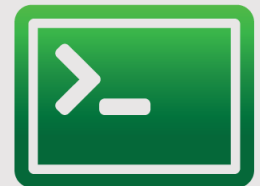


Tutorial: OpenFlow in GENI

GENI Project Office



Design/Setup



Execute



Finish

“The current Internet is at an impasse because new architecture cannot be deployed or even adequately evaluated” [PST04]

[PST04]: Overcoming the Internet Impasse through Virtualization, Larry Peterson, Scott Shenker, Jonothan Turner
Hotnets 2004

Modified slide from: <http://cenic2012.cenic.org/program/slides/CenicOpenFlow-3-9-12-submit.pdf>

- Enables innovation in networking
- Changes practice of networking



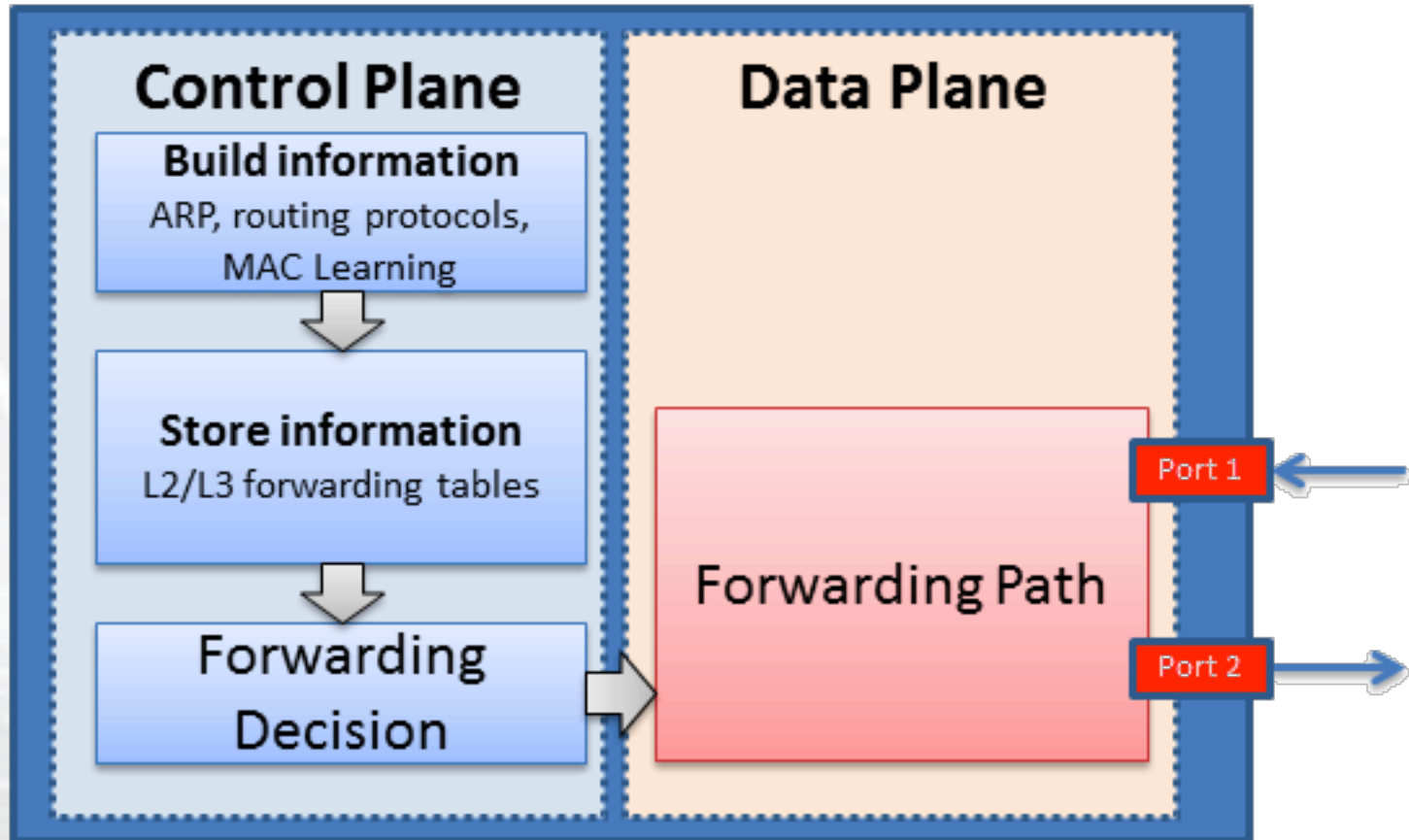
Google's SDN WAN

OpenFlow basics

How OpenFlow works ... (1.0)

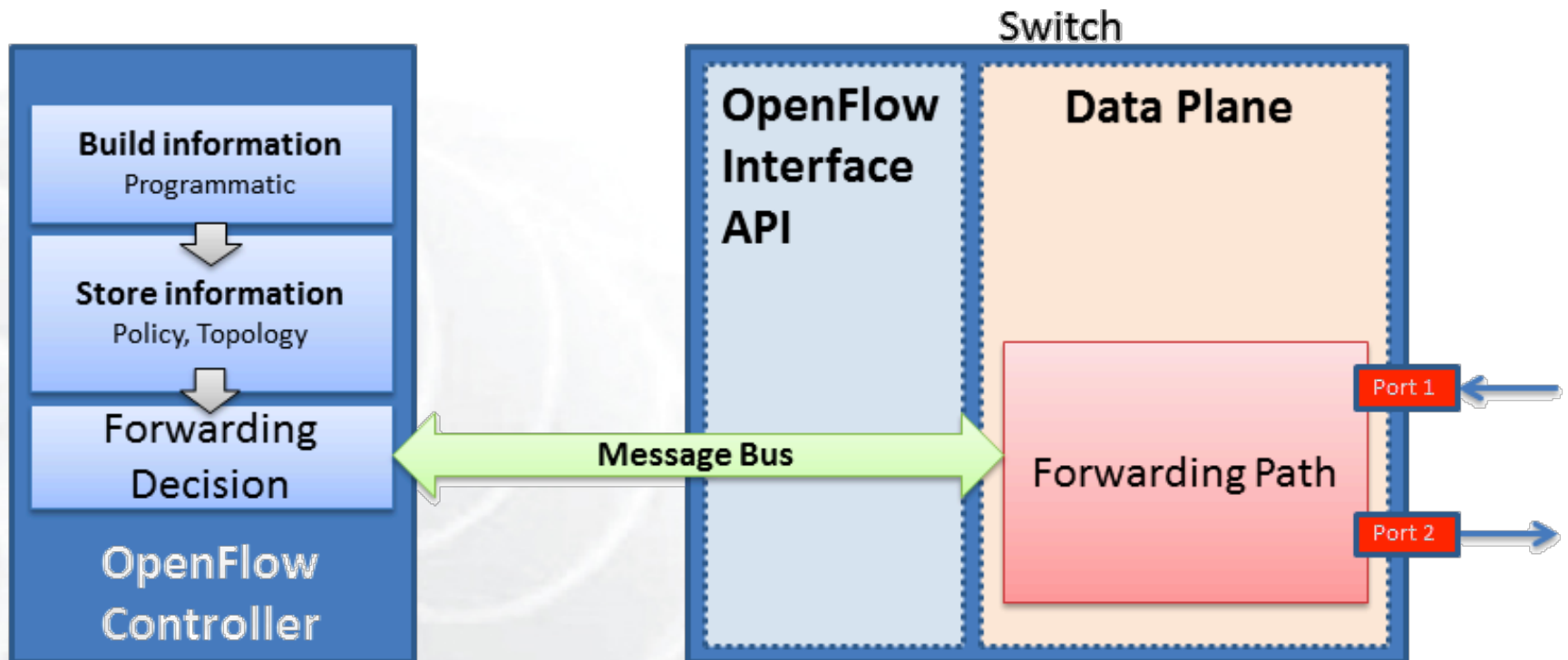
Hand's on tutorial

Switch



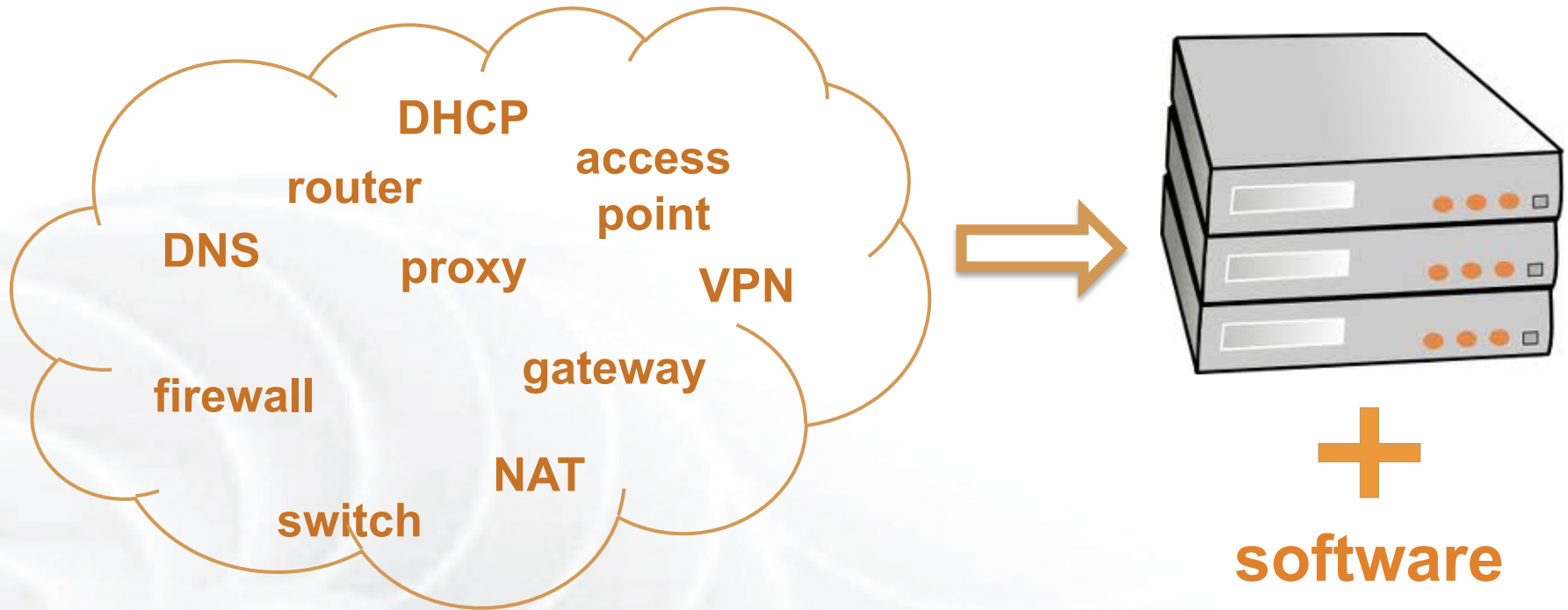
BRAD HEDLUND .com

Externally controlled Switch



- Control how packets are forwarded
- Implementable on COTS hardware
- Make deployed networks programmable
 - not just configurable
- Makes innovation easier





Any network device can be OpenFlow enabled

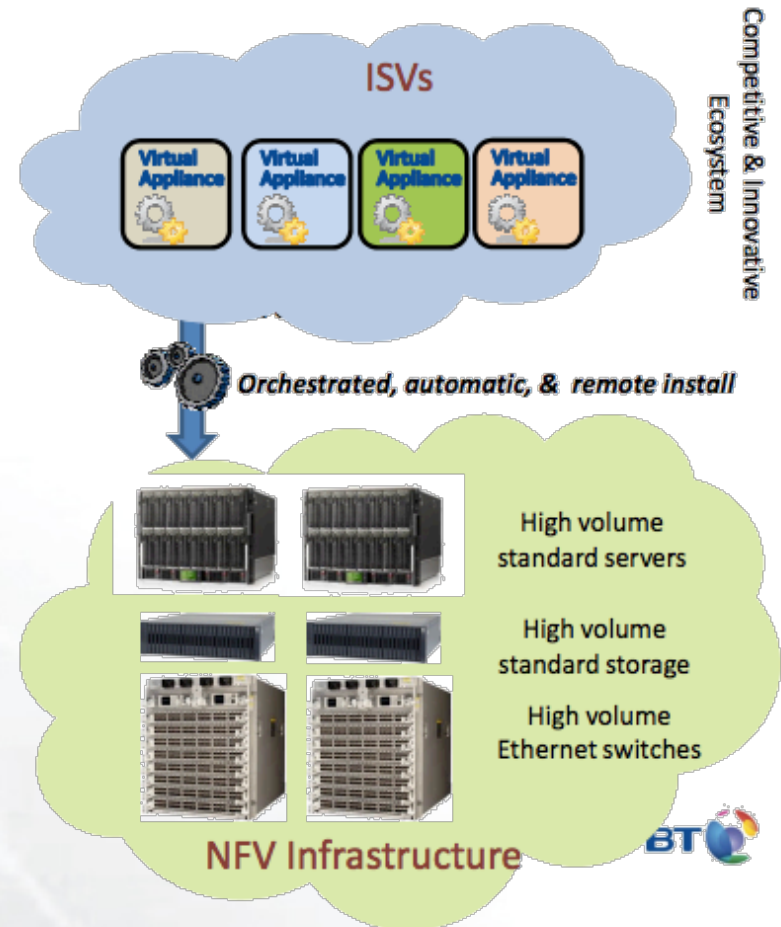
Classical Network Appliance Approach



- Fragmented non-commodity hardware.
- Physical install per appliance per site.
- Hardware development large barrier to entry for new vendors, constraining innovation & competition.

© 2013 High Telecomms Europe plc

NFV Approach



Slide from: http://docbox.etsi.org/Workshop/2013/201304_FNTWORKSHOP/S07_NFV/BT_REID.pdf

- **External control**
 - Enables network Apps
 - General-purpose computers (Moore's Law)
 - Deeper integration
 - Network hardware becomes a commodity

- **Centralized control**
 - One place for apps to interact (authentication, auth, etc)
 - Simplifies algorithms
 - Global Optimization and planning

[1]: OpenFlow: A radical New idea in Networking, Thomas A. Limoncelli CACM 08/12 (Vol 55 No. 8)

Campus

Multiple buildings, heterogeneous IT, groups of users, campus backbone

Enterprise Data Centers

Security, various sizes, storage, WAN optimizations

Data Centers – Clouds

Multi-tenant, virtualization, disaster recovery, VM mobility

WAN

Diversity, multiple domains/carriers/users

Google global private WAN [1]

Connects dozens of datacenters worldwide with a long-term average of 70% utilization over all links

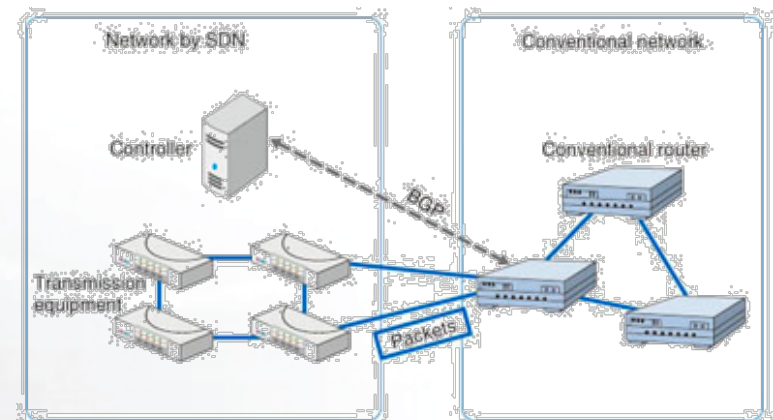
Stanford Campus deployment

Part of Stanford campus migrated to OpenFlow

NTT's BGP Free Edge

Internet 2 - AL2S

Can build Layer 2 circuits between any Internet 2 end-points

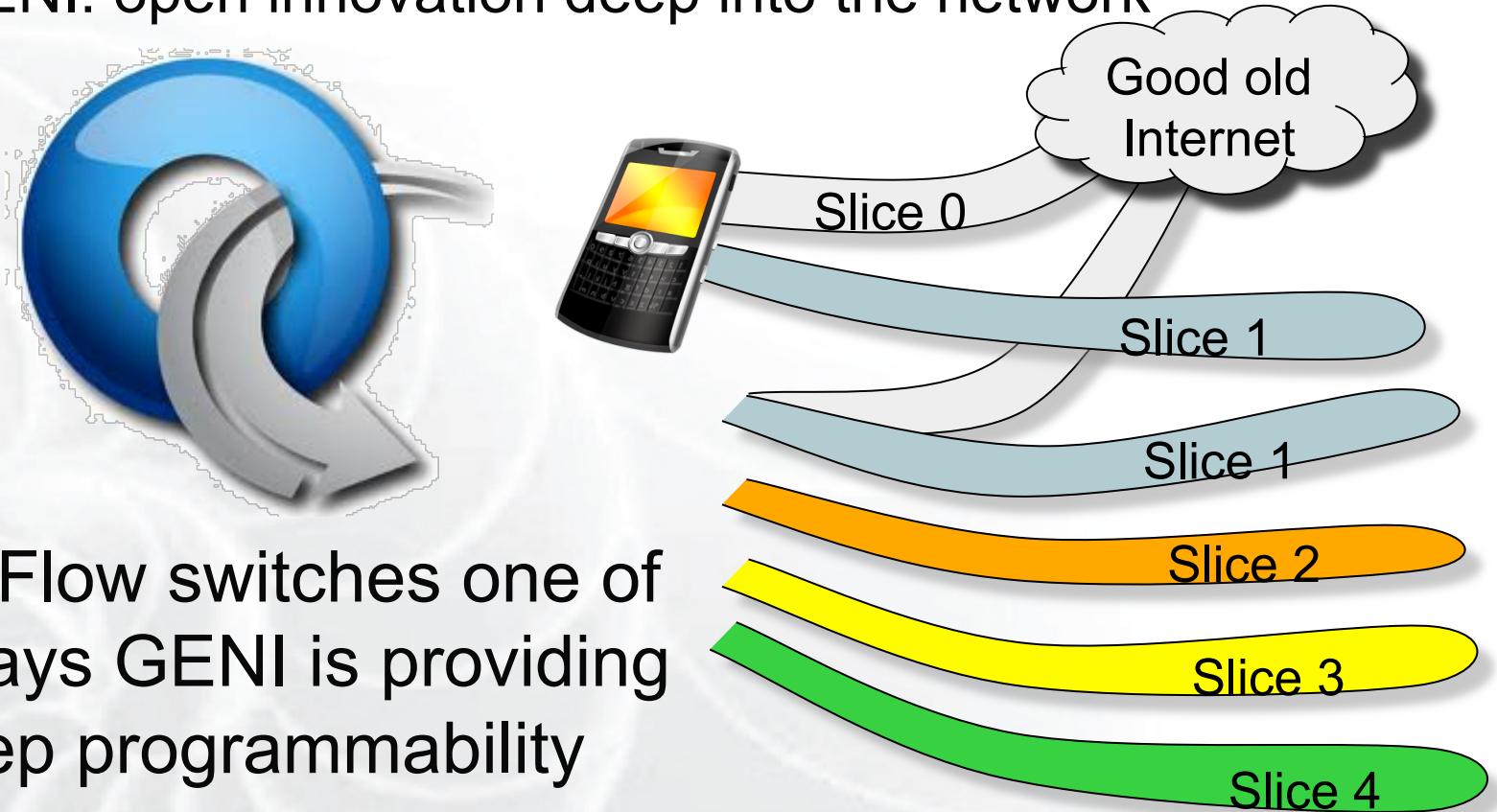


<https://www.ntt-review.jp/archive/ntttechnical.php?contents=ntr201310fa3.html>

[1] *B4: Experience with a Globally-Deployed Software Defined WAN*, SIGCOMM'13, Jain et al.

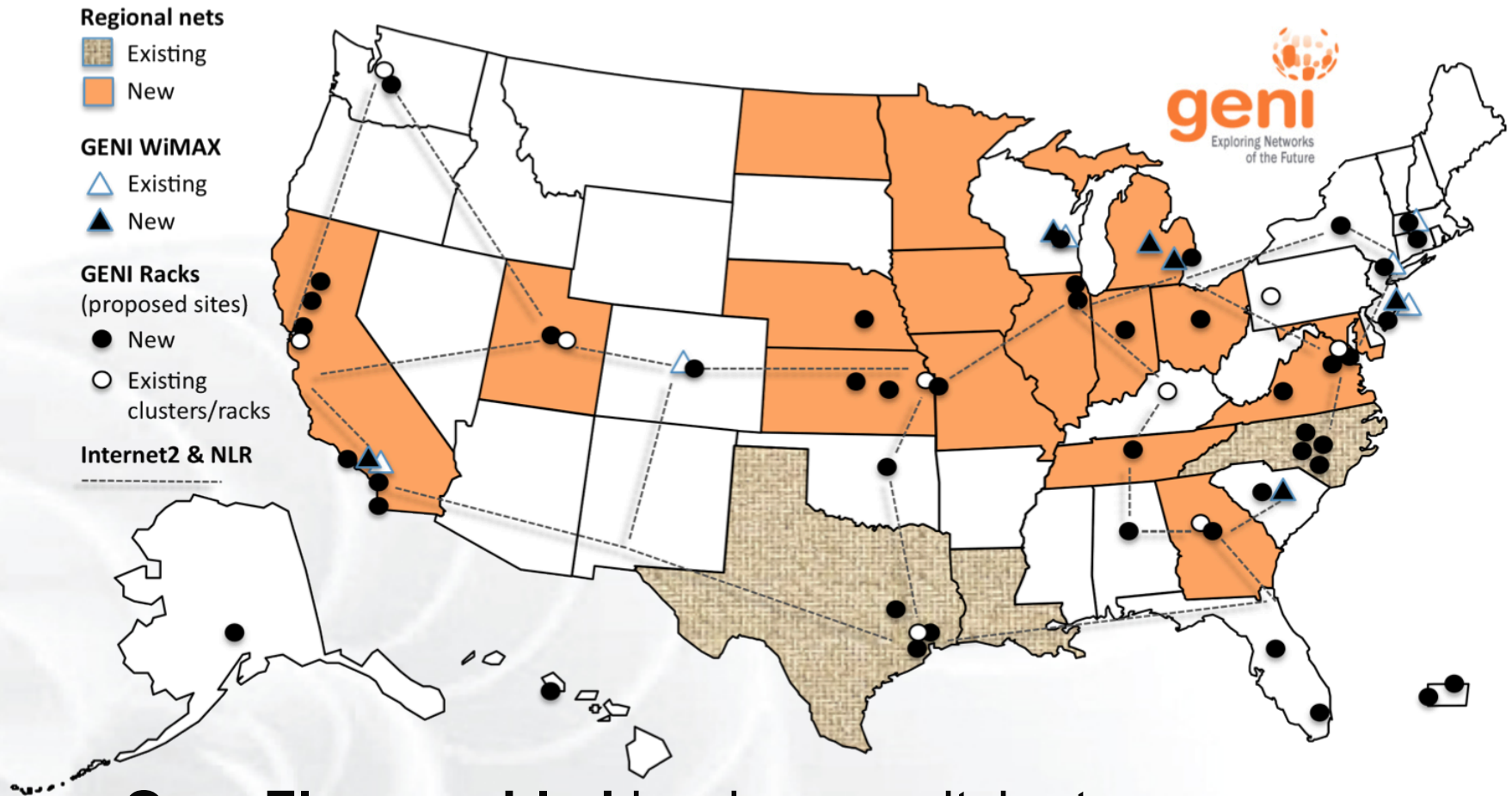
GENI and OpenFlow deployment

- Key GENI concept: slices & deep programmability
 - Internet: open innovation in application programs
 - GENI: open innovation deep into the network



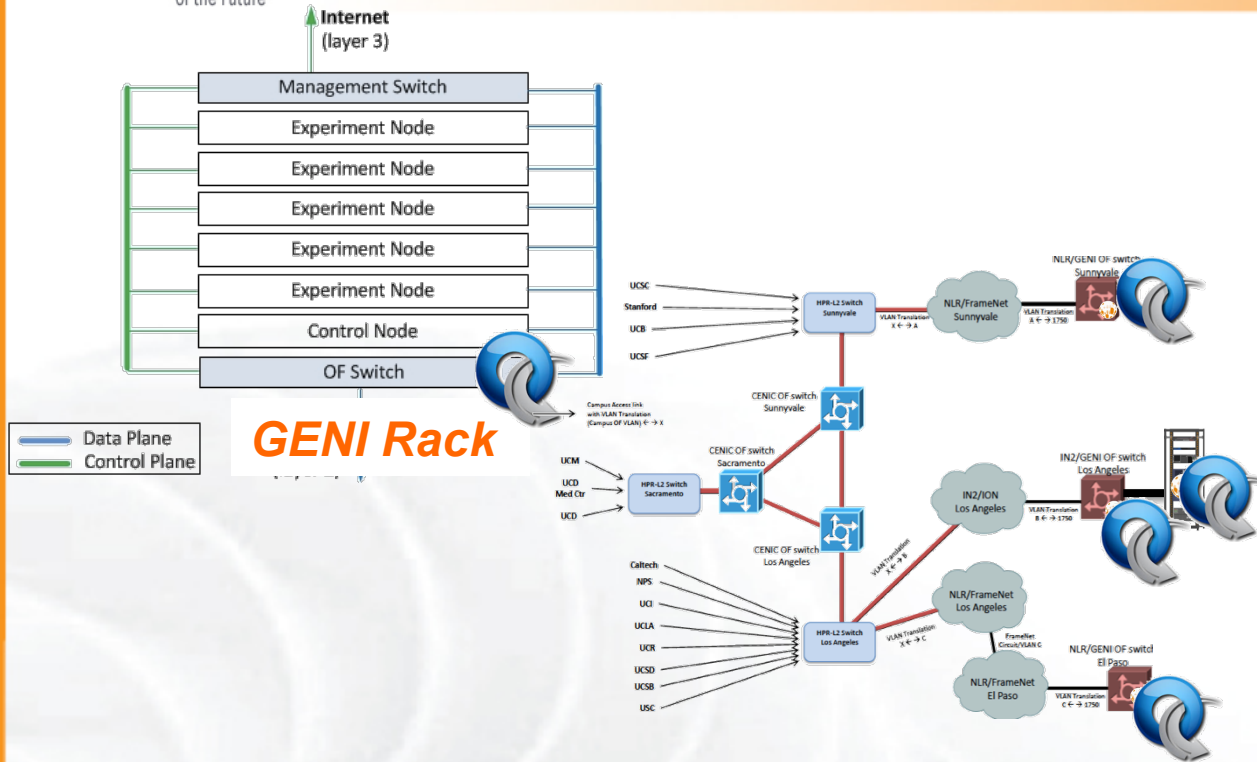
OpenFlow switches one of the ways GENI is providing deep programmability

GENI OpenFlow Deployment

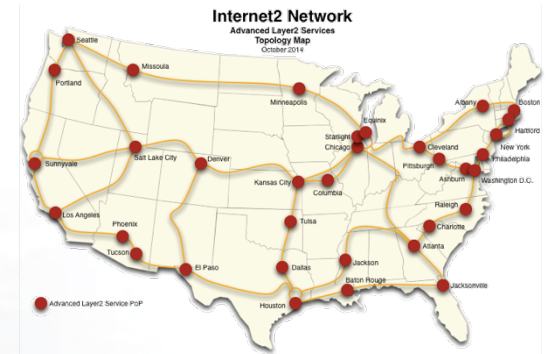


OpenFlow-enabled hardware switch at:

- Each GENI Rack
- Backbone and regional networks



*GENI-enabled regionals
e.g. CENIC*



Internet2 AL2S and ION

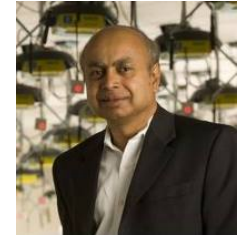
GENI OpenFlow Experiments



VDC: real-time load-balancing functionality deep into the network to improve QoE

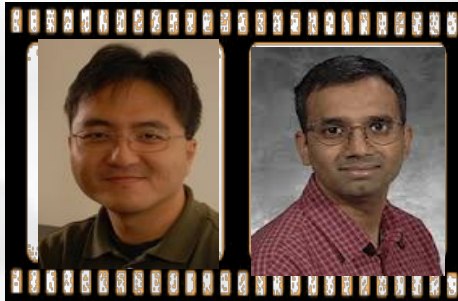
Prasad Calyam, Missouri

MobilityFirst: A new architecture for the Internet designed for emerging mobile/wireless service requirements at scale



Dipankar (Ray)
Raychaudhuri, Rutgers,
leads MobilityFirst

*Parmesh Ramanathan
U. Wisconsin*



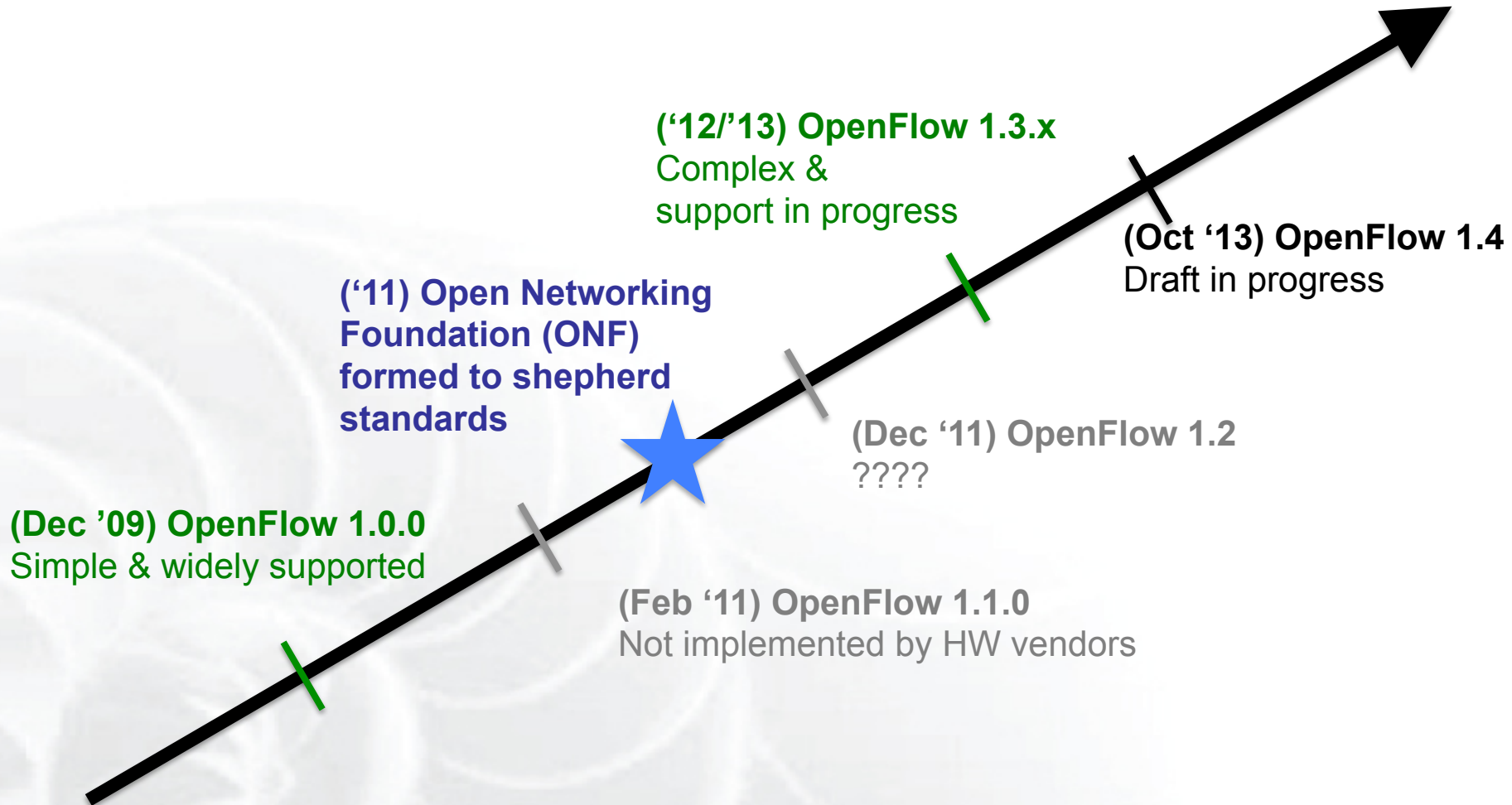
GENI Cinema: Efficient live video broadcasting

*KC Wang
U. Clemson*

OpenFlow basics

How OpenFlow works ... (1.0)

Hands-On tutorial



(Dec '09) OpenFlow 1.0.0
Simple & widely supported

**('11) Open Networking
Foundation (ONF)
formed to shepherd
standards**

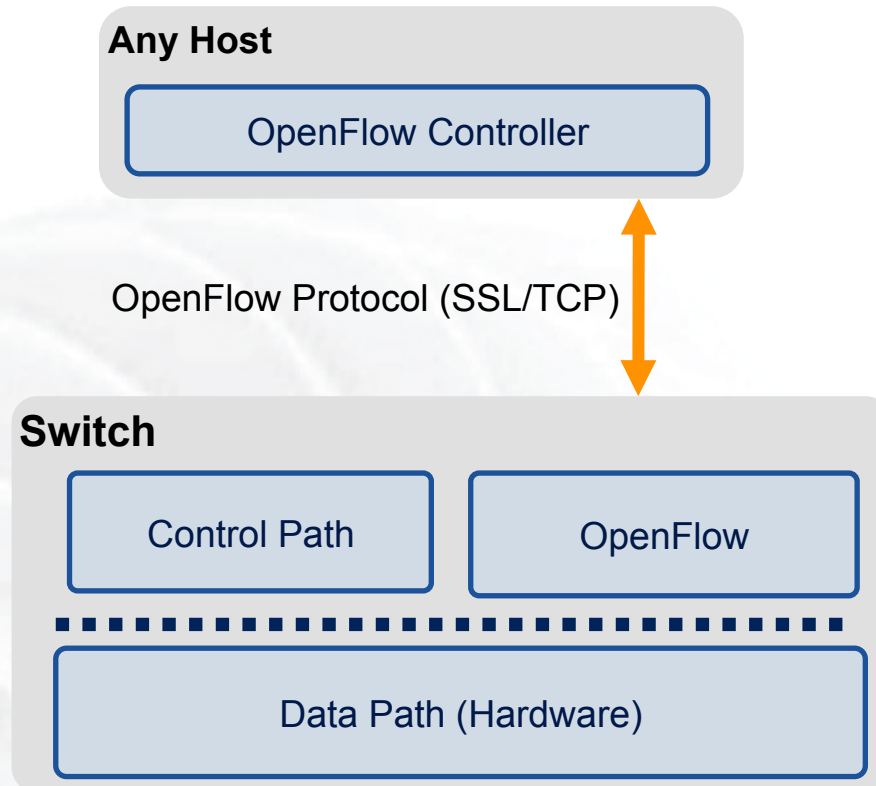
('12/'13) OpenFlow 1.3.x
Complex &
support in progress

(Feb '11) OpenFlow 1.1.0
Not implemented by HW vendors

(Dec '11) OpenFlow 1.2
First ONF standard

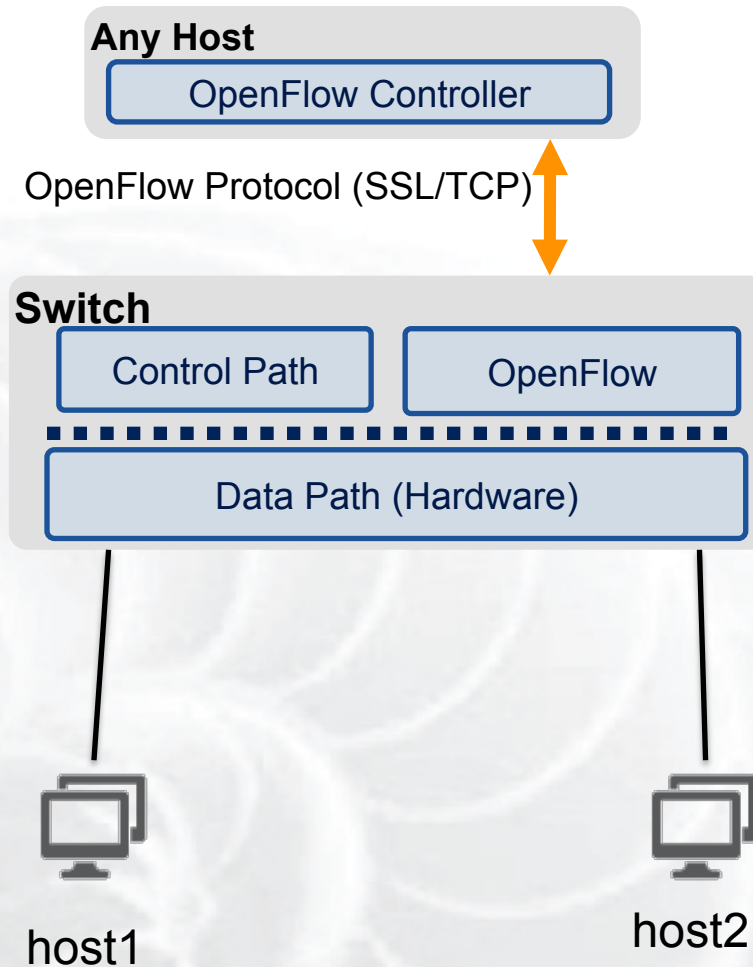
(Oct '13) OpenFlow 1.4
(Nov'13) OpenFlow 1.0.2

- **Open source controller frameworks**
 - NoX – *C++*
 - PoX - *Python*
 - OpenDaylight - *Java*
 - FloodLight - *Java*
 - Trema – *C / Ruby*
 - Maestro - *Java*
 - Ryu - *Python*
- **Production controllers**
 - Mostly customized solutions based on Open Source frameworks
 - ProgrammableFlow - NEC



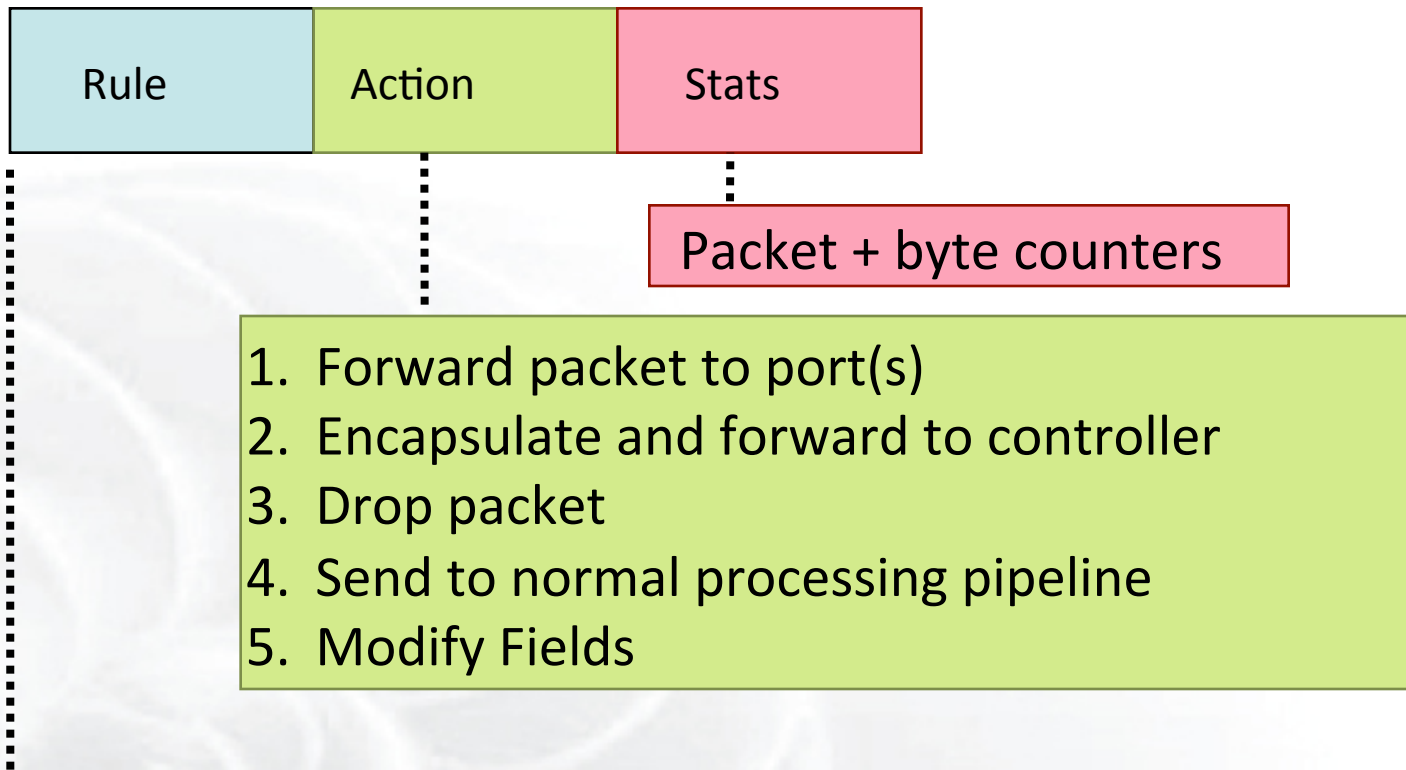
- The controller is responsible for populating forwarding table of the switch
- In a table miss the switch asks the controller

OpenFlow in action



- Host1 sends a packet
- If there are no rules about handling this packet
 - Forward packet to the controller
 - Controller installs a flow
- Subsequent packets do not go through the controller

OpenFlow Basics (1.0)



Switch Port	VLAN ID	VLAN PCP	MAC src	MAC dst	Eth type	IP Src	IP Dst	IP Prot	IP ToS	TCP sport	TCP dport
-------------	---------	----------	---------	---------	----------	--------	--------	---------	--------	-----------	-----------

+ mask what fields to match

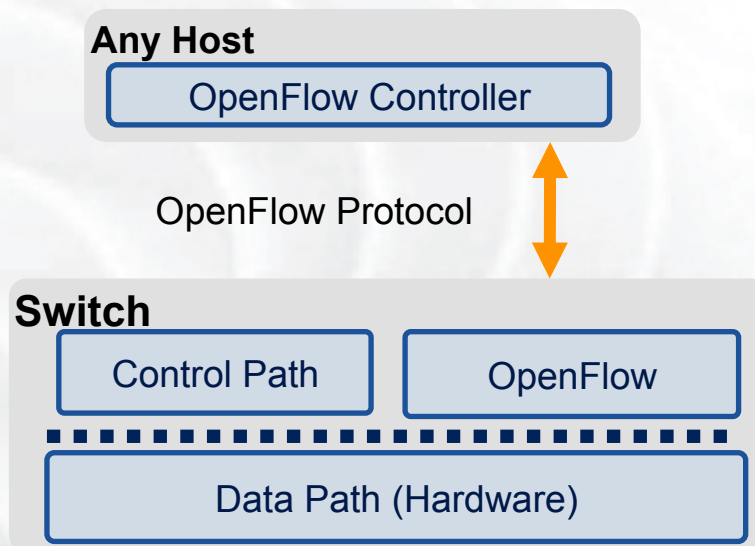
- Going through the controller on every packet is inefficient
- Installing Flows either proactively or reactively is the right thing to do
- A Flow Mod consists of :
 - A **match** on any of the 12 supported fields
 - A **rule** about what to do matched packets
 - Timeouts about the rules:
 - Hard timeouts
 - Idle timeouts
 - The packet id in reactive controllers
 - Priority of the rule

OpenFlow common PitFalls

- Controller is **responsible for all traffic**, not just your application!
 - ARPs, DHCP, LLDP
- Reactive controllers
 - Cause additional latency on some packets
 - UDP – many packets queued to your controller by time flow is set up
- Performance in hardware switches
 - Not all actions are supported in hardware
- No STP to prevent broadcast storms

OpenFlow enabled devices are usually referred to as ***datapaths*** with a unique ***dpid***

It is not necessary that 1 physical device corresponds to 1 dpid



Different OpenFlow modes

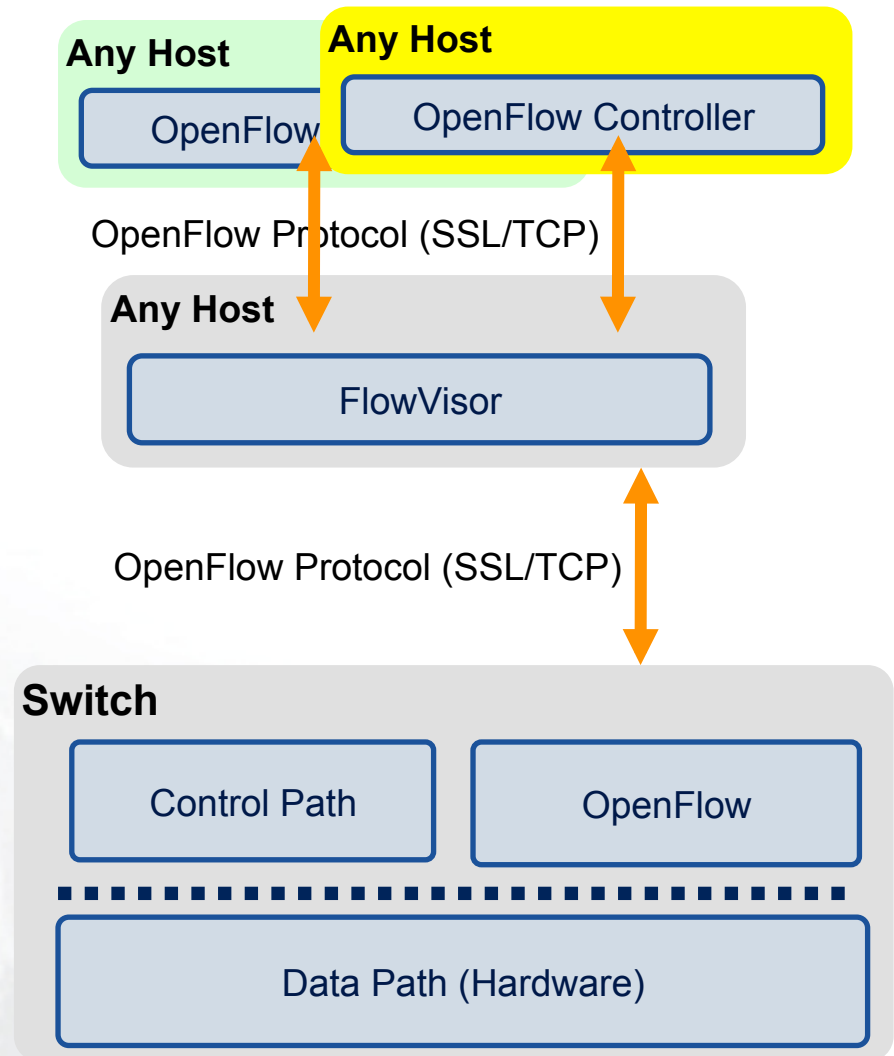
- switches in **pure OF** mode are acting as one datapath
- **Hybrid VLAN switches** are one datapath per VLAN
- **Hybrid port switches** are two datapaths (one OF and one non-OF)

Each Datapath can point to only one controller at a time!

- Only one controller per datapath
- FlowVisor is a proxy controller that can support multiple controllers

FlowSpace describes packet flows :

- **Layer 1**: Incoming port on switch
- **Layer 2**: Ethernet src/dst addr, type, vlanid, vlanpcp
- **Layer 3**: IP src/dst addr, protocol, ToS
- **Layer 4**: TCP/UDP src/dst port



- An OpenFlow Aggregate Manager
- It's a GENI compliant reservation service
 - Helps experimenters reserve flowspace in the FlowVisor
- Speaks AM API v1 and AM API v2
- RSpecs GENI v3, OpenFlow v3 extension

In GENI:

- Slice by VLAN for exclusive VLANs
- Slice by IP subnet and/or eth_type for shared VLANs

In FIRE:

- On iMinds testbed
 - Slice by inport
- On OFELIA testbed
 - Slice by VLAN

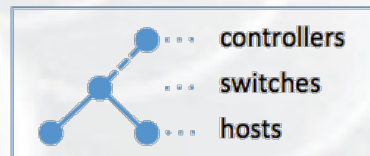
OpenFlow Experiments

Debugging OpenFlow experiments is hard:

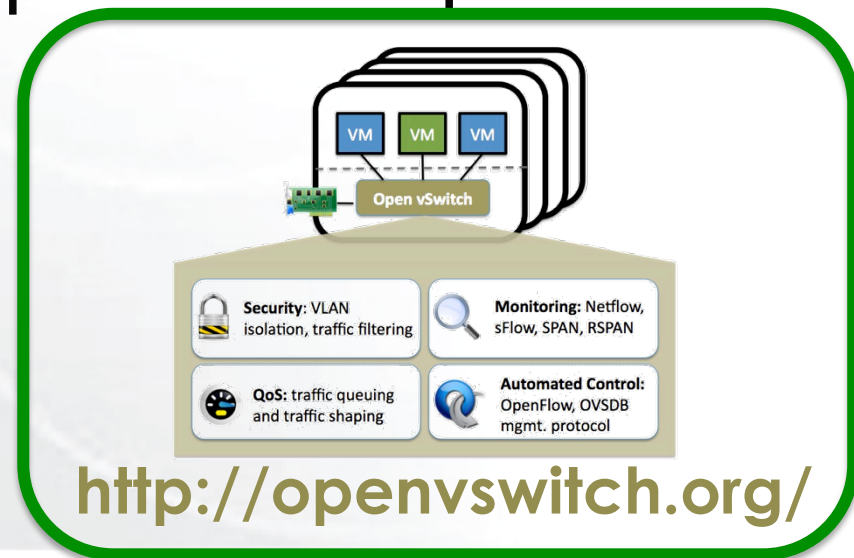
- Network configuration debugging requires coordination
- Many networking elements in play
- No console access to the switch

Before deploying your OpenFlow experiment
test your controller.

```
> sudo mn
```



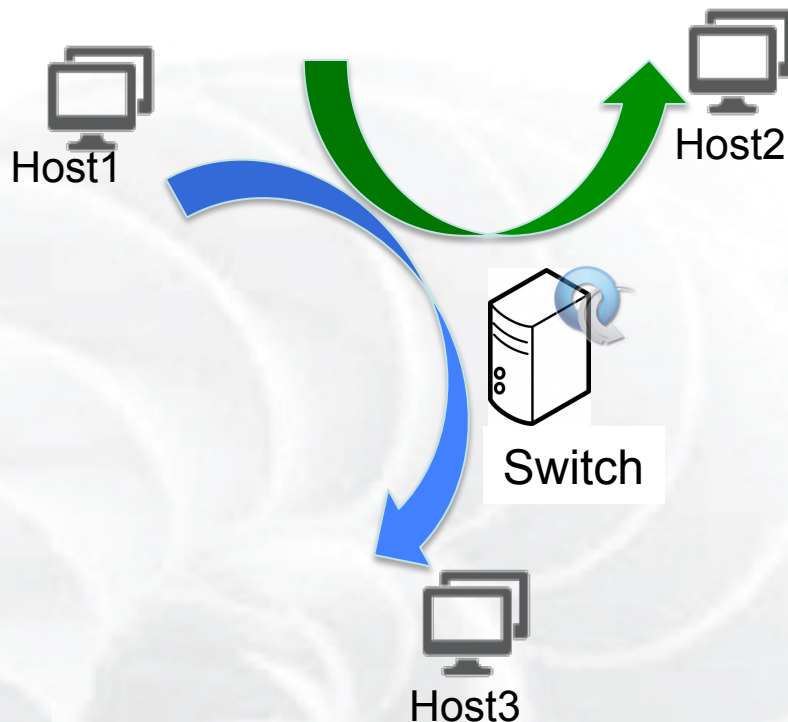
<http://mininet.github.com/>



OpenFlow basics

How OpenFlow works ... (1.0)

Hands on tutorial



- Traffic forwarding
 - OVS switch-based exercise
 - Replace OVS switch with a hardware switch
 - Use OVS switch but move one of the hosts to another rack (stitched link)
- Network Function Virtualization
 - OpenFlow based NAT
 - OpenFlow based Firewall

<http://tinyurl.com/msu-geni>
QUESTIONS?