MAX Mid-Atlantic Crossroads

Cluster B Networking Meeting

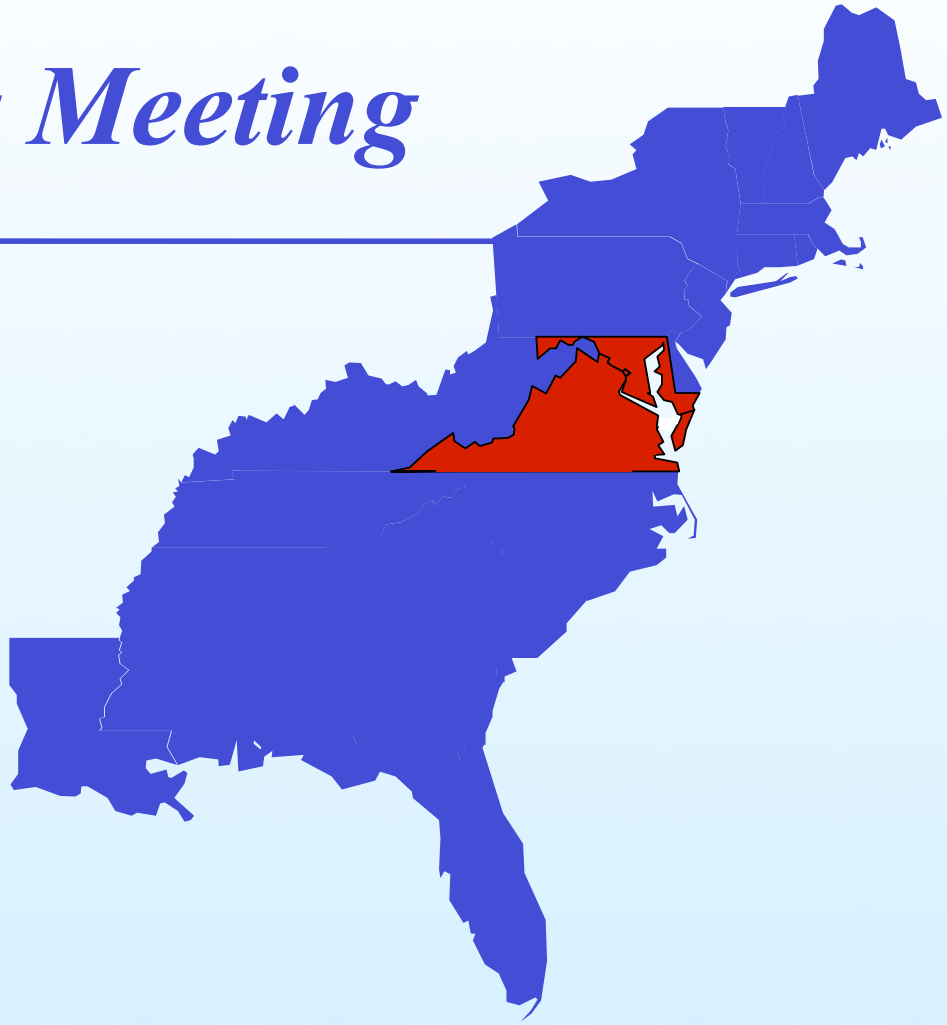
Chris Tracy

Mid-Atlantic Crossroads
(MANFRED)

Cluster B Participant

February 13, 2009

Denver, CO



Overview and Update

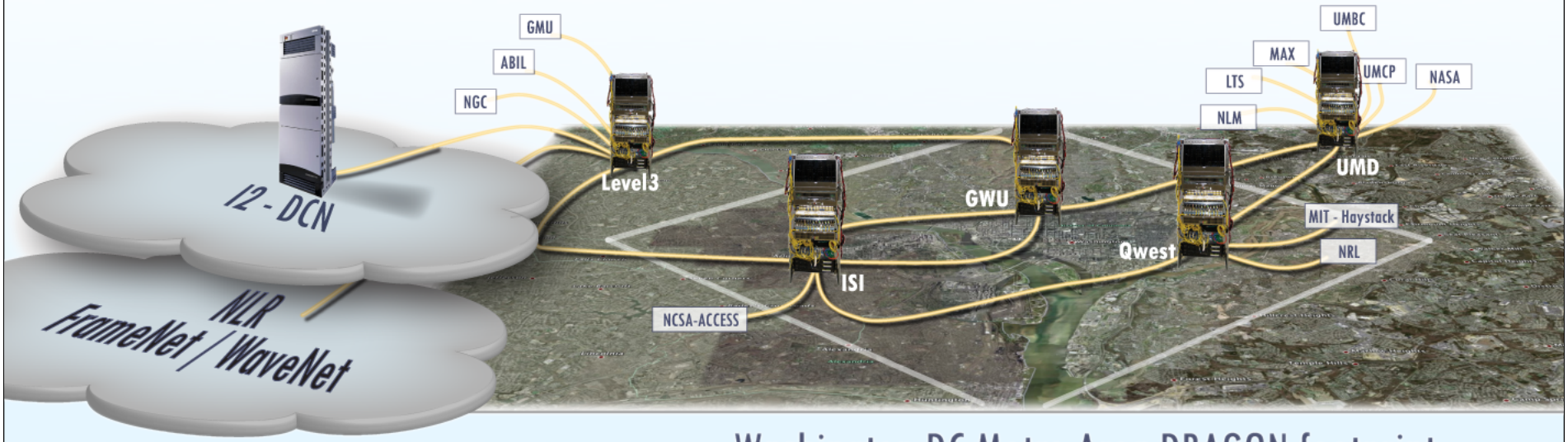
- DRAGON will function as a GENI network aggregate
- We have "sliceable" PlanetLab nodes hosted at 3 sites:
  - University of Maryland (College Park, MD)
  - University of Southern California ISI-East (Arlington, VA)
  - MAX Headquarters (College Park, MD)
    - » MyPLC-native is also hosted here and is functional
- Each PlanetLab node has two GigE NICs
  - eth0 to MAX Layer 3 w/ public IP address
  - eth1 to dynamic DRAGON Layer 2 infrastructure **
- Many other non-PlanetLab hosts connected similarly
  - file servers, HDTV capture/display PCs, UML/Xen VMs

Aggregate/Component Manager (AM/CM)

- DRAGON network has running code which will act as the internal AM/CM for *dynamic network resources*
  - open-source GMPLS stack (under development since 2004)
  - can provision Ethernet VLANs, SONET circuits, lambdas
  - provisioning interfaces consist of:
    - » Web-based User Interface
    - » Web Services API (expects a signed SOAP message)
    - » Text-based User CLI (either via telnet or command-line tool)
    - » GMPLS API (using RSVP-TE signaling and OSPF-TE for routing)
- Capabilities are analogous to what PlanetLab does for compute resources
  - PlanetLab can “slice” PCs, DRAGON can slice networks...

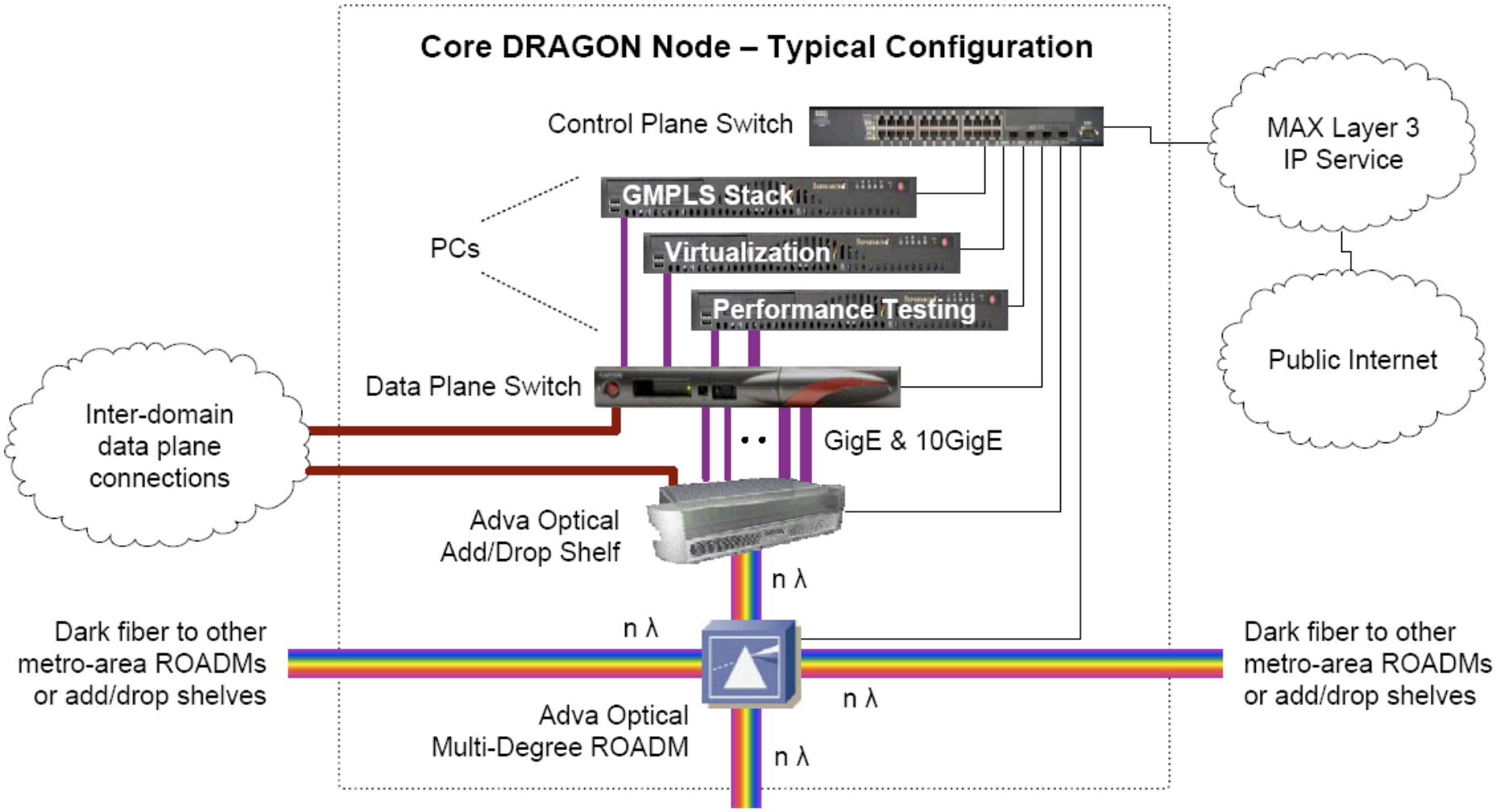
MAX Mid-Atlantic Crossroads

The DRAGON Testbed



Washington DC Metro Area DRAGON footprint

- Over 150 miles of dark fiber
- 5 multi-degree ROADMs (four 4-degree, one 3-degree)
- 12 OADMs (up to 40 channels, some transponders are tunable)
- 10 Ethernet switches (10GigE, GigE)
- Lambdas and Ethernet VLANs provisioned exclusively using GMPLS
- Interconnects to national backbones and many regional campuses
- Control PCs, performance and virtualization nodes, compute clusters


Core DRAGON Node – Typical Configuration
Control Plane Switch
MAX Layer 3
IP Service
GMPLS Stack
PCs
Virtualization
Performance Testing
Public Internet
Data Plane Switch
Inter-domain
data plane
connections
GigE & 10GigE
Adva Optical
Add/Drop Shelf
n λ
n λ
Dark fiber to other
metro-area ROADMs
or add/drop shelves
n λ
Dark fiber to other
metro-area ROADMs
or add/drop shelves
Adva Optical
Multi-Degree ROADM
n λ

Typical DRAGON core node

Adva Optical Multi-degree ROADM →

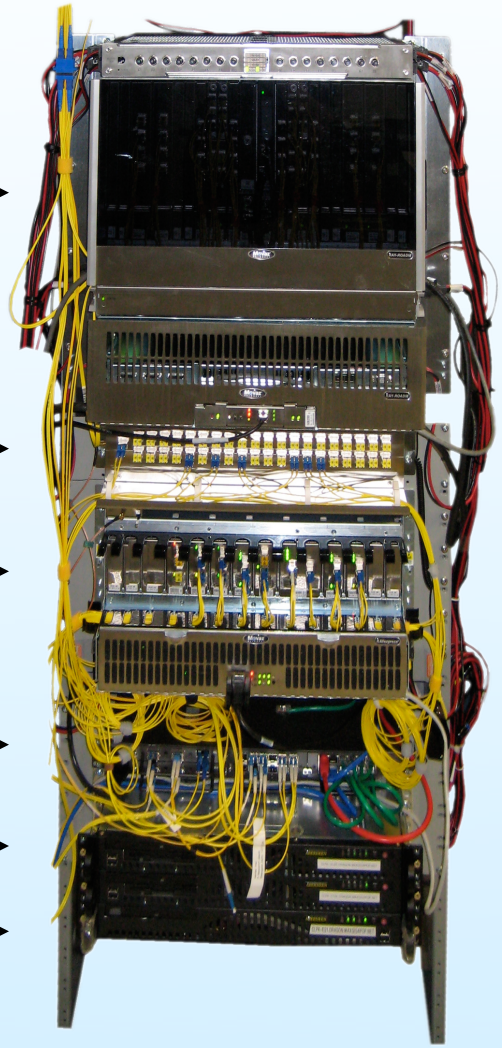
Adva Optical 40 λ mux/demux →

Adva Optical RayExpressII OADM →

Raptor ER-1010 Ethernet switch →

Virtual Label Switching Router (VLSR) →

Perf PCs, virtualization nodes, etc. →



SFA and RSpec Concerns

- Compute resources versus non-compute resources
  - current slice operations seem centered around compute resources
  - support for network resources seems limited to IP interfaces
    - » interface bandwidth/address (max_kbyte/min_rate/max_rate/addr)
    - » seems to assume all interfaces will be carrying IP traffic
  - Doesn't contain elements to express complex network topology
    - » switching capability
    - » encoding
    - » interface-specific switching capability descriptors (e.g. VLAN IDs)
  - How should network resources be represented/provisioned?
    - » Since 2006, DICE and OGF (NMWG) have developed international standard for expressing complex network topology (http://controlplane.net)

MyPLC Concerns

- Level of effort to adapt MyPLC to work with Xen/etc
  - How tightly coupled is PlanetLab to:
    - » Linux VServers?
      - * What if we wanted to use Xen, OpenVZ or UML ?
    - » CentOS / RedHat / Fedora?
      - * We would prefer to use Debian, but MyPLC assumes CentOS/Fedora
- XML-RPC vs Web Services (SOAP)
  - Our preference is to use signed SOAP messages over SSL
  - Can admittedly be more complex, but allows for user-defined complex data types
    - » this was a significant feature for us in terms of developing inter-domain resource scheduling, signaling, monitoring, etc.
  - Python ZSI (Zolera Soap Infrastructure) works well

VServer Concerns

- Limitations associated with Linux VServer
  - using raw sockets requires source modification/recompilation
    - » must bind to TCP or UDP port to claim ownership before reading
    - » slices share a single public IP address, so what happens if two slices on the same physical host want to bind a raw socket for other IP protocols?
    - » e.g. RSVP (IP proto 46) or OSPF (IP proto 89) (Quagga/dragon-sw/etc)
    - » PL-VINI may have solved some of these issues with virtual topologies?
  - "networkability" (as compared with VDE2, for example)

VServer Concerns

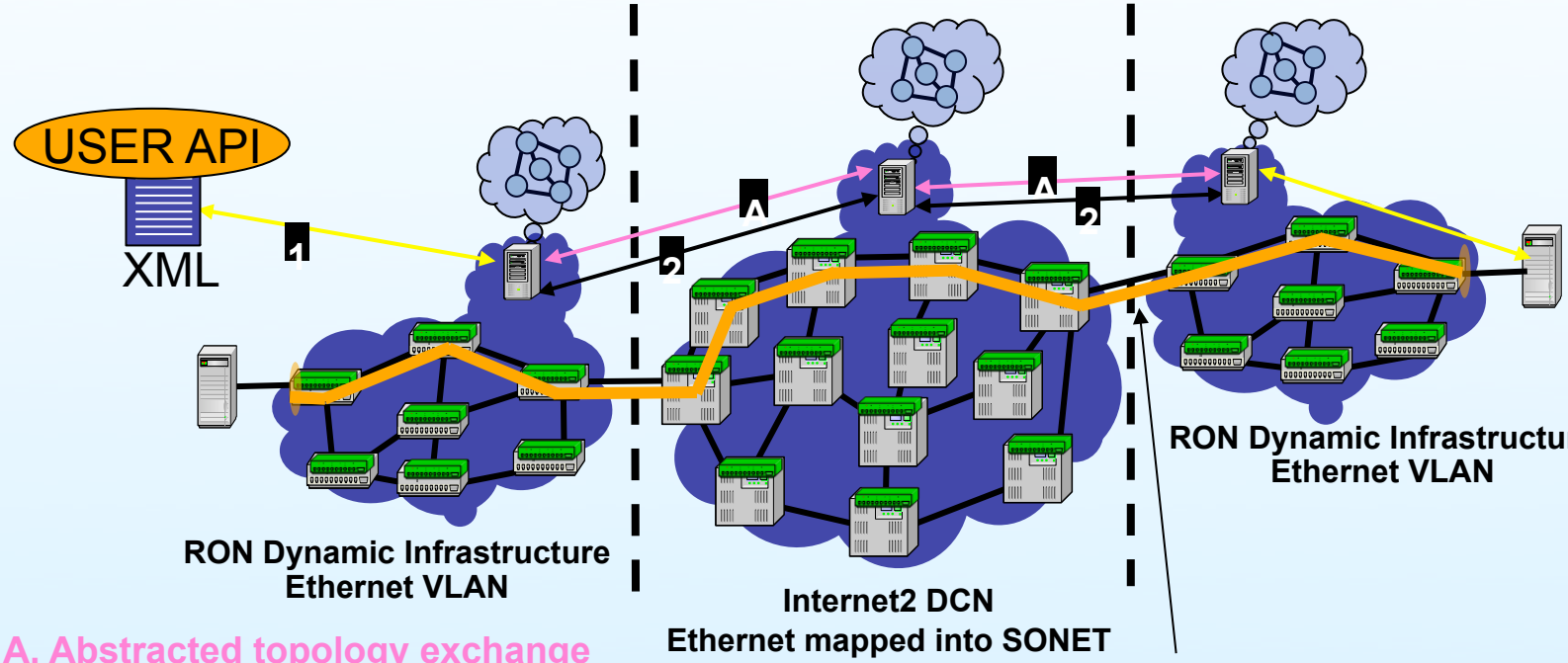
- Tagged 802.1Q VLAN interfaces
  - Glad to see that Linux VServers do support tagged VLAN logical sub-interfaces (e.g. eth1.2001)
  - Would like to see the PlanetLab API extended to support them
    - » support creation of any number of tagged VLAN interfaces when a VServer is instantiated or "booted"
    - » allowing for dynamic reconfiguration of tagged interfaces on running slices would be even better, in our opinion
  - Can "manually" add tagged sub-int to PLC-instantiated VServer
    - » add tagged VLAN in root context using *vconfig*
    - » manipulate /etc/vservers/[...]/interfaces/ directory, restart vserver
  - Limitation: uses IP isolation, so a particular VLAN ID can only be used on one slice at a time

Other Concerns

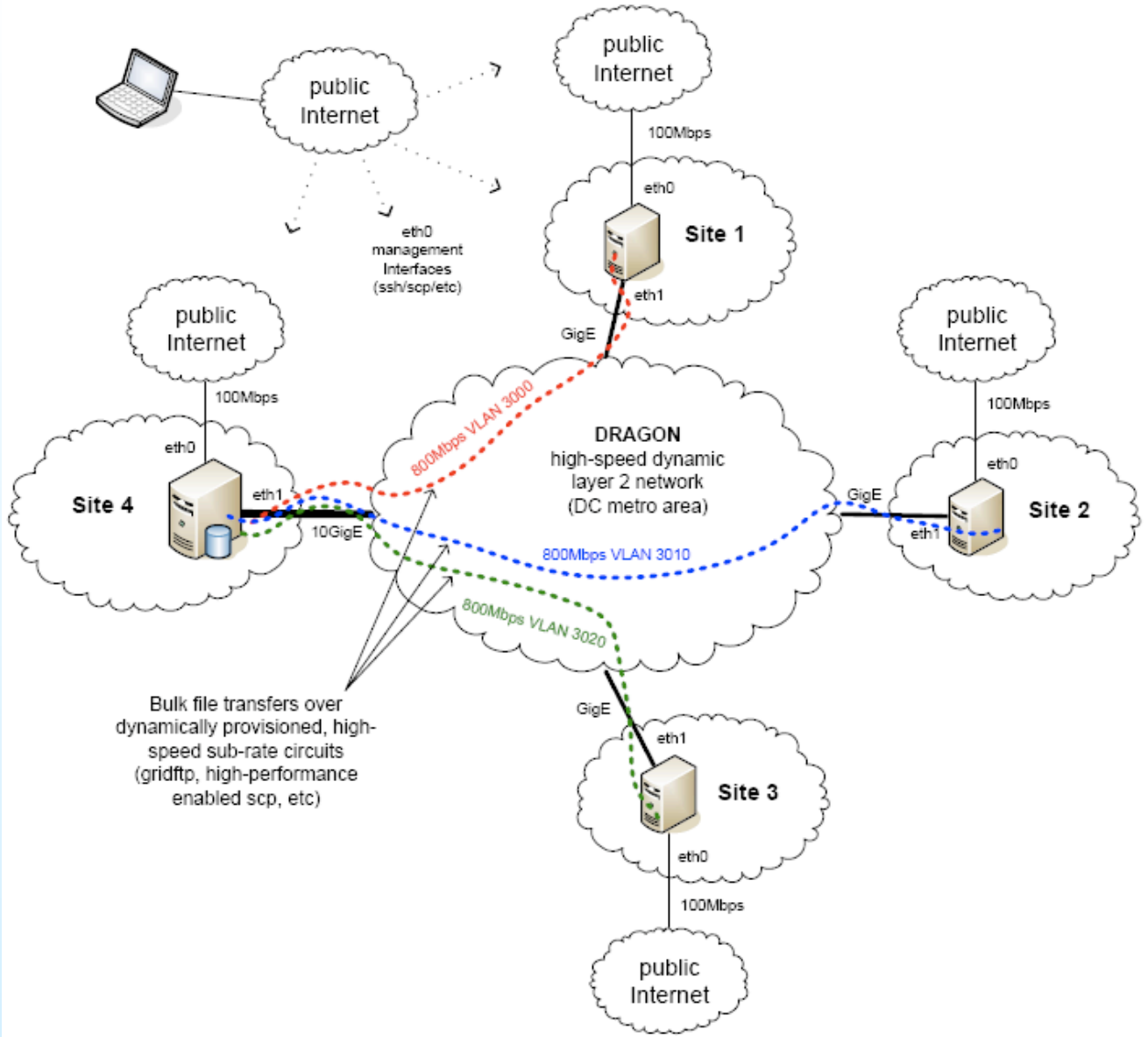
- We prefer to run UML, OpenVZ, or Xen under Debian
  - easy to simulate large, complex networks on a few PCs
    - » GMPLS routers, Ethernet switches, end hosts, etc
  - using *netem*, can simulate real-world WAN conditions
    - » delay, packet loss, duplication, re-ordering, rate control, etc
  - using *vde*, can emulate arbitrary network connections
    - » essentially provides a "patch-cable" like mechanism between VMs
  - virtual machines can be setup to act as Ethernet bridges
    - » brconfig, ebtables, vde_switch
    - » this does not appear to be possible with Linux VServers?

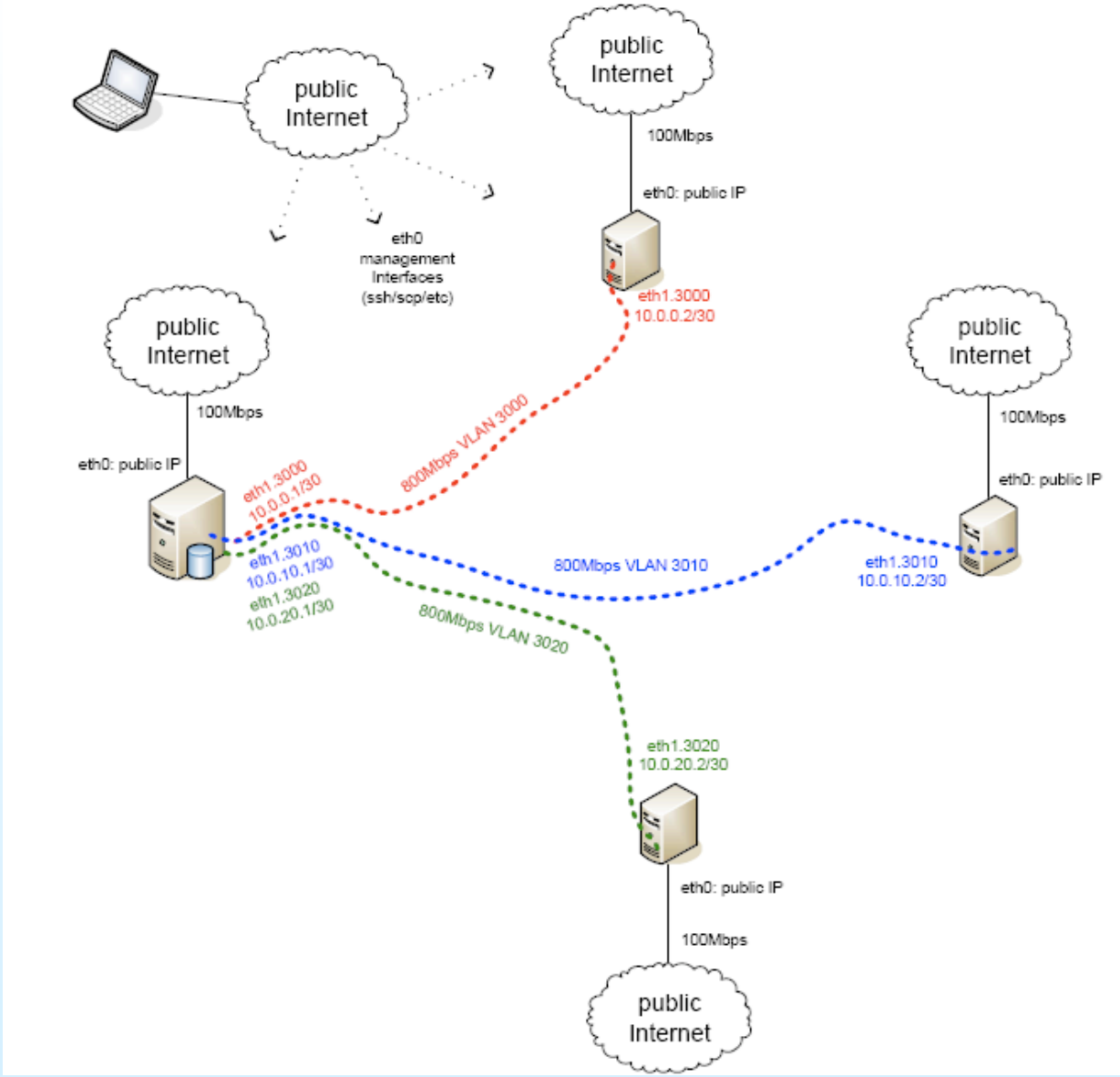
Backup Slides

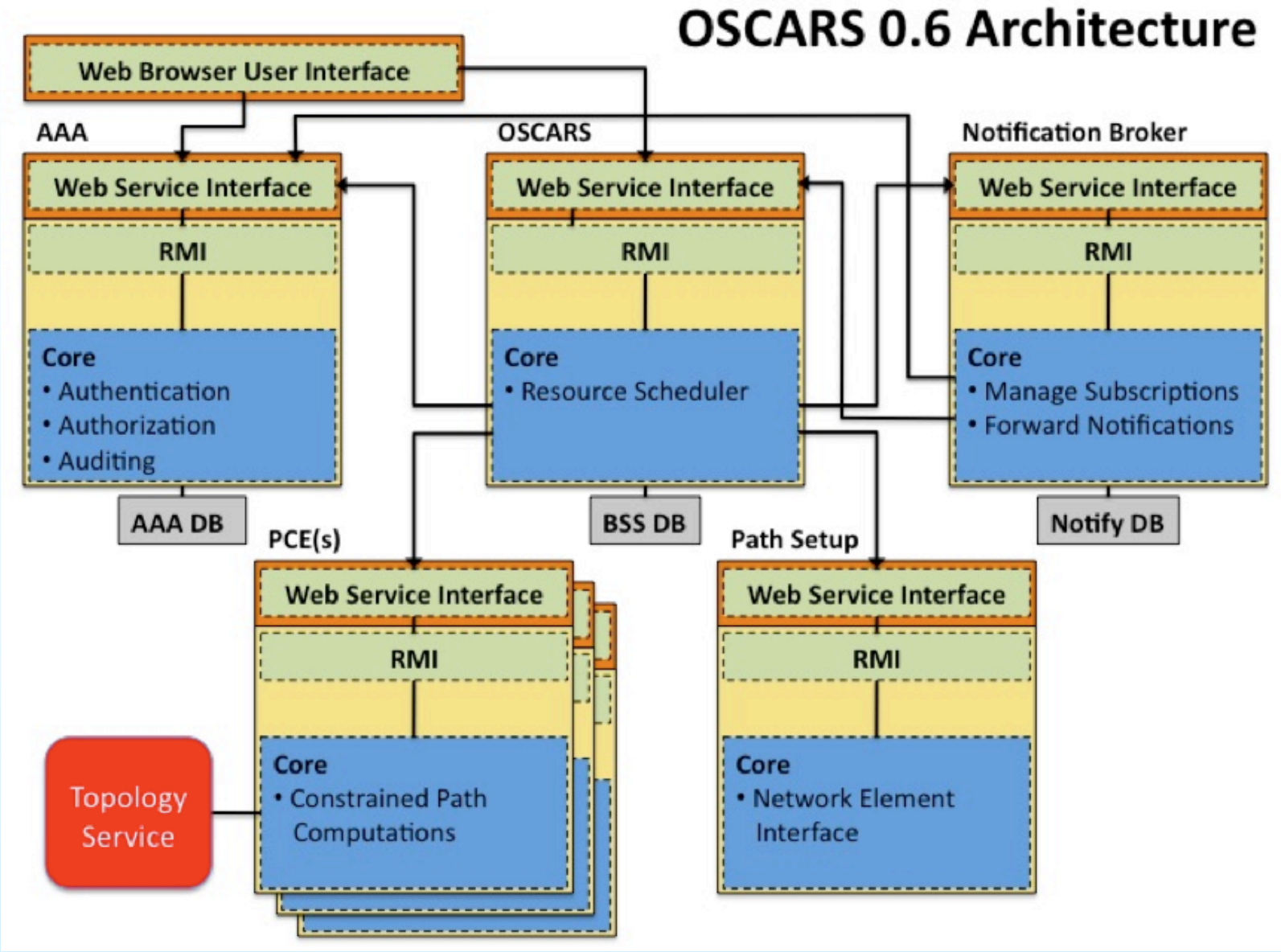
- No difference from a client (user) perspective for InterDomain vs IntraDomain



A. Abstracted topology exchange
1. Client Service Request
2. Resource Scheduling
5. Service Instantiation (as a result of Signaling)


public Internet
public Internet
100Mbps
eth0
Site 1
eth0 management Interfaces (ssh/scp/etc)
eth1
GigE
public Internet
100Mbps
eth0
Site 4
eth1
10GigE
800Mbps VLAN 3000
DRAGON
high-speed dynamic layer 2 network
(DC metro area)
800Mbps VLAN 3010
800Mbps VLAN 3020
public Internet
100Mbps
eth0
GigE
Site 2
eth1
Bulk file transfers over dynamically provisioned, high-speed sub-rate circuits (gridftp, high-performance enabled scp, etc)
GigE
eth1
Site 3
eth0
100Mbps
public Internet


public Internet
public Internet
100Mbps
eth0: public IP
eth0 management Interfaces (ssh/scp/etc)
eth1.3000 10.0.0.2/30
public Internet
100Mbps
eth0: public IP
eth1.3000 10.0.0.1/30
800Mbps VLAN 3000
eth1.3010 10.0.10.1/30
eth1.3020 10.0.20.1/30
800Mbps VLAN 3010
800Mbps VLAN 3020
public Internet
100Mbps
eth0: public IP
eth1.3010 10.0.10.2/30
eth1.3020 10.0.20.2/30
eth0: public IP
100Mbps
public Internet


OSCARS 0.6 Architecture
Web Browser User Interface
AAA
OSCARS
Notification Broker
Web Service Interface
RMI
Core
• Authentication
• Authorization
• Auditing
Core
• Resource Scheduler
Core
• Manage Subscriptions
• Forward Notifications
AAA DB
BSS DB
Notify DB
PCE(s)
Path Setup
Topology Service
Core
• Constrained Path Computations
Core
• Network Element Interface