SELECTED TOPICS IN ADVANCED NETWORKING FOR SCIENTIFIC APPLICATIONS

Artur Barczyk/Caltech

Varenna School

Varenna, July 29th, 2014

Agenda

- Introduction

- Part I: Application-Aware Networks
  - Dynamic Circuits
  - OGF Network Services Interface
  - Examples (ANSE)

- Part II: Software Defined Networking
  - Introduction to SDN
  - OpenFlow
  - Programmable Networks
  - Use cases

- Part III:
  - Content Centric Networking

- Additional Resources
  - Networks for experimentation

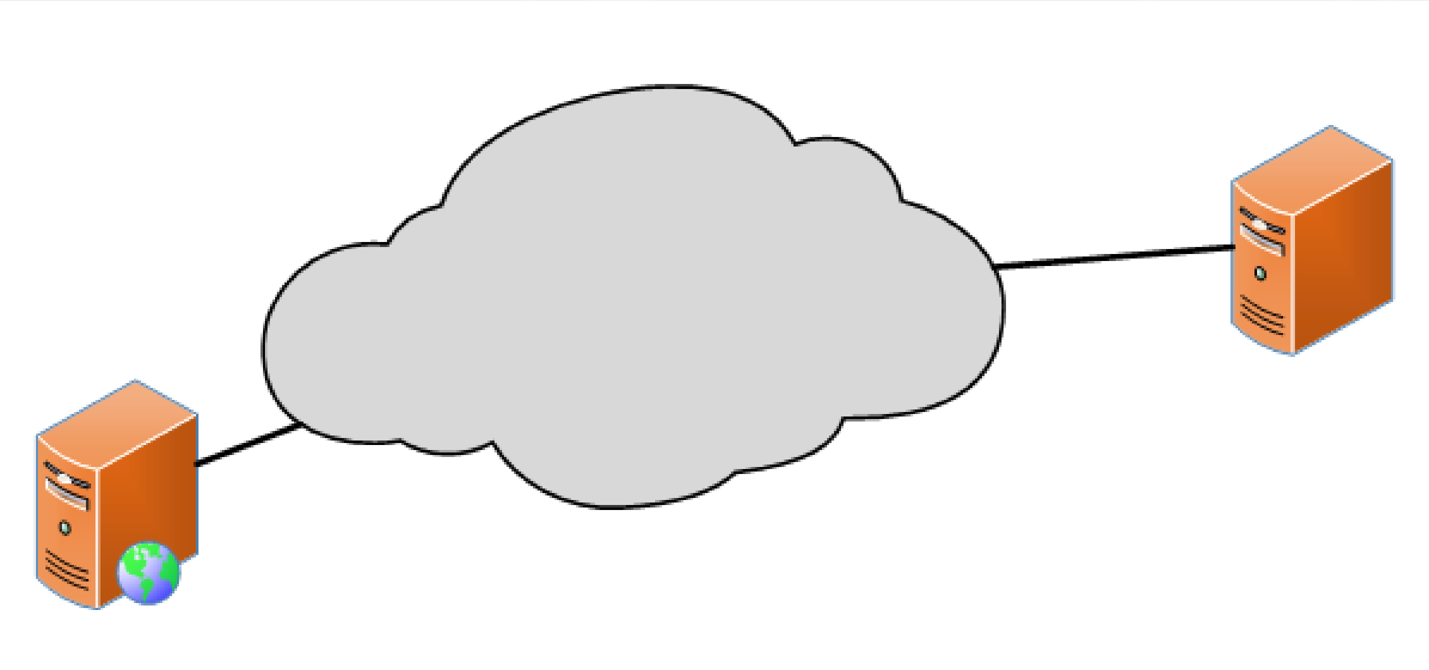
INTRODUCTION

Different Views of a Network

CALTECH HEP NETWORKING

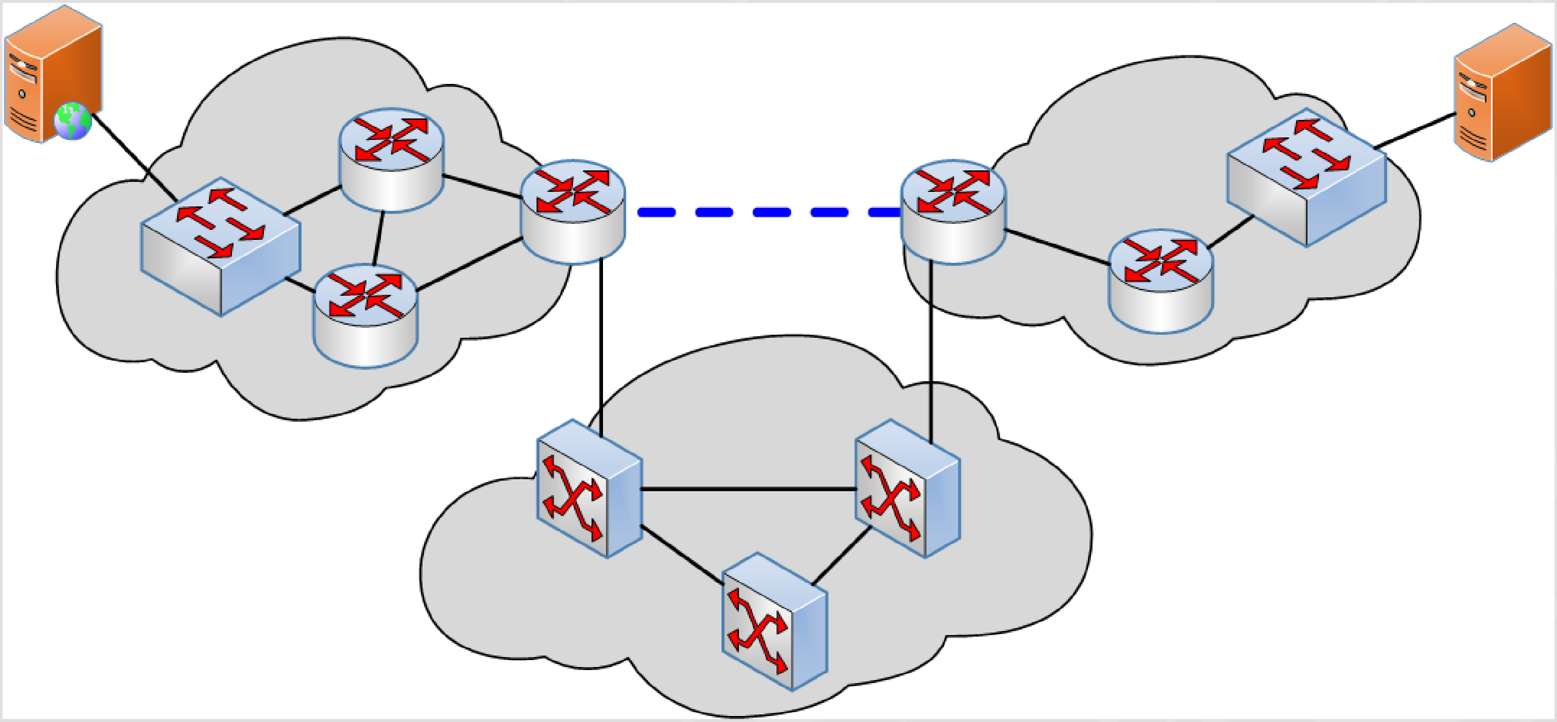
US LHCNet

… what a user might see:



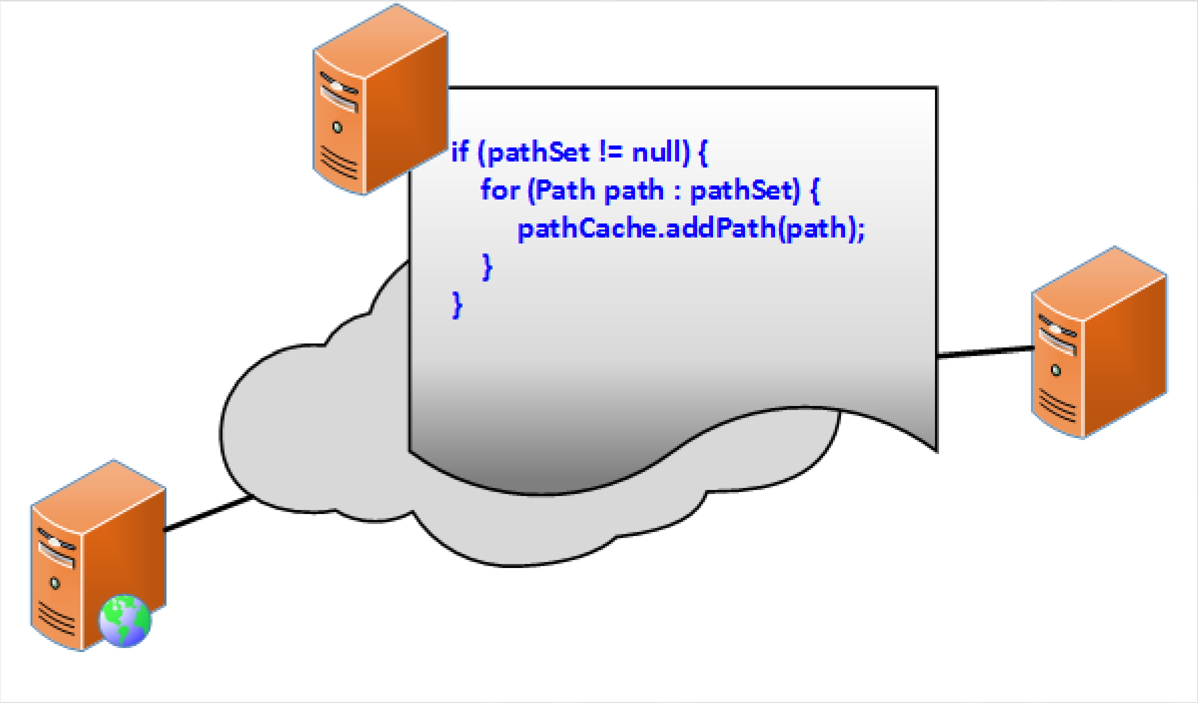
Different views of a Network

… what a network engineer would see:



Different Views of a Network

… what an SDN network engineer would see:

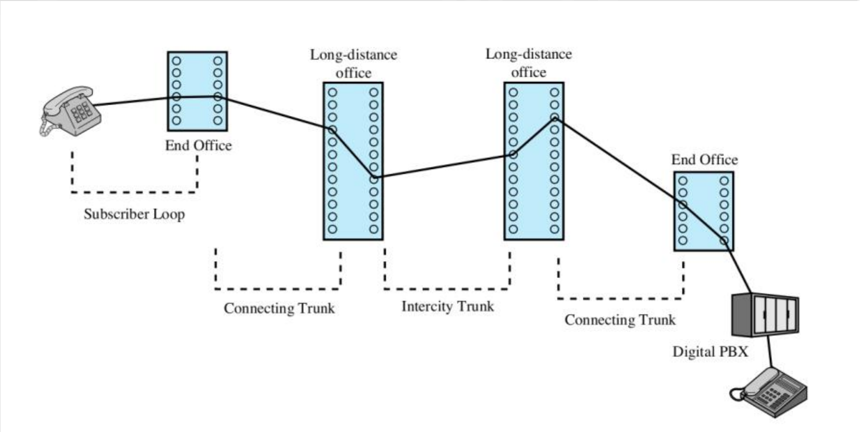


But first…

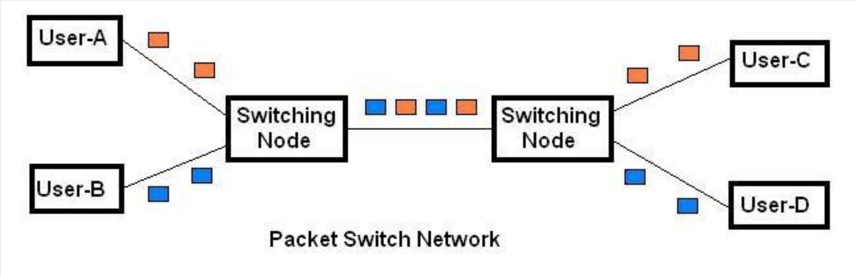
- A refresher of basics and terms…
  (not a formal course on networking, just a collection of terms and definitions needed for the discussion later)

Circuit vs Packet Switched Networks

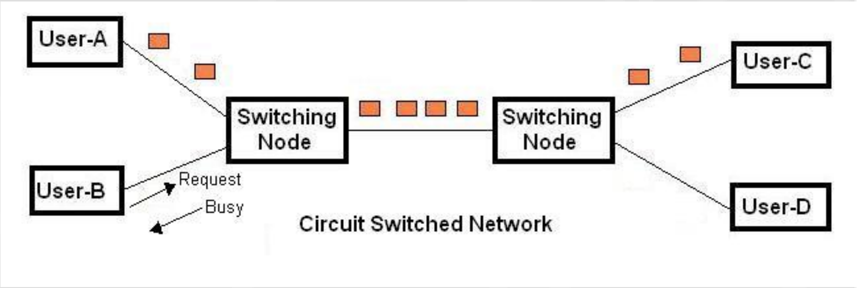
- Circuit Switching
  - Dedicated communication path between two stations
  - Set up prior to data exchange
  - Usually through several nodes in the network
  - Example: telephone network



- Packet Switching
  - Data sent in packets
  - Each packet's header is inspected at each network node
  - Packets are passed from node to node based on header information and (local) routing database
  - Example: IP network



- Virtual Circuit Switching
  - Emulation of circuit switching on packet switched infrastructure



Network Layers

CALTECH HEP NETWORKING

US LHCNet

- Communication happens between corresponding entities in a layered structure

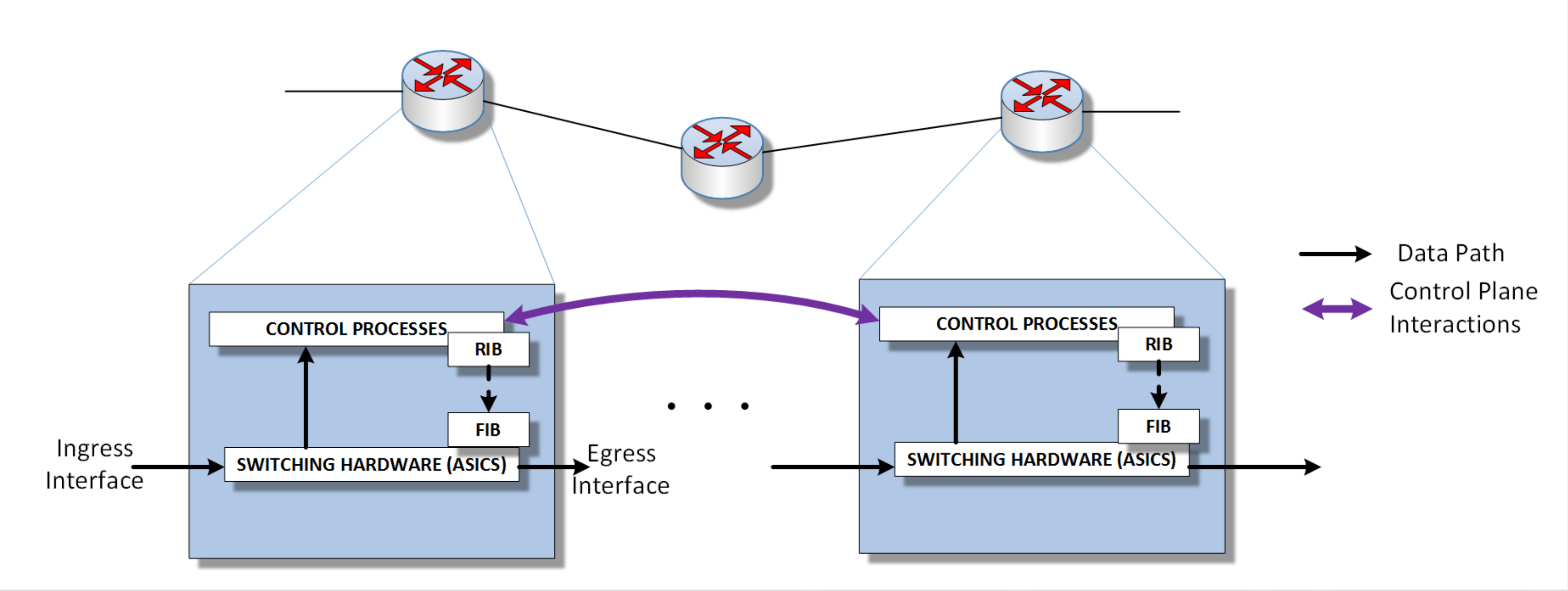
OSI Reference Layers

End System	Packet Router	End System	Function	
7. application		7. application	Interface between application and communication sw	http
6. presentation		6. presentation	Data formats, encryption	
5. session		5. session	Start/control/stop sessions	
4. transport		4. transport	Segmentation and reassembly, error recovery	TCP, UDP
3. network	3. network	3. network	End-to-end delivery of packets	IP
2. data link	2. data link	2. data link	Data delivery across a link or medium	Ethernet
1. physical	1. physical	1. physical	Physical characteristics	SONET

Packet Router

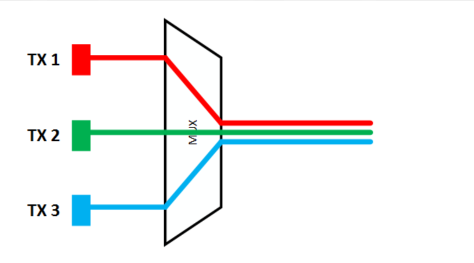
Control & Data Planes

- **Data Plane**: processing of incoming data packets
  - Inspect, forward or drop
- **Control plane**: processes to build topology (RIB) and forwarding tables (FIB)
  - Needed to populate Forwarding Information Base used in the data plane
- In a traditional networks, each node operates processes in both control and data plane

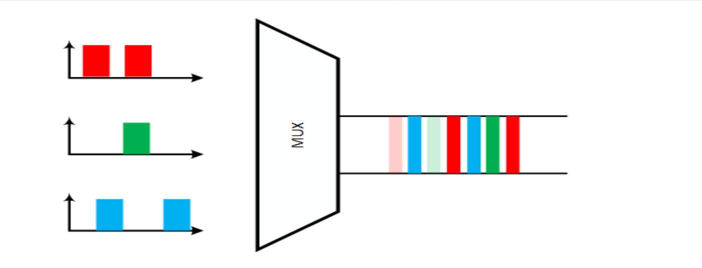


Multiplexing

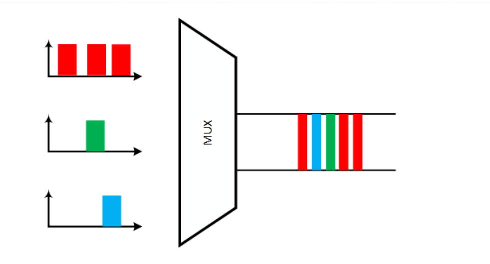
- Multiplexing is used to enable sharing of transmission medium between multiple devices
- Most common multiplexing schemes:
  - Wavelength-Division multiplexing



  - Time-Division Multiplexing



  - Statistical multiplexing



  - But also Space-Division multiplexing…

Circuit switching revisited

- Optical circuit switching equipment operate at Layer 1
  - Or even at "Layer 0" like e.g. MEMS switches

- Layer 1 optical equipment can switch based on wavelengths - Called Lightpath or Lambda-switching

- **Virtual circuit** connections above physical layer

  - SONET/SDH: TDM channels with defined capacity

  - MPLS emulates circuit connections using bandwidth profiles

  - TCP: a logical circuit connection between two end-systems
    - As opposed to UDP's datagrams

Packets, Circuits and Flows

- Packet switches and routers forward based on each individual packet's header information
  - In IP networks, typically only IP Destination address is matched against the Routing Information Base
    - plus QoS fields
    - sometimes also source address (PBR)

- Flow-based forwarding on the other hand…
  - Flow definition based on a set of parameters , such as e.g. {IP_SRC, IP_DST, TCP_PORT}
  - Network device forwards packets based on forwarding database information for that flow – each packet in the flow takes the same path

- Flow-based forwarding is encountered in e.g. Link Aggregation scenarios (LAG, ECMP), as well as being the basis of OpenFlow

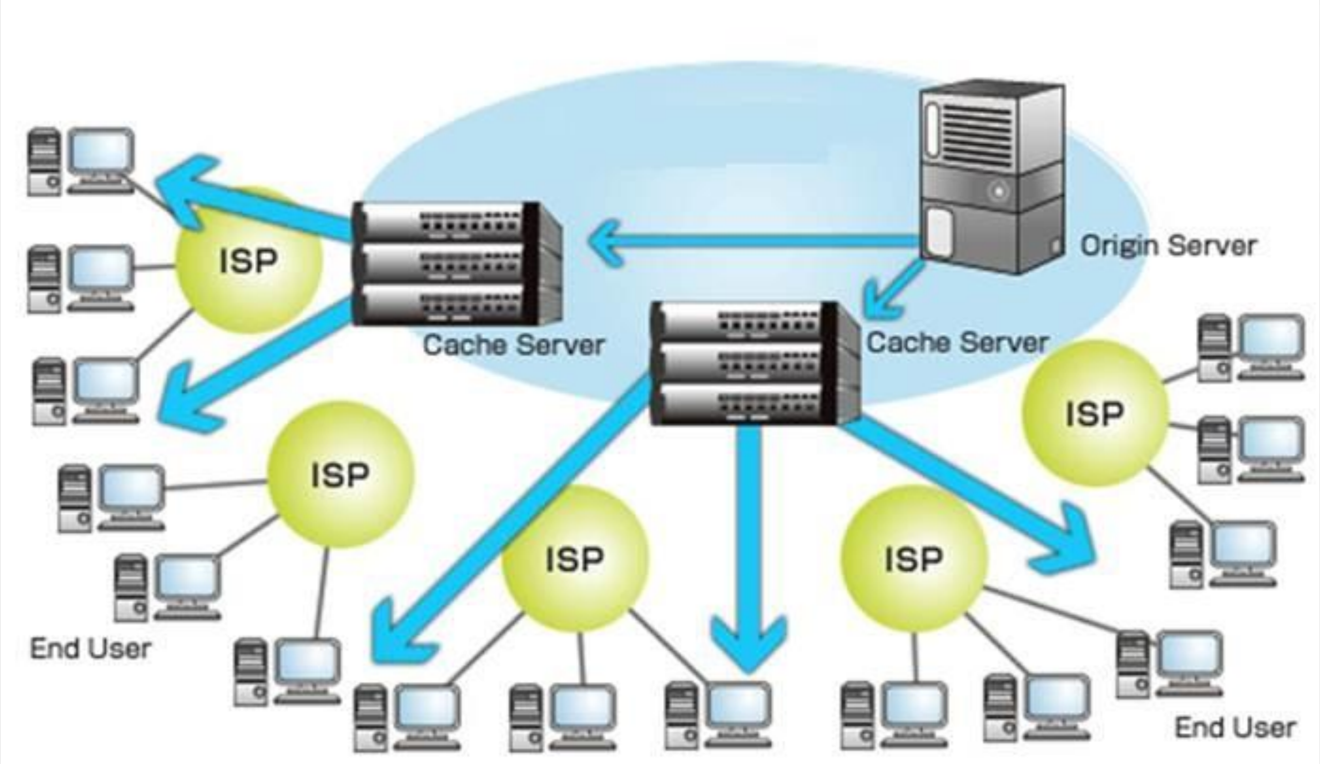
Other contents of a network environment

- In practical deployment, networks do not just forward packets
- Other services needed to function:
  - DNS
  - Possibly DHCP
  - AAA
    - NIS, LDAP, Shibboleth, etc.
  - Monitoring

- Networks deliver content…

Content Delivery Networks / CDN

- Goal: reduce WAN latencies for data delivery
- Strategically placed Cache Servers
- Data replicated from the Origin Server(s)
- Application-level technology
- Usually an overlay on top of existing IP network infrastructure



APPLICATION AWARE NETWORKING

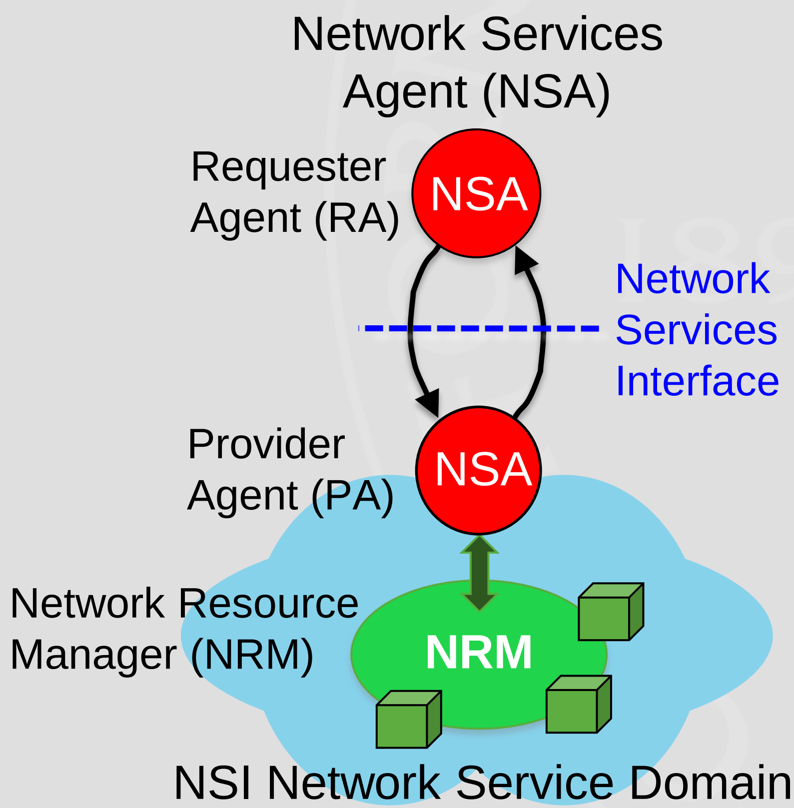
And network-aware applications

Network – Application Interface

- Any distributed system needs some form of network interaction
- Basic programming interface: Sockets
  - "puts bits on wire"
  - Restricted QoS
- Network Control
  - Reserve capacity
    - usually a NOC procedure, unless BoD system used
  - Prioritize traffic
- Network Monitoring and Analytics
  - To base smart decisions on
    - Reachability
    - Topology
    - Available capacity

OGF Network Services Interface (NSI)

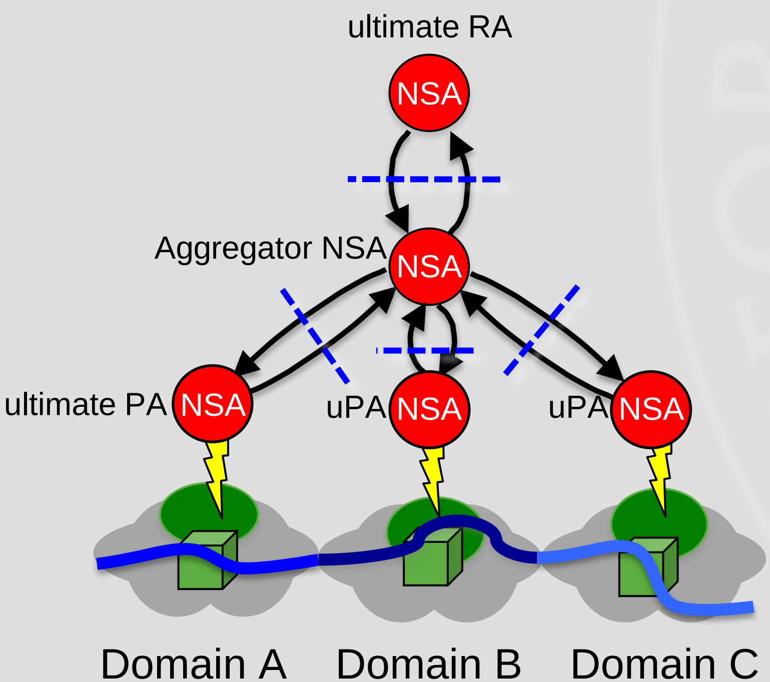
- A standardized service interface between network domains
  - Note: A computing site is also a network domain!
- Open Grid Forum Working Group (NSI-WG)

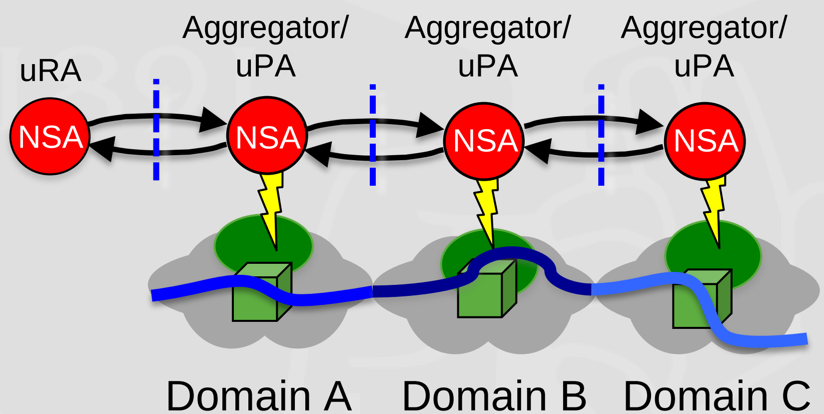


picture C. Guok, ESnet

NSI multi-domain service construction

- Two ways defined for “chaining” services: tree and chain
- Note: “ultimate Requester Agent” can be an end-user app





OpenGridForum

pictures C. Guok, ESnet

NSI Services

- Currently foreseen services:
  - Connection Service (NSI-CS)
  - Topology Service (NSI-TS)
  - Discovery Service (NSI-DS)
  - Switching Service (NSI-SS)

- Future Services:
  - Monitoring Service
  - Protection Service
  - Verification Service
  - …

NSI Connection Service

- NSI-CS: first NSI service standardized, currently v2.0
- Advance-reservation protocol
  - Mandatory Reservation parameters:
    - A-point, Z-point
  - Optional parameters:
    - Start time, end time
    - Bandwidth
    - Labels/VLAN IDs

- V2.0 supports optional modification of a reservation
  - Start time, end time and bandwidth

NSI Request Messages

NSI CS Message (abbreviation)	**SM**	**Synch. /Asynch.**	**Short Description**
reserve (rsv.rq)	RSM	Asynch	The *reserve* message allows an RA to send a request to reserve network resources to build a Connection between two STP's.
reserveCommit ***(***rsvcommit.rq)	RSM	Asynch	The *reserveCommit* message allows an RA to request the PA commit a previously allocated Connection reservation or modify an existing Connection reservation.
reserveAbort (rsvabort.rq)	RSM	Asynch	The *reserveAbort* message allows an RA to request the PA to abort a previously requested Connection that was made using the *reserve* message.
provision (prov.rq)	PSM	Asynch	The *provision* message allows RA to request the PA to transition a previously requested Connection into the Provisioned state. A Connection in Provisioned state will activate associated data plane resources during the scheduled reservation time.
release (release.rq)	PSM	Asynch	The *release* message allows an RA to request the PA to transition a previously provisioned Connection into Released state. A Connection in a Released state will deactivate the associated resources in the data plane. The reservation is not affected.
terminate (term.rq)	LSM	Asynch	The *terminate* message allows an RA to request the PA to transition a previously requested Connection into Terminated state. A Connection in Terminated state will release associated resources and allow the PA to clean up the RSM, PSM and all related data structures.

Full messages listing in http://www.ogf.org/documents/GFD.212.pdf

OpenGridForum

NSI Service Agent implementations

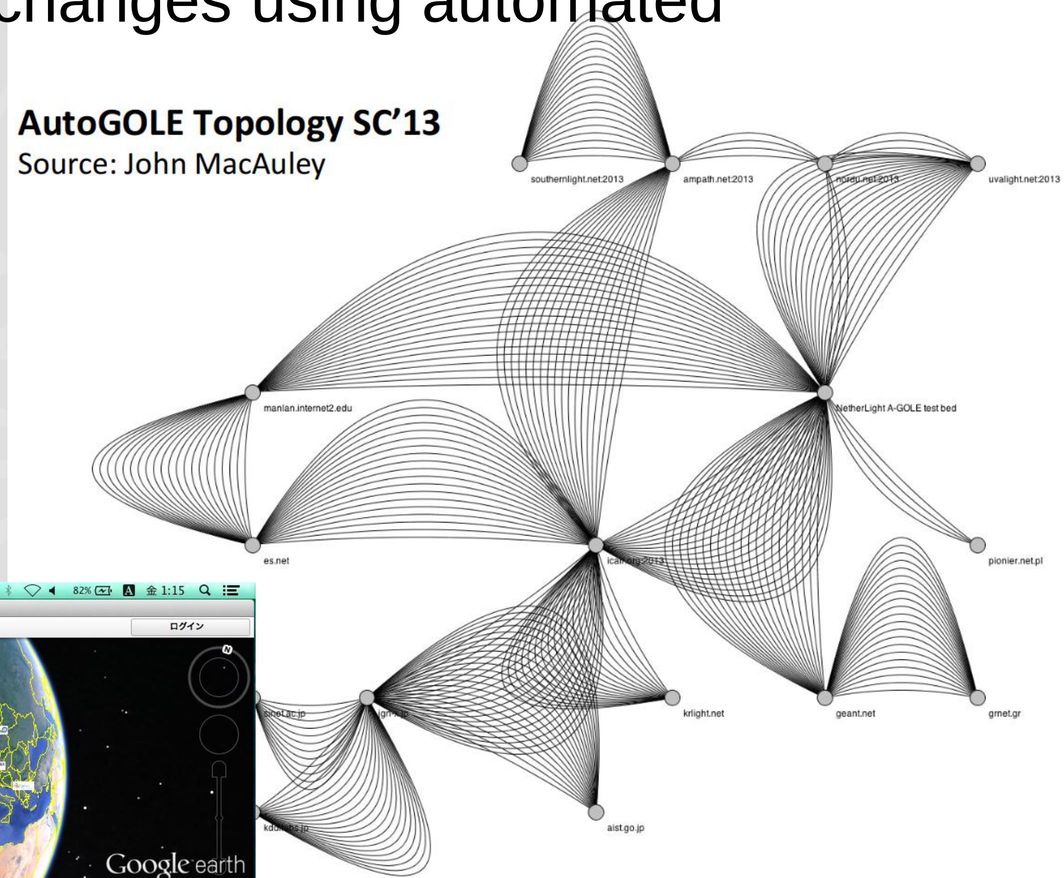
- **AutoBAHN** : GÉANT (EU)
- **BoD** : SURFnet (NL)
- **DynamicKL** : KISTI (KR)
- **G-LAMBDA-A** : AIST (JP)
- **G-LAMBDA-K** : KDDI Labs (JP)
- **OpenNSA** : NORDUnet (DK, SE, NO, FI, IS)
- **OSCARS** : ESnet (US)

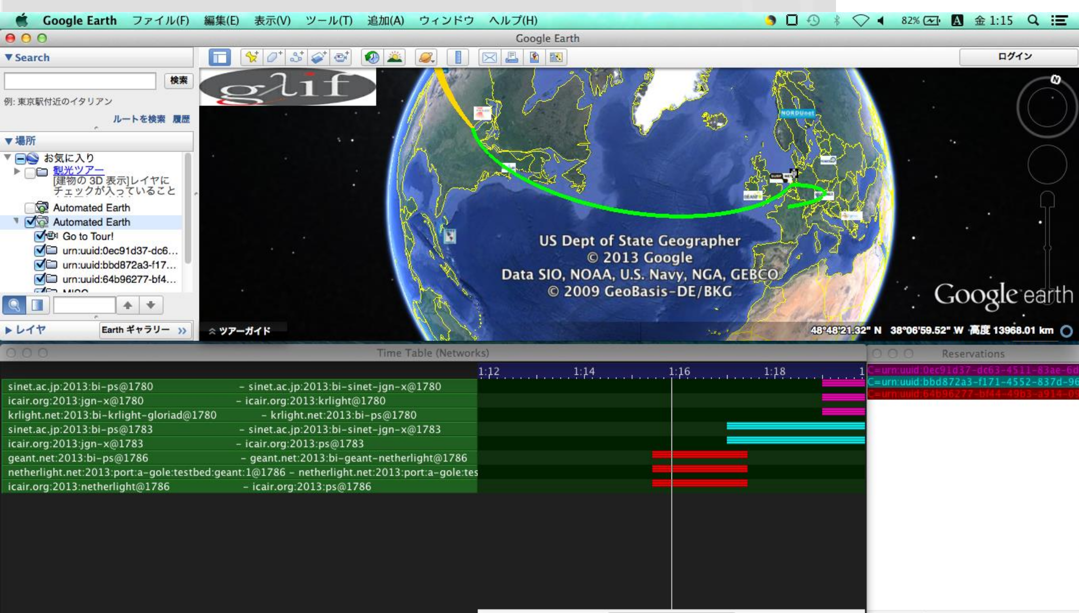
Automated GOLE Fabric

John MacAuley, ESnet

NSI-CS in Action: GLIF AutoGOLEs

- GLIF Open Lightpath Exchanges using automated provisioning
- Currently in R&D
- Demonstrated e.g. at SC'13
- Next demo planned for GLIF and SC'14





NSI in Action: eVLBI

CALTECH HEP NETWORKING
US LHCNet

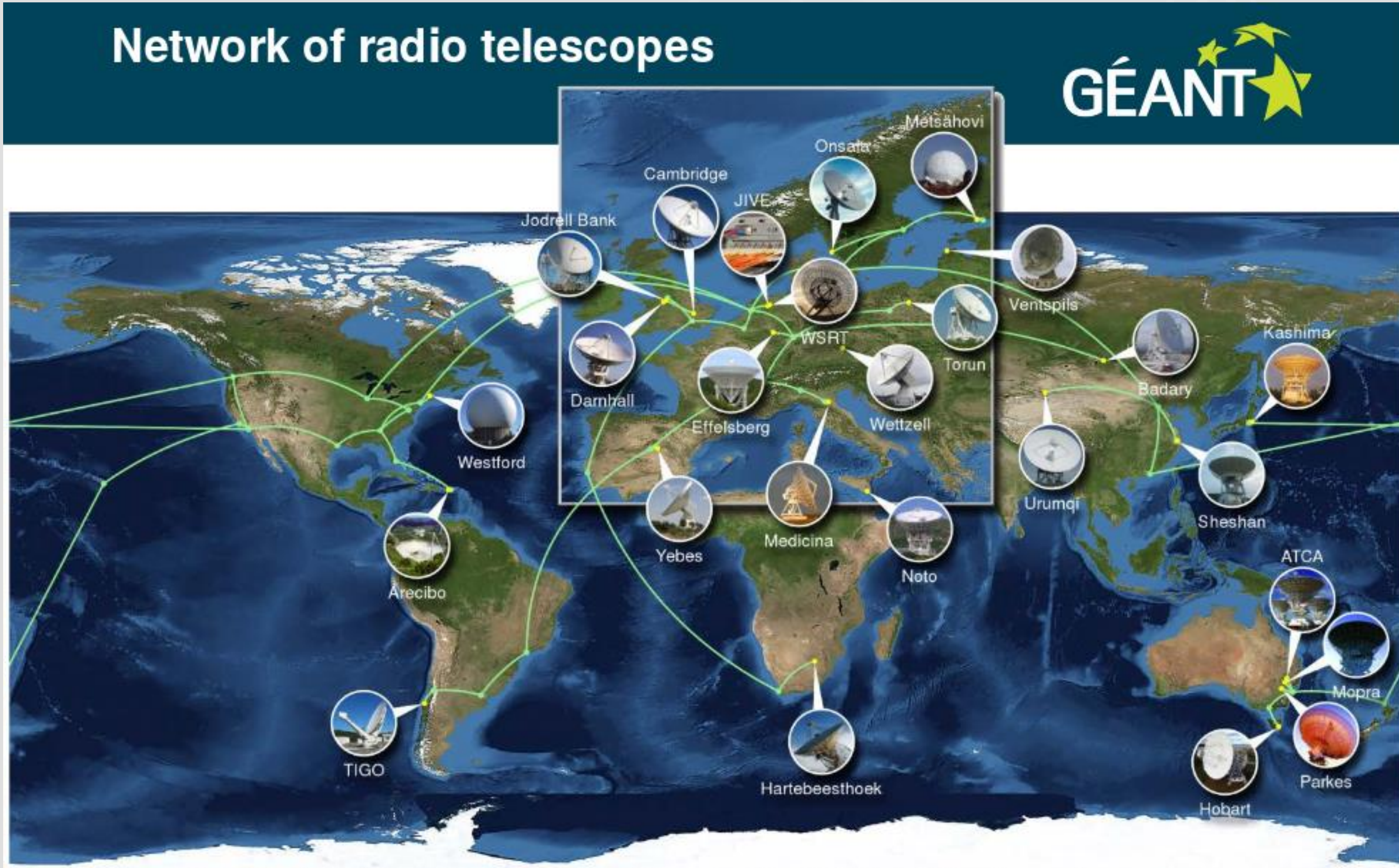


Image by Paul Boven (boven@jive.nl). Satellite image: Blue Marble Next Generation, courtesy of Nasa Visible Earth (visibleearth.nasa.gov).

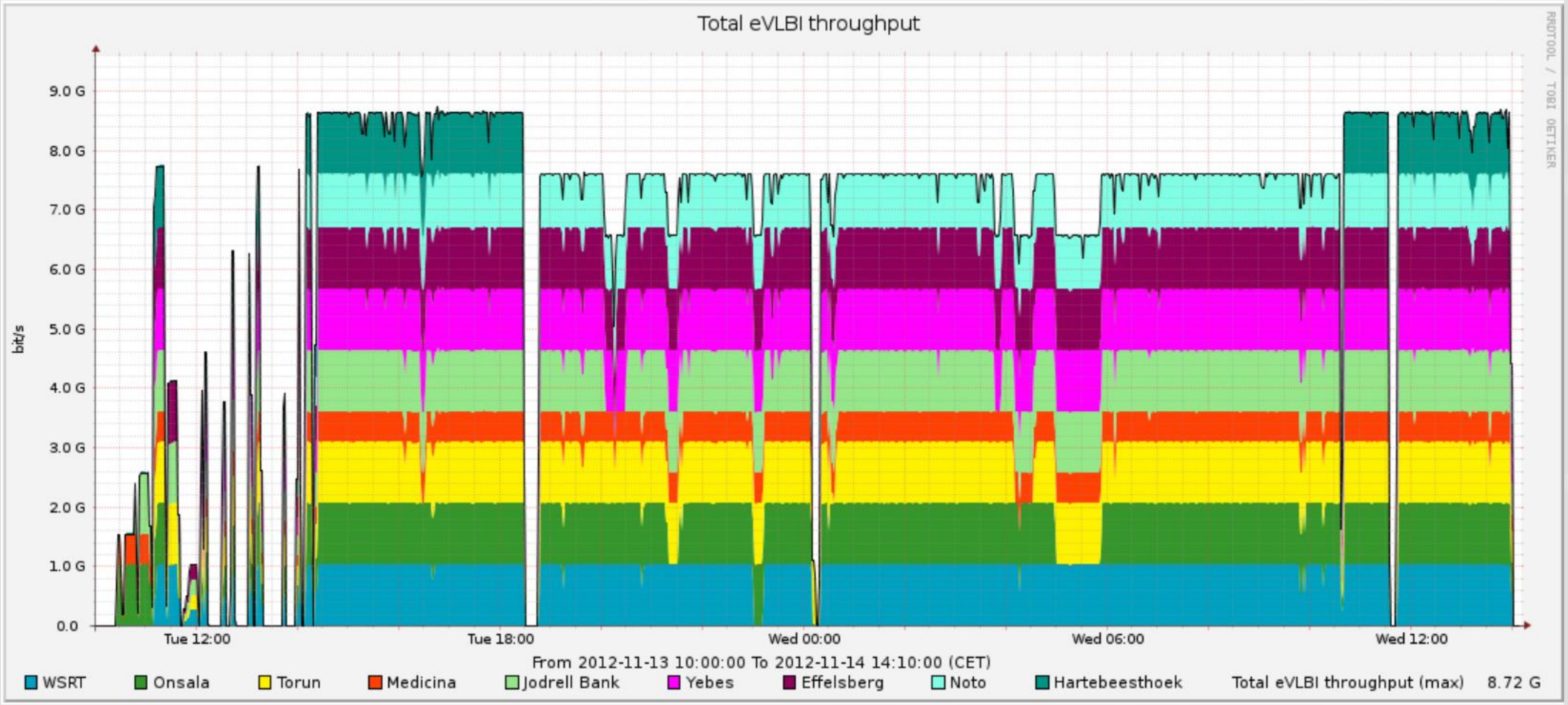
More info on http://www.evlbi.org/evlbi/evlbi.html

A typical eVLBI run

CALTECH HEP NETWORKING

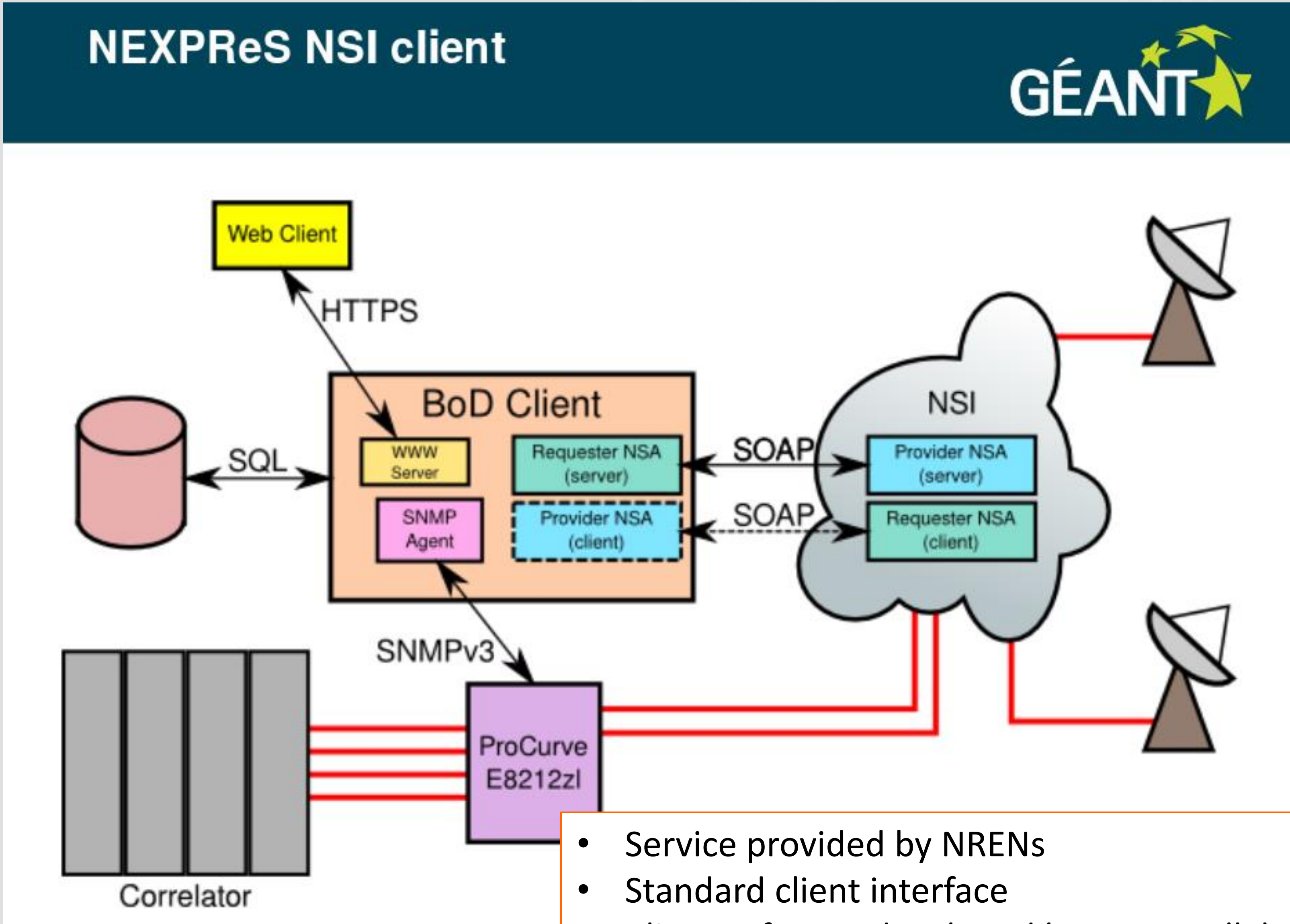
US LHCNet

- 8-12 radio telescopes
- 1Gbps per telescope (future: 4Gbps)
  - Steady streams of data from antennas to correlator
  - Low jitter very important
- 8-12 hours
- 30-65 TB



Paul Boven, JIVE

NSI in Action: eVLBI



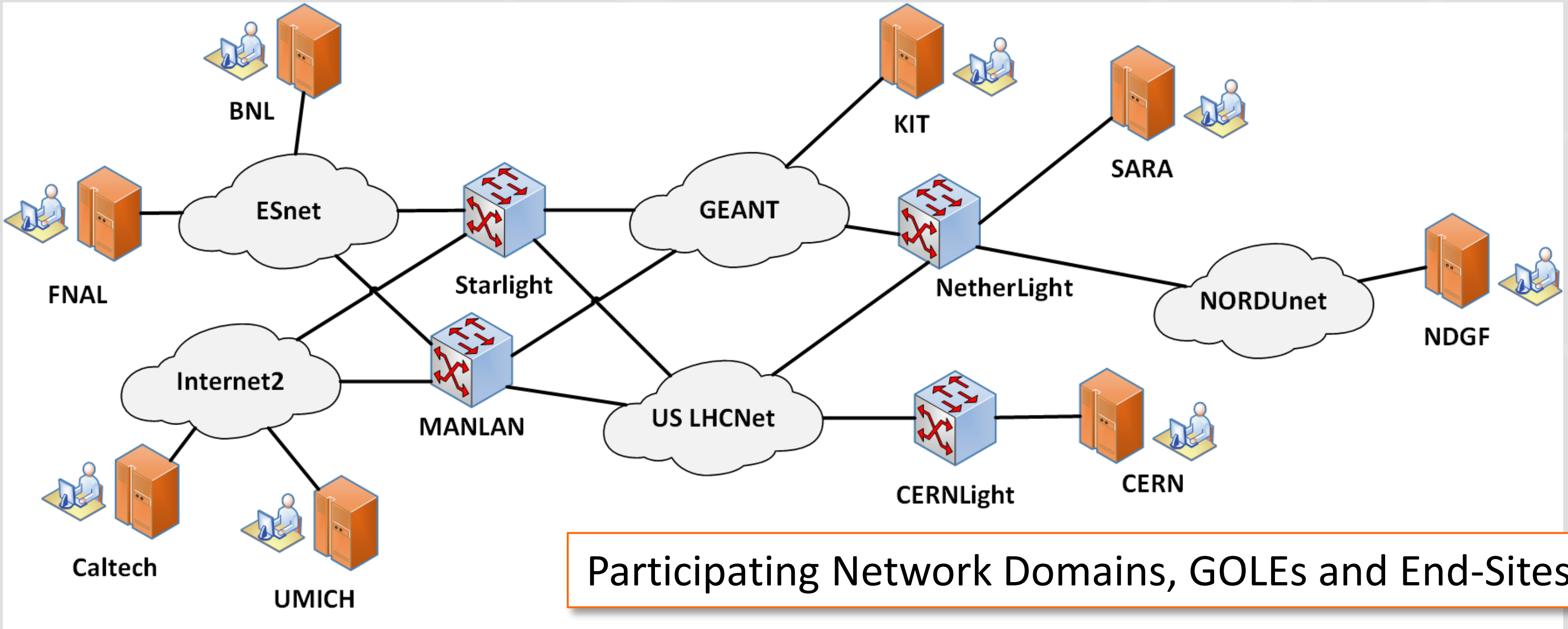
- Service provided by NRENs
- Standard client interface
- Client software developed by eVLBI collaboration

NSI for LHC: LHCONE Point-to-Point Service

CALTECH HEP NETWORKING

US LHCNet

- LHC Open Network Environment
  - VRF for current multipoint production use
  - Experiment/demonstration: Bandwidth-on-Demand
- Target: demonstrate multi-domain bandwidth reservation capability



Participating Network Domains, GOLEs and End-Sites

- Status: under construction
  - Multi-domain service created first, then connect end-sites

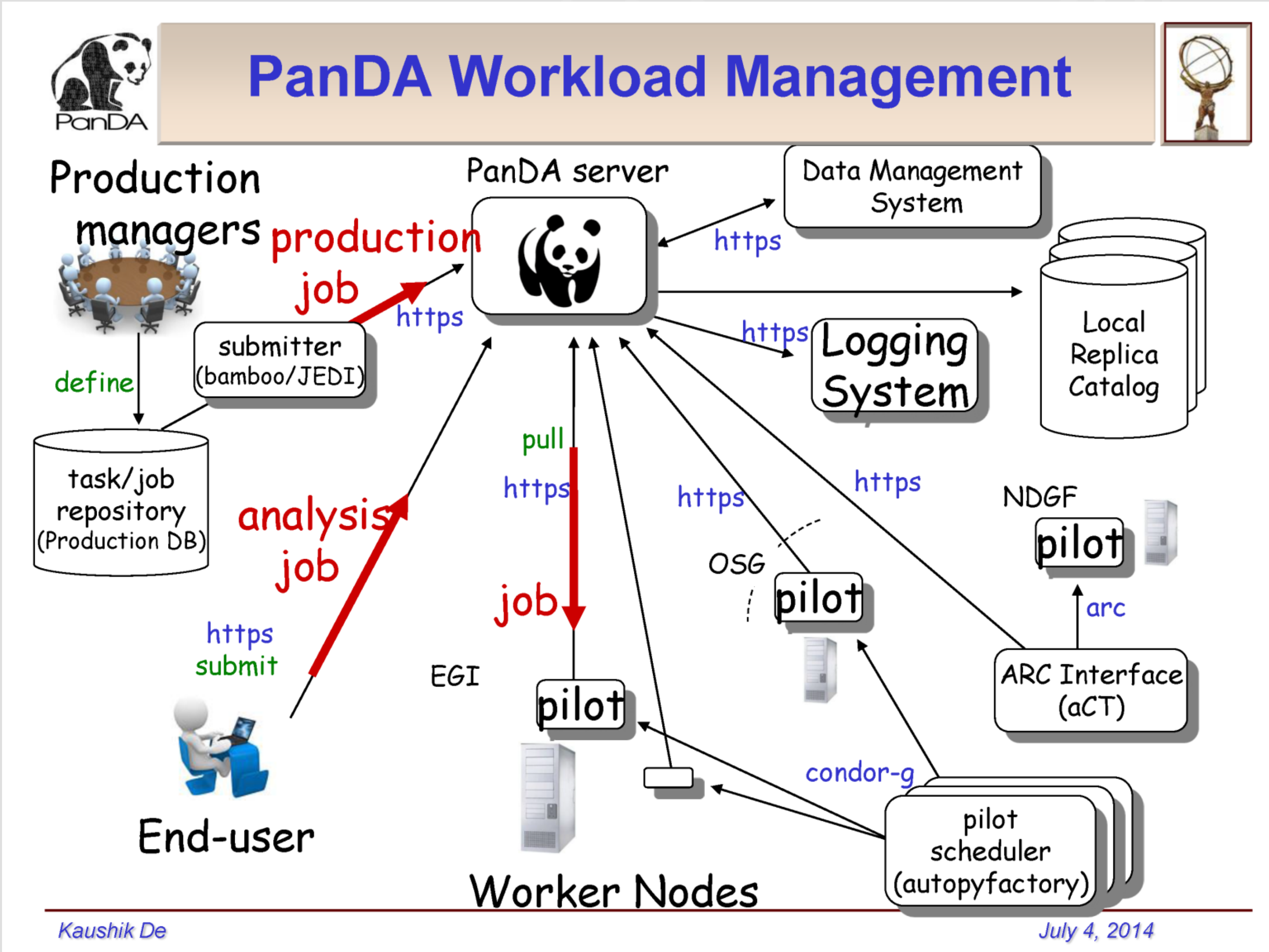
Building Network Awareness in LHC

CALTECH HEP NETWORKING

US LHCNet

- General: network monitoring in OSG and WLCG
- Based on information from monitoring systems such as MonALISA and PerfSONAR
- Specific project in CMS and ATLAS Experiments: **Advanced Network Services for Experiments**
  - Network Integration into
    - Workflow management (PanDA)
    - Data movement management (PhEDEx)
  - Measurement: PerfSONAR and MonALISA integration
  - Control: interface to provisioning systems (DYNES/OSCARS, NSI)

ANSE/Panda

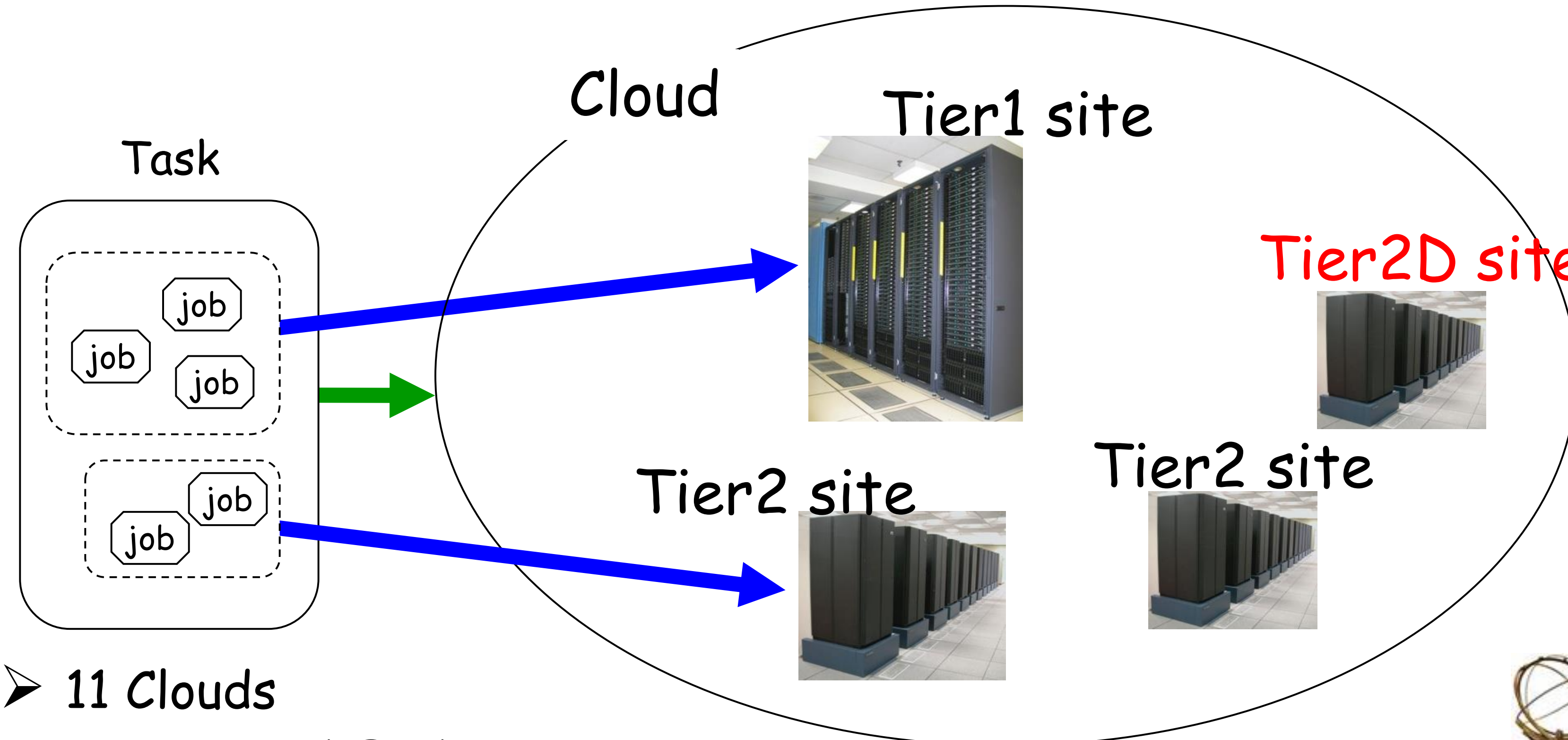


PanDA Workload Management
PanDA
Production managers
PanDA server
Data Management System
production job
https
submitter (bamboo/JEDI)
define
task/job repository (Production DB)
Local Replica Catalog
Logging System
analysis job
pull
https
job
https
submit
EGI
pilot
OSG
pilot
NDGF
pilot
arc
ARC Interface (aCT)
condor-g
pilot scheduler (autopyfactory)
End-user
Worker Nodes
Kaushik De
July 4, 2014

Network Integration in PanDA

- Concept: utilize network as a resource like other resources such as CPU, disk storage
  - Use network information for FAX brokerage
    - Brokerage should use concept of nearby sites
    - Jobs are sent to site with best weight, not necessarily the site with local data or available CPUs
  - Use network information for cloud selection
    - Best T2D site should be selected based on throughput from T1 to T2D measurements
- Network measurements are available at SSB (Site Status Board, Network view)
  - FAX xrdcp rate metric used for FAX brokerage
  - DDM Sonar metrics used for cloud selection

ATLAS Computing Model



- 11 Clouds
  - 10 T1s + 1 T0 (CERN)
  - Cloud = T1 + T2s + T2Ds (except CERN)
  - T2D = multi-cloud T2 sites
- 2-16 T2s in each Cloud

Task → Cloud
  Task brokerage
Jobs → Sites
  Job brokerage

Kaushik De, UTA

ANSE/Panda Use Cases

- Based on
  - Important to PanDA users
  - Enhance workload management through use of network
  - Should provide clear metrics for success/failure

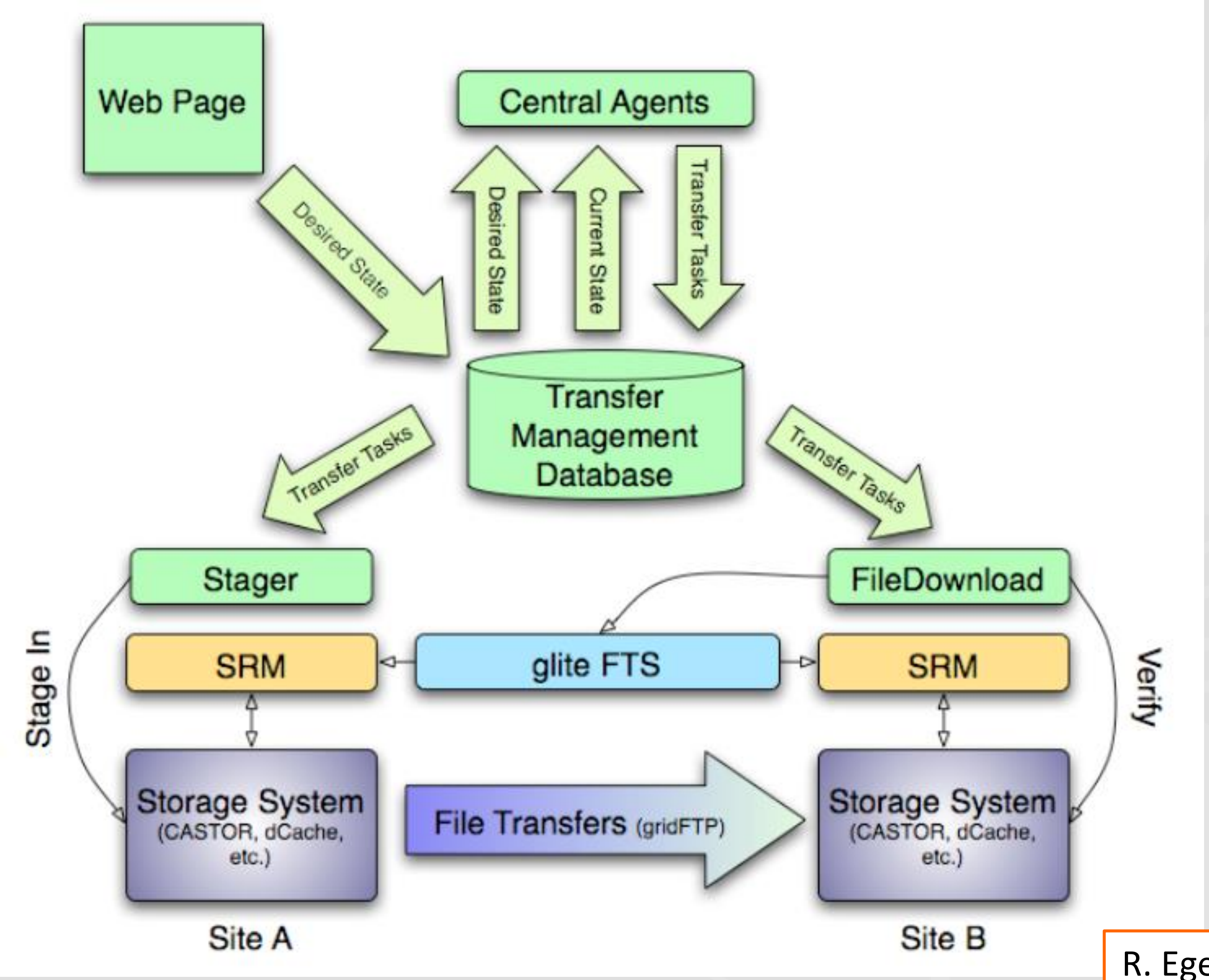
1. Improve User Analysis Workflow
   - Include network information for routing of jobs to T1/T2 sites

2. Cloud Selection:
   - Optimize choice of T1-T2 pairings
   - Automate using network information

- Both use cases are development and testing phase

ANSE/PhEDEx

- PhEDEx is the CMS data management toolkit

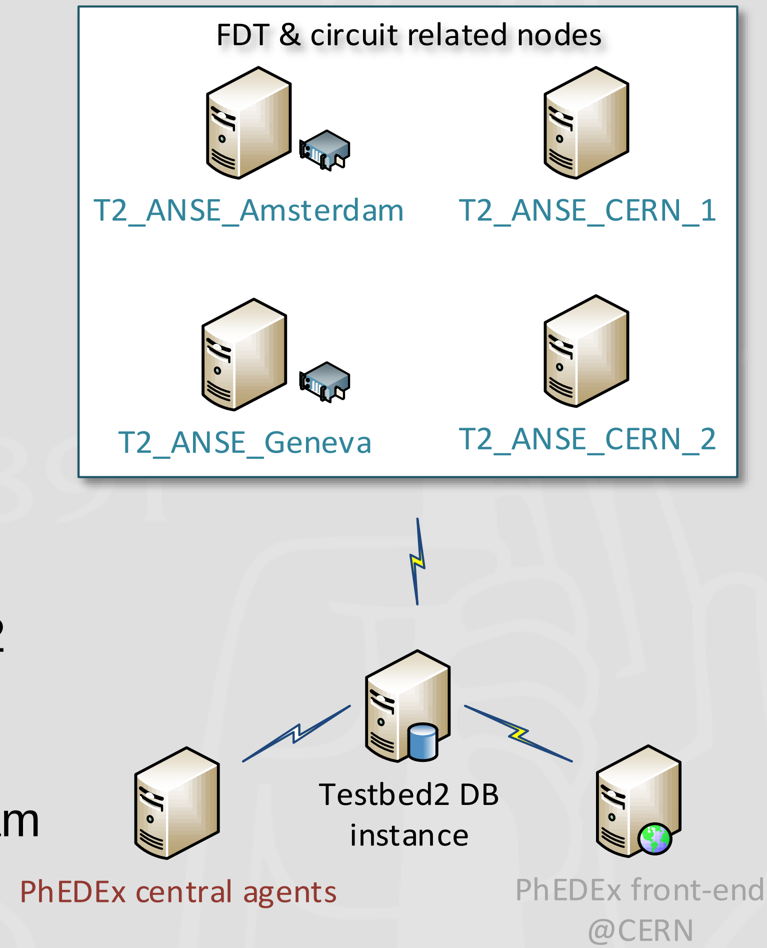


R. Egeland, 2008

ANSE/PhEDEx – Dynamic Circuits

- Several points of circuit integration into PhEDEx
  - At the transfer-job level
  - At the link level (FileDownload agent)
  - At the instance level (FileRouter agent)

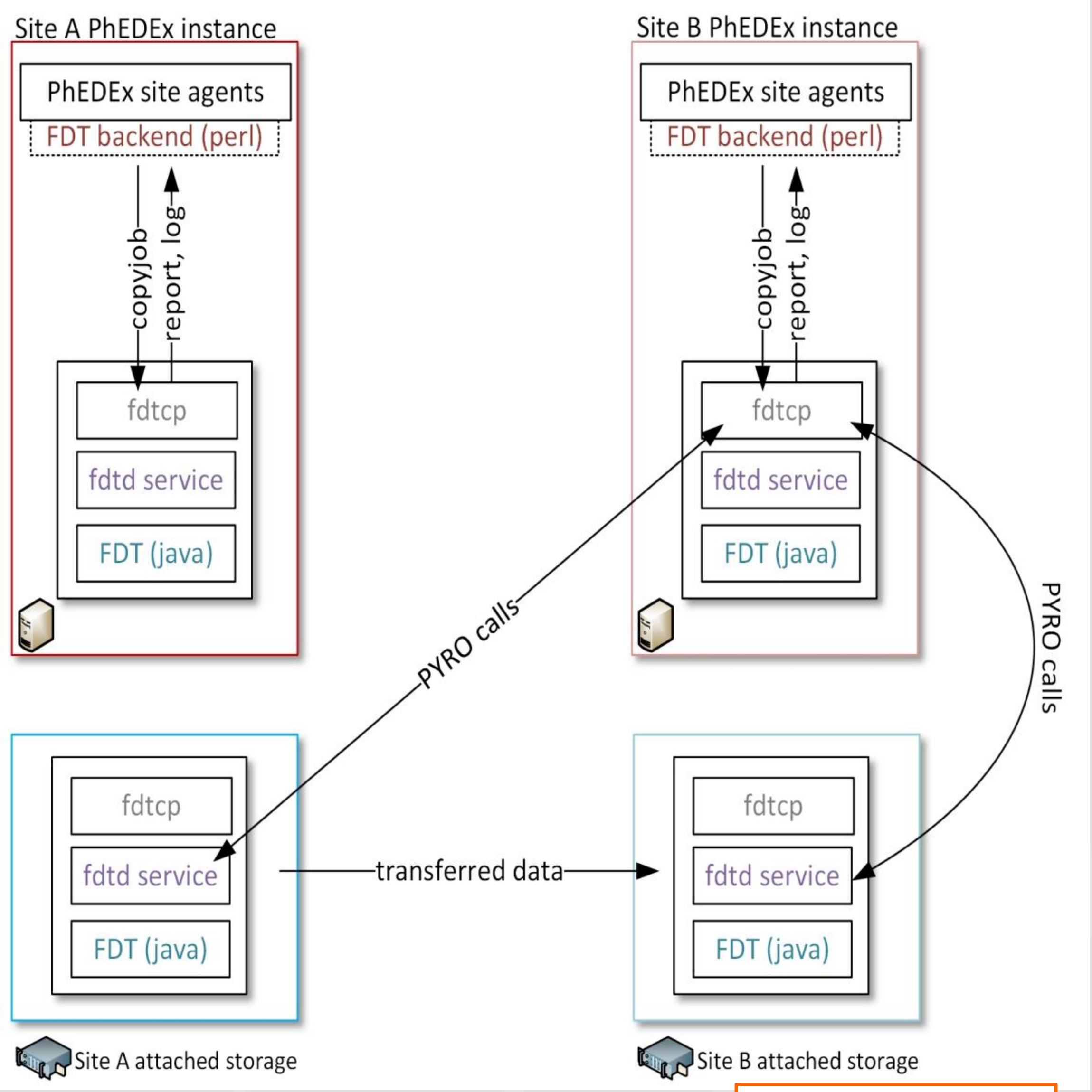
- Currently using a distributed testbed
  - T2_ANSE_CERN_1 & T2_ANSE_CERN_2
    - Both PhEDEx and storage nodes
  - T2_ANSE_Geneva & T2_ANSE_Amsterdam
    - PhEDEx and storage nodes separate
    - High speed link between AMS & GVA
    - 4x4 SSD software RAID 0 arrays



V. Lapadatescu, T. Wildish

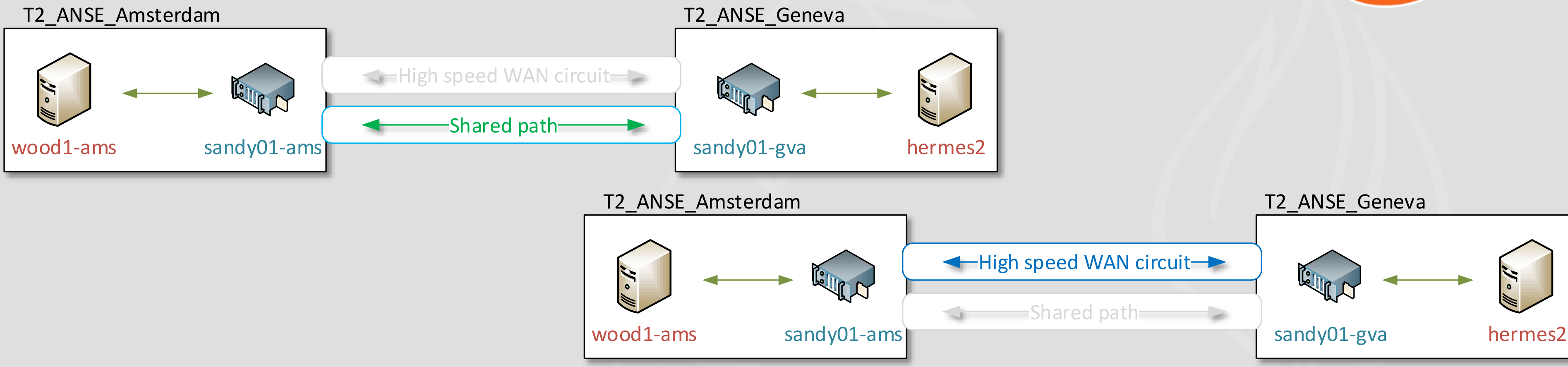
Circuits in PhEDEx at transfer level

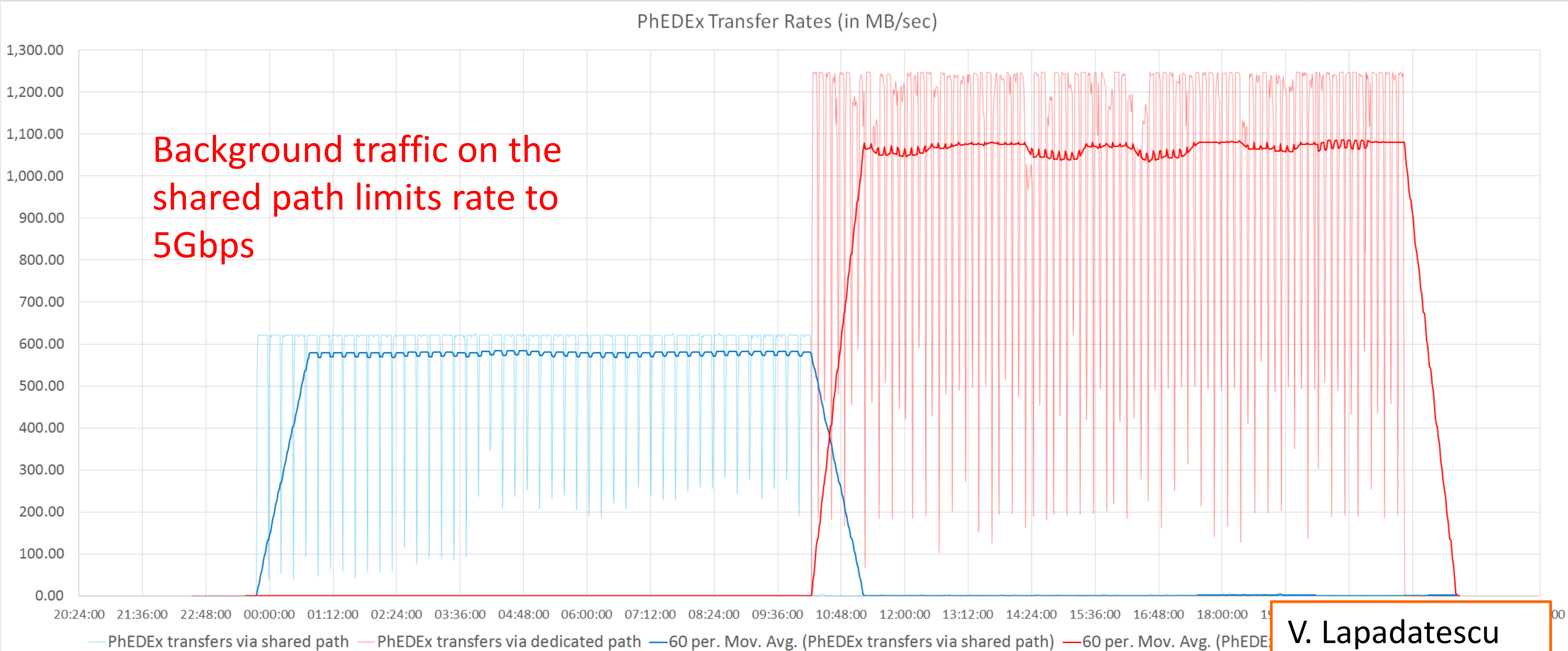
- FDT transfer tool integrates IDCP (OSCARS) calls
- Integrating FDT as transfer tool in PhEDEx naturally includes BoD capability
- Work ongoing on integration at FileDownload agent level



V. Lapadatescu

PhEDEx BoD Trials – FileDownload agent





V. Lapadatescu

Network-Aware Applications Summary

- Network awareness can improve overall system performance
  - through acting on precise, real-time data on network state
  - through creating application-specific topologies such as point-to-point virtual circuits

- Network Services Interface (NSI) standard released, several implementations in development at many NRENs

- More examples of network-application interaction in the SDN part later

SDN
SOFTWARE DEFINED NETWORKING

Where we encounter OpenFlow and intelligent networks

Network business the traditional way

- Proprietary hardware, proprietary software
  - IPR
  - provide business edge
  - vendor lock-in

- Effects:
  - closed software
  - innovation slow, driven by vendors only
  - difficult to develop and evaluate new ideas

Drivers behind SDN

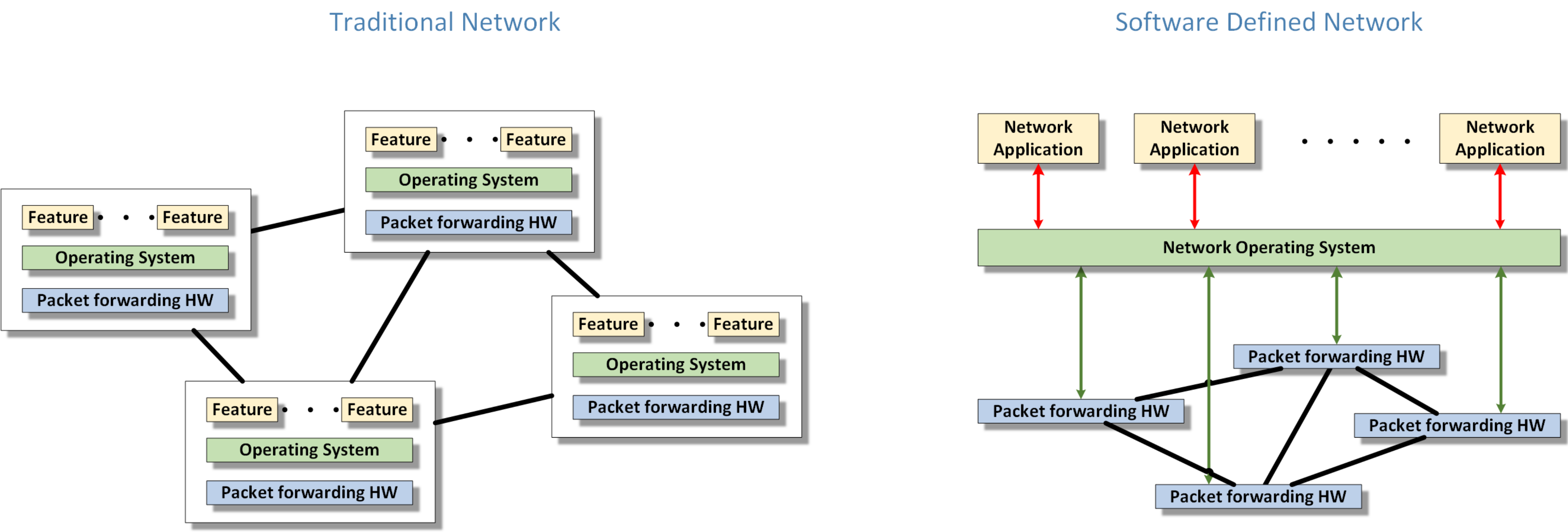
- Change in traffic patterns
  - away from single client - single server
  - local as well as wide area
- Appearance of cloud services
  - need for security, flexibility, scalability
- Manage complexity on large scales
  - Appearance of huge data centers
    - Multi-tenant facilities
  - Often global connectivity requirements

More in ONF SDN whitepaper at http://www.opennetworking.org

Software Defined Networking

CALTECH HEP NETWORKING US LHCNet

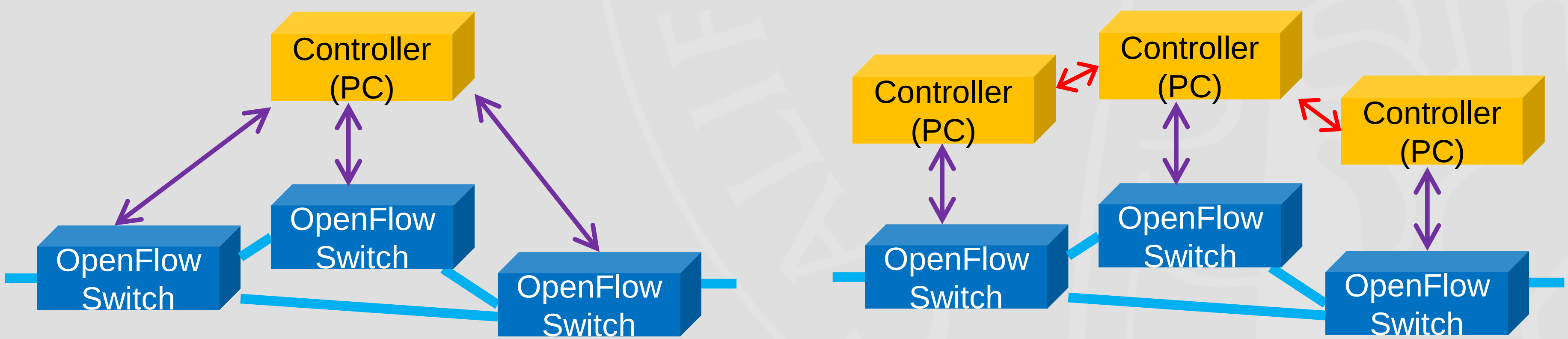
- Basic SDN Paradigm:
  Separation of Network control plane from the data plane



- Enables network control by applications; provides an API to **programmatically** define network functionality

Central control

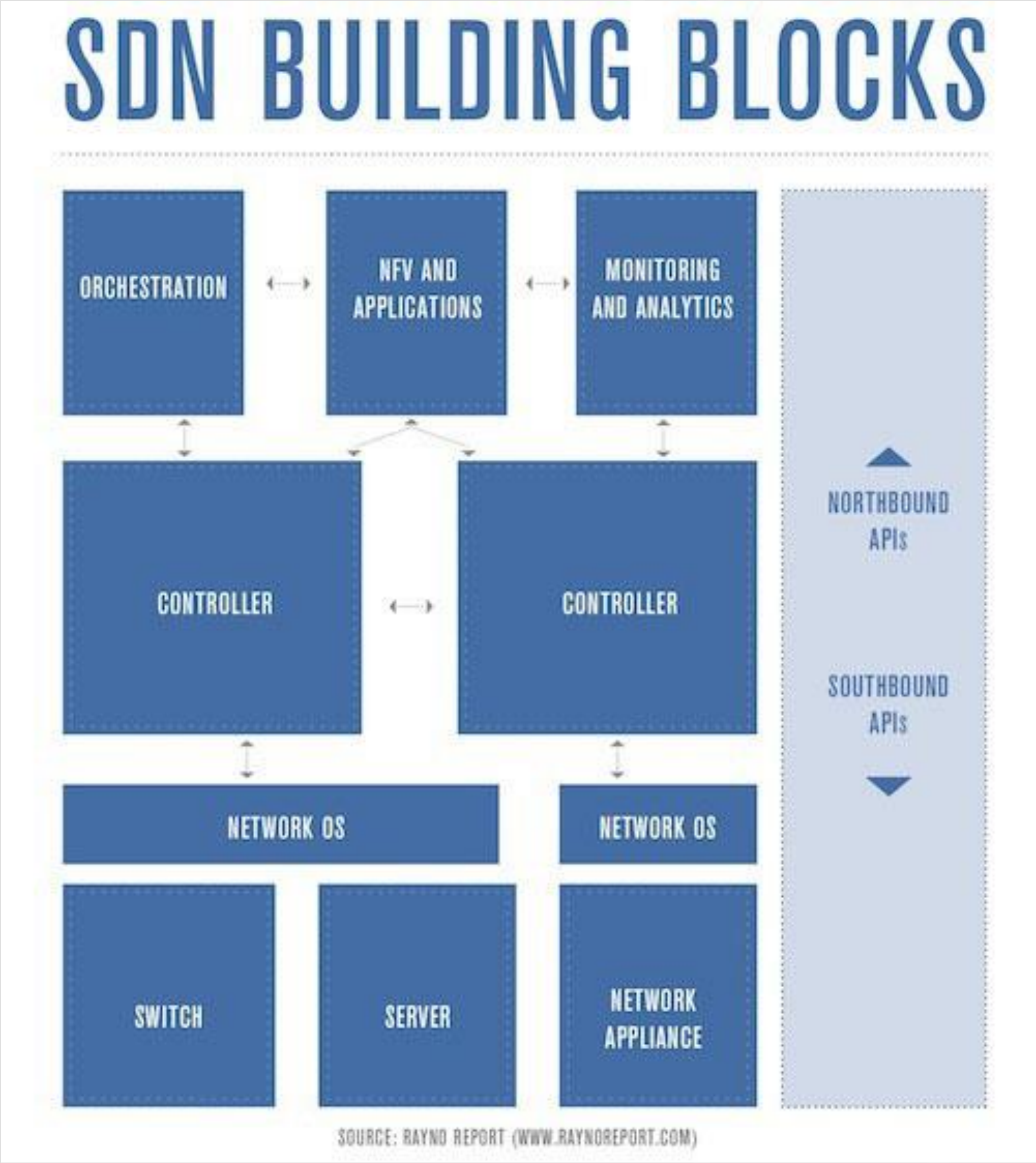
- Logically centralized control:
  - simplified operation
  - cost reduction
  - faster reconfiguration -> increased efficiency
- Physically distributed infrastructure:
  - scalability
  - redundancy



Network Programmability

- Network devices expose interface to third-party applications
- Applications provide the value
  - Networks applications:
    - Routing
    - Traffic Engineering
    - Flow Management
    - Network load balancing
  - End-user and service provider applications:
    - Access control and filtering
    - Computing resource load balancing
- Standards provide uniformity across vendor platforms

The Building Blocks



What is SDN good for

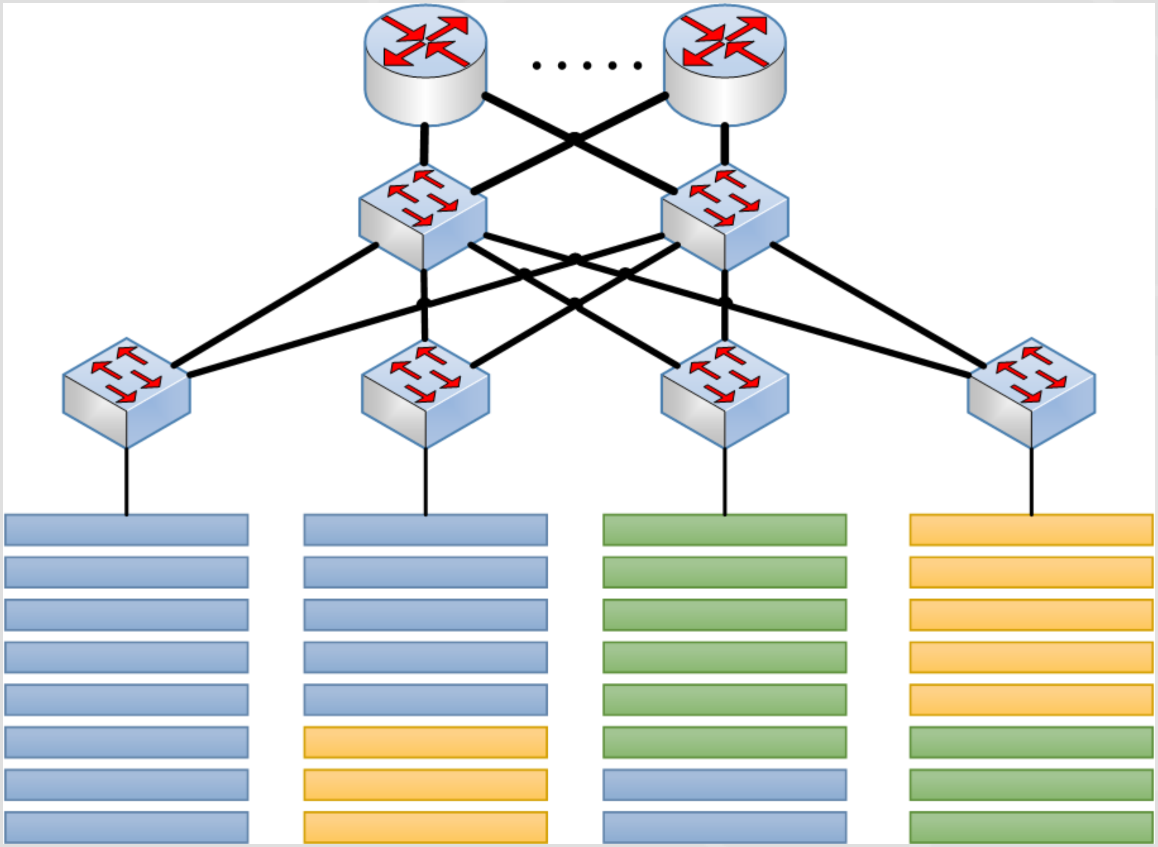
- SDNs are used for
  - Network virtualization
    - Scalability
    - Robustness
    - Security
    - Logical separation (multi-tenant environments)
  - Centralization of management
    - Simplify operational aspects and workload
  - R&D
    - Fast development and deployment of new or non-IP protocols

- SDNs are/can/will be used in
  - Data center networks
  - Cloud systems (intra-/inter-site)
  - WANs
  - Transport networks

Example: Multi-Tenant Datacenter

CALTECH HEP NETWORKING

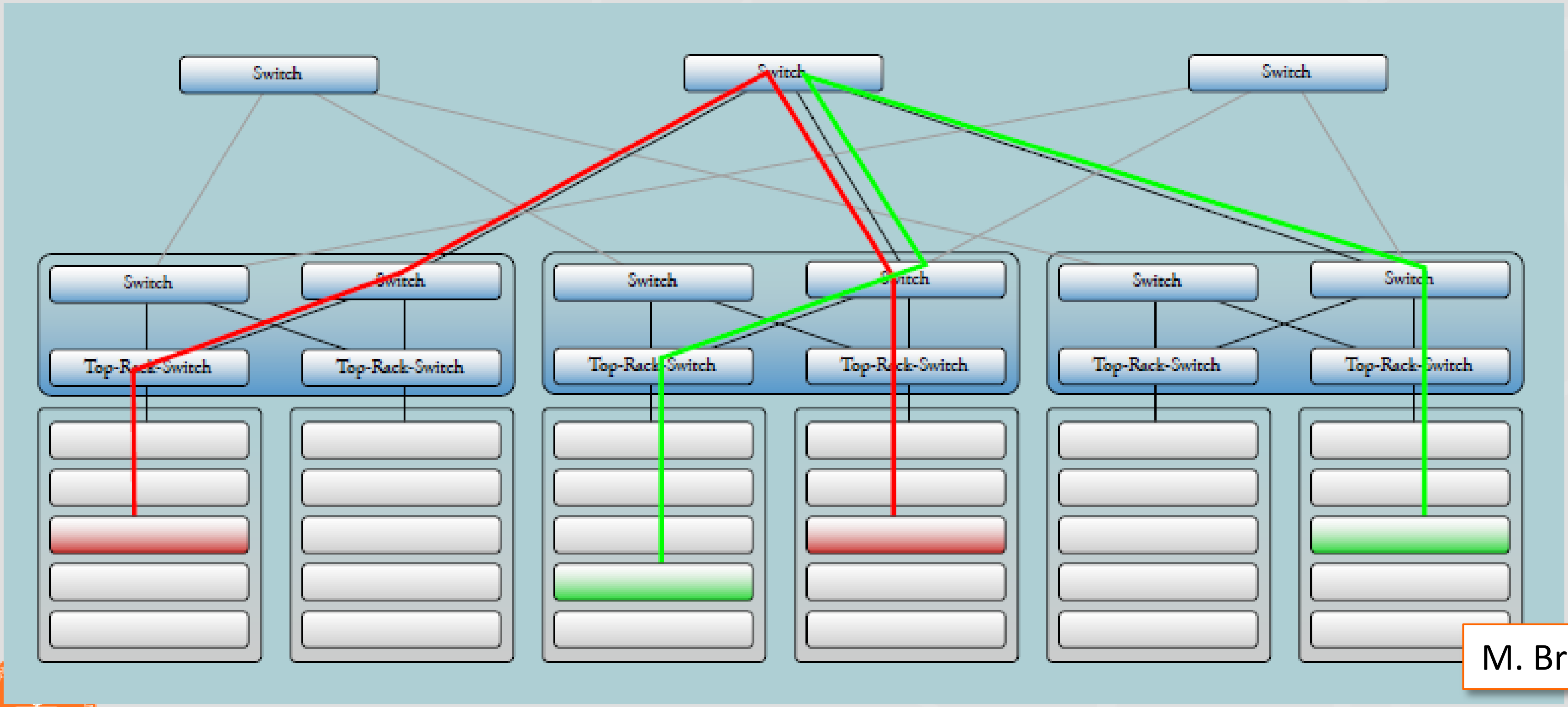
- Some challenges in multi-tenant large data centers are
  - scalability
  - change management in large/complex deployments
  - elasticity, fast
  - …



Data Center Example

CALTECH HEP NETWORKING US LHCNet

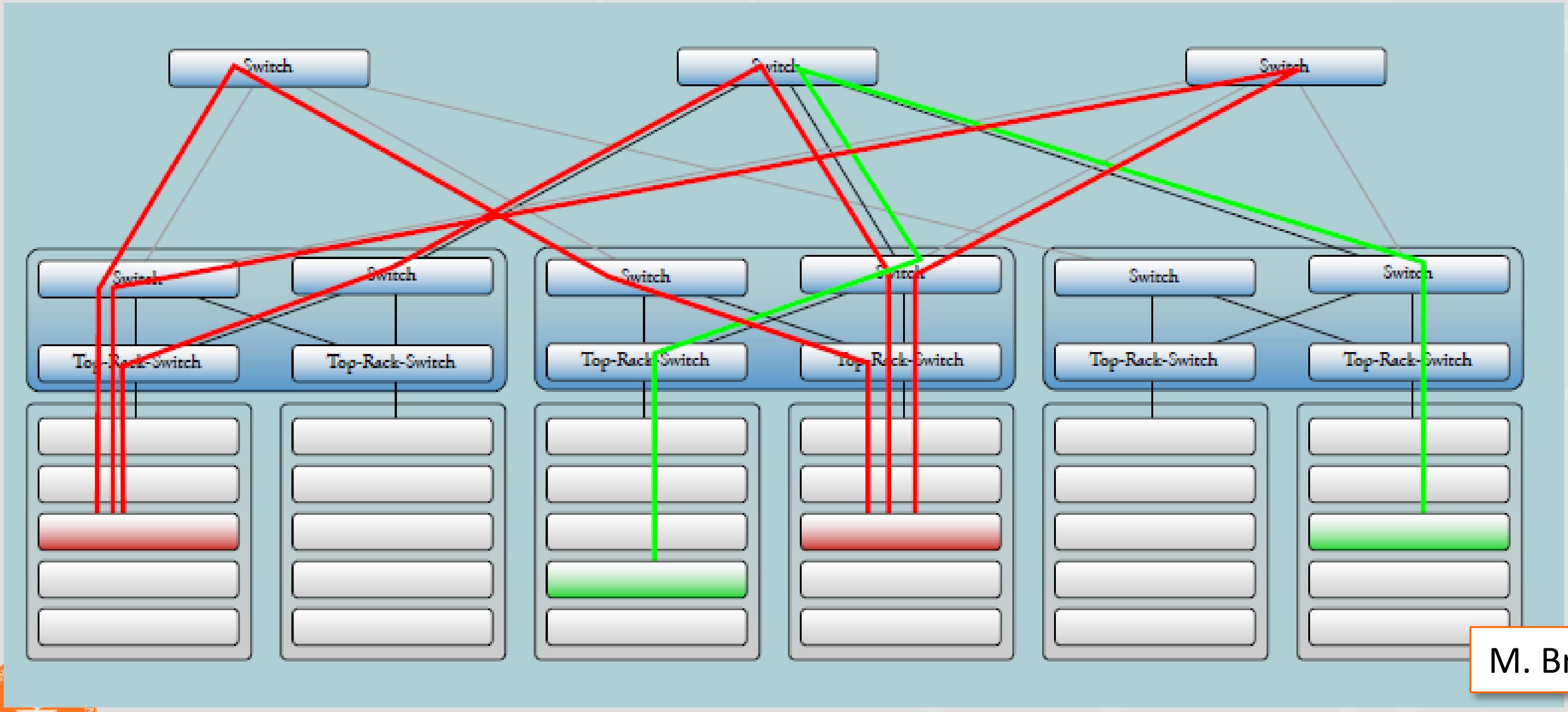
- Current techniques are limiting performance:
  - Spanning Tree for loop avoidance
  - LAGs are link-local
  - scaling up involves much configuration work on each involved device



M. Bredel

Multipath in Data Center

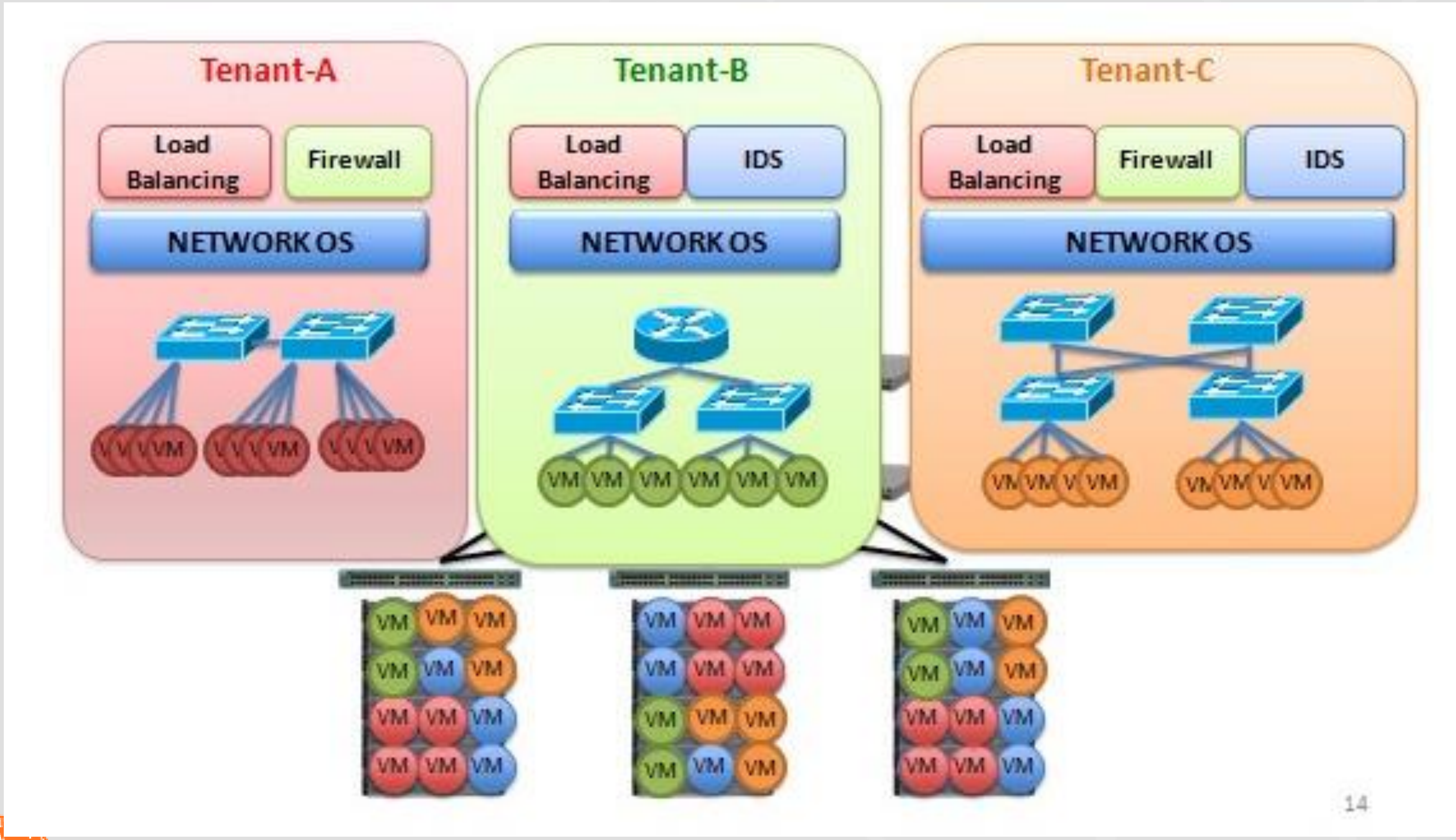
- Multipath can be achieved in several ways, e.g.
  - Multipath-TCP (IETF RFC 6824)
  - TRILL (IETF RFC 6325)
  - SPB (IEEE 802.1aq)
  - **And/Or Load Balancing algorithms in SDN!**



M. Bredel

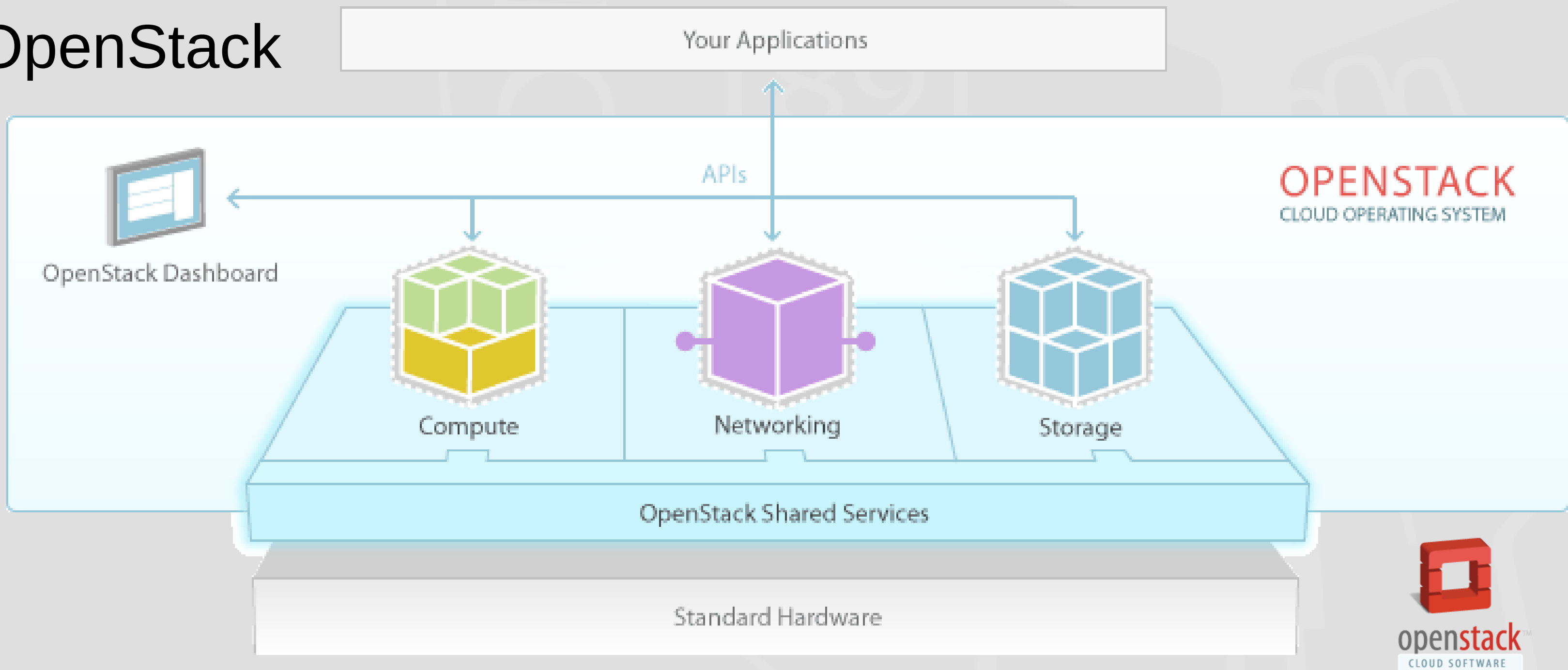
Example: Multi-Tenant Datacenter

- In addition, virtualization enables
  - host sharing
  - client-specific topologies



Example: Orchestration

- (Wikipedia: "…automated arrangement, coordination, and management of complex computer systems, middleware, and services")
- For full service deployment need to orchestrate Storage, Compute and Network resources
- E.g. OpenStack



OpenFlow – SDN’s favourite protocol

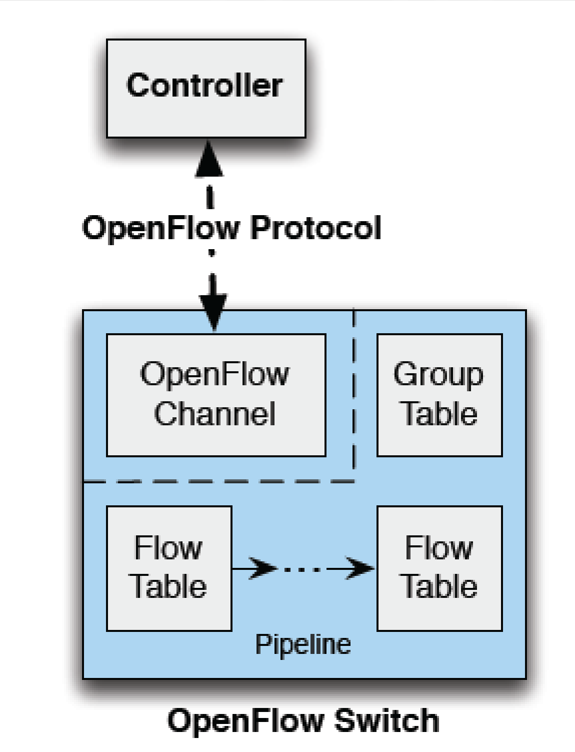
- OpenFlow ≠ SDN
  - – SDN is (technically, and with limitations) possible with SNMP, CLI, etc.

- OpenFlow = open standard for
  - – Protocol for controller – device communication
  - – Definition of packet processing in the switch

- Standardized by the Open Networking Foundation

OpenFlow switch components

- For packet look-up and forwarding
  - Flow Tables
  - Group Tables
- Control Channel
  - add, update, remove flow table entries
- OpenFlow Switch Ports:
  - Physical
  - Logical
    - e.g. LAG, tunnel, etc.
  - Reserved
    - ALL, CONTROLLER, TABLE, etc.



ONF http://www.opennetworking.org

Flow Tables

Match fields	Priority	Counters	Instructions	Timeouts	Cookie	Flags

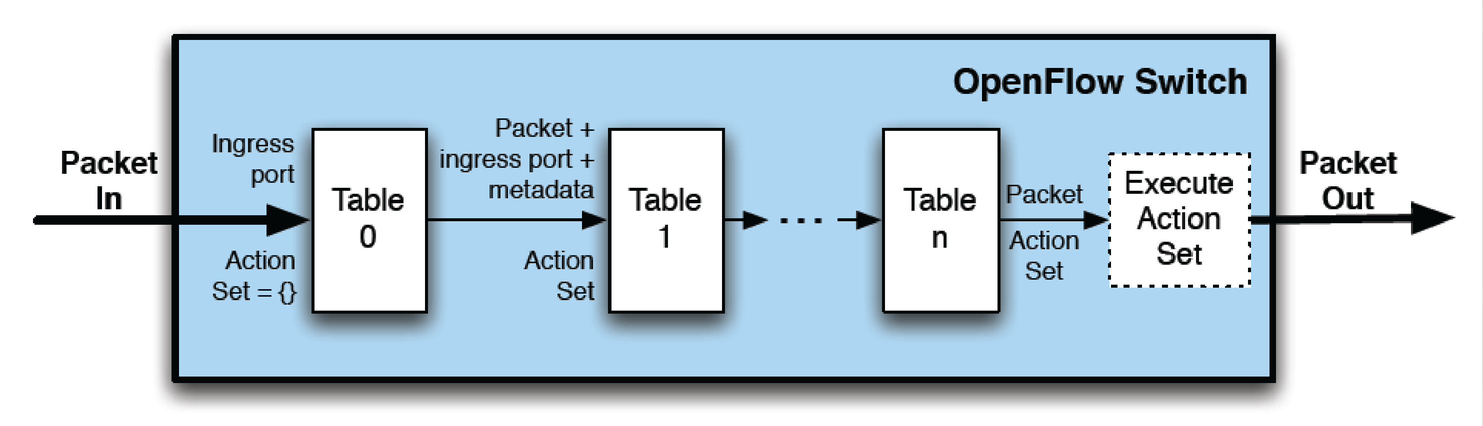
MAC src	MAC dst	IP src	IP dst	TCP dport	…	Count	Instructions	…
*	50:25:.	*	*	*		531	Out port 7	
*	*	*	1.2.3.*	80		77	local	
*	*	*	*	*	*	2755	Controller	*

- Matching fields at Layer 2, 2.5, 3 and 4
- Wildcards allowed
- Table miss entry - default action:
  - forward to controller, port or drop (default)

It is what the controller writes into the flow tables that determines the network behaviour

Table Pipeline

- Tables are processed in a pipeline



- For each table:
  1. Find highest priority matching flow entry
  2. Apply Instructions
     1. apply actions
     2. update action set
     3. update metadata
  3. Send match data and action set to next table

Instructions and Actions

CALTECH HEP NETWORKING US LHCNet

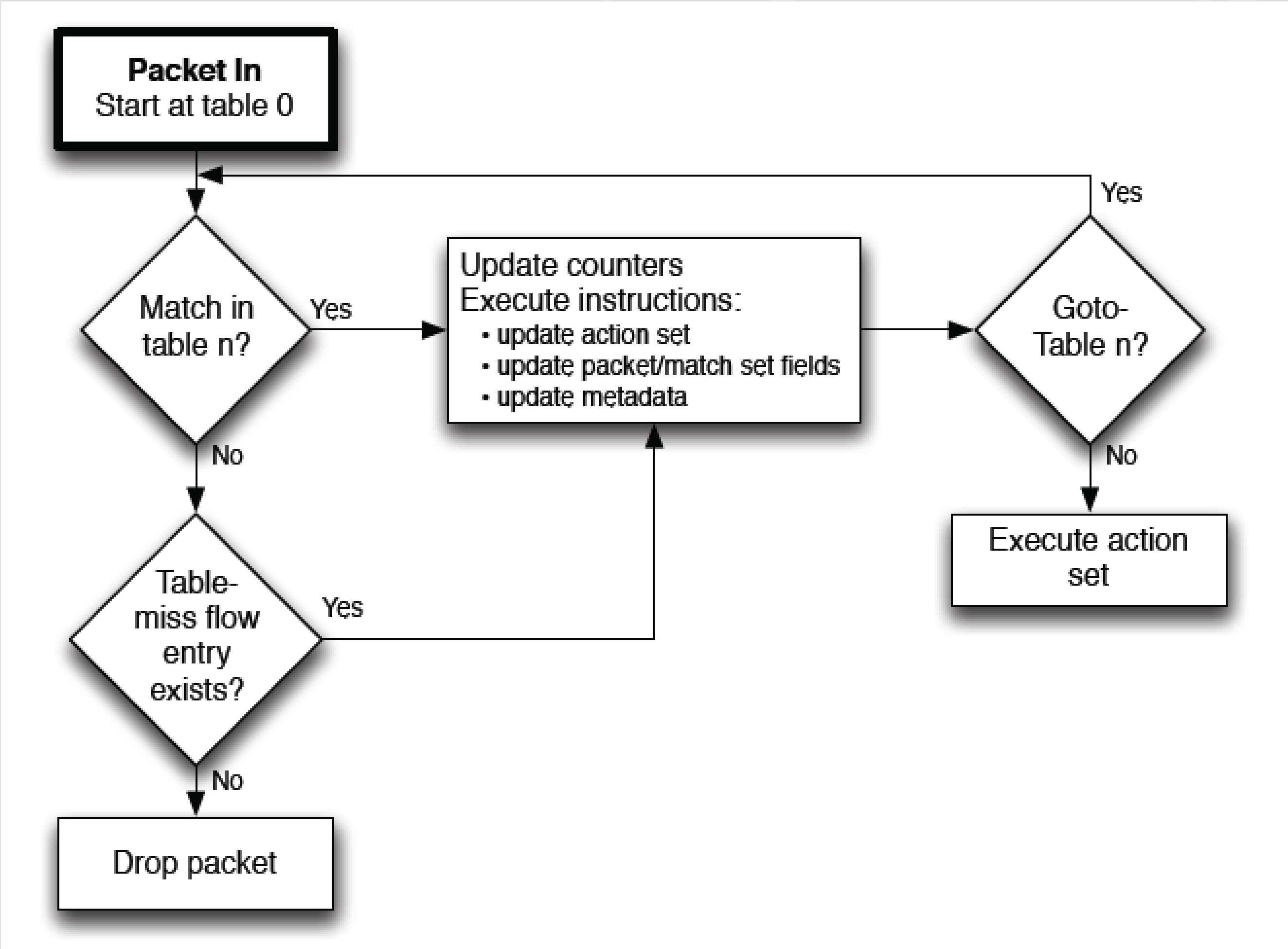
- (Some) Instructions:
  - Apply actions <actions>
  - Clear actions <actions>
  - Write actions <actions>
  - Meter <actions>
  - Goto <table>

- (Some) Actions:
  - Output <port>
  - Drop
  - Push tag
  - Pop tag

- Tags specified (v1.3) can be
  - VLAN
  - MPLS
  - PBB

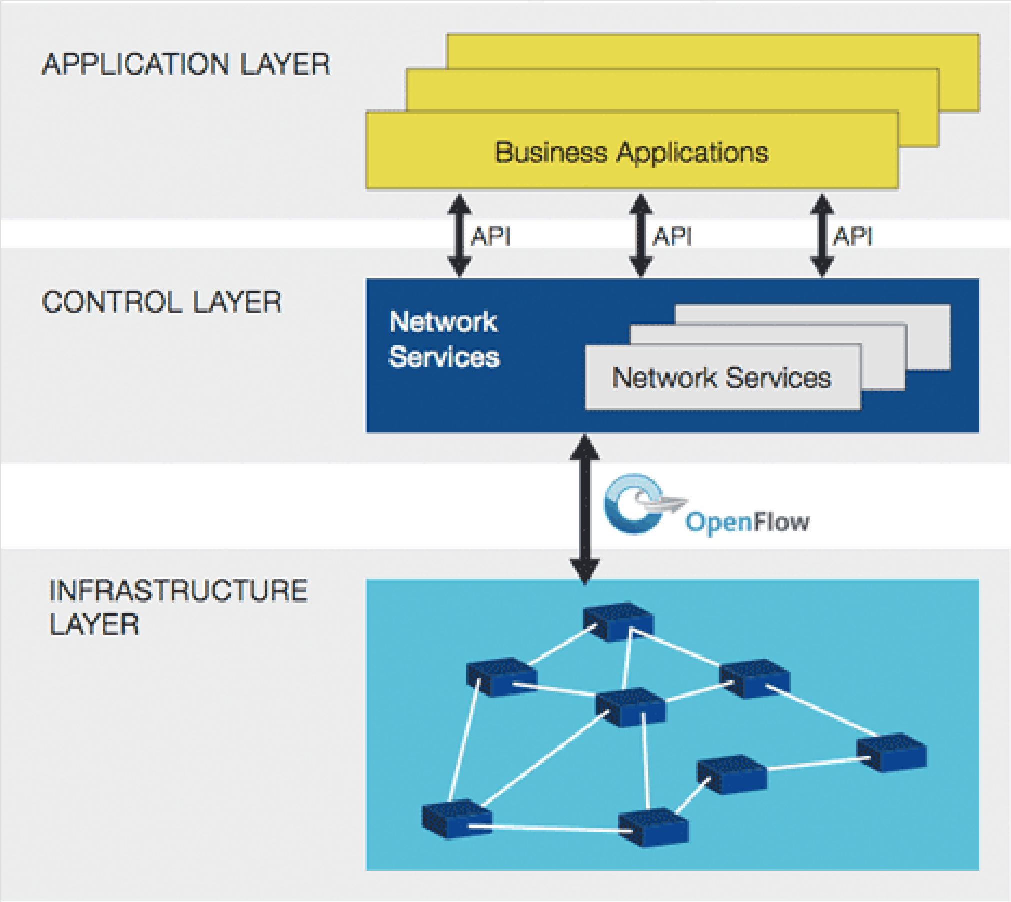
ONF OpenFlow Standard v1.3
For full document, see http://www.opennetworking.org

Packet Processing in OpenFlow Switch



OpenFlow - The Controller

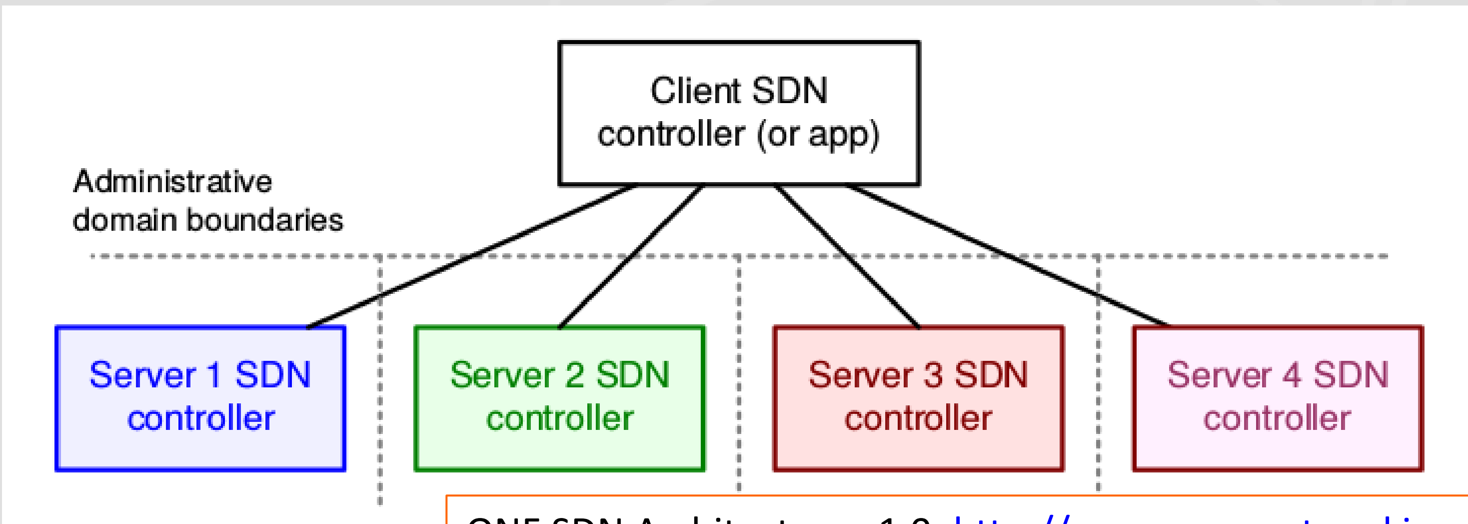
- Software typically running on commodity hardware
- Provides the API to user applications
  - Aka Northbound interface



http://www.opennetworking.org

Path to multi-domain SDN

- Not to forget: interactions between administrative domains



ONF SDN Architecture v1.0; http://www.opennetworking.org

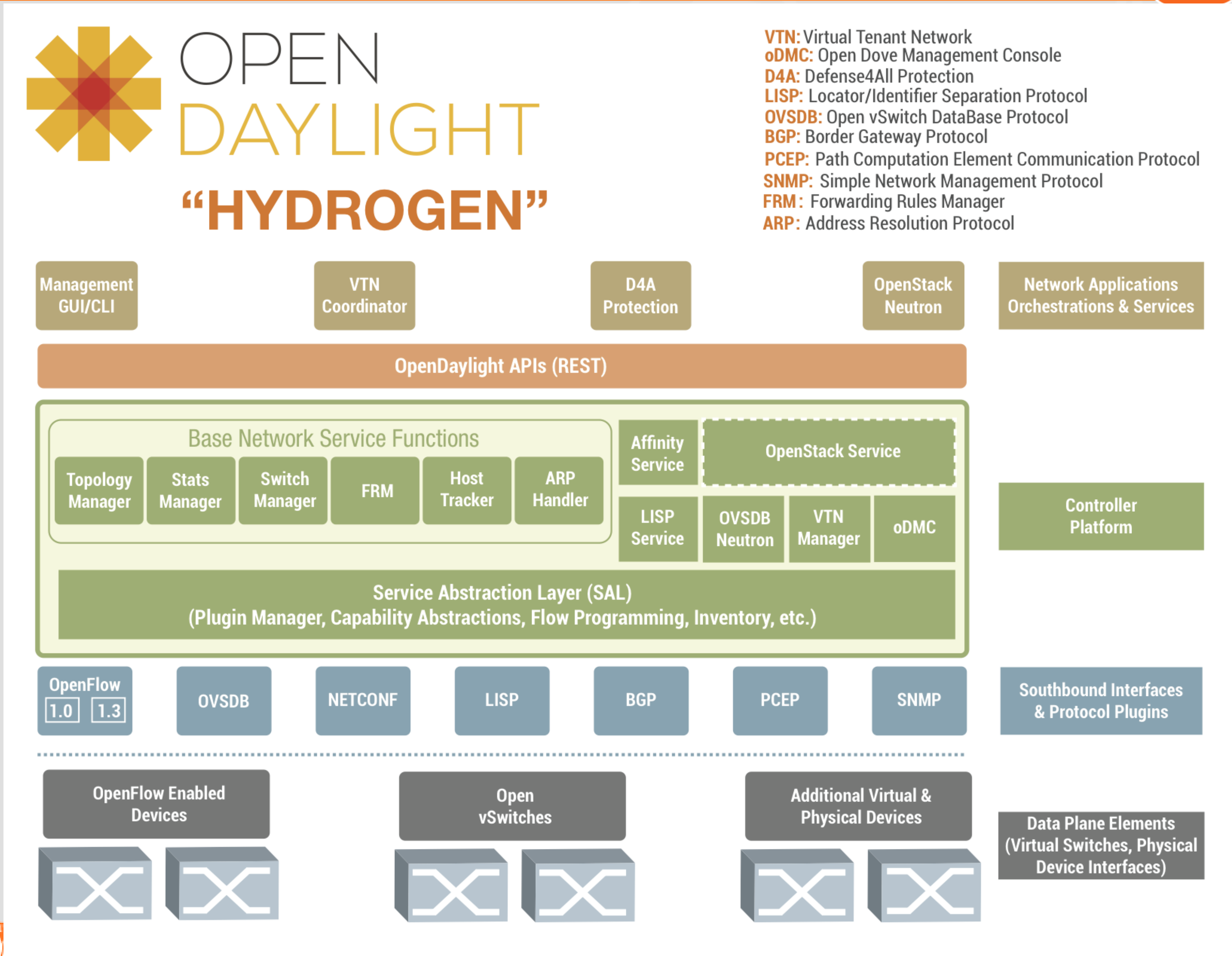
Popular Controller Examples

- NOX (C++)
  - http://www.noxrepo.org/
- POX (Python)
  - http://www.noxrepo.org/
- Ryu (Python)
  - http://osrg.github.io/ryu/
- Floodlight (Java)
  - http://www.projectfloodlight.org/floodlight/
- OpenDaylight (Java)
  - http://www.opendaylight.org/

OpenDaylight – Industry Driven

BROCADE, CISCO, CITRIX, ERICSSON, HP, IBM, JUNIPER NETWORKS, Microsoft, redhat, QOSMOS Your Network is Information, radware, VERSA NETWORKS, ZTE中兴, FUJITSU, guavus, H3C, HUAWEI, INOCYBE TECHNOLOGIES, intel, 6WIND, A10 Networks, ADVA Optical Networking, ARISTA, AVAYA, ciena, midokura, nuagenetworks, ORACLE, CONTEXTREAM, Coriant, CYAN, PANTHEON technologies, PLEXXI, PLUMgrid, DELL, Extreme networks, FLEXTRONICS, NEC, vmware

OpenDaylight – OpenFlow and beyond



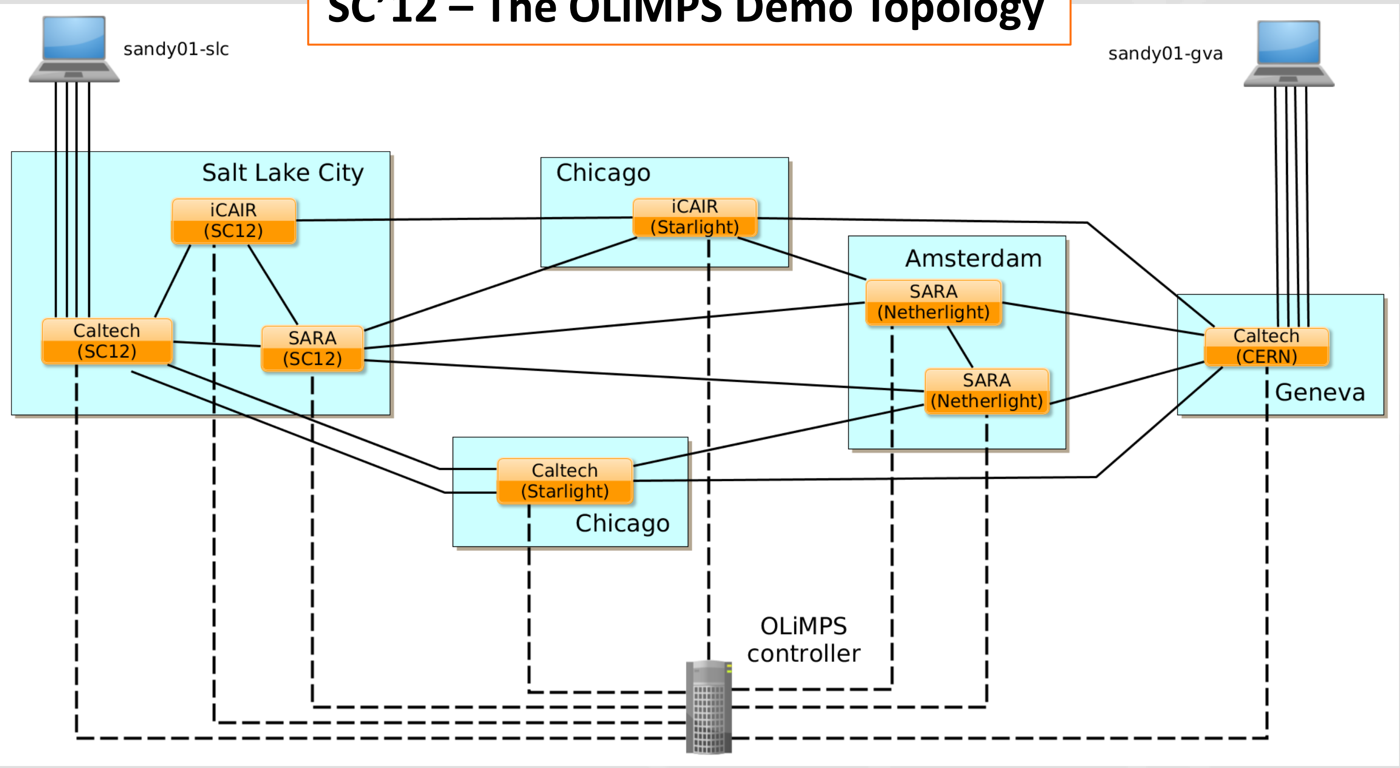
Some Important Components

- Northbound interface: REST and OSGi
- BGP-LS (BGP-Link State)
- PCEP (Path Computation Engine Protocol)
- Southbound interface supporting OpenFlow and non-OpenFlow devices

Application Example: Multipath Controller

- **O**penFlow **Li**nk-layer **M**ulti-**P**ath **S**witching, OLiMPS
- DOE funded project
- Extending capabilities of the Floodlight controller
- Load-balancing traffic over multiple possible end-to-end paths

SC’12 – The OLiMPS Demo Topology



Meshed Networks, Multipath

CALTECH HEP NETWORKING

US LHCNet

- Data connections (TCP) are point-to-point
- Classical IP routing constrains flows to a single path
- Reality: Networks are meshed, many paths
  possible, only one used

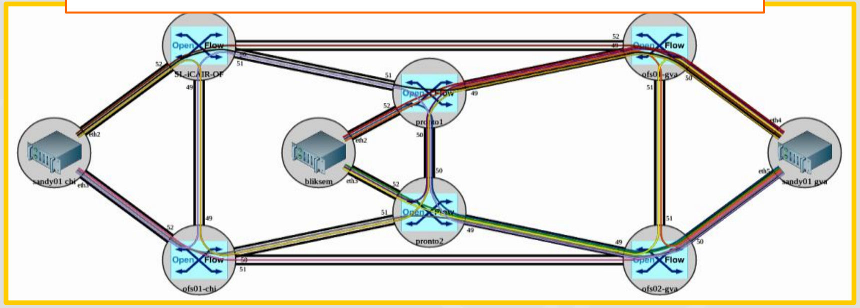
- Multipath forwarding helps increasing network efficiency

- Application “telling” the network controller its intentions increases efficiency even further

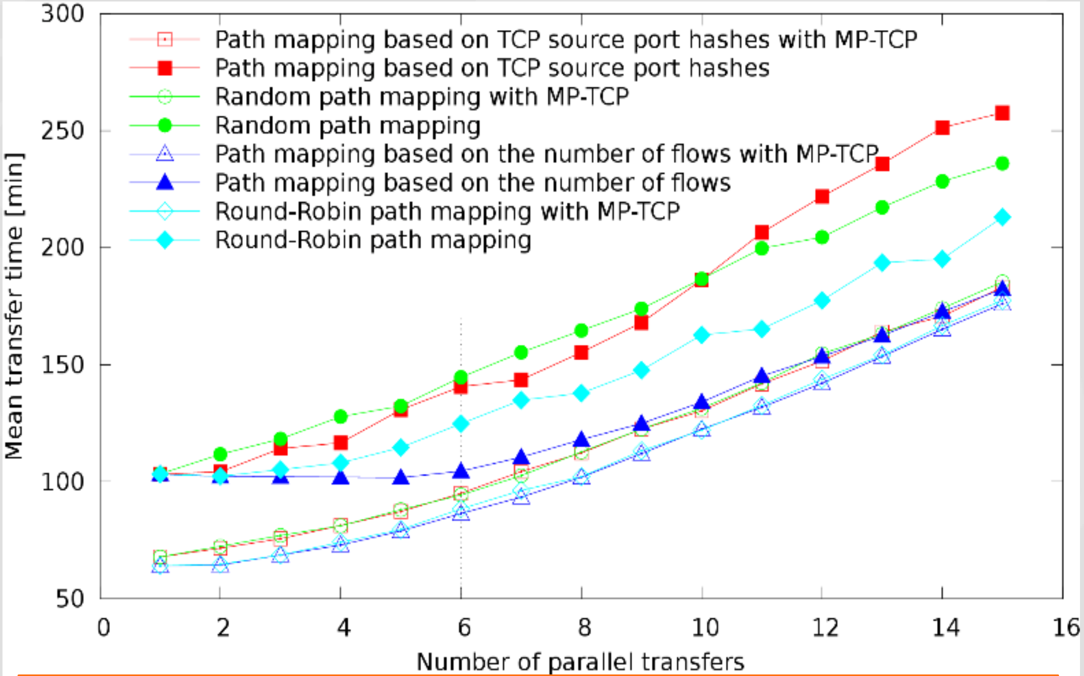
- Implemented using Floodlight controller

- Paper to be presented at HotSDN 2104

Example of WAN demo topology



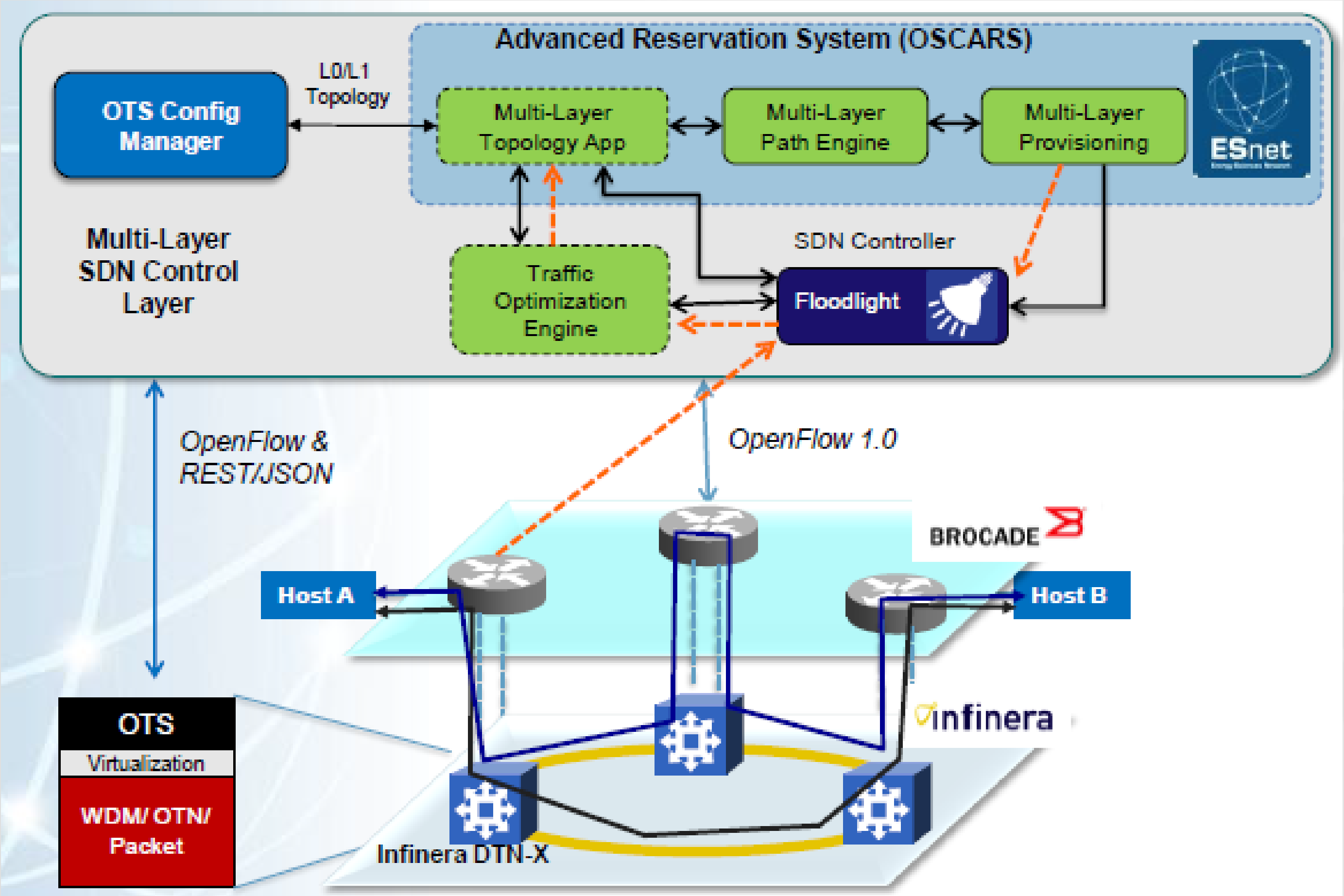
From OLiMPS project – multipath with openflow



Local testbed with Application-Network interface

SDN + Dynamic Circuits I

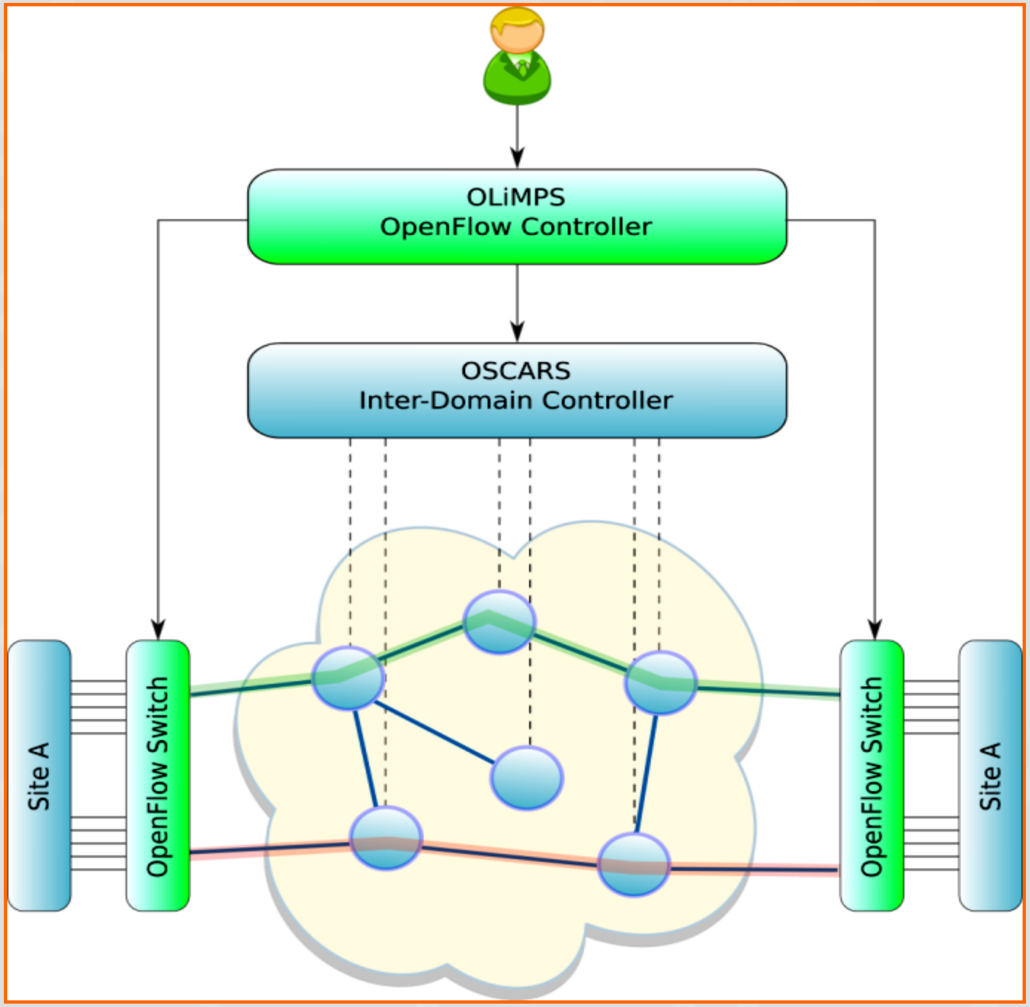
- ESnet's OSCARS management system using OpenFlow controller for traffic optimization



Demo: http://www.sdncentral.com/events/brocade-infinera-esnet-sdn-demo/

SDN + Dynamic Circuits II

- Caltech's OLiMPS project created an interface between Floodlight OpenFlow controller and the OSCARS dynamic circuit system
- Additional capability of the controller:
  Create additional paths between OpenFlow devices
- I.e. create a topology optimized to the load distribution in the network
- Fits OpenDaylight architecture



SDN Summary

- SDN provides a new possibility for programmatic network interaction

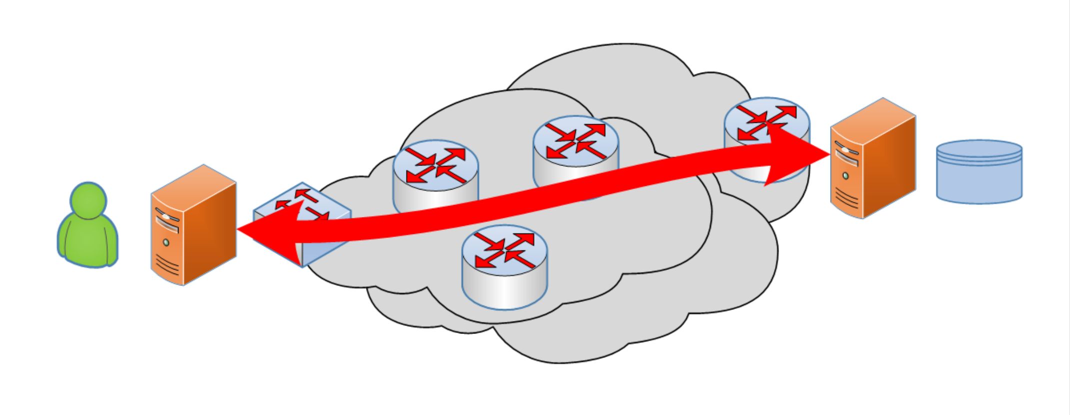
- HEP computing should be involved in defining services provided by the networks, built on SDN

CONTENT CENTRIC NETWORKING

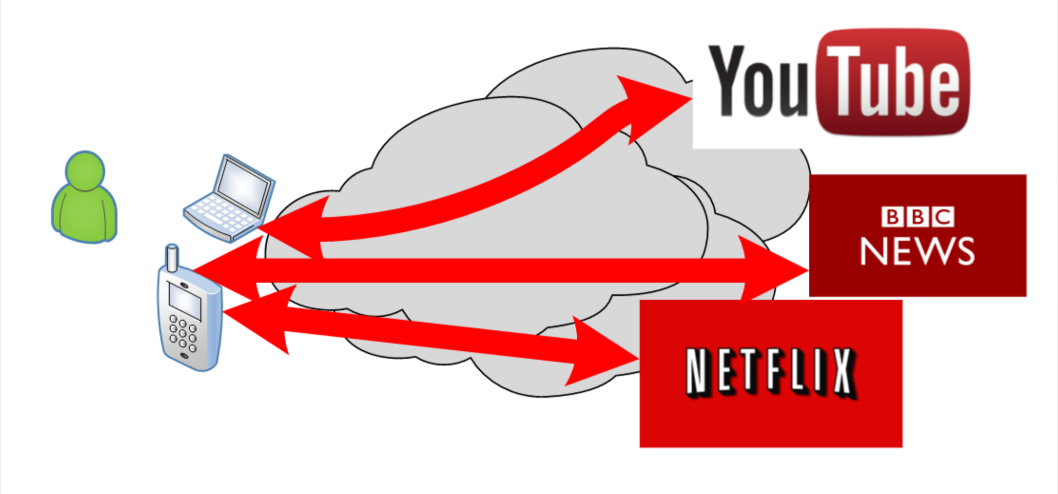
Where we meet CCN, NDN and friends

CCN Background

- When Internet was invented, it was connecting resources
  - TCP/IP: point-to-point connections between two entities
  - IP: delivering packets to destination **hosts**



- Today's applications, ours included, care about **content**



CCN Background, cont.

- Applications deal with “what”, while the network deals with “where”
  - Lots of middleware needed to match these
  - Web services, CDNs, P2P, …

- Complexity arises when dealing with failover, security, etc.
  - E.g. if the server at A.B.C.D does not respond, it’s the application to react and possibly find a backup source for the data
  - E.g. you trust the server, but it’s the content that’s potentially dangerous

- Lot of the work in CDNs, redirection, caching deals with this mismatch

- Can we do a better design instead?
- Identify data rather than hosts?

The Future Internet

- CCN is one of the **Future Internet Architectures** being developed and studied
- Specific projects include
  - Content-Centric Networking – CCN
    - Project at PARC
    - Code base developed: CCNx
  - Named Data Networking – NDN
    - NSF funded project since 2010, recently extended
    - Collaboration including PARC
  - and several other similar projects

- I will focus on the Named Data Networking (NDN) project in the following

Named Data Networking

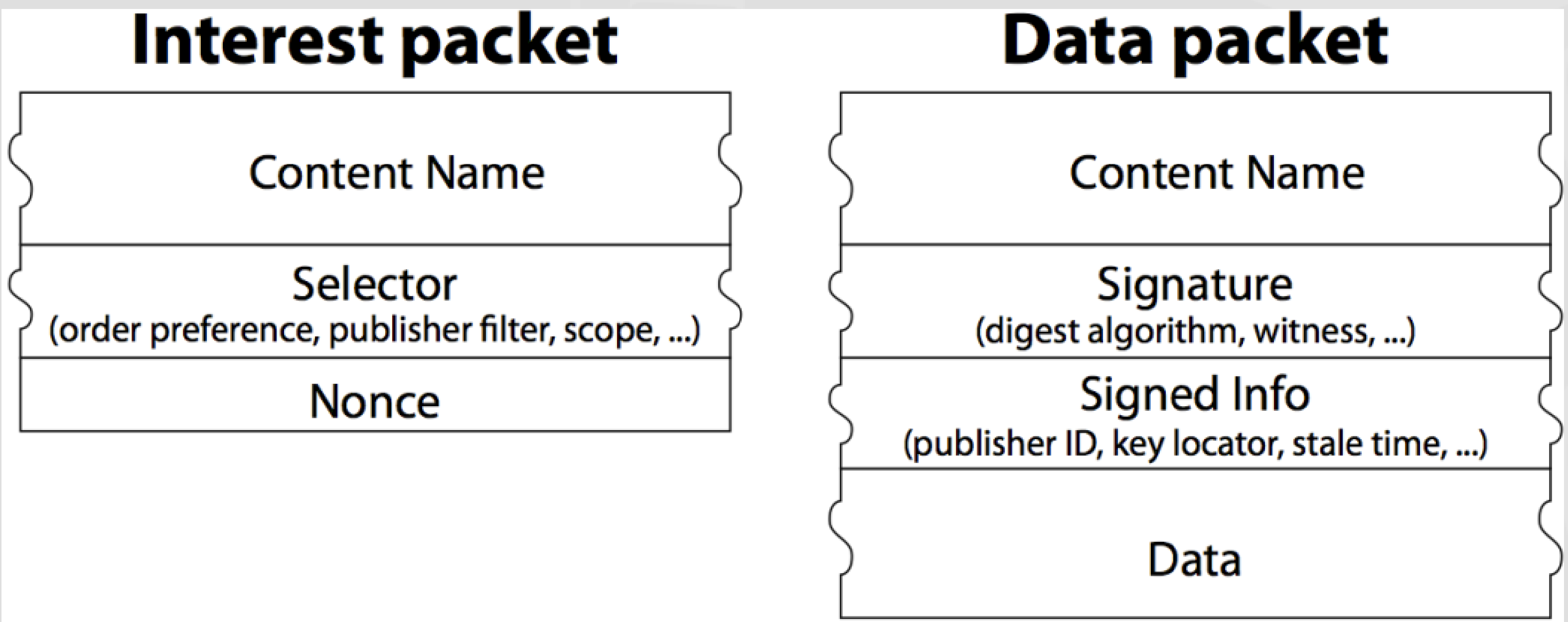
- Basic Principle: Name Data instead of naming end-hosts

- Today's Internet delivers packets to a destination address
- NDN delivers content identified by a given name to the client

- This is a basic change in semantics of the network service

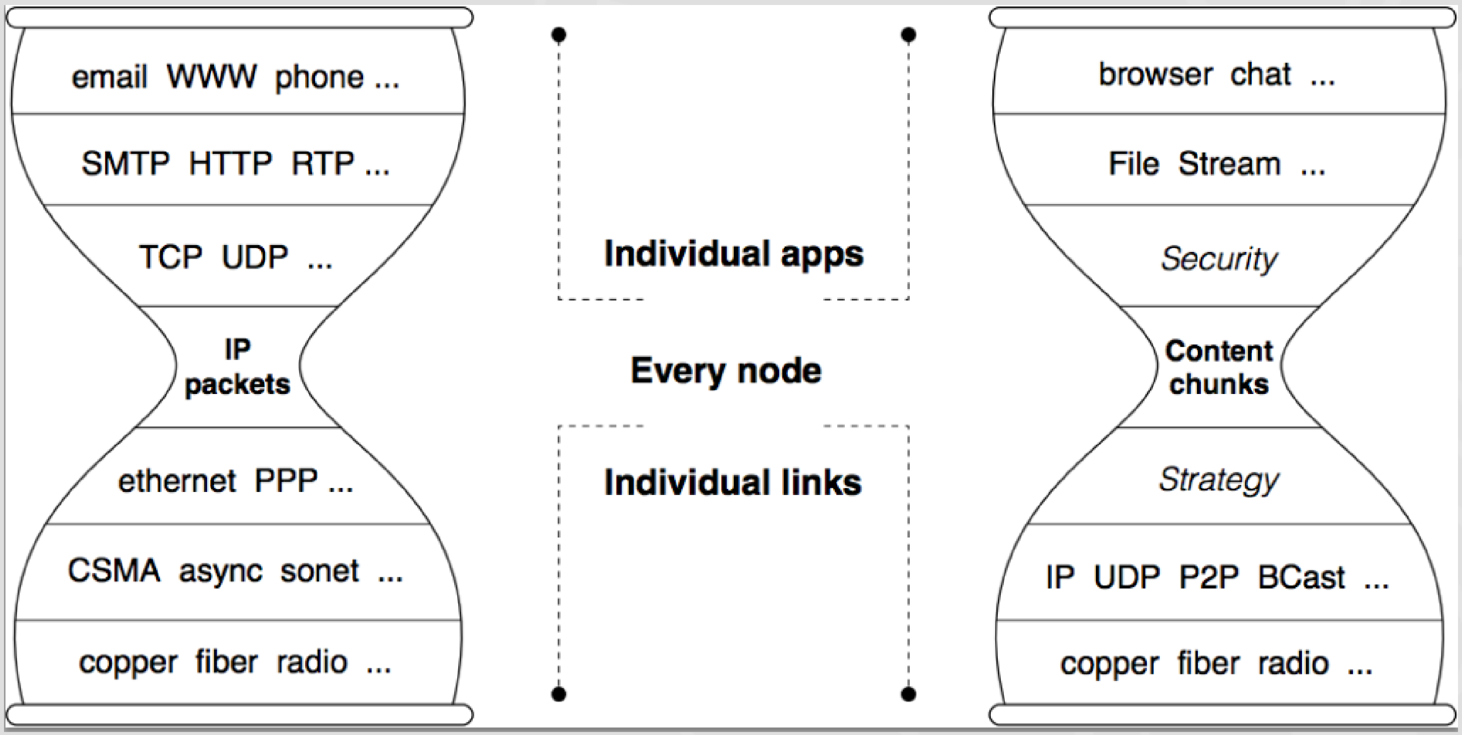
Two NDN packet types

- Communication is driven by the receiving end
- **Interest Packets:**
  - Sent out by the data consumer, identifies desired data
- **Data Packets:**
  - Sent back by the node which has the desired content



Protocols

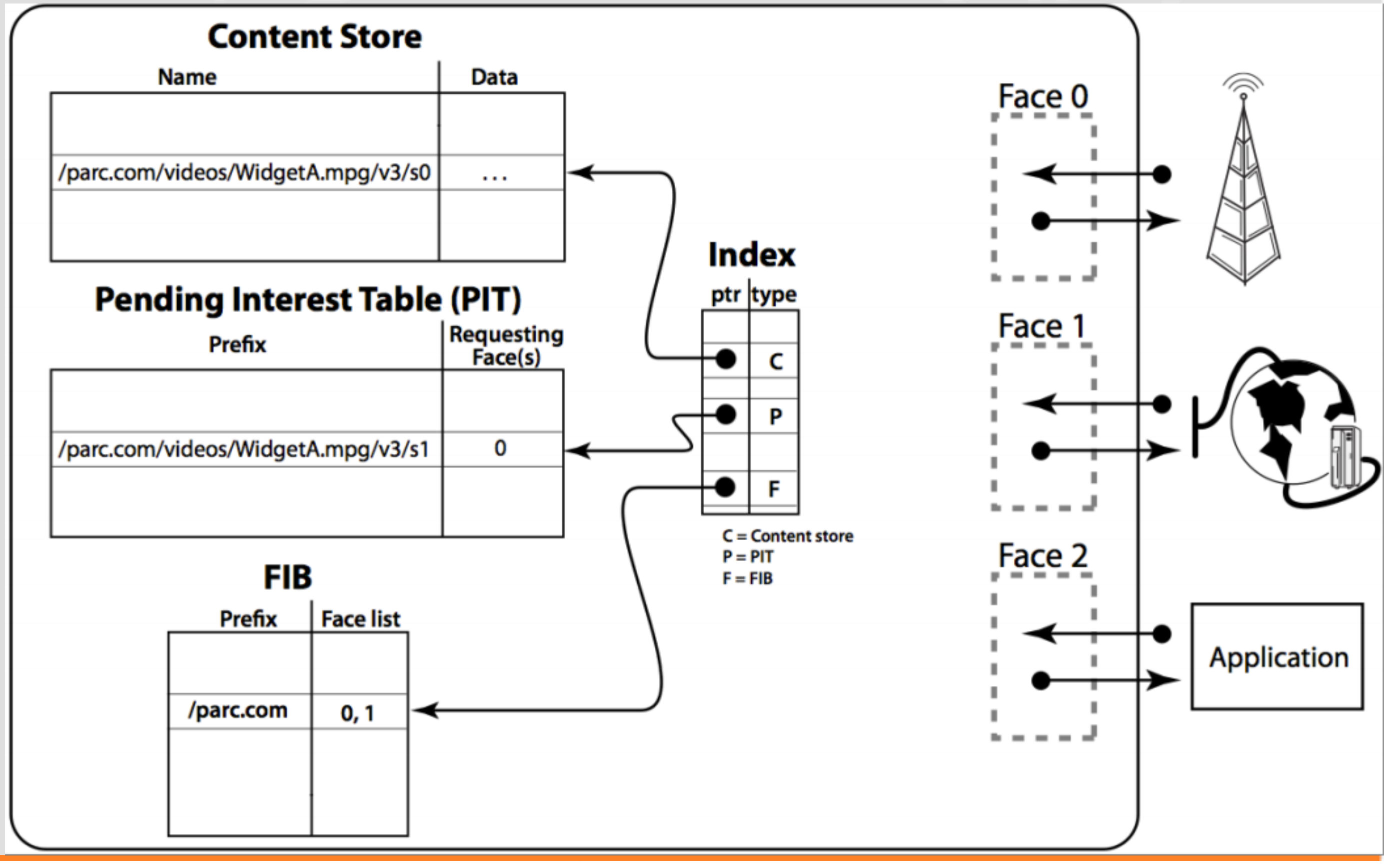
- Basis for NDN:
  - Named data replaces named end-points
  - Keeping the thin waist approach



TCP/IP NDN

Network node operation

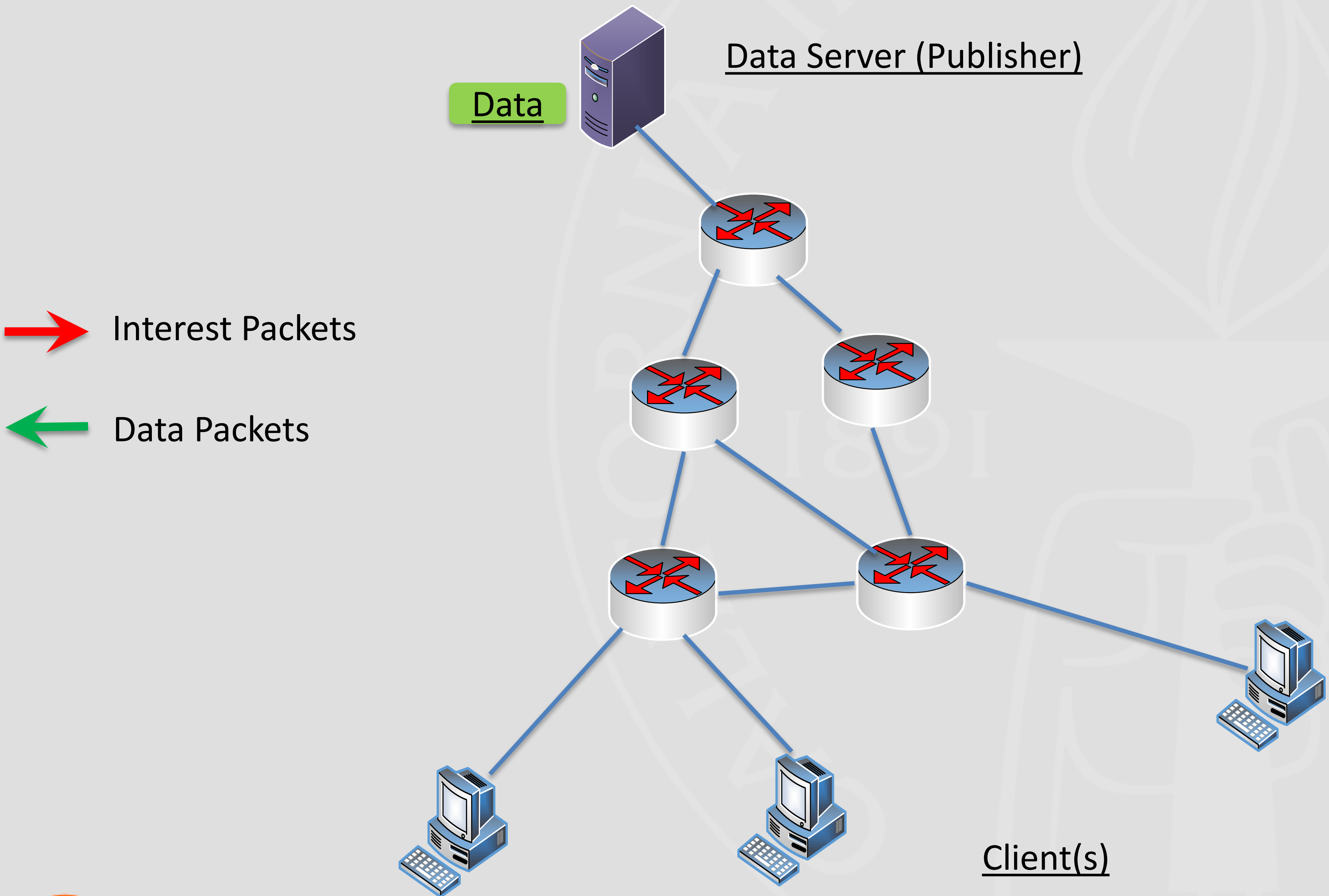
- NDN routers remember the interface the request came in the PIT
- Forwards the Interest Packet looking up the name in the FIB
  - Populated by routing protocol
- Once the Interest Packet reaches a node that has the content, a Data Packet is sent back following the reverse path (as stored in the PIT)



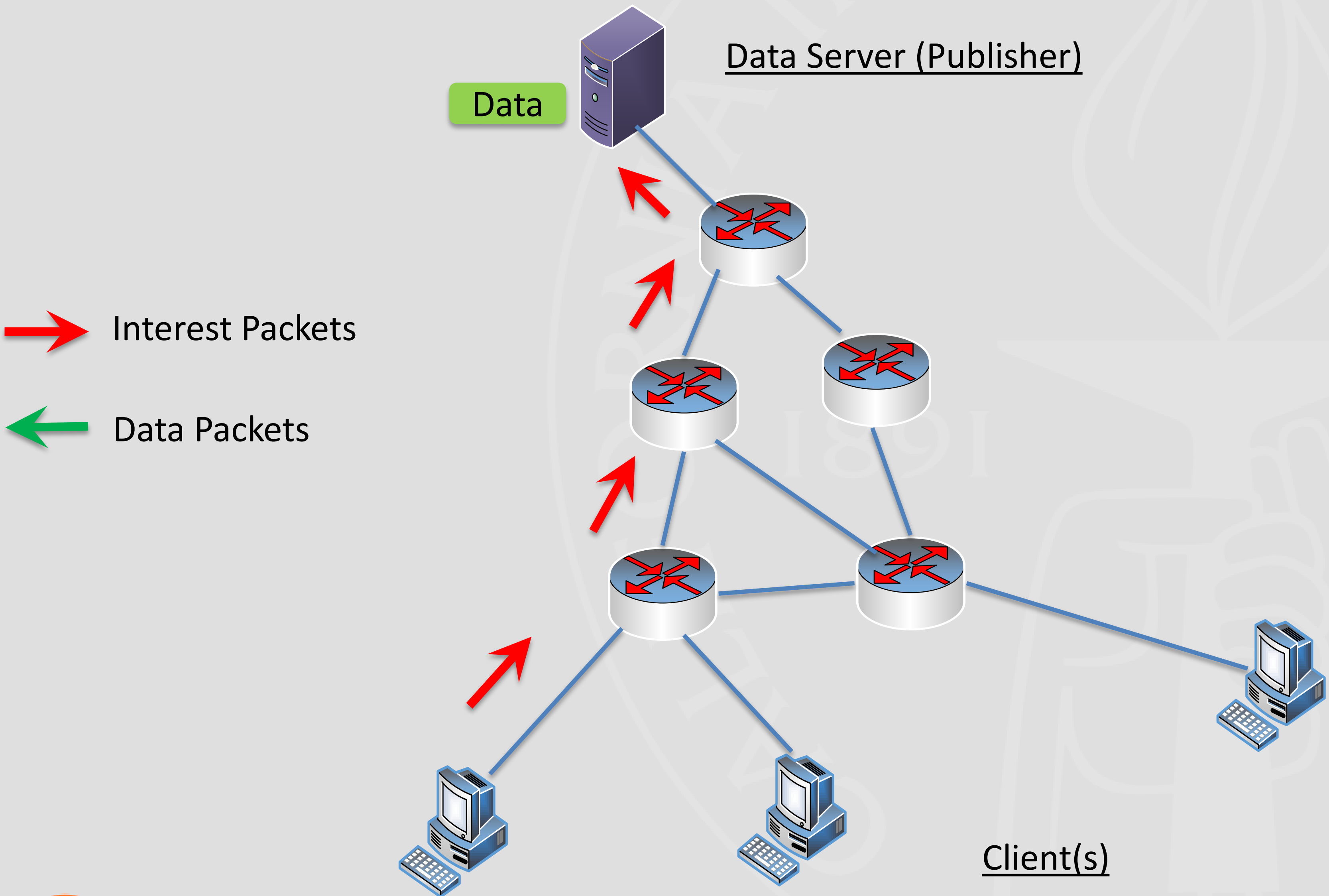
Better-than Multicast

- Data Packets are cached in the routers' Content Store
- Data Packets are then forwarded to all interfaces with registered interest in the router's PIT
  - i.e. if multiple IPs received for same content – multicast!
- When next interest packet arrives for a named data in the content store (cache), a Data Packet is sent from the router, rather than forwarding the IP to the data source
- This provides for additional multicast-like operation
- With one big difference: no multicast request or protocol is necessary
- Added benefit: because of caching, destinations do not have to be sychronised - fits a pull model as opposed to push

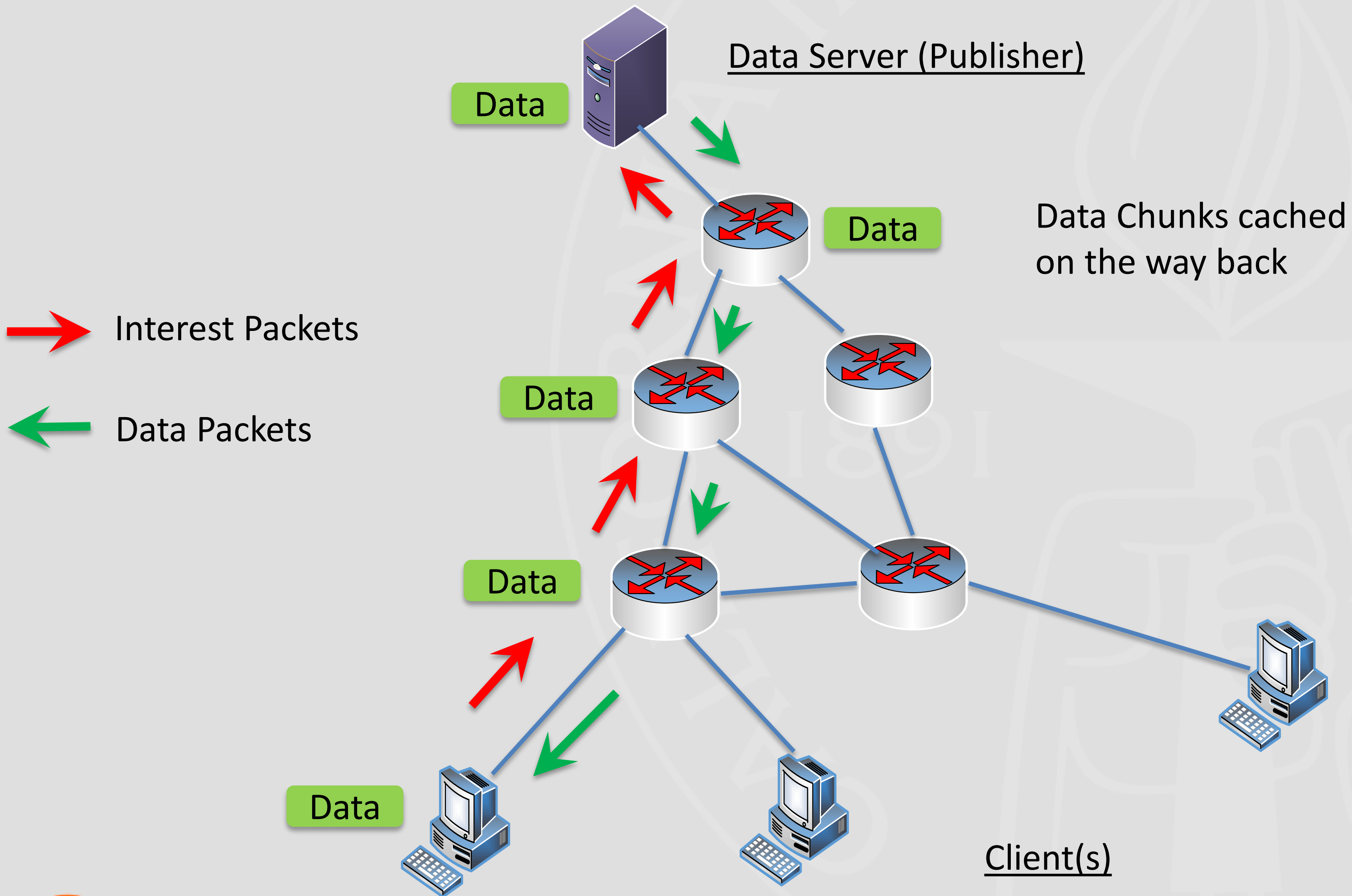
NDN Operation



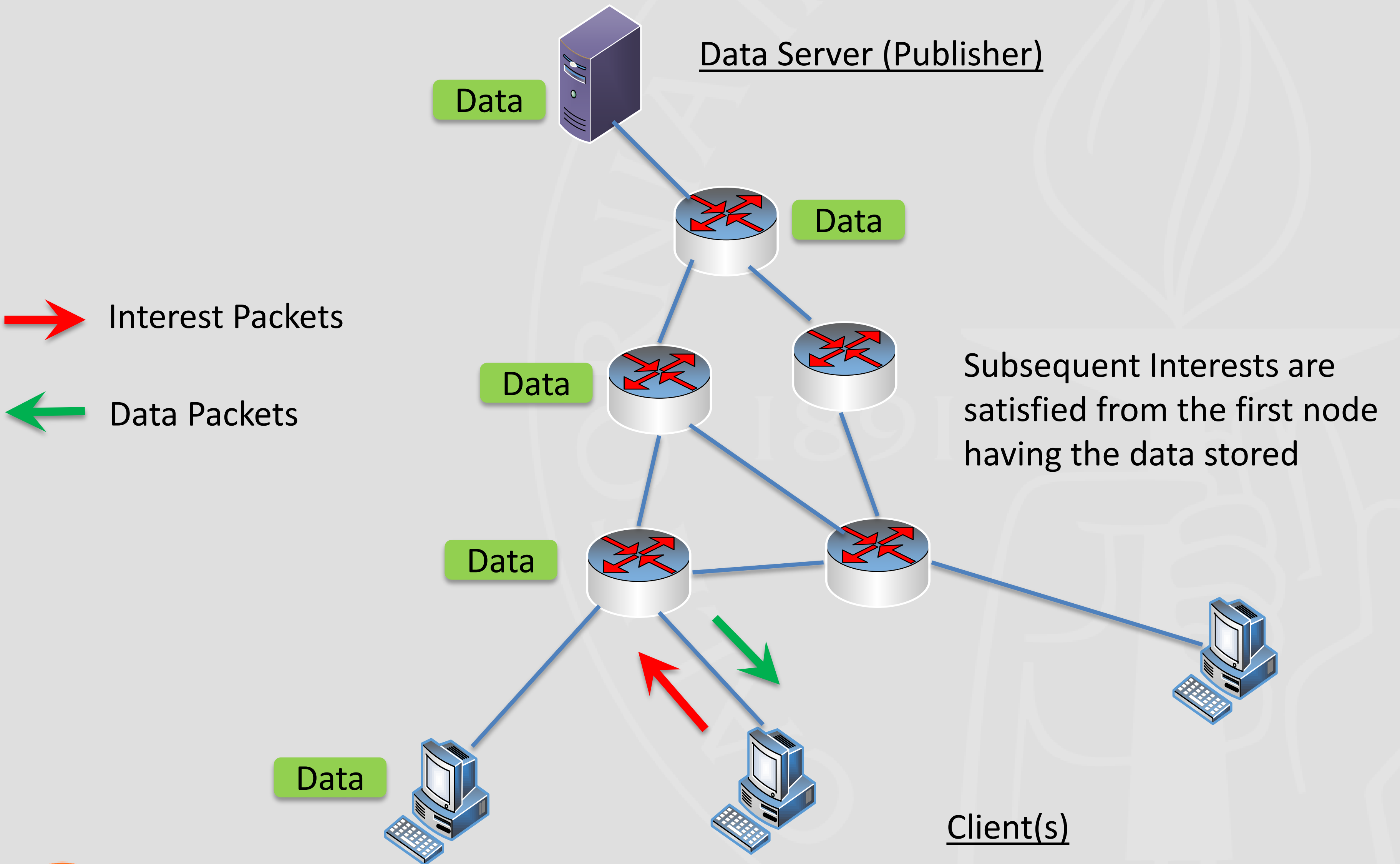
NDN Operation – New request



NDN Operation – New request



NDN Operation – New request



Why should we investigate use of NDN?

- A potential candidate technology to solve several issues, but do so at the network layer:
  - Optimal data distribution
  - Data caching
  - Popularity based data placement
  - Latency optimization for remote data access

- NDN could simply be the way the Internet works in the future

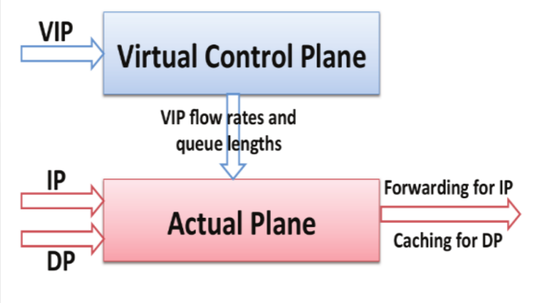
- How will this change the way we access and process data?

Some topics for investigation

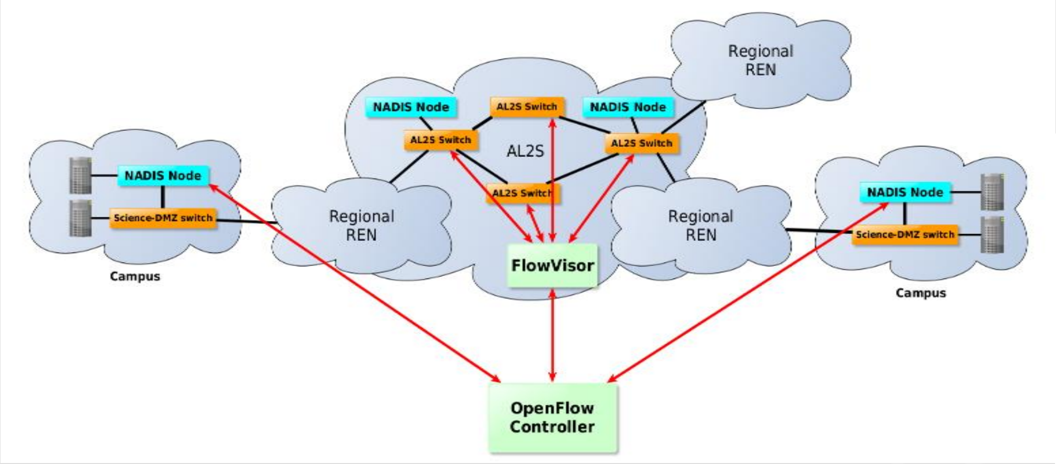
- Caching strategies
  - Could we reduce the storage at end-sites to only permanent copies?
  - Rely on caching in the network?
- What is the correct data chunk?
  - File? Block? Event?
- Bulk data transfer strategies
  - E.g. multipath, multi-source, multicast
- Multipath forwarding
- Network-Application interface
  - Sockets?
  - Calendaring?
- Impact on workflows and job scheduling
  - Reduced latency through caching
  - Rely on remote data access as default?
- QoS and flow prioritization
- …

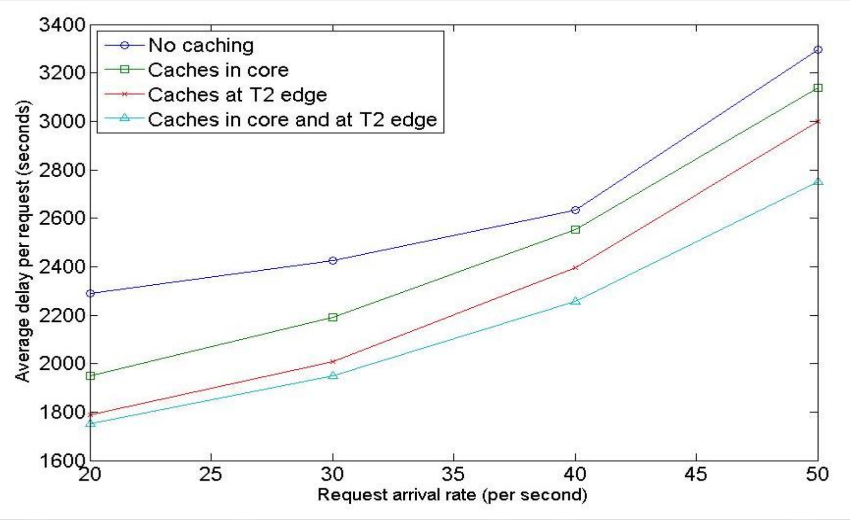
Combining SDN and NDN

- For starters, SDN can be an easy way to create a high performance NDN test bed
- In which we want to investigate a possible design suitable for HEP data (and other data intensive science fields)



Caltech and Northeastern proposal to NSF:
NDN Architecture for Data Intensive Science (NADIS)





SDN/NDN - Summary

- Software Defined Networking provides a powerful way to interact with the network
  - Needs engagement and collaboration with the network operators

- Named Data Networking is a fresh approach at the design of the Internet of the future
  - Designed with the content rather then end-points as basis for communication
  - Has many features which can benefit LHC data processing
  - Despite it being rather new, basic implementation and a test bed are available
  - **The underlying ideas match very well with distributed data and computing models as in HEP computing**
  - Impact on the LHC data processing models needs careful study

NETWORK TESTBEDS

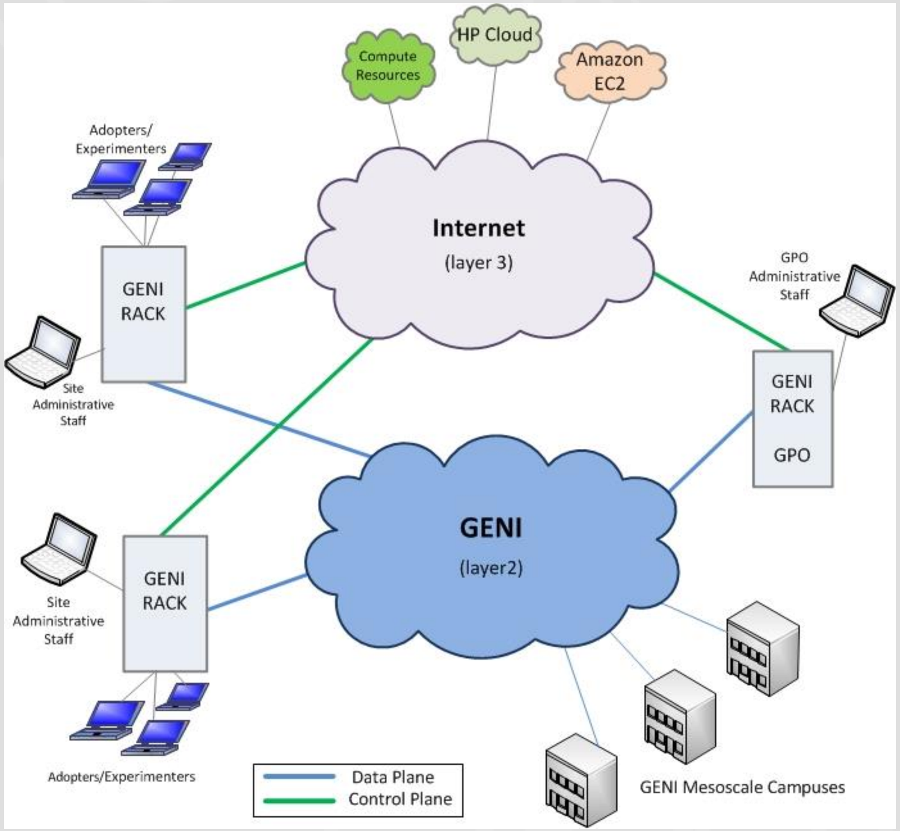
A non-exclusive list of examples for people interested in practical network innovation

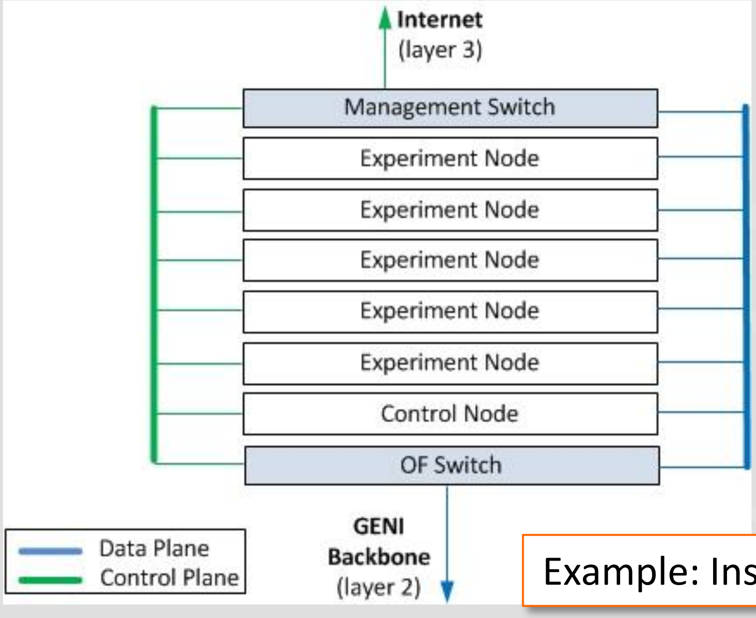
Network Testbeds

- If you want to test new network ideas, several testbeds might be available:
  - GENI – generic network testbed, mostly US
  - OFELIA – European Openflow testbed
  - GEANT OpenFlow test facility
  - FELIX – EU-JAPAN testbed for FI research
  - …

GENI

- Global Environment for Network Innovations
- “virtual laboratory for networking and distributed systems research and education”
- Mainly US based initiative, but not only
- GENI racks (3 “models”) installed on several campuses



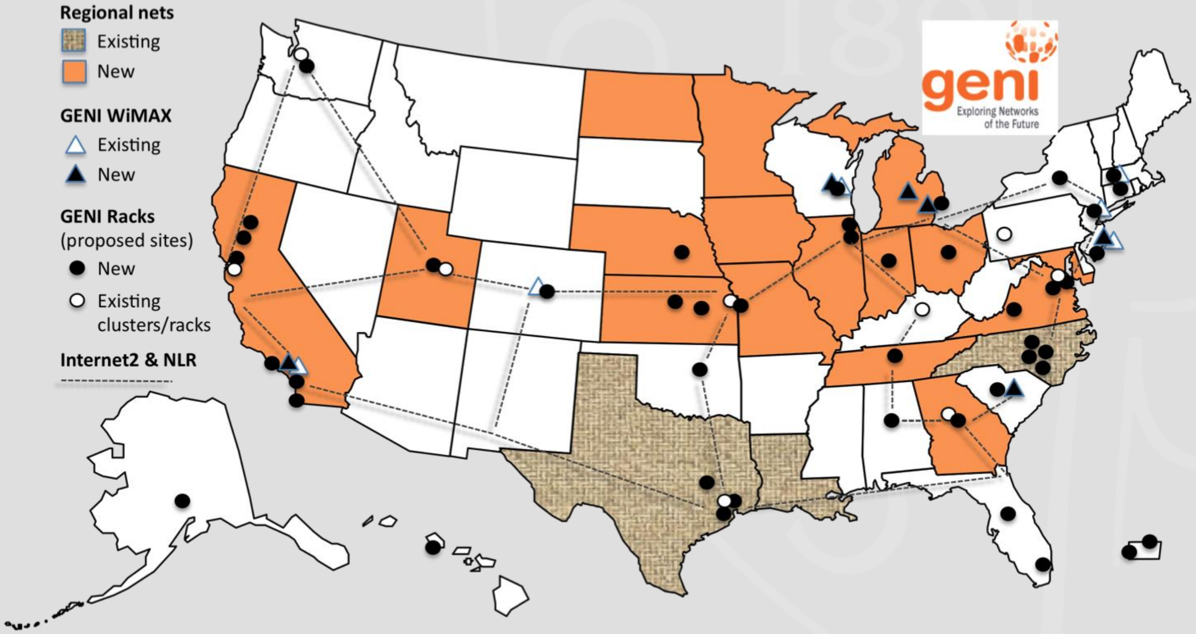


Example: InstaGENI rack layout

http://www.geni.net

GENI, cont.

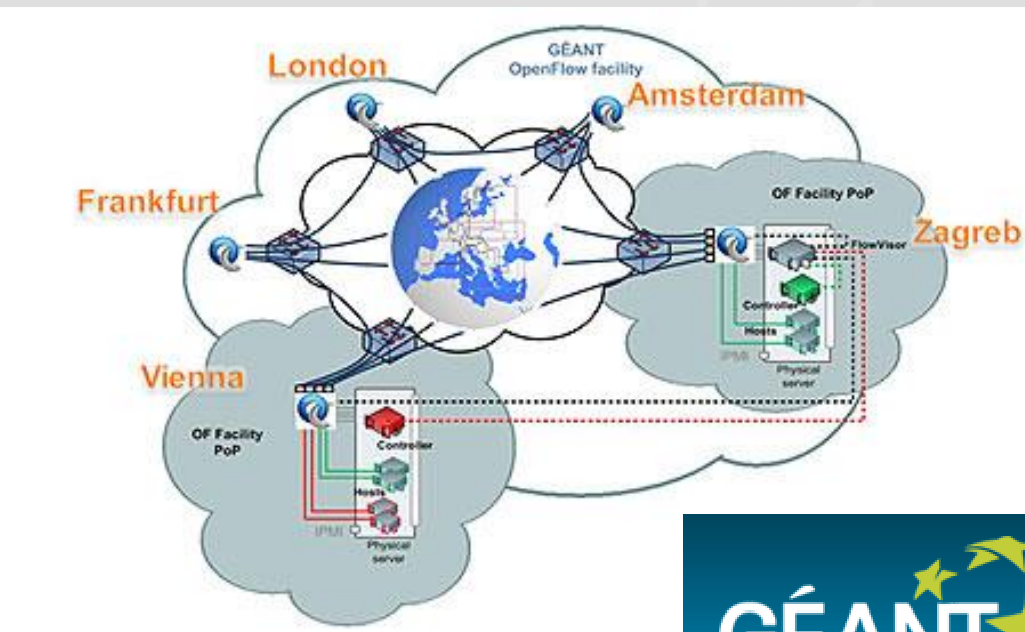
- GENI allows experimenters to:
  - Obtain compute resources from locations around the US
  - Connect compute resources using Layer 2 networks in topologies best suited to their experiments
  - Install custom software or even custom operating systems on these compute resources
  - Control how devices in their experiment handle traffic flows
  - Run their own Layer 3 and above protocols



http://www.geni.net

OFELIA/GEANT

- European FP7 Project
- European OpenFlow Testbed Facility
- Project ended in 2013, but the GEANT Openflow Facility continues



FELIX

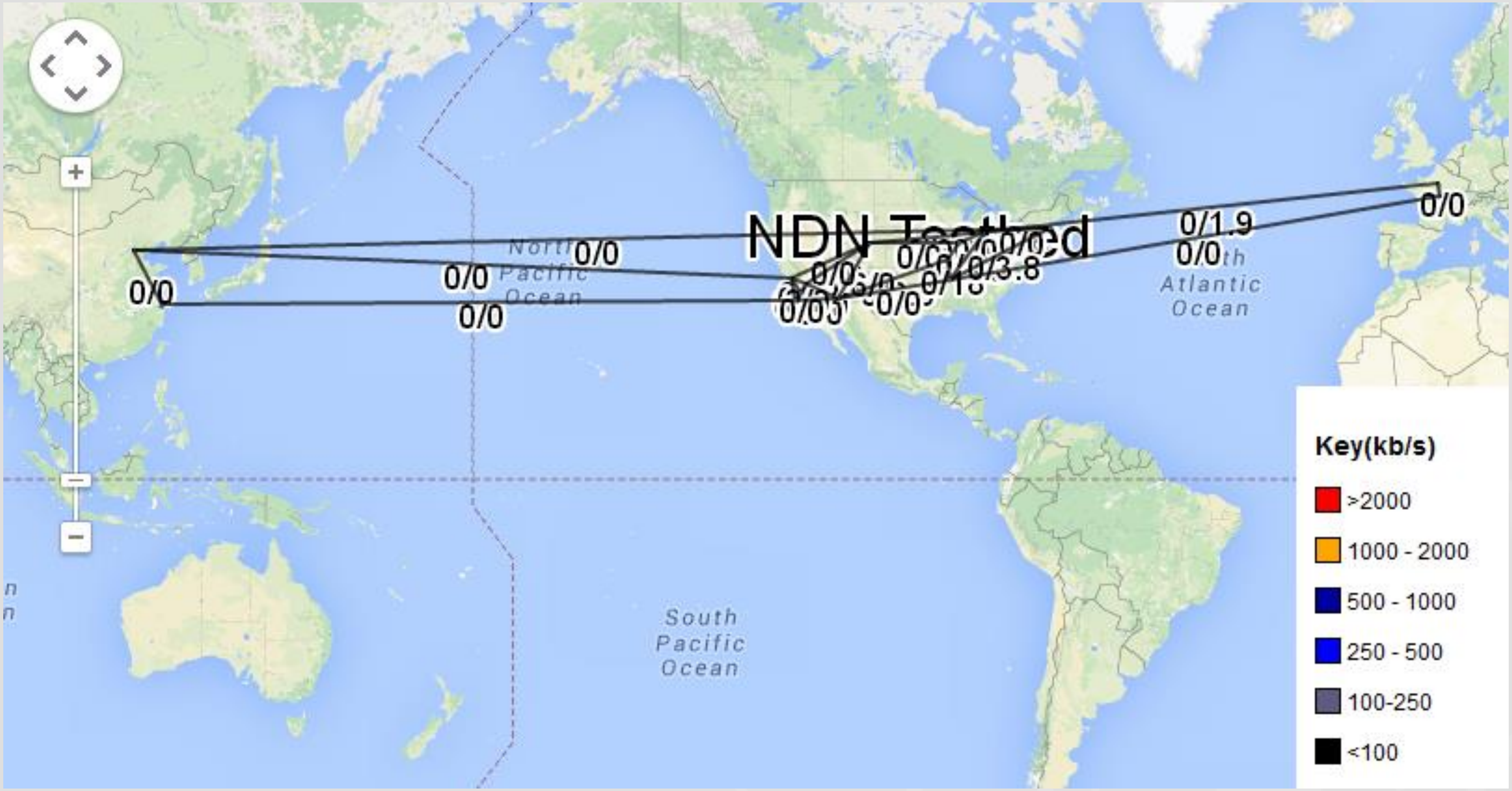
- FEderated Test-beds for Large-scale Infrastructure eXperiments
- EU-JAPAN Project
- SDN-oriented service architecture for federating Future Internet facilities like OFELIA and JGN-X RISE
- Use high-capacity NSI-enabled networks as substrate
  – JGN-X, GEANT, GLIF, NRENs
- On-demand setup of “OpenFlow based network slices”
  – including network, compute and storage

http://www.ict-felix.eu

NDN Testbed

CALTECH HEP NETWORKING

US LHCNet



Summary and Conclusions

- Networks are not any more providing only transmission of bits between a pair of hosts
- New developments are in areas above providing bandwidth
- In development of distributed computing systems, we should leverage the new capabilities of the network systems
- Engagement with the network service providers (NRENs) is necessary in order to benefit most from it

QUESTIONS & DISCUSSION

Artur.Barczyk@cern.ch

Some Resources

1) OGF NSI WG http://redmine.ogf.org/projects/nsi-wg
2) Open Networking Foundation: http://www.opennetworking.org
3) Floodlight controller: http://www.projectfloodlight.org/floodlight/
4) OpenDaylight: http://www.opendaylight.org/
5) Named Data Networking: http://named-data.net/
6) CCNx: http://www.ccnx.org/