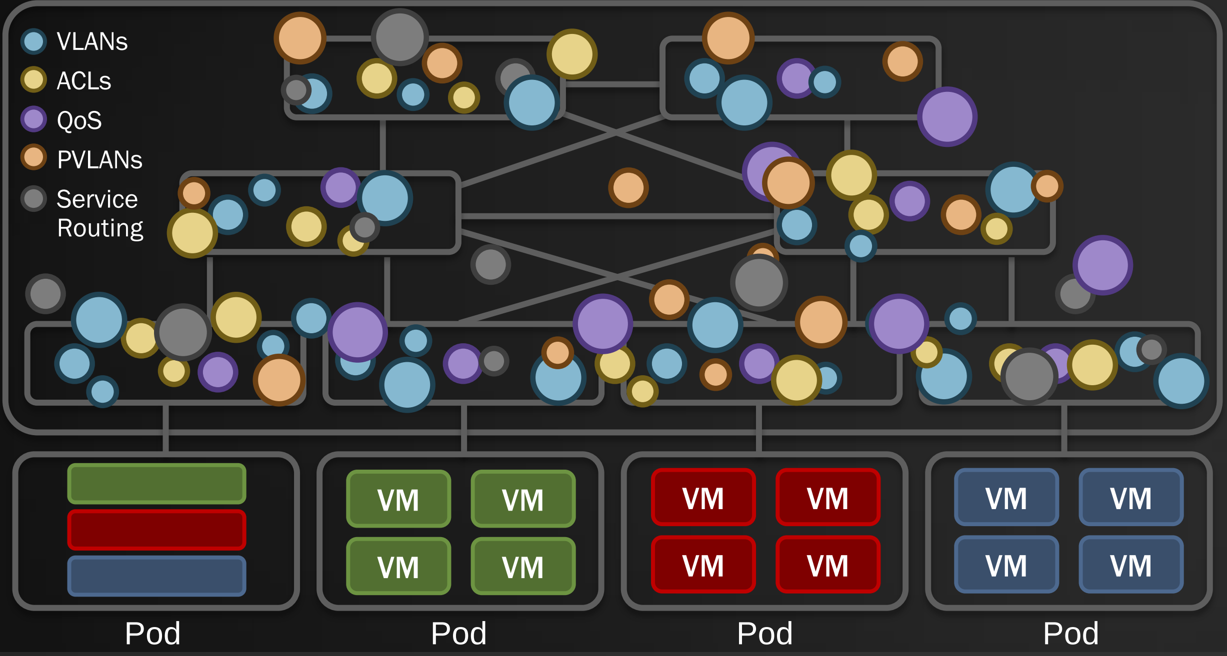
SDN: Openflow & Internet2

BROCADE

Why is This so Difficult to Support Today?

EXAMPLE: DATA CENTER NETWORK OFFERING MULTI-TENANT CLOUD



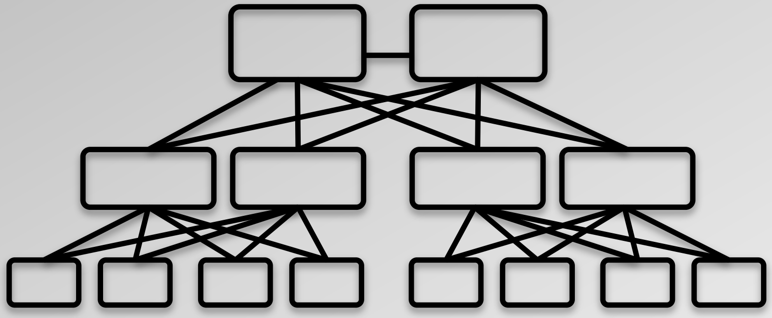
MULTI-TENANCY REQUIREMENTS

- Isolation
- Security
- SLAs
- Shared Services
- Service Insertion
- ...

Today's Networks are very <u>difficult to manage</u>
Have not evolved to support the demands of Multi-Tenant Cloud

Why can't you do these things today?

Traditional Network



Hierarchical

Monolithic

Closed

North/South-optimized

Inflexible

- Network changes are difficult, slow, and risky
- Can't handle rapid swings in traffic demands
- New services requires adding expensive specialized skills
- Unlimited funds needed to solve issues

What if you could ...

Build your next data center optimized for **highest demands** in **reliability and scale**

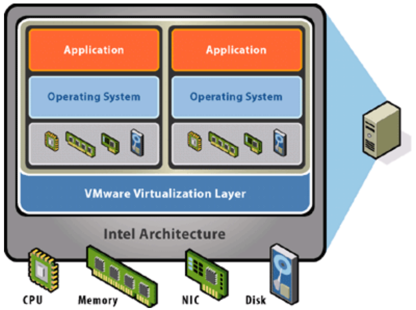
Virtualize your network starting now for greater responsiveness and increased asset utilization

Create and deliver **customized services and new offerings** at the **speed of customer need**

Unlock the intelligence from your network for **real-time orchestration and analytics**

SDN Goals

Server Analogy



Virtualization

Server: VMs
Network: Tunnels,…,

Automation

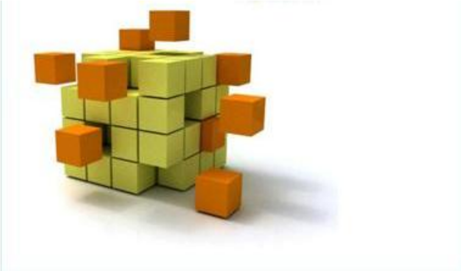
Server: vCloud Director, AWS
Network: ??

Standardization

Server: x86 instruction set
Network: OpenFlow

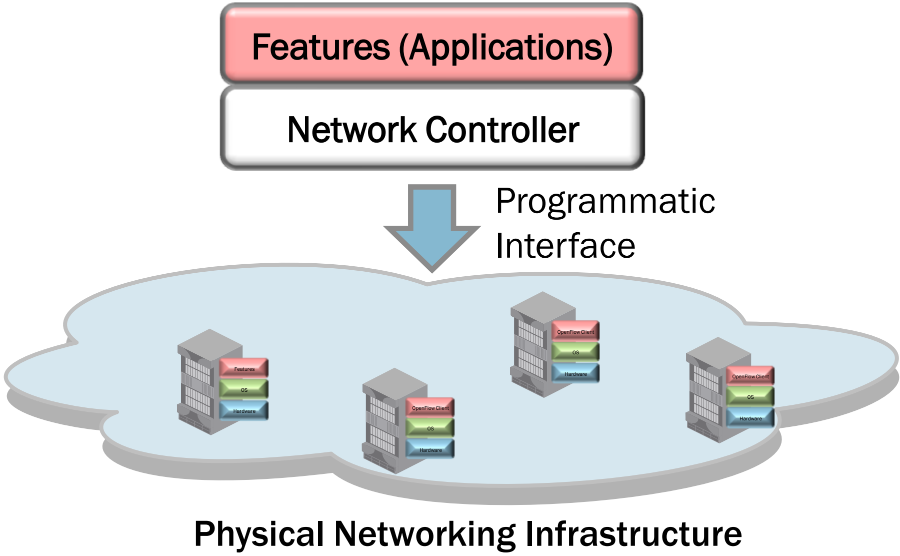
Disaggregation

Server: ISVs, Middleware
Network: ??



What is Software-Defined Networking (SDN)?

- Software abstraction layer (network controller) on top of networking infrastructure
  - Abstracts physical network
  - Makes networking hardware vendor independent
- Standardized programmatic interface
  - Hardware vendor independence achieved using standardized interface to physical network
  - Programmatic interface allows the networking functionality to be defined by the software abstraction layer
- Key customer benefits
  - Decouples network application innovation from dependency on new router OS releases
  - Accelerates automation of network changes to increase service velocity

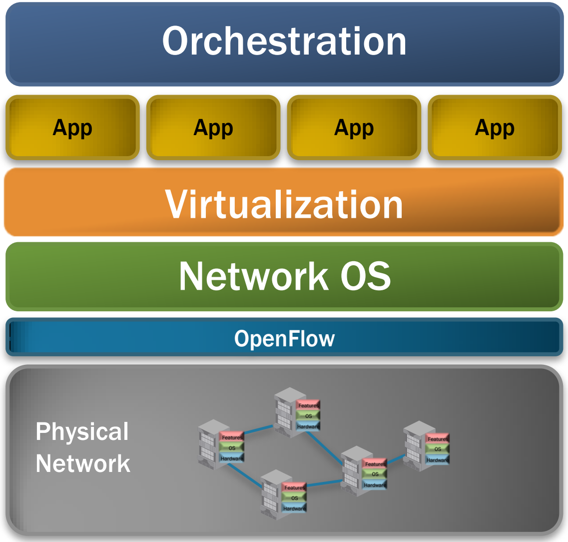


Who is behind Software Defined Networking?

Open Networking Foundation (ONF)

- ONF launched publicly in March, 2011
- Support from more than 70 major companies
- The ONF defines OpenFlow and API specifications
- Founding members of ONF:

Deutsche Telekom Microsoft facebook verizonwireless Google YAHOO!

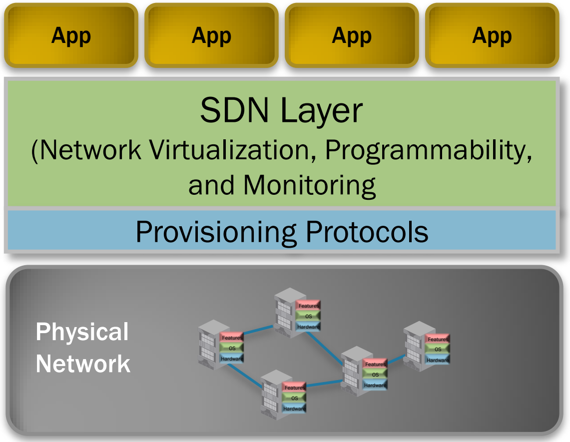


ONF SDN Model
(simplified)

Software Defined Networking and IETF

Similar Goals, Similar Architecture

- "SDN Problem Statement and Use Cases for Data Center Applications"
  - draft-pan-sdn-dc-problem-statement-and-use-cases-02.txt
- Flexible in terms of provisioning protocols
  - OpenFlow, NetConf, PCE, etc



SDN-enabled Network

ONF Background in a nutshell

- Initial goal of SDN: enable experimentation/innovation at scale
  - Started by academics, attracted industry where legacy efforts were failing
  - OpenFlow spec goes back to 0.2 in 2008, but ONF only started in 2011
- Other mainstream orgs had not taken up the challenge
  - E.g. IETF worked on ForCES for 10 yrs, but not much has happened
  - SDN advocates felt that vendors were impeding OpenFlow/SDN
  - This prompted a new “customer-driven” org with very broad goals
- Obviously an ambitious move
  - Will vendors really follow? Answer: mostly yes!
  - New org means new processes and culture, growing pains
  - Broad goals increase challenge: consensus is slower, architecture trickier

History of Forwarding Abstractions Working Group

- March 2011, ONF formed
- May 2011, first ONF meeting, people already talking about issues in OpenFlow 1.1
  - The multi-table pipeline is both “too flexible” and “not flexible enough”
  - “Too flexible” means hard to implement on ASIC/merchant silicon platforms
- October 2011, Google members showed a bold “OF2.0” proposal to the ONF TAG
- November 2011: “Future Discussion Group” was created, which I co-led
- In April 2012, Future DG submitted charter for “FPMODS” to the TAG
  - Board had just chosen to slow spec development, push for adoption of OF1.3
  - TAG asked us to focus first on implementation challenges on ASICs / merchant silicon
- In August 2012, the ONF board approved the revised charter: FAWG launched

Key drivers for FAWG

- OF1.0 was simple and was well-adopted, but…
- Complex forwarding needs resulted in OF1.1
- Unfortunately, two aspects of OF1.1 (etc) are hard
  - Mapping of behavior must be handled at run-time (as with 1.0)
  - Incremental behavior messages (instead of end-to-end) (new)
  - FAWG members view these as 1) arbitrary and 2) undesirable in production
- Altering framework make it practical
  - Share end-to-end behavior, not piecemeal “do it this way”
  - Share the behavior long before run time
    - Eliminates non-deterministic run-time events, makes run-time predictable
    - Also allows market participants to know which products interoperate

FAWG Phase 1 Goals

- Enhance OpenFlow adoption on hardware-based forwarding targets
  - While maintaining good adoption on SW forwarding targets
- Push broad adoption of common “Table Type Patterns” (TTPs)
  - Eases controller side implementation
  - Provides a more meaningful basis for testing/certification
  - Makes product interoperability easier to determine
- Encourage adoption of member-owned TTPs to drive convergence
- Ensure that existing OpenFlow capability is still supported
  - Base on OpenFlow 1.3 (adjust to 1.4 if need be)
  - When no TTP has been negotiated, the connection will use earlier OpenFlow versions
    - If a TTP has been negotiated, some minor changes are okay

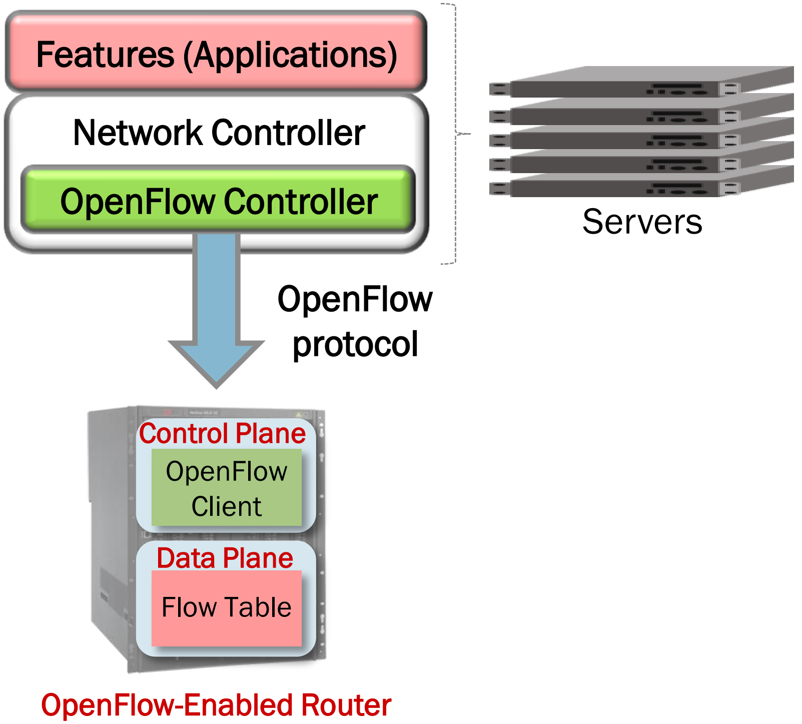
OpenFlow Versions

- OpenFlow 1.0 (03/2010) (R5.4)
  - Most widely used version
  - Layer 2 and Layer 3 (IPv4) matching fields
  - Single Flow Table
- OpenFlow 1.1 (02/2011)
  - Add MPLS label/EXP matching fields
  - Multiple tables
  - Group table (LAG)
- OpenFlow 1.2 (12/2011 - ONF)
  - Add IPv6 matching fields, extensible expression
- OpenFlow 1.3 (2012 – ONF)
  - Topology discovery
  - Test processes
  - Test suites
- OpenFlow 1.4 (2012 - ONF)
  - Improve capability discovery
  - Test labs
- OpenFlow 2.0 (TBD – ONF)
  - Complete redesign

OpenFlow Basics

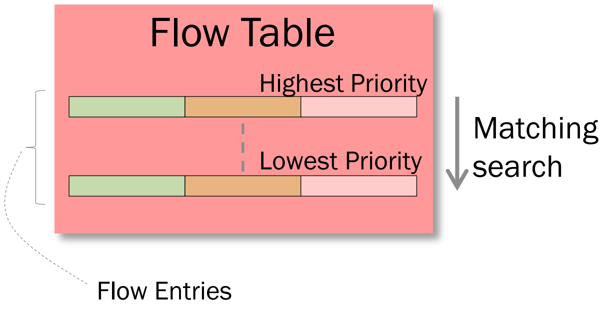
OpenFlow Introduction

- OpenFlow-enabled router supports an OpenFlow Client (control plane software)
- OpenFlow Client communicates with an OpenFlow Controller using the OpenFlow protocol
  - The term "OpenFlow Controller" is sometimes (incorrectly) used in the industry as being the same as (Network Controller + Feature / Application)
  - An OpenFlow Controller may be just a software layer that supports APIs to a Network Controller, which in turn supports APIs to Applications
- OpenFlow-enabled routers support the abstraction of a <u>Flow Table</u>, which is manipulated by the OpenFlow Controller



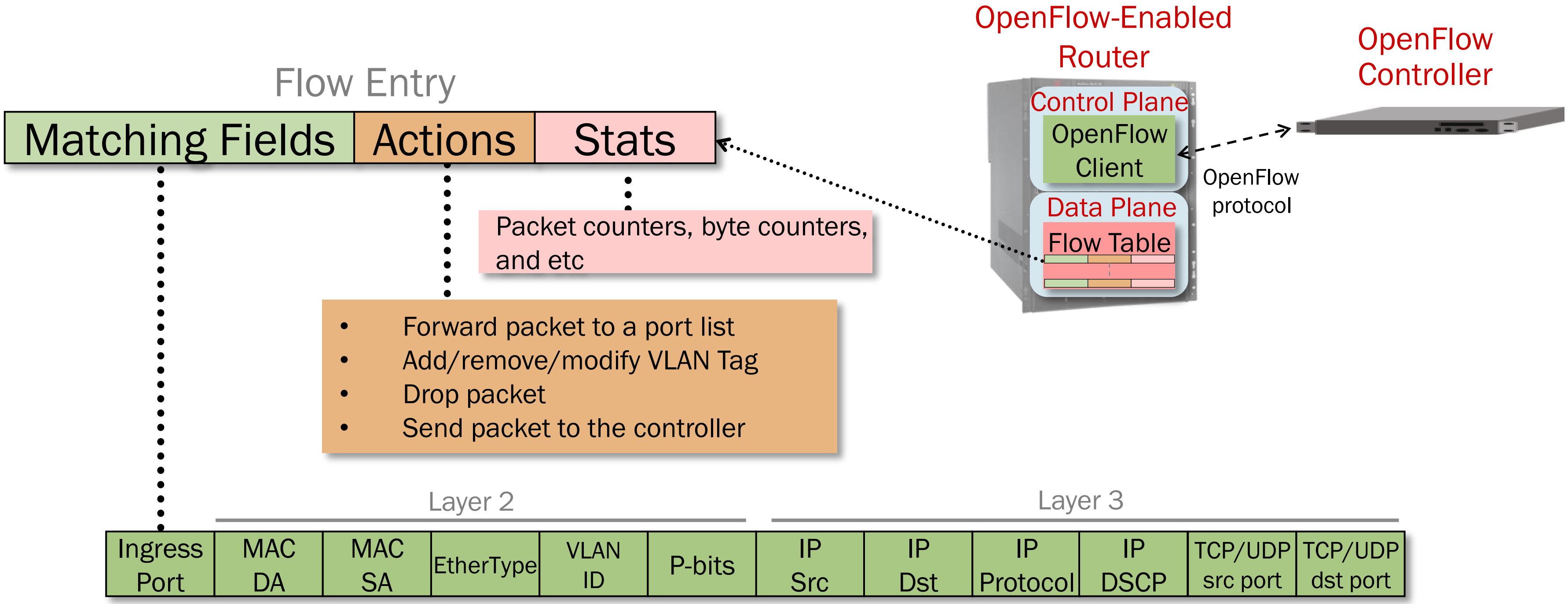
OpenFlow-Enabled Router Operation

- Flow Table contains Flow Entries
  - Each Flow Entry represents a Flow, e.g., packets with a given destination IP address
- The flow table is sorted by flow priority, which is defined by the controller
  - Highest priority flows are at the top of the Flow Table
- Incoming packets are matched against the flow entries (in order)
  - Matching means: Does the packet belong to this Flow?
- If there is match, flow matching stops, and the set of <u>actions</u> for that flow entry are performed
- Packets that don't match any flow entry are typically dropped



Flow Table Entry

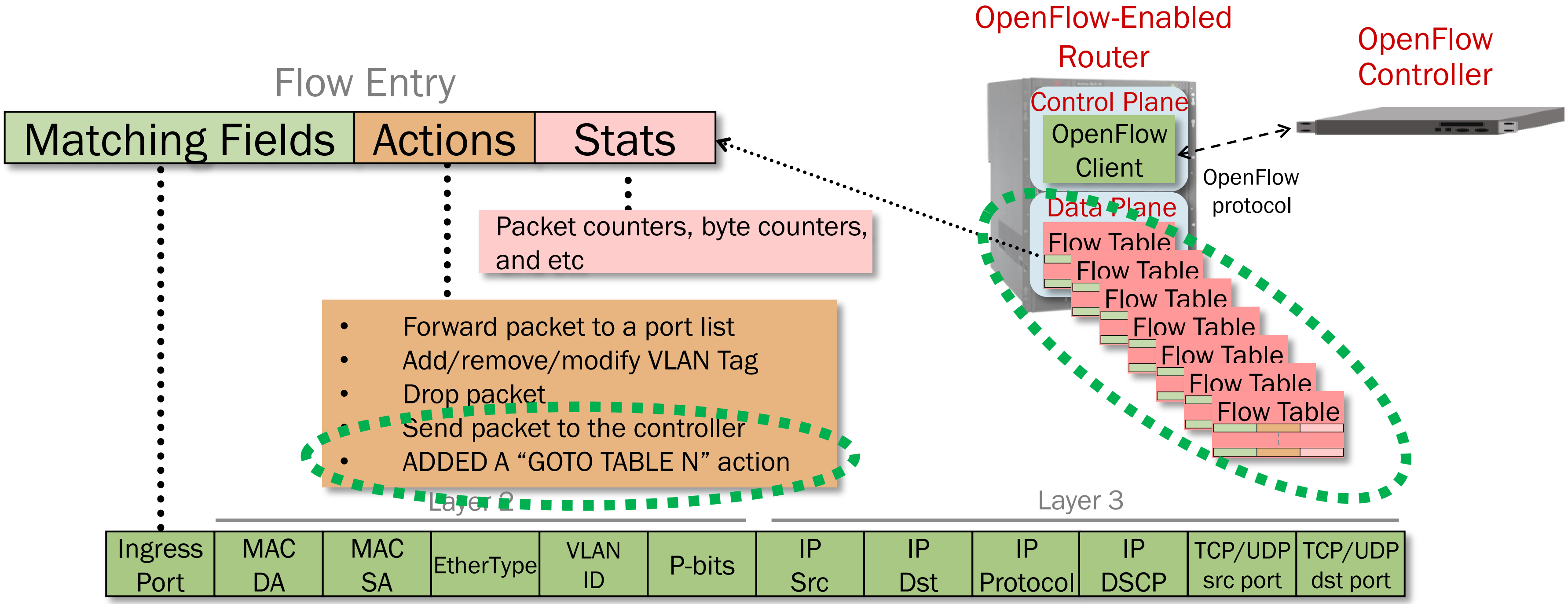
OpenFlow 1.0



- Each flow table entry contains a set of rules to match (e.g., IP src) and an action list to be executed in case of a match (e.g., forward to port list)

In 1.1 (1.2, 1.3), OpenFlow added “multiple tables”

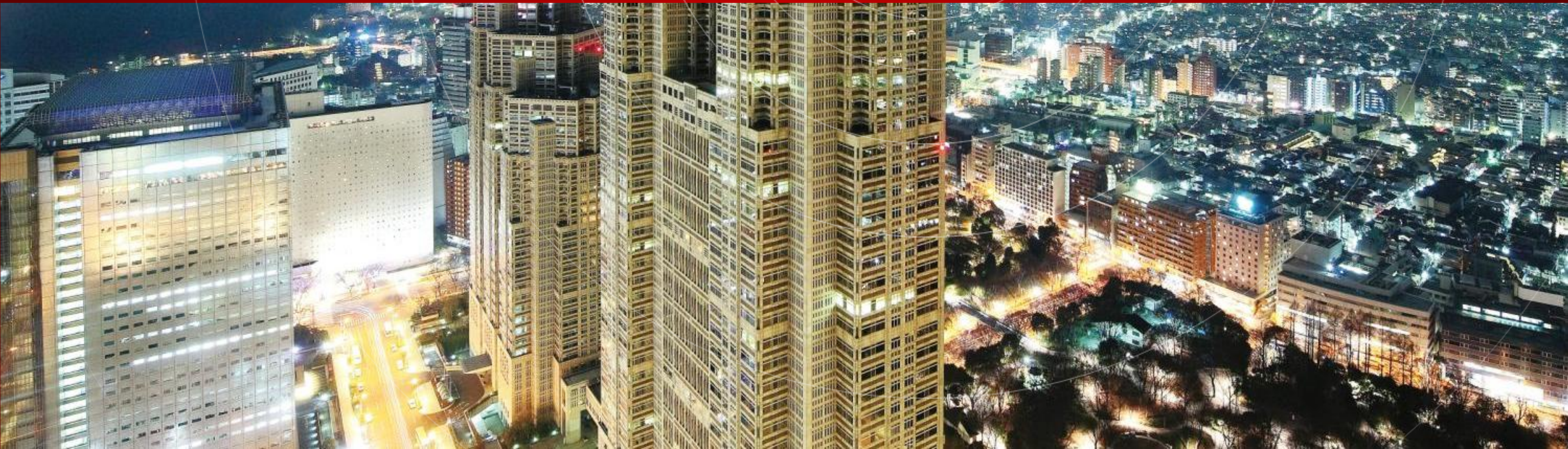
Plus “group tables” and various new actions for IPv6 support, MPLS, etc



- Each flow table entry contains a set of rules to match (e.g., IP src) and an action list to be executed in case of a match (e.g., forward to port list)

BROCADE

INTERNET2



Internet2

BROCADE OPENFLOW ENABLED 100G NATIONWIDE BACKBONE

INTERNET 2

IN SUPPORT OF U.S.UCAN

man LAN
Exchange Point

Internet 2

- 49 Custom Location Facilities
- 15,500 miles of dark Fiber
- 8.8 Tbps of Optical Capacity
- Hybrid Mode with protected OpenFlow traffic

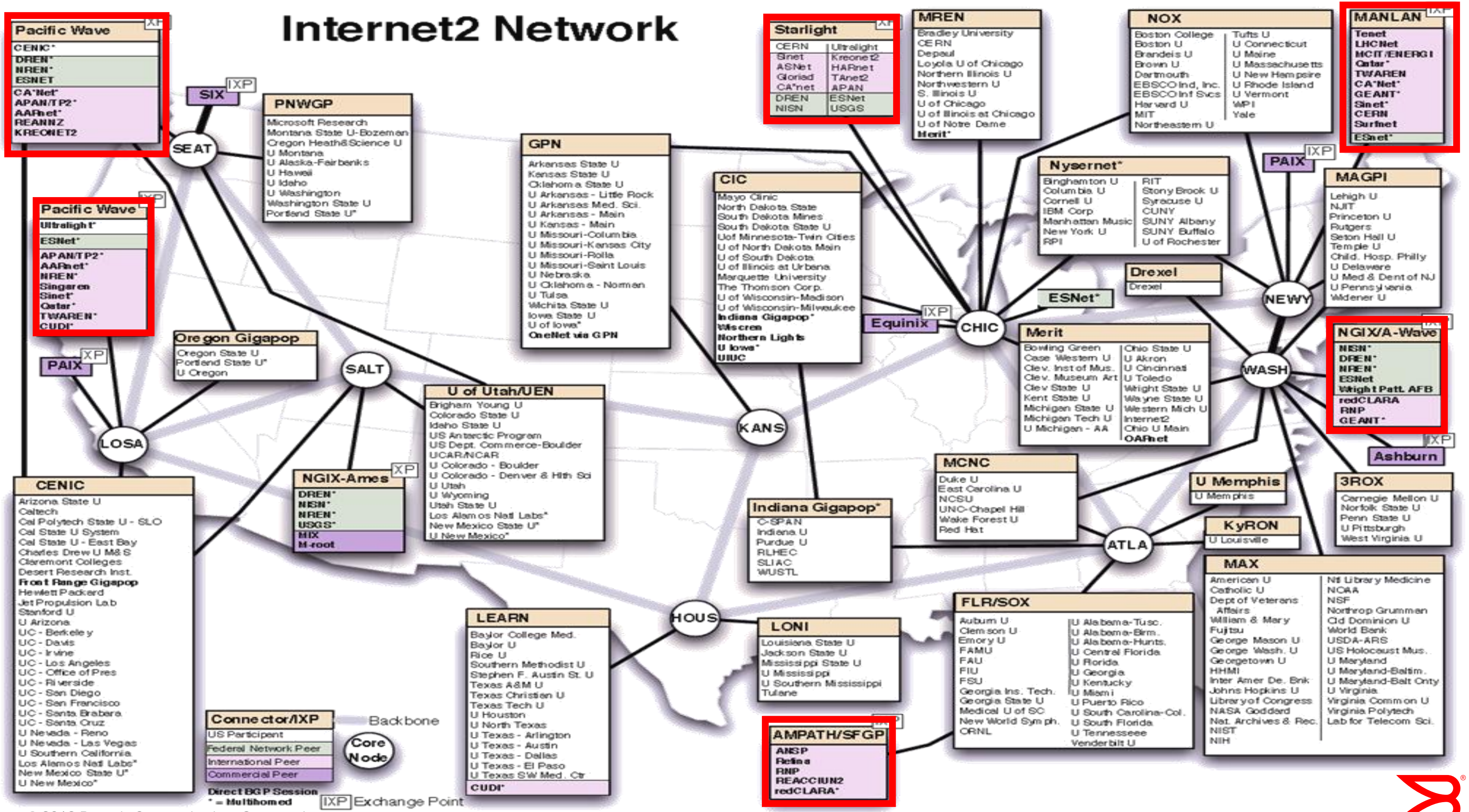
Indiana University - OESS

Openflow enabled SDN nodes

Benefits

- Network Development and Deployment Initiative (NDDI) Project
- Open Science, Scholarship and Services Exchange (OS3E)
- Network Platform and Complementary Software tested with Brocade NetIron MLXe
- International Interconnects
- Offerings
  - Layer2 Service with Vlan identifiers
  - Preservation of Vlans across WAN
  - Preservation of QoS
  - Protected Paths across WAN Infrastructure
  - Programmable through Web based GUI and Openflow

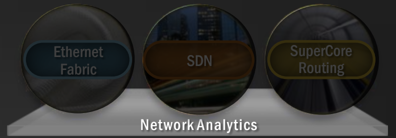
Internet2 Network



Current Trials

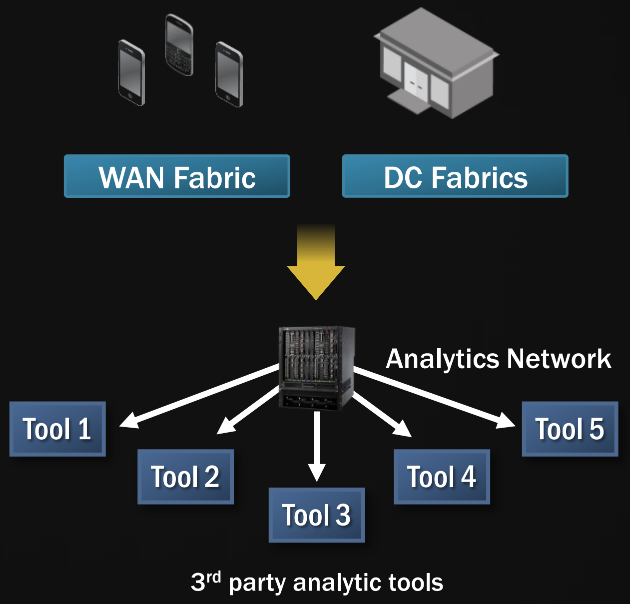
68 trials/deployments spanning 13 countries

NEXT….



SDN and Network Analytics

- Objectives
  - Real-time network statistics collection and alerting
  - Summary of normal and abnormal traffic
  - Detect network performance issues in advance of customer complaints
- Approach
  - SDN Network Analytics application to map analytic policies to traffic filtering and replication requirements using OpenFlow



BROCADE

BROCADE

¡Gracias!
Thank You