

Network Abstractions: The first step towards a programmable WAN

Inder Monga

TIP 2013

January 15th, 2013



Ongoing Science Revolution



Data-intensive Science

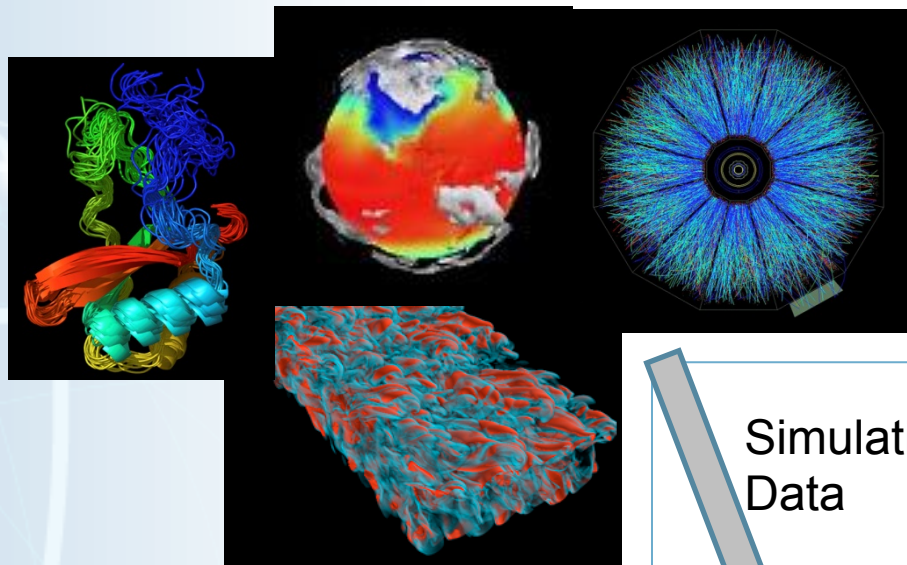
- Era of 'Big-Data'

Exascale, HPC and Future Data Centers

- Optics to the end

Data Integration

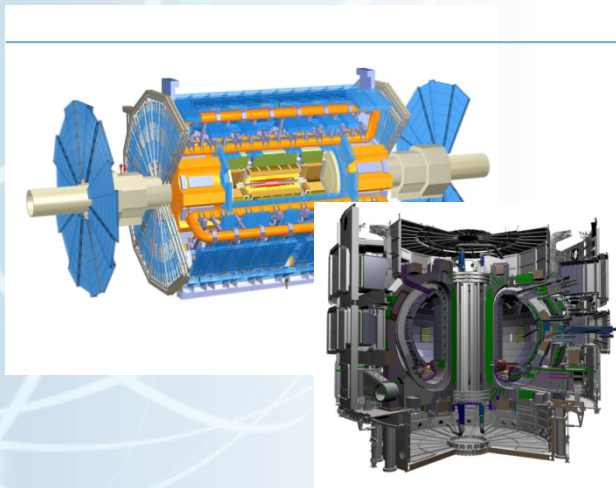
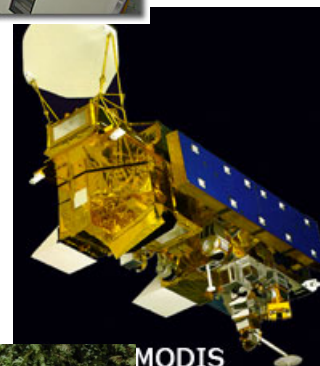
Bringing together 'big-data' from various sources



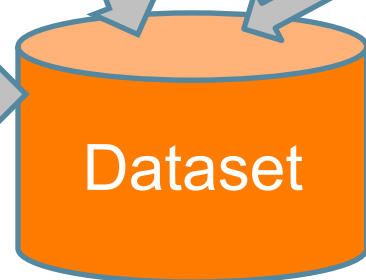
Simulation Data



Observation Data

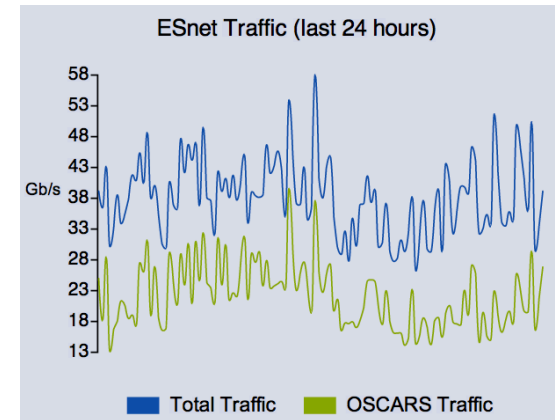
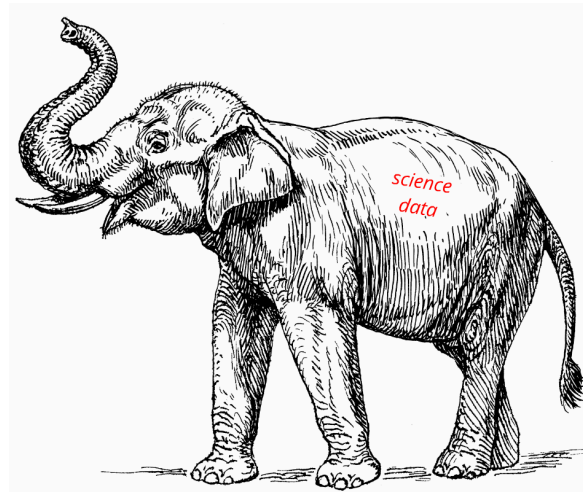
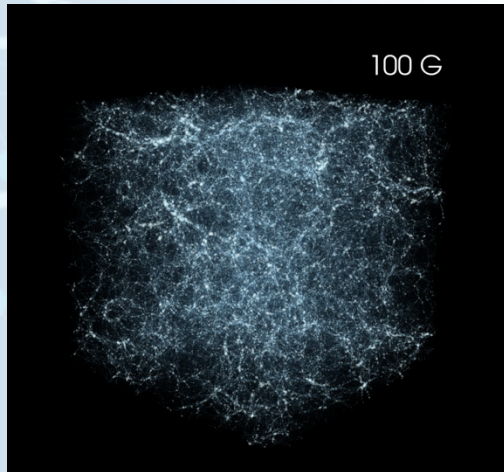


Experiment Data



Source: Kathy Yelick

Distributed Data, consolidated computation



Elephant Flows: 'big-data' movement for Science, end-to-end

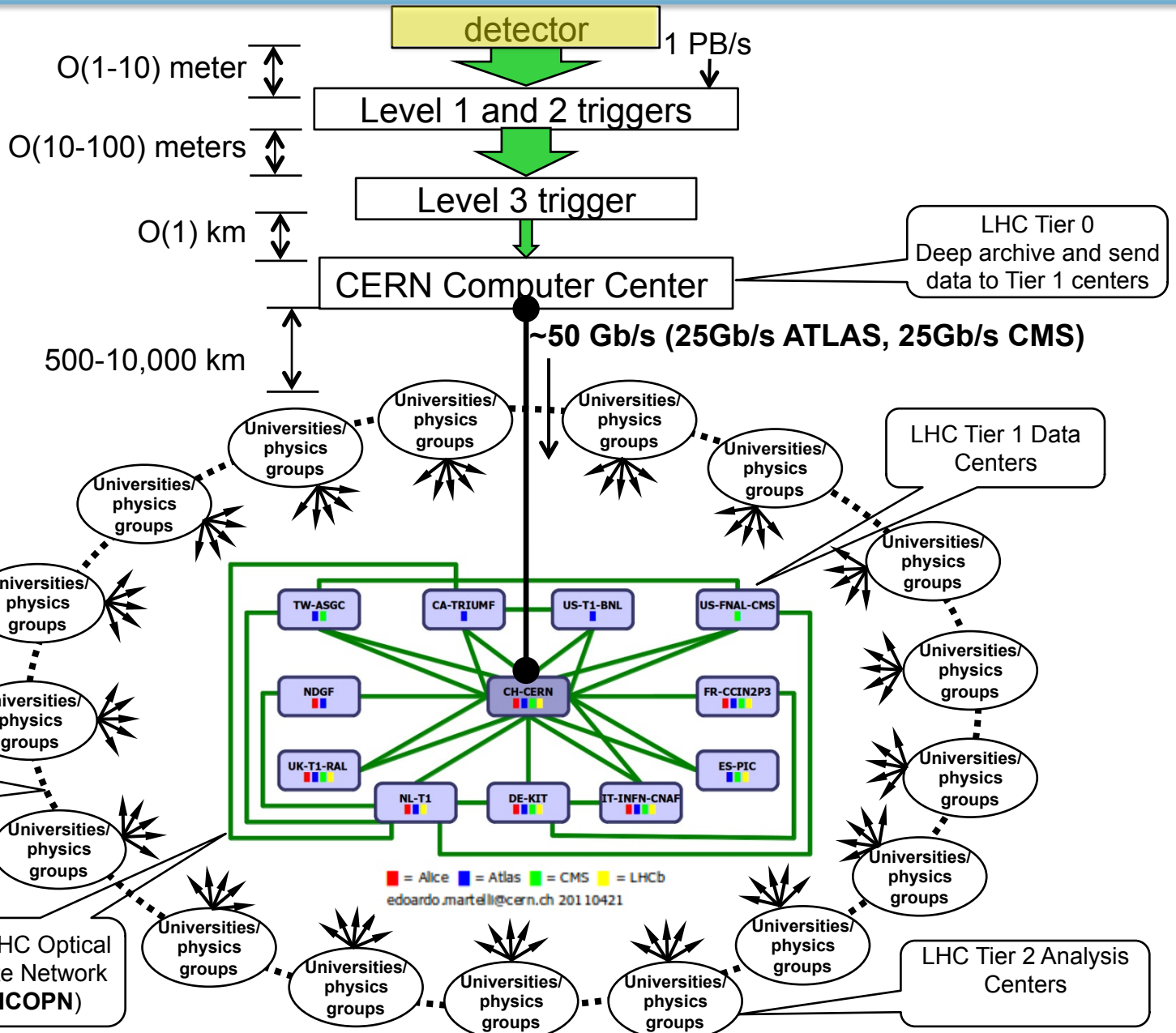
Network "weather" conditions may vary

- Long latencies (RTT), Multi-domain, Multi-vendor, Multi-technology

Packet loss is the "enemy" of high-throughput

Large Instruments, Complex Simulations, Global Collaborations

CERN → T1	mile s	kms
France	350	565
Italy	570	920
UK	625	1000
Netherlands	625	1000
Germany	700	1185
Spain	850	1400
Nordic	1300	2100
USA – New York	3900	6300
USA - Chicago	4400	7100
Canada – BC	5200	8400
Taiwan	6100	9850



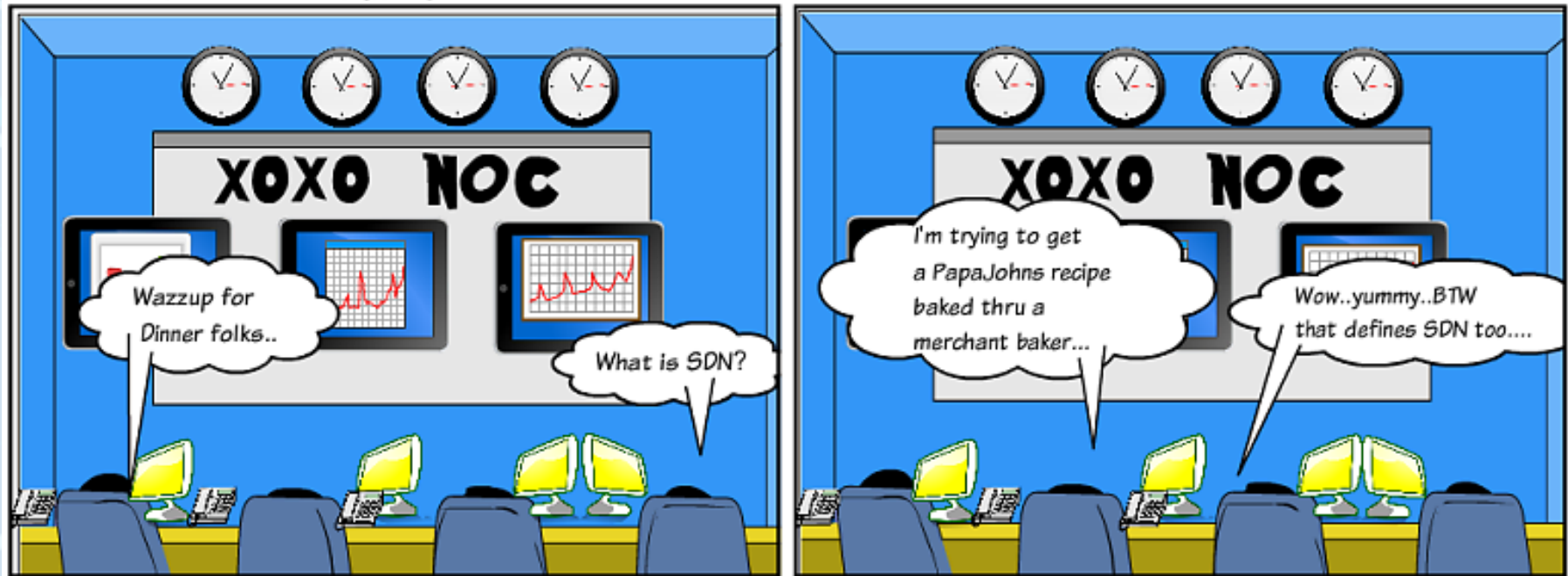
Source: Bill Johnston

Software-Defined Networking and Network Functions Virtualization....



SDN AND PIZZA - BY VRAJ09

WWW.TOONDOO.COM



What is Software-Defined Networking?

(as defined by Scott Shenker, October 2011)

<http://opennetsummit.org/talks/shenker-tue.pdf>



“The ability to master complexity is not the same as the ability to extract simplicity”

“Abstractions key to extracting simplicity”

“SDN is defined precisely by these **three abstractions**

- Distribution, forwarding, configuration “

Fundamental Network Abstraction: a end-to-end circuit



Wavelength, PPP, MPLS, L2TP, GRE, NSI-CS...



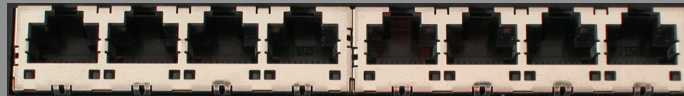
Switching points, store and forward, transformation ...

Simple, Point-to-point, Provisionable

A new WAN Abstraction: “WAN Virtual Switch”



WAN Virtual Switch



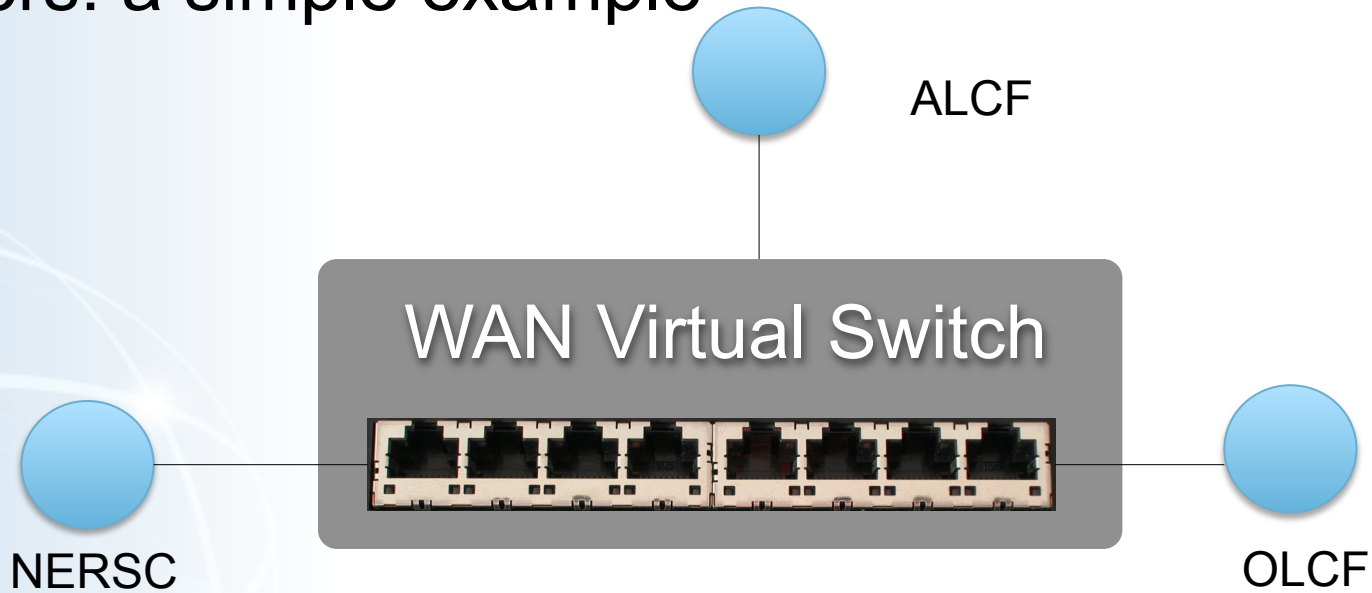
Simple, Multipoint, Programmable

Configuration abstraction:

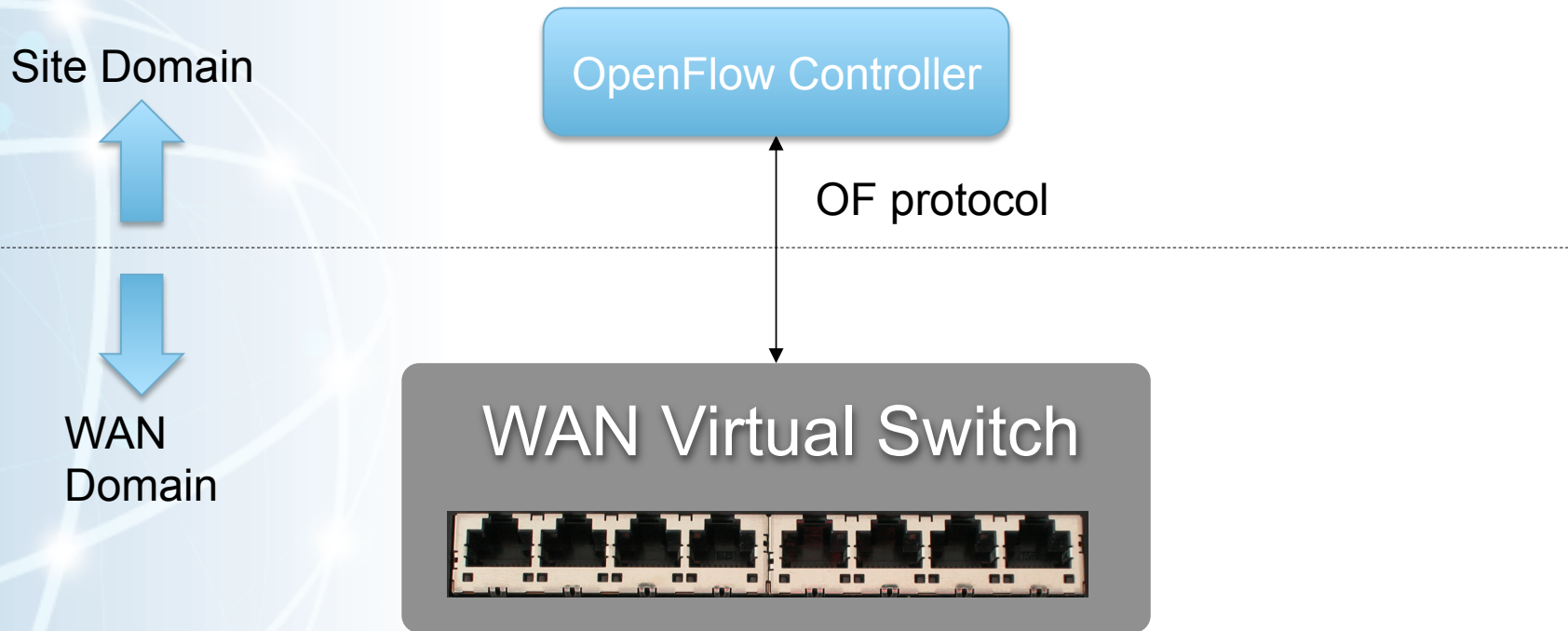
- Expresses desired behavior
- Hides implementation on physical infrastructure

It is not only about the concept, but implementation

One Virtual Switch connecting the SC centers: a simple example



Programmability



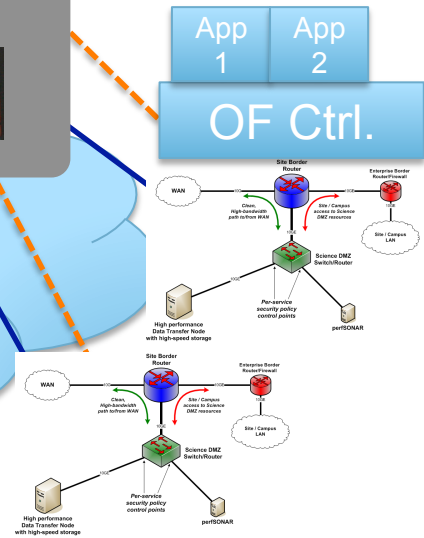
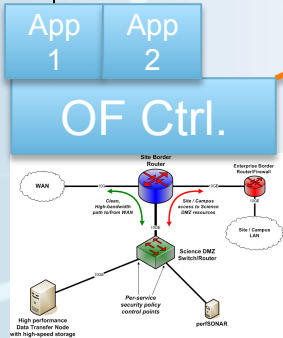
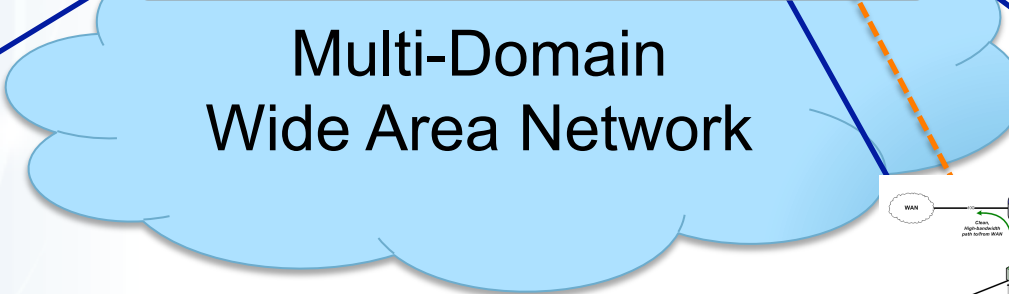
Expose 'flow' programming interface leveraging standard OF protocol

“Flow Programmability” by end-sites

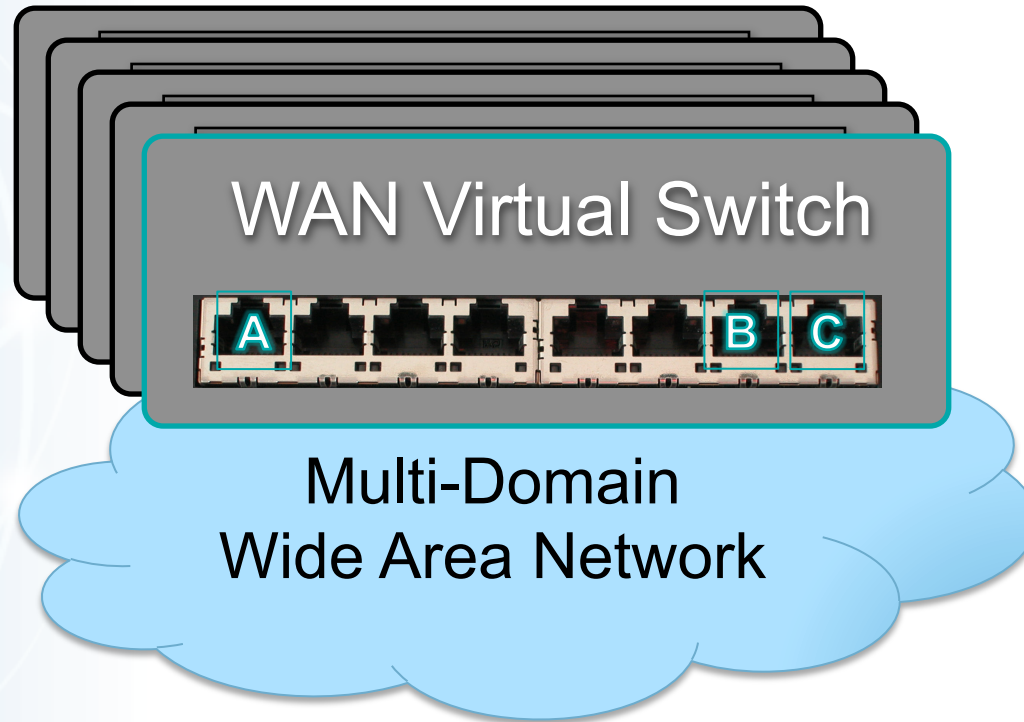
Science Flow1: A → B, QoS, Label

Science Flow2: A → C, VLAN

Science Flow3: A → B,C

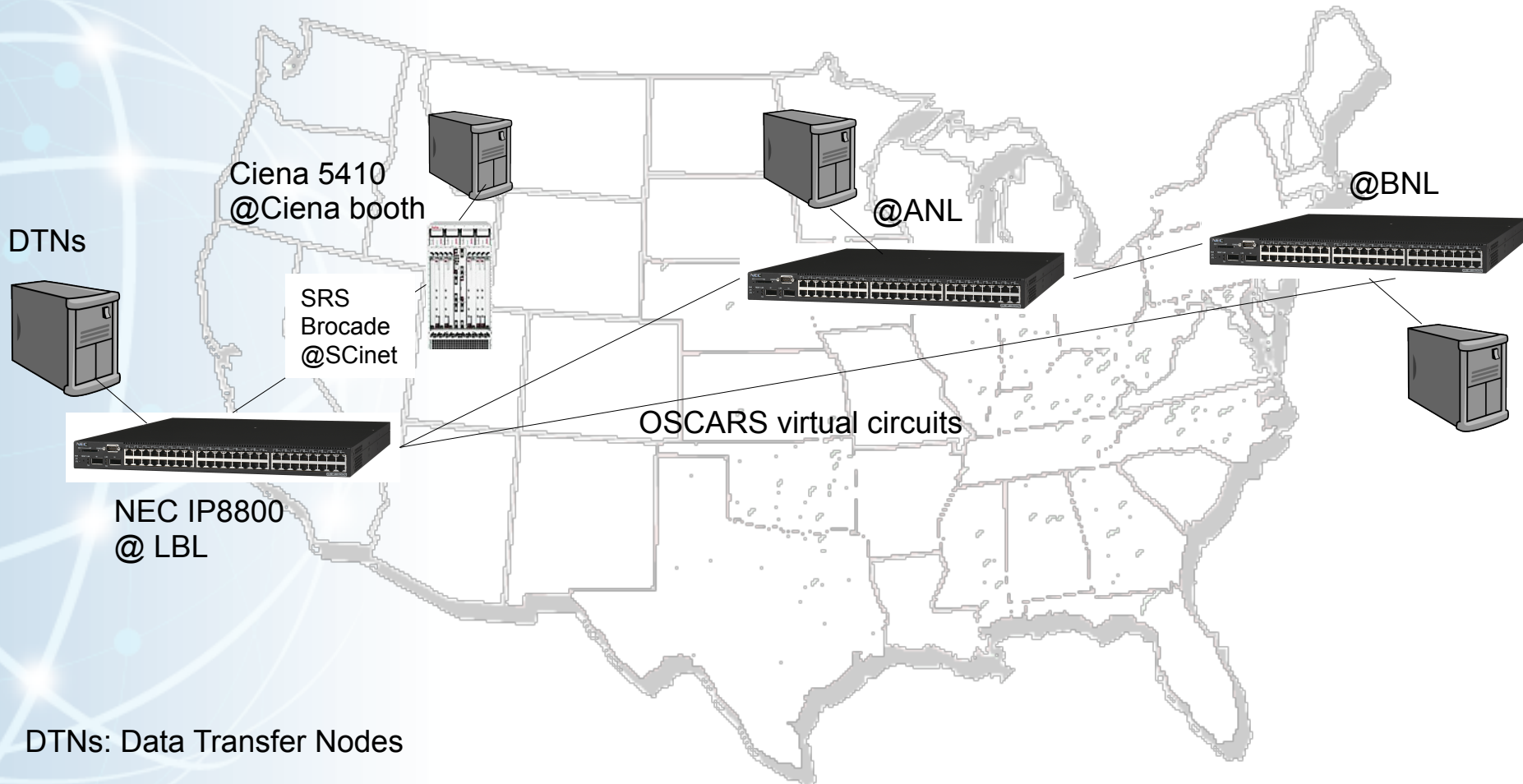


Collaborations, Virtual Switches and Multi-domain: A many-many relationship



SRS Demonstration Physical Topology

ciena



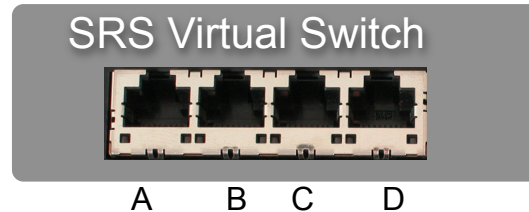
DTNs: Data Transfer Nodes

Virtual Switch Implementation: Mapping abstract model to the physical



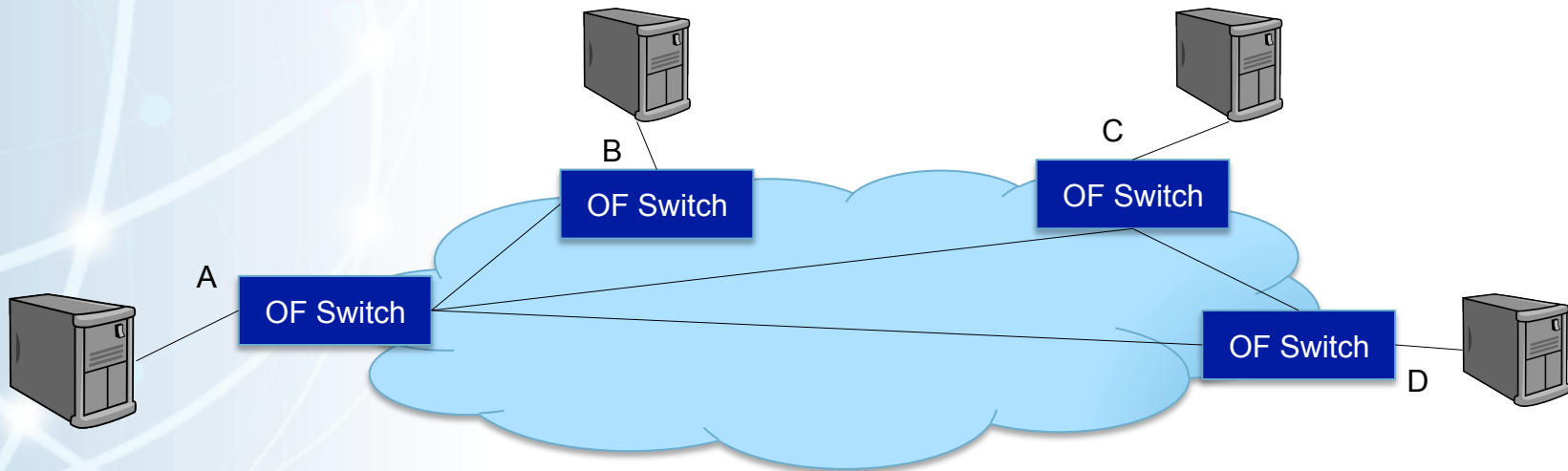
Create Virtual switch:

- Specify edge OF ports
- Specify backplane topology and bandwidth
- Policy constraints like flowspace
- Store the switch into a topology service



Virtual

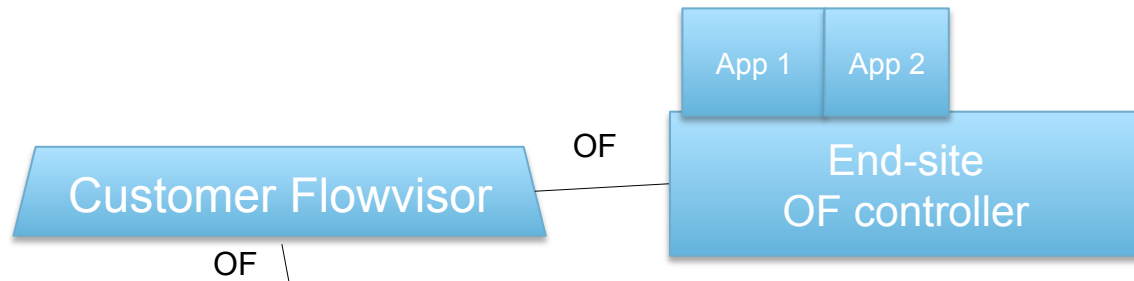
Physical



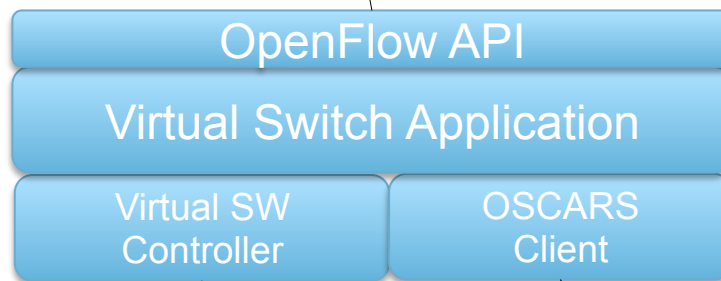
WAN Virtual Switch: Implementation architecture



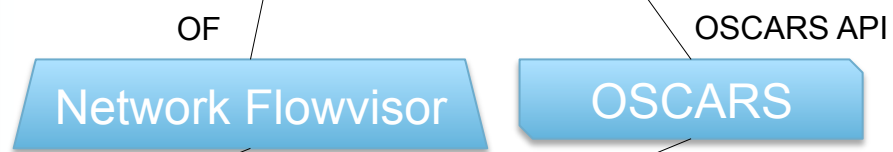
Policy/Isolation of customer OF control



Virtual Switch Software stack



Infrastructure Software, Slicing and provisioning

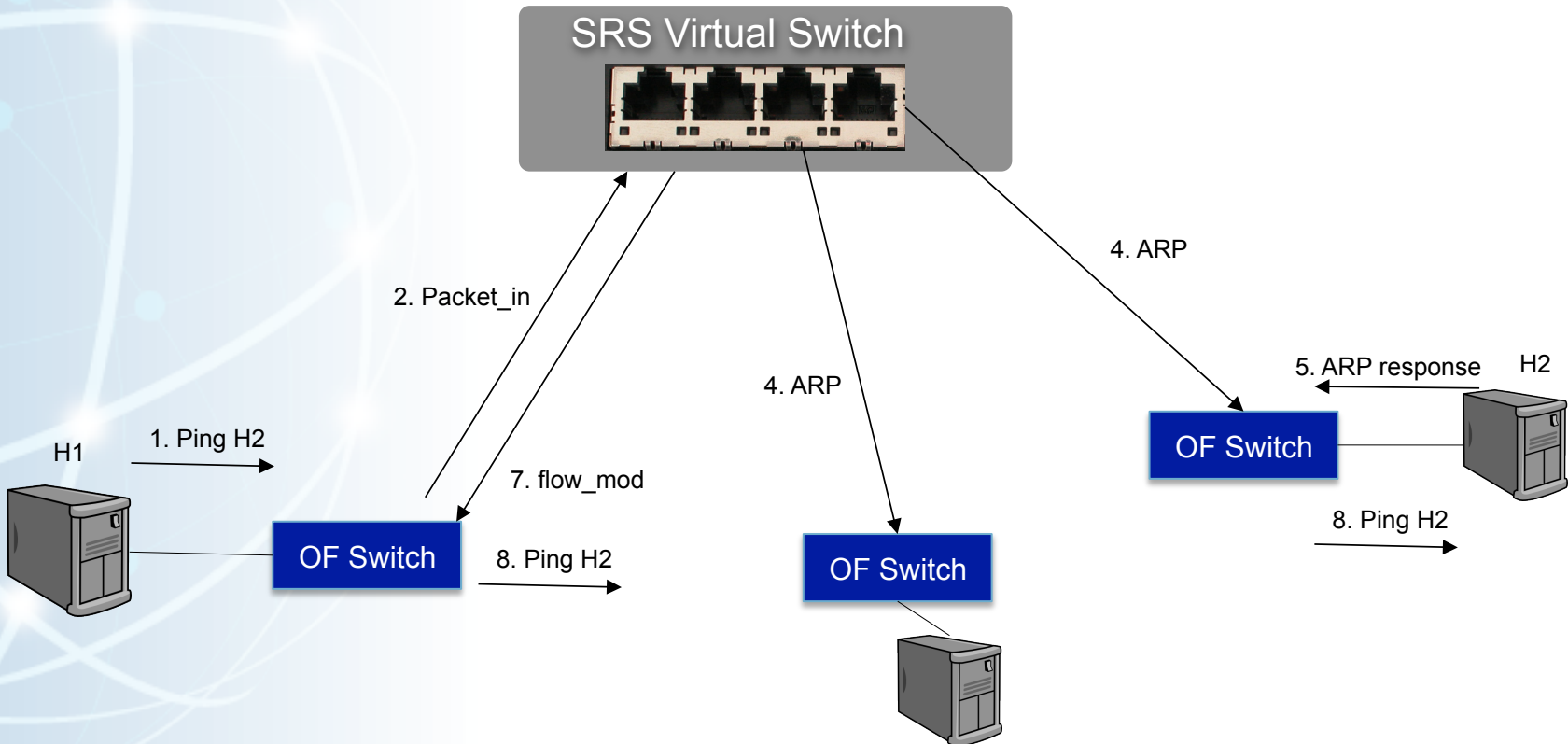


Mandatory Ping example

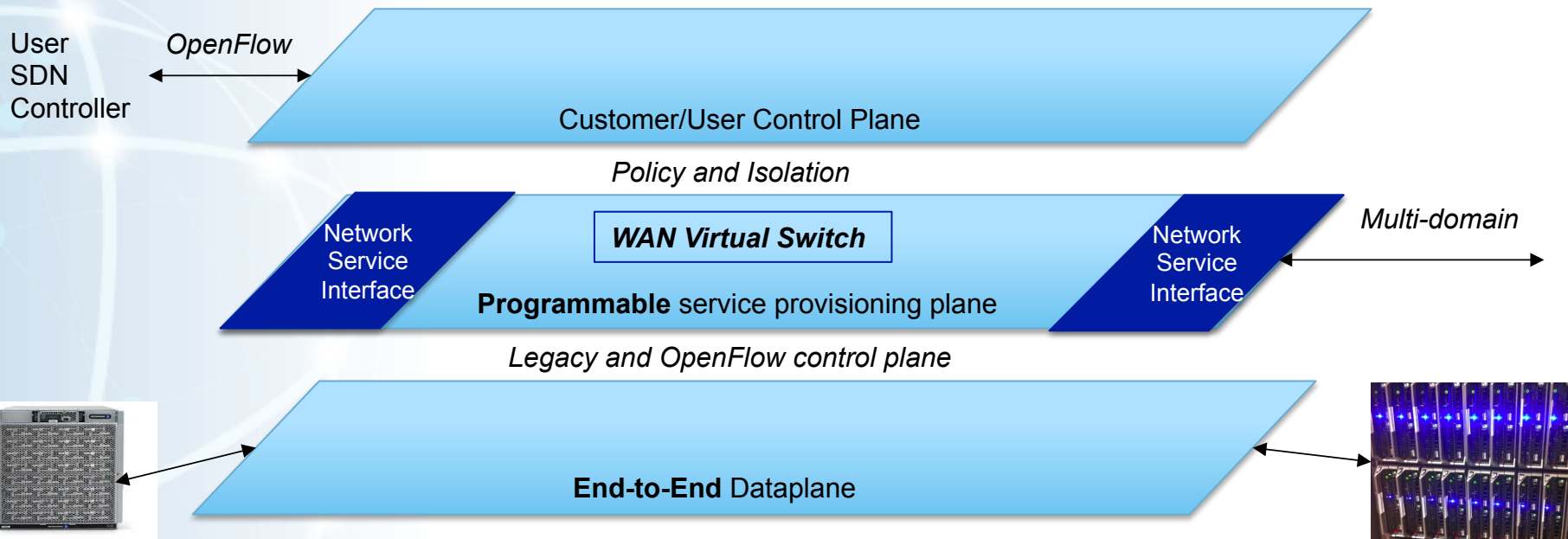


3. Need to ARP

6. virtual-mac-addresses learned and mapped to real flow



Programmability in the layers



- Creation of a programmable network provisioning layer
- Sits on top of the “network OS”



Summary

- Powerful network abstraction
 - Files / Storage
- Benefits
 - **Simplicity** for the end-site
 - Works with off-the-shelf, open-source controller
 - Topology simplification
 - **Generic** code for the network provider
 - Virtual switch can be layered over optical, routed or switched network elements
 - OpenFlow support needed on edge devices only, core stays same
 - **Programmability** for applications
 - Allows end-sites to innovate and use the WAN effectively



Future Work

Harden the architecture and software implementation

- Move from experiment to test service

Verify scaling of the model

- Using virtual machines, other emulation environments

Automation and Intelligent provisioning

- Work over multi-domain
- Wizards for provisioning
- Dynamic switch backplane

Create recurring abstractions

- Virtual switch in campus
- How do we deal with a “network” of virtual switches

Acknowledgements



My co-collaborators: Eric Pouyoul and Chin Guok

Many folks at ESnet who helped with the deployment and planning

- Sanjay Parab (CMU), Brian Tierney, John Christman, Mark Redman, Patrick Dorn, Eli Dart among other ESnet NESG/OCS folks

Ciena Collaborators:

- Rodney Wilson, Marc Lyonnais, Joshua Foster, Bill Webb

DOE ASCR research funding that has made this work possible



Questions – please contact imonga at es.net

www.es.net

Thank you!

Computer virtualization

