR** – policy based routing

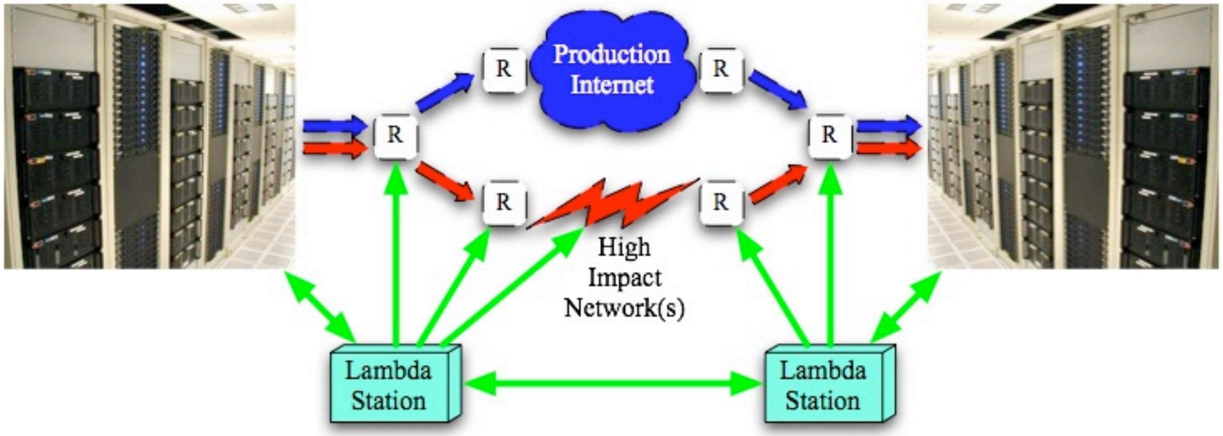
- **PBR Client** – a system or cluster and applications running on it sourcing traffic flows that can be subject for policy based routing

- **Flow -** a stream of packets with some attributes in common such as endpoint IP addresses (or range of addresses), protocols, protocol's ports if applicable and differentiated services code point (**DSCP**).

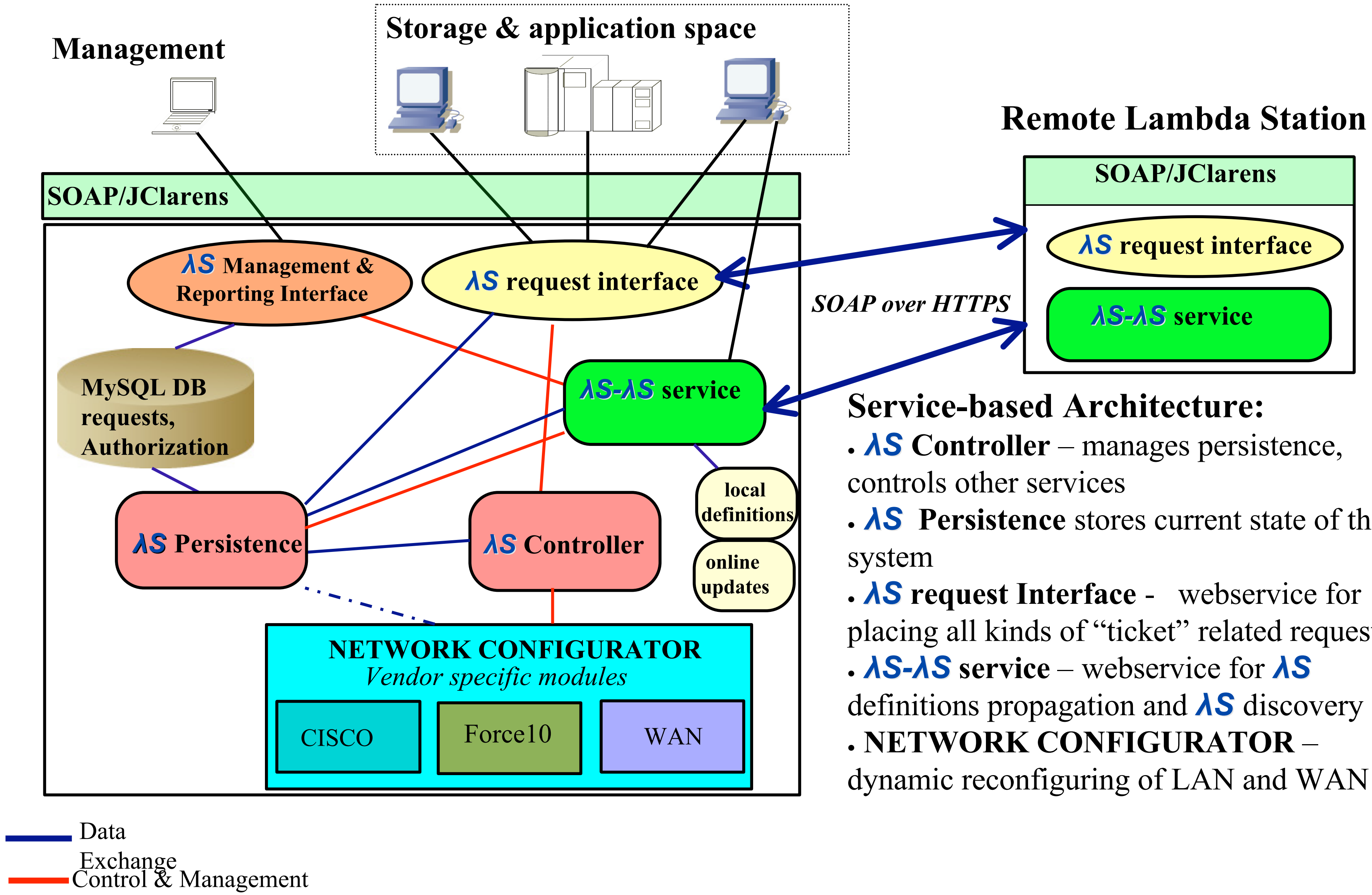
The goal of the project

The main goal of **Lambda Station** project is to design, develop and deploy a network path selection services to interface production storage and computing facilities with advanced research networks.

- selective forwarding on a ***per flow*** basis
- alternate network paths for high impact data movement
- access control in site edge routers for those selected flows
- on-demand from **applications** (authentication & authorization)
- current implementation based on policy-based routing & including the support of **DSCP** marking



Lambda Station Building Blocks



Service-based Architecture:

- **λS Controller** – manages persistence, controls other services
- **λS Persistence** stores current state of the system
- **λS request Interface** -   webservice for placing all kinds of “ticket” related requests
- **λS-λS service** – webservice for **λS** definitions propagation and **λS** discovery
- **NETWORK CONFIGURATOR** – dynamic reconfiguring of LAN and WAN

Network Configurator (Netconfig) Module

- dynamically modifies the configurations of local network devices
- a vendor dependent component
- implemented in perl

- Configuring **PBR** on **Cisco™** routers
  - IOS version with support for sequencing type of named **ACLs**
  - interface on which **PBR** is applied needs to be configured with ***"ip policy route-map"*** statement
    - route map needs to be configured as ordered list of match/action statements
    - match criteria need to be associated with **ACLs**

Basic λS requests

- openSvcTicket
  - Major ***λS*** operational request, places alternative path reservation (“**ticket**”)
  - Accepts **svcTicket** element as an argument, validated by XML schema
  - Returns udpated **svcTicket** XML element with **ticket ID**
- updateFlowSpecs
  - updates flow specification for the “**ticket**”
  - Accepts **svcTicket** XML element as an argument, validated by XML schema
  - Returns **boolean**
- getTicket
  - get **svcTicket** XML element with full information about placed “**ticket**”
  - Accepts **“ticket”** ID
  - Returns **svcTicket** XML element
- cancelTicket
  - cancel existing “**ticket**”, ticket will be closed and network topology will be changed back to production path
  - Accepts “**ticket**” ID
  - Returns **boolean**

“**ticket**” reservation Operational modes

All modes are subject to TLS/SSL based authentication and rules based authorization

- **new** ticket
  - create a new “**ticket**”
  - client must be authorized for local ***λS*** and station must be authorized for remote ***λS***
- **join** ticket
  - join already active “**ticket**” (in case of multiple requests for the same flow)
  - existing “**ticket**” parameters will be reused
- **extend** ticket
  - extend already active “**ticket**”
  - **endtime** will be extended

Java API

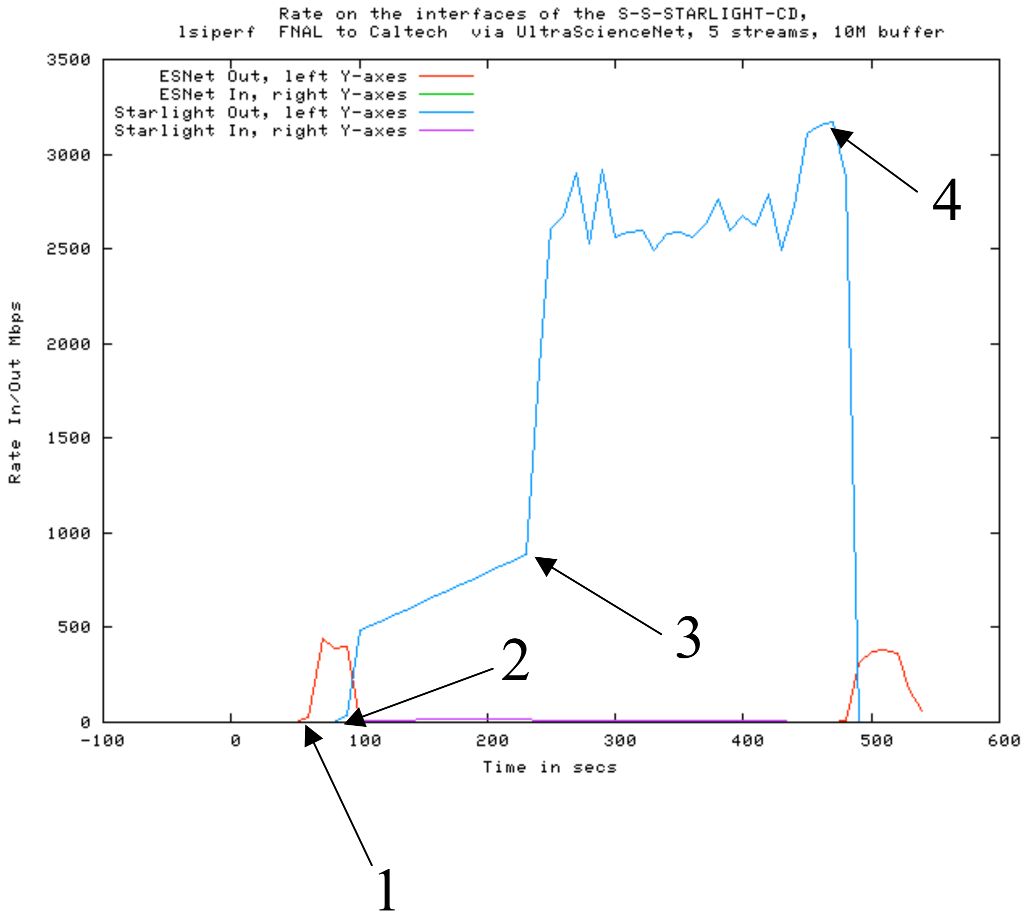
- Service Oriented Architecture, interfaces described by **WSDL**
- utilized **JClarens** and **Axis** framework as a web-services toolkit
- messages are defined and strongly validated by XML schema
- ***λS*** service is multi-threaded, one thread for ***λS*** **Controller**, one thread for ***λS-λS*** service and threads pool for **openSvcTicket** requests
- ***λS-λS*** and client-***λS*** authentication is based on **gLite** library and supports standard **Grid proxies** and **KCA**-issued certificates
- Authorization is based on rules set
- General framework persistence is accomplished by MySQL DB backend
- secure document/literal wrapped SOAP messages, Web Services Interoperability Profile (WS-I Basic Profile Version 1.1)

Java API *(continued)*

- Automated ***λS*** and **PBR** client configuration management
- Automated deployment (one can install on any Linux box)
- ***λS*** **Controller**, ***λS-λS*** , ***λS*** AAA, ***λS*** client interface are ready for deployment. Supported **Java** and **perl** clients.
- Some interest from ANL to support **C** client for **Globus** toolkit
- **Network Configurator** calls implemented in interface and may relay requests to **perl** service (**SOA** at work)
- Currently deployed and work (exchanging **PBR** and ***λS*** configurations) at **Fermilab** and **Caltech**

LSiperf End-to-End Test

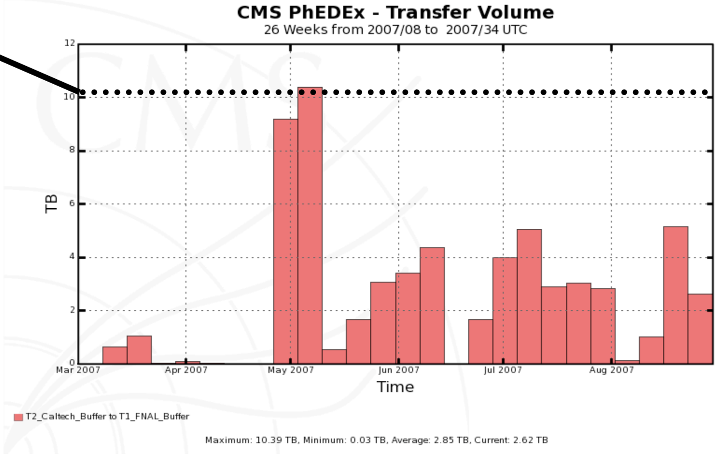
1. Data transfer started:
   - 10GE host; 5 tcp streams
   - Network path is via ESnet
     - OC12 bottleneck…
   - Path MTU is 1500B
   - LambdaStation openSvcTicket is placed
2. LambdaStation changes network path to USN
3. Host path MTUD check detects a larger path MTU
4. LambdaStation service ticket expires:
   - Network path changed back to ESnet



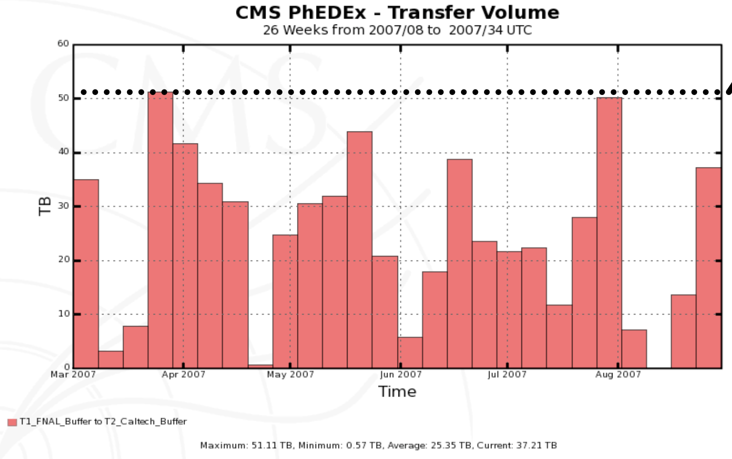
SRM production environment

- At Fermilab
  - 100s of read/write pool nodes, ~ 1PB of tape-backed disk
  - more than 100TB in resilient storage, about 650 worker nodes
- At Caltech
  - about 75 pool nodes
  - about 55TB in resilient storage

10TB

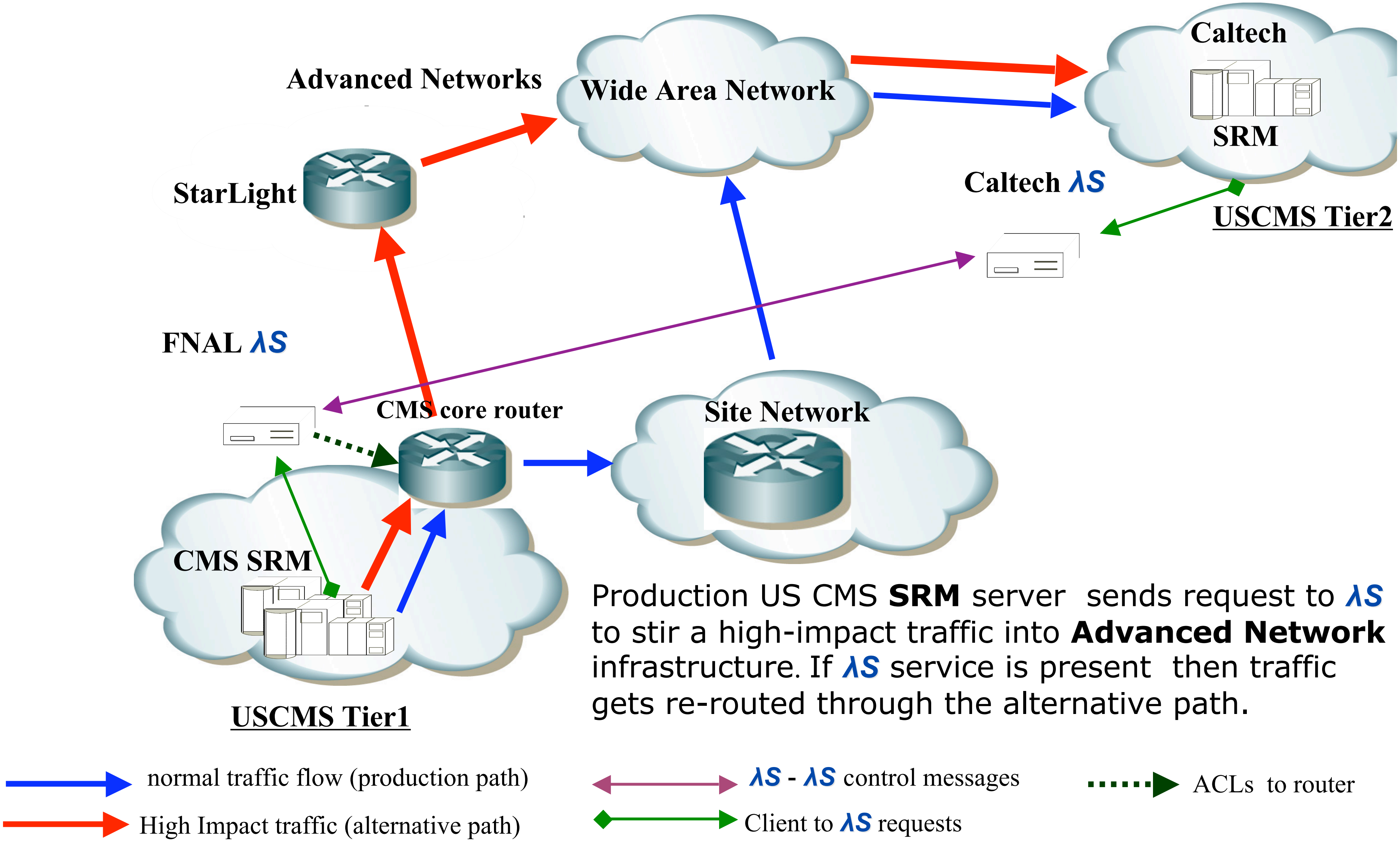


50TB



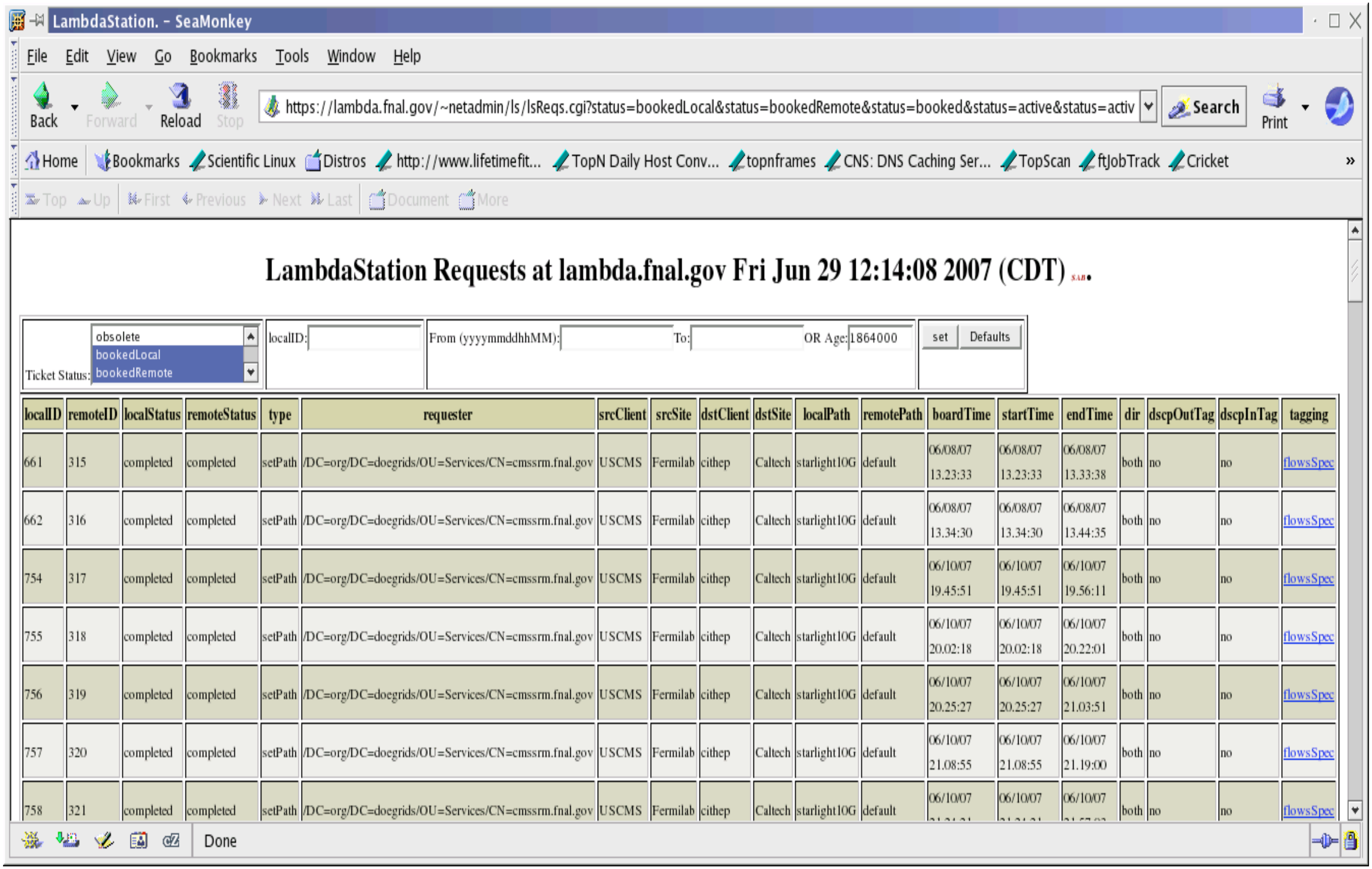
about 500 requests per day to LS (randomly distributed)

SRM/dCache 1.7 LS-awareness



Production US CMS **SRM** server sends request to ***λS*** to stir a high-impact traffic into **Advanced Network** infrastructure. If ***λS*** service is present then traffic gets re-routed through the alternative path.

LS in production SRM environment



LambdaStation. - SeaMonkey
File Edit View Go Bookmarks Tools Window Help
Back Forward Reload Stop
Search
Print
Home Bookmarks Scientific Linux Distros http://www.lifetimefit... TopN Daily Host Conv... topnframes CNS: DNS Caching Ser... TopScan ftJobTrack Cricket
Top Up First Previous Next Last Document More
Done

https://lambda.fnal.gov/~netadmin/ls/lsReqs.cgi?status=bookedLocal&status=bookedRemote&status=booked&status=active&status=activ

LambdaStation Requests at lambda.fnal.gov Fri Jun 29 12:14:08 2007 (CDT)

Ticket Status: obsolete / bookedLocal / bookedRemote

localID:

From (yyyymmddhhMM): To: OR Age: 1864000

set Defaults

localID	remoteID	localStatus	remoteStatus	type	requester	srcClient	srcSite	dstClient	dstSite	localPath	remotePath	boardTime	startTime	endTime	dir	dscpOutTag	dscpInTag	tagging
661	315	completed	completed	setPath	/DC=org/DC=doegrids/OU=Services/CN=cmssrm.fnal.gov	USCMS	Fermilab	cithep	Caltech	starlight10G	default	06/08/07 13.23:33	06/08/07 13.23:33	06/08/07 13.33:38	both	no	no	flowsSpec
662	316	completed	completed	setPath	/DC=org/DC=doegrids/OU=Services/CN=cmssrm.fnal.gov	USCMS	Fermilab	cithep	Caltech	starlight10G	default	06/08/07 13.34:30	06/08/07 13.34:30	06/08/07 13.44:35	both	no	no	flowsSpec
754	317	completed	completed	setPath	/DC=org/DC=doegrids/OU=Services/CN=cmssrm.fnal.gov	USCMS	Fermilab	cithep	Caltech	starlight10G	default	06/10/07 19.45:51	06/10/07 19.45:51	06/10/07 19.56:11	both	no	no	flowsSpec
755	318	completed	completed	setPath	/DC=org/DC=doegrids/OU=Services/CN=cmssrm.fnal.gov	USCMS	Fermilab	cithep	Caltech	starlight10G	default	06/10/07 20.02:18	06/10/07 20.02:18	06/10/07 20.22:01	both	no	no	flowsSpec
756	319	completed	completed	setPath	/DC=org/DC=doegrids/OU=Services/CN=cmssrm.fnal.gov	USCMS	Fermilab	cithep	Caltech	starlight10G	default	06/10/07 20.25:27	06/10/07 20.25:27	06/10/07 21.03:51	both	no	no	flowsSpec
757	320	completed	completed	setPath	/DC=org/DC=doegrids/OU=Services/CN=cmssrm.fnal.gov	USCMS	Fermilab	cithep	Caltech	starlight10G	default	06/10/07 21.08:55	06/10/07 21.08:55	06/10/07 21.19:00	both	no	no	flowsSpec
758	321	completed	completed	setPath	/DC=org/DC=doegrids/OU=Services/CN=cmssrm.fnal.gov	USCMS	Fermilab	cithep	Caltech	starlight10G	default	06/10/07	06/10/07	06/10/07	both	no	no	flowsSpec

Project accomplishments

- Software version 1.0 (a fully functional prototype supporting whole cycle of ***λS*** functionality)
- positive results of testing between **Fermilab** and **Caltech**
- ***lsiperf, lsTraceroute*** – wrappers around well known applications to add ***λS*** awareness (based on prototype version 1.0)
- **SRM/dCache** integration added in production **SRM 1.7.0** release
- ***λS***-aware production **SRM/dCache** runs at **Fermilab**'s **US CMS Tier1** site and **Caltech Tier2** site
- Interoperable **Java** implementation of the ***λS'*****s** major components (**perl**, **Java** clients available)

Problems and challenges

- Traffic Asymmetry is bad for high performance applications
- Making applications **λS**-aware is very complex task
- Definition of **PBR Client** is a complex issue, auto definition is not yet available, although configuration management is available

Plans

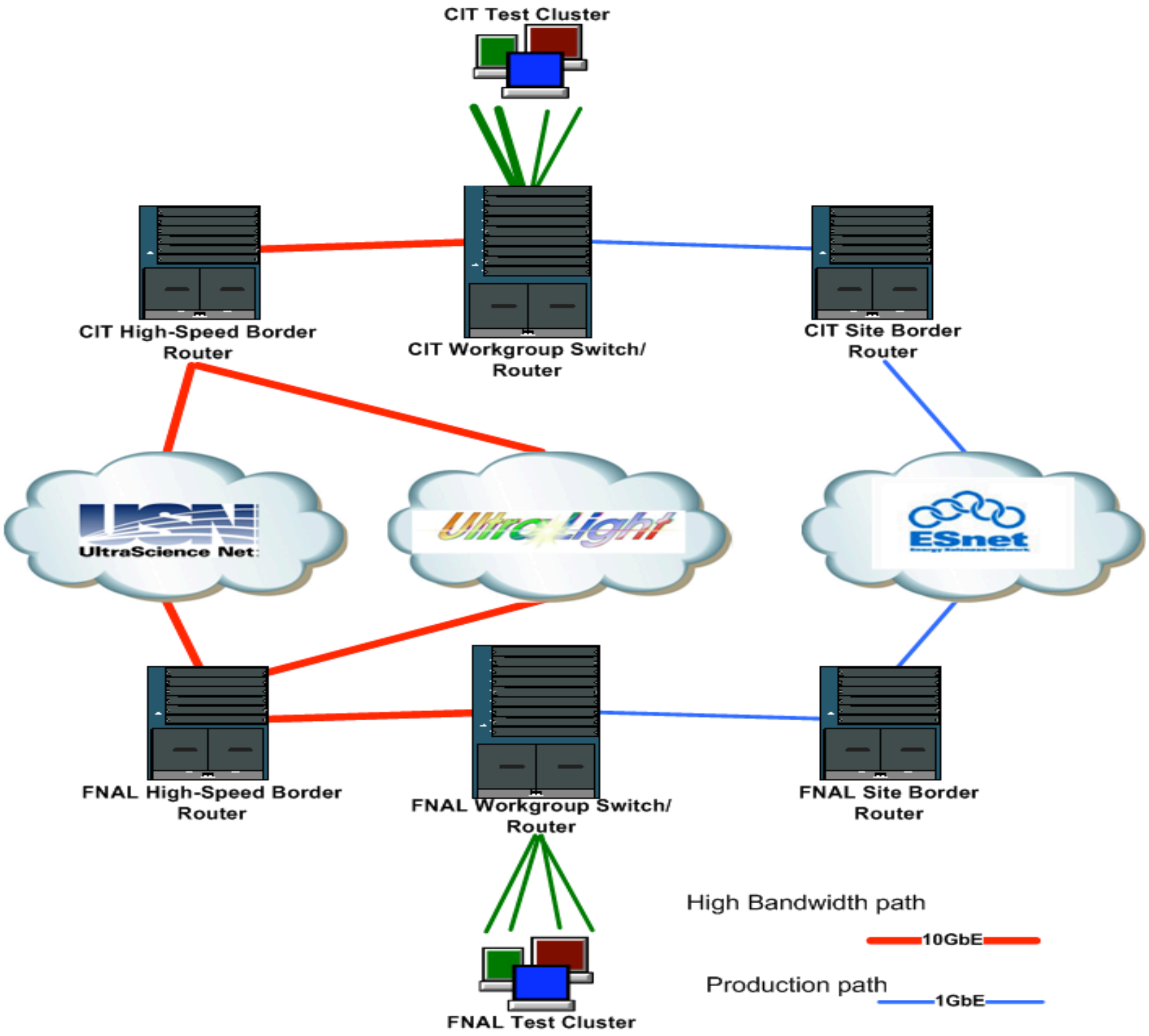
- release fully functional Java **λS** API
- add Java client **λS** API into production **SRM/dCache**
- add real-time monitoring of utlized resources (**perfSONAR** ?)
- add **WAN** control plane module
- integration with **OSCARS, DRAGON** and **Terapaths** (pushing idea of unified Network Path Reservation Model)

Links

- Lambda Station project: http://www.lambdastation.org/
- SRM Wiki: https://srm.fnal.gov/twiki/bin/view/SrmProject/WebHome
- Wiki page on LambdaStation, OSCARS, TeraPaths integration: https://wiki.internet2.edu/confluence/display/CPD/LambdaStation+and+TeraPaths

Questions ?

Lambda Station Testbed



Office of Science
U.S. DEPARTMENT OF ENERGY

Fermilab

Flows and DSCP tagging

Any combination of flow's attributes can be used by Lambda Station (**LS**) software to identify flows on per-ticket basis.

Typical steps of alternative path reservation:

- client API sends request for service to local LS
- local LS negotiates service and parameters with remote site LS (*optional*)
- local LS configures local and *wide area network (in future plans)*
- client API starts marking traffic (if specified).

Current LS software is capable to complete all these steps within 3 – 5 mins. That is why it is desirable to know flow selection parameters before transferring is started:

- endpoint IP addresses
- DSCP

DSCP Tagging

Complexity of using DSCP tagging:

- preservation of DSCP is not guaranteed in WAN

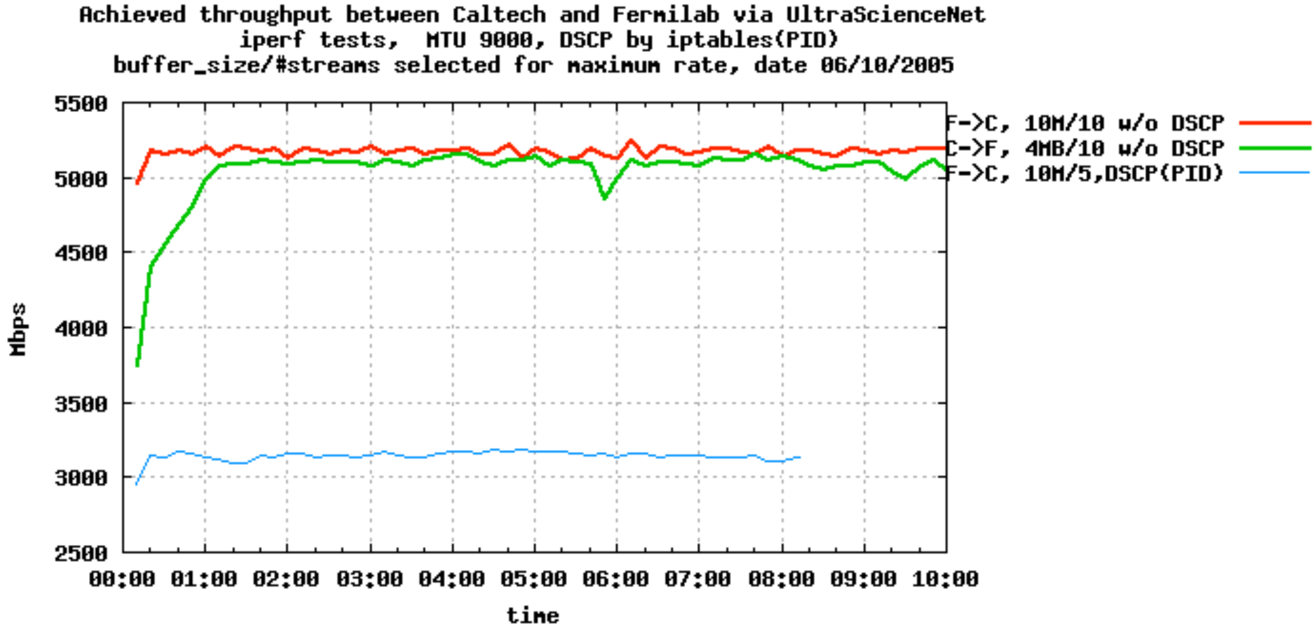
- DSCP tagging needs to be synchronized between sites for dynamically configurable networks (asymmetry is bad for high-performance transfer)

LS software does support two different modes of DSCP tagging :

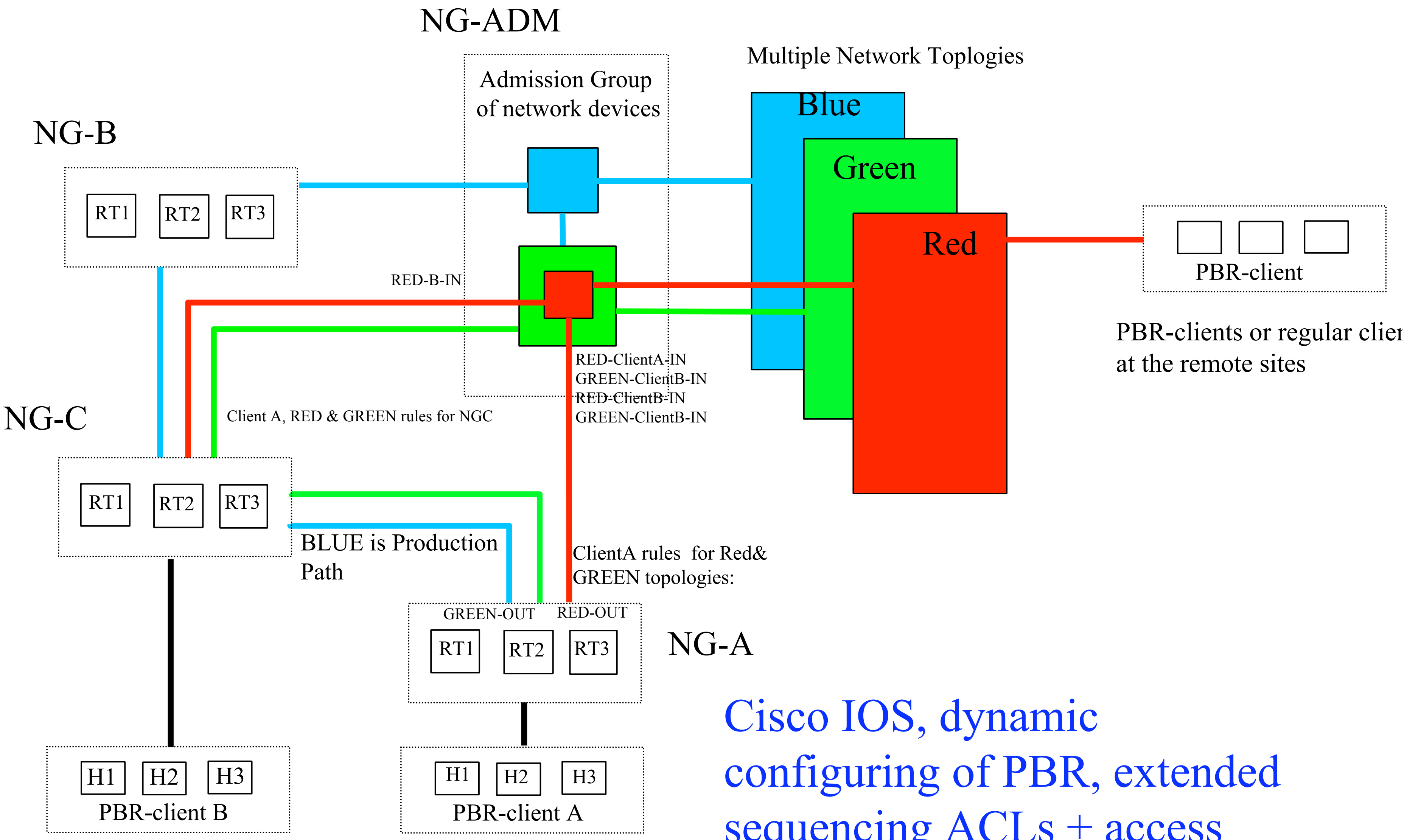
- fixed DSCP values to identify site's traffic.

- DSCP value is assigned dynamically on per ticket base.

Effect of DSCP tagging with IPTables

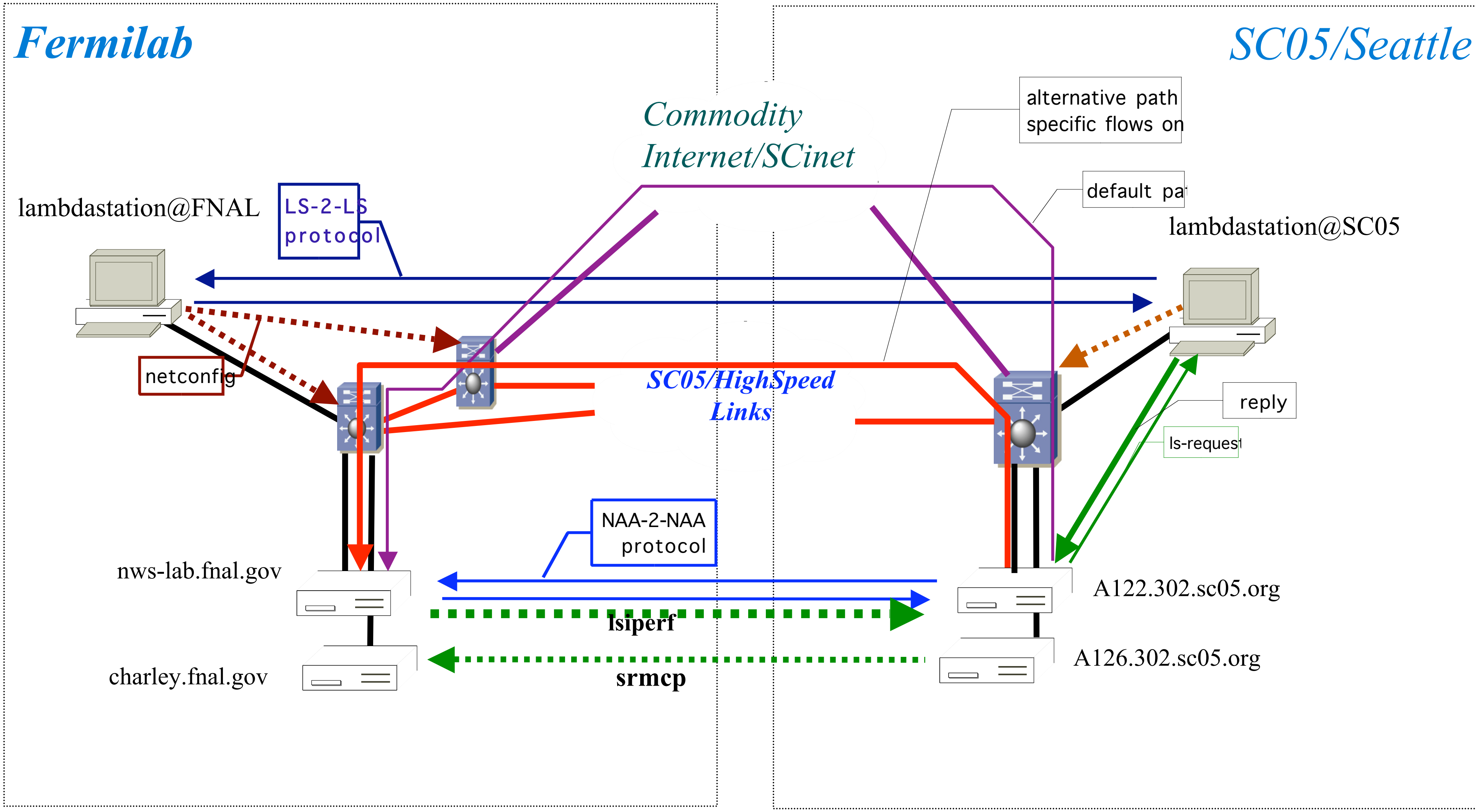


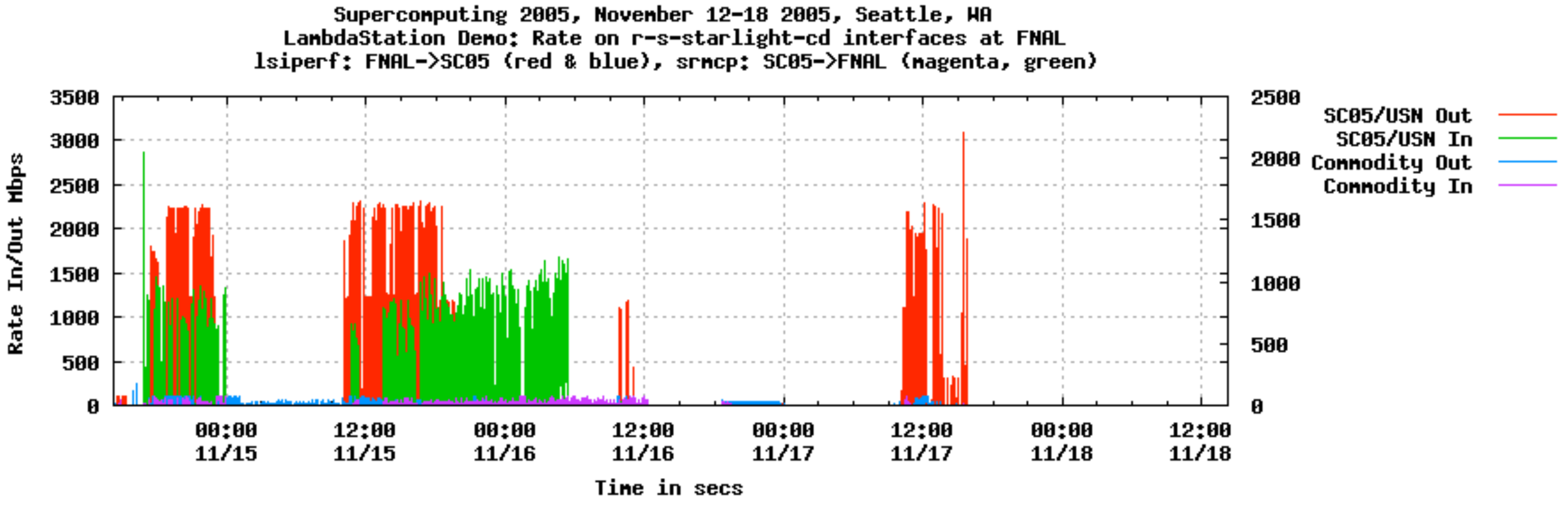
LS multitopology network model



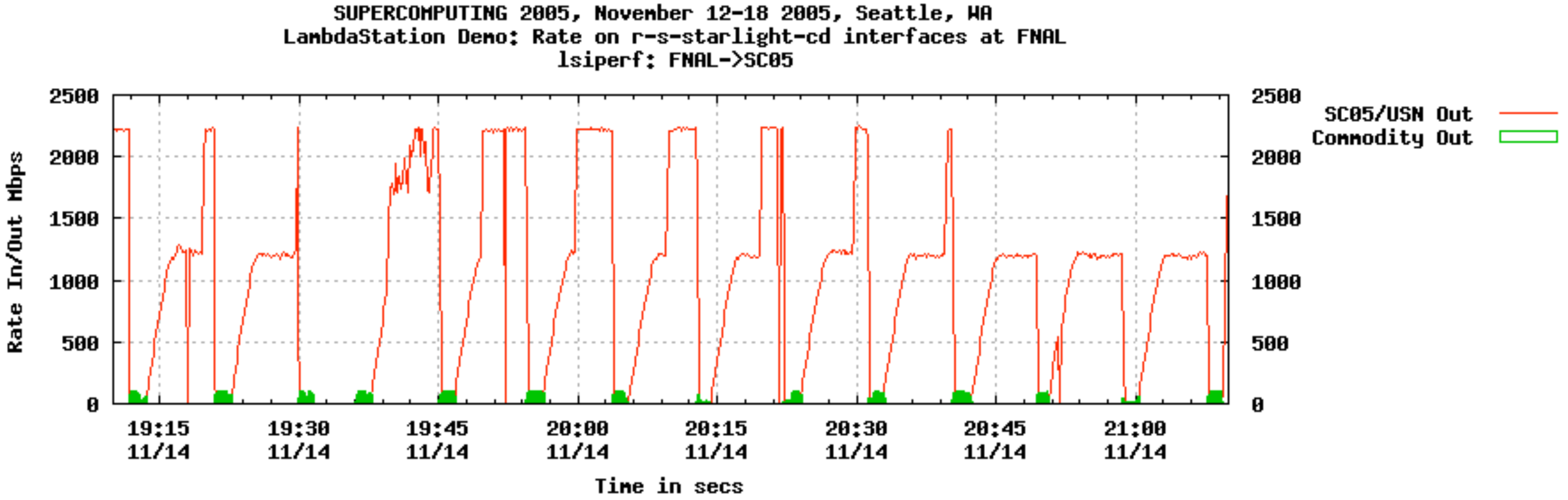
Cisco IOS, dynamic configuring of PBR, extended sequencing ACLs + access policy ACLs

LambdaStation SC05 Demo

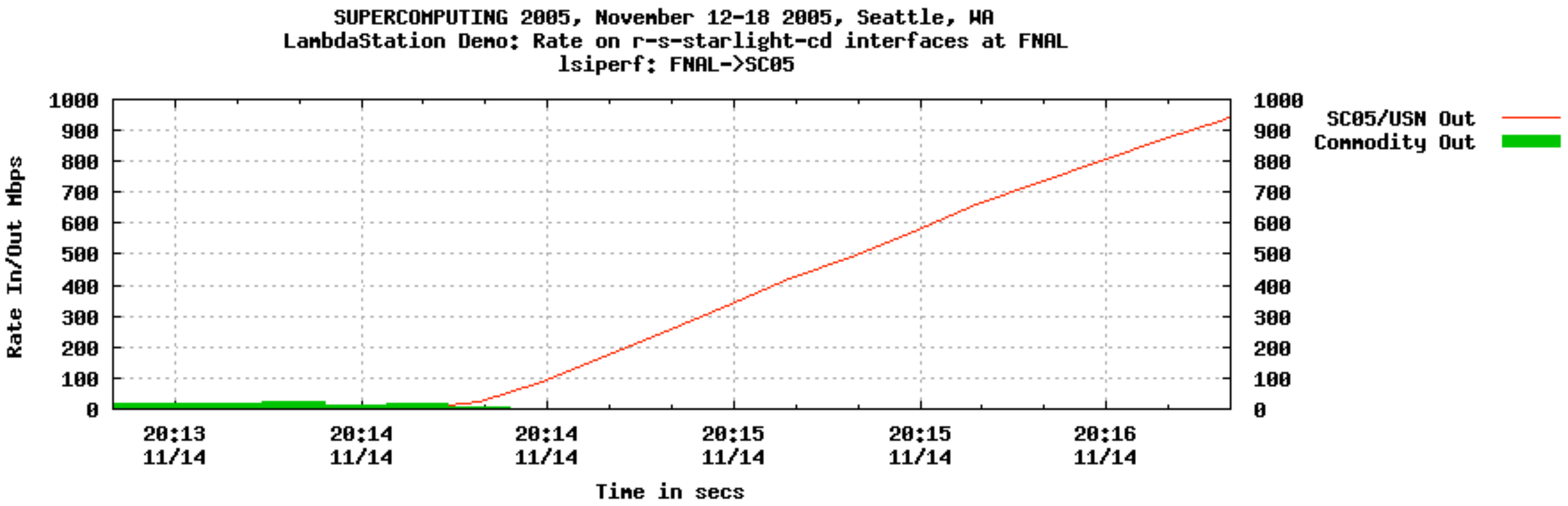


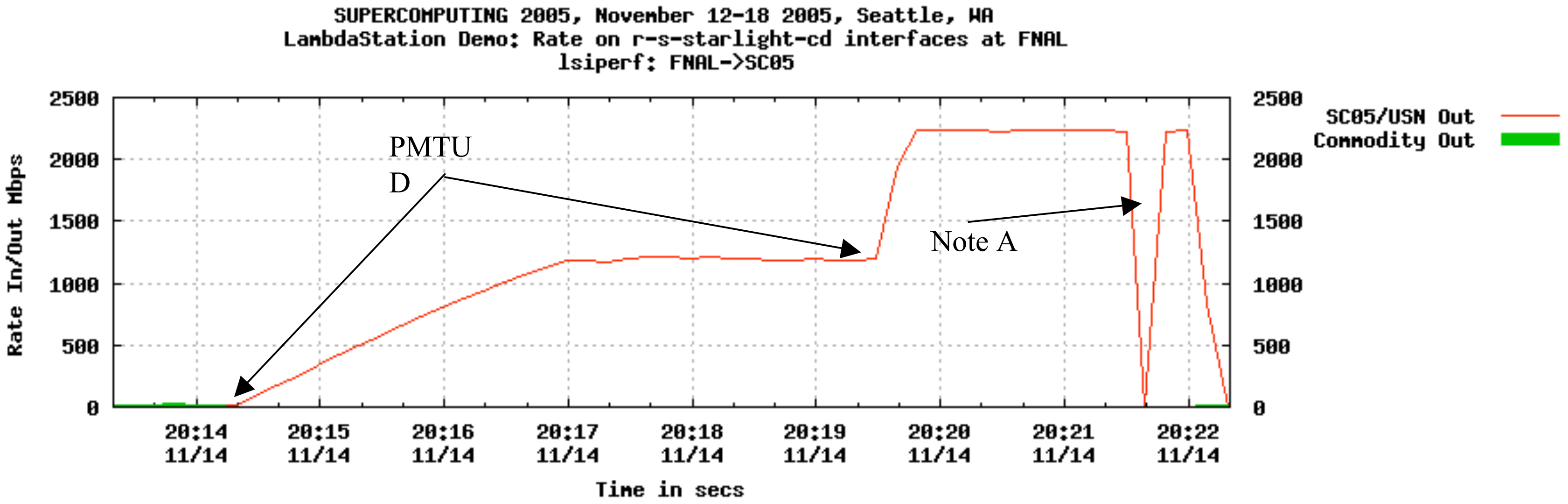


Supercomputing 2005, November 12-18 2005, Seattle, WA
LambdaStation Demo: Rate on r-s-starlight-cd interfaces at FNAL
lsiperf: FNAL->SC05 (red & blue), srmcp: SC05->FNAL (magenta, green)

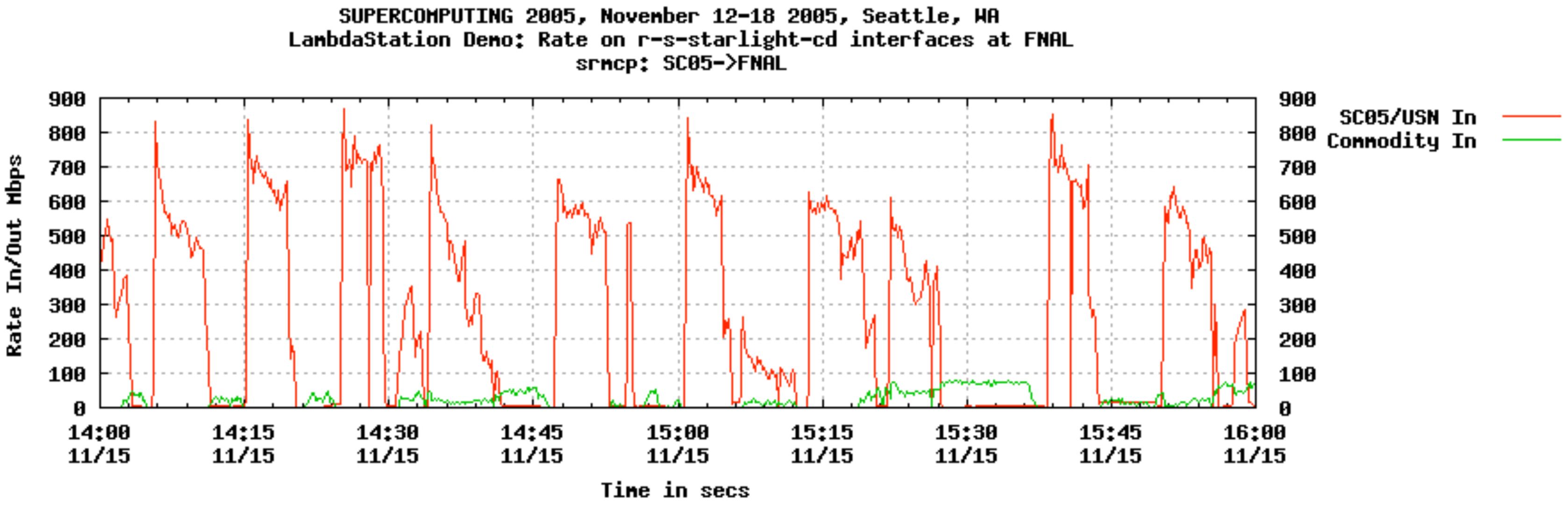


SUPERCOMPUTING 2005, November 12-18 2005, Seattle, WA
LambdaStation Demo: Rate on r-s-starlight-cd interfaces at FNAL
lsiperf: FNAL->SC05

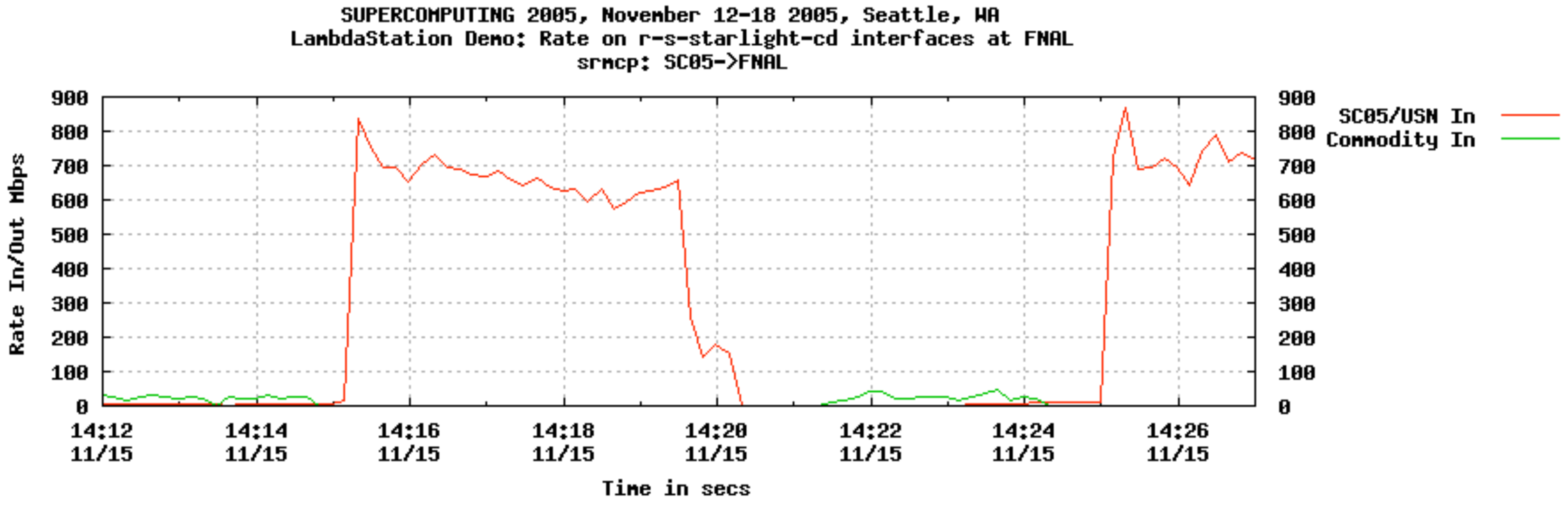

SUPERCOMPUTING 2005, November 12-18 2005, Seattle, WA
LambdaStation Demo: Rate on r-s-starlight-cd interfaces at FNAL
lsiperf: FNAL->SC05


SUPERCOMPUTING 2005, November 12-18 2005, Seattle, WA
LambdaStation Demo: Rate on r-s-starlight-cd interfaces at FNAL
lsiperf: FNAL->SC05

Note A: We believe it is a HW/ASIC problem with SNMP monitoring, a time to time SNMP -get returns the same counters as in previous cycle.



SUPERCOMPUTING 2005, November 12-18 2005, Seattle, WA
LambdaStation Demo: Rate on r-s-starlight-cd interfaces at FNAL
srmcp: SC05->FNAL



SUPERCOMPUTING 2005, November 12-18 2005, Seattle, WA
LambdaStation Demo: Rate on r-s-starlight-cd interfaces at FNAL
srmcp: SC05->FNAL