Introduction to OpenFlow:

Bringing Experimental Protocols to a Network Near You!

Chris Tracy, Network Engineer

ESnet Engineering Group

ESnet
Energy Sciences Network

U.S. DEPARTMENT OF **ENERGY**
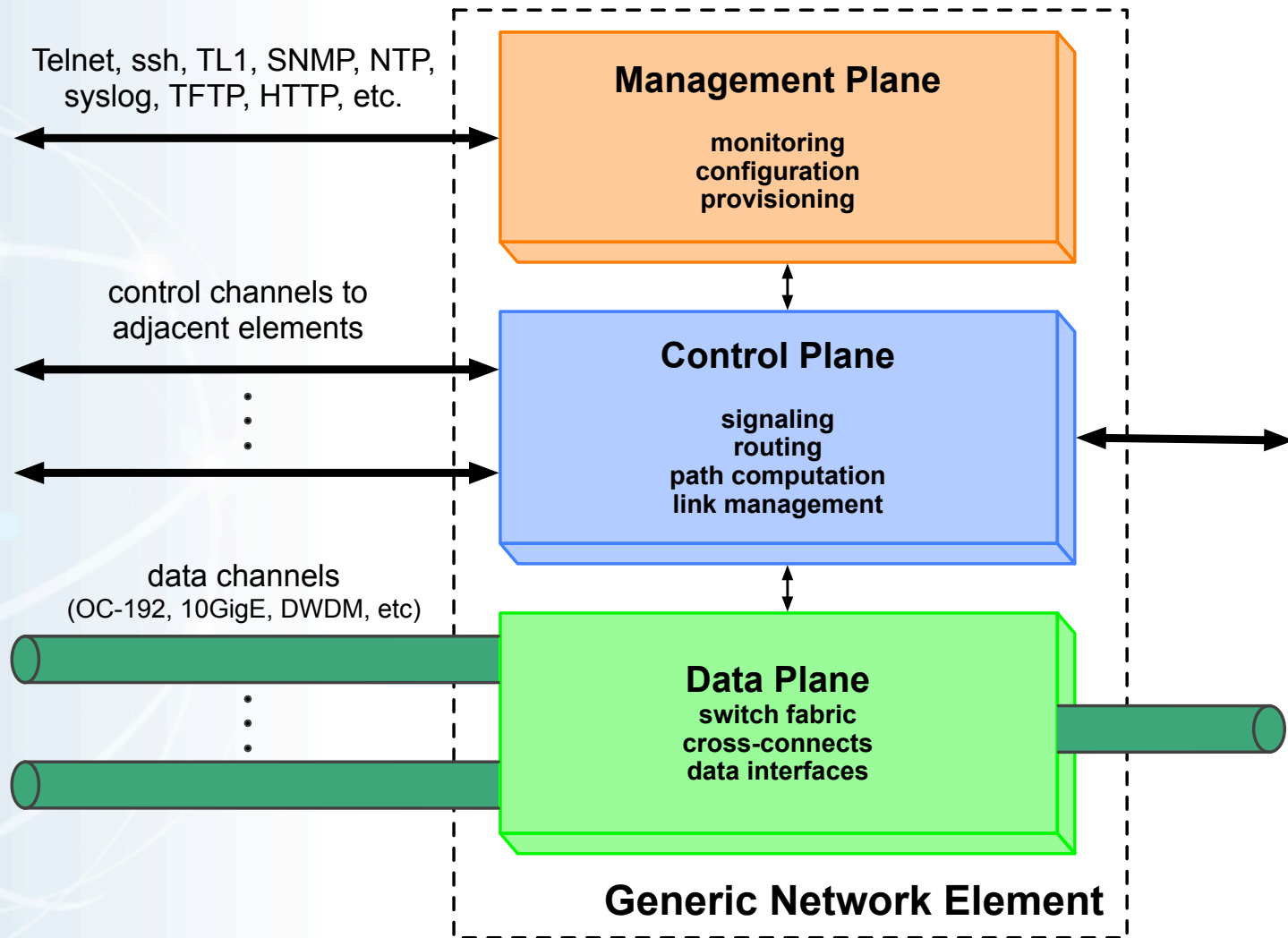Office of Science

BERKELEY LAB

Overview

- Fundamentals and Terminology

- Hardware Options for Deploying OpenFlow

- ESnet OpenFlow Deployment

Fundamentals and Terminology

General view of a router, switch, ROADM

Telnet, ssh, TL1, SNMP, NTP,
syslog, TFTP, HTTP, etc.

Management Plane

**monitoring
configuration
provisioning**

control channels to
adjacent elements

Control Plane

**signaling
routing
path computation
link management**

data channels
(OC-192, 10GigE, DWDM, etc)

**Data Plane
switch fabric
cross-connects
data interfaces**

Generic Network Element

Generic Network Element

This model can be used to generalize many devices:

- IP/MPLS routers

- TDM (SONET/SDH/OTN) or Ethernet switches

- ROADM or wavelength selectable switches

- "hybrid" switches that combine multiple interface switching capabilities in a single box
  - combined lambda + Ethernet + TDM switching in one platform

Right... So how does this relate to OpenFlow?

- OpenFlow is primarily focused on the **data plane**

- exploits the data plane similarities in most networking devices

- each vendor has some kind of **"flow-table"** and a way to program it to establish connectivity through the data plane

OpenFlow in a Nutshell

Directly from the OpenFlow white paper:

- "provides an open protocol to program the **flow-table** in different switches and routers" [1]

How does OpenFlow compare to NETCONF [2] ?

- OpenFlow
  - very focused on opening up access to the flow-table
  - has developed a protocol to do this in a standard way

- NETCONF
  - programmatic way to configure any aspect of a device
  - includes some vendor-independent data schema
  - in reality, a "wrapper" for passing vendor- and platform-specific commands in XML-format (better than `expect` hitting the CLI)
  - same operation for two devices looks different on the wire

What is an OpenFlow Switch?

Must include the following components: [1] [3]

- **Flow Table**: defines how the switch will process each flow
- **Secure Channel**: connects to remote control process (controller)
- **OpenFlow Protocol**: standard for switch--controller interaction

meets OpenFlow Switch Specification [3]
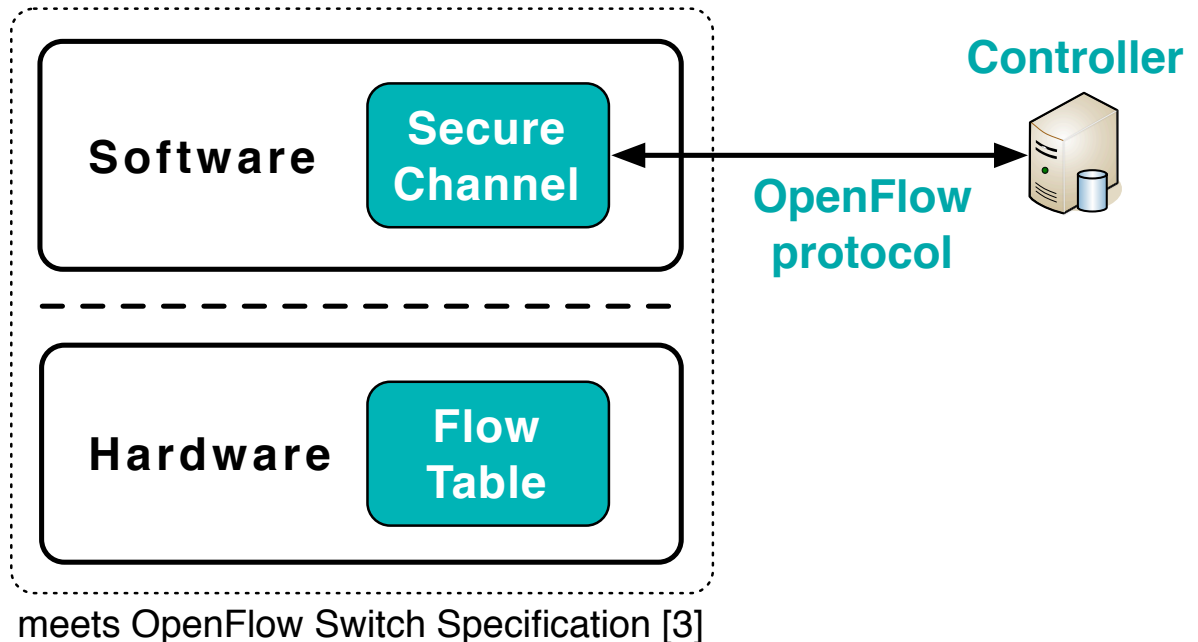
What is an OpenFlow Switch?

Can be classified as either "OpenFlow-only" or "OpenFlow-enabled"
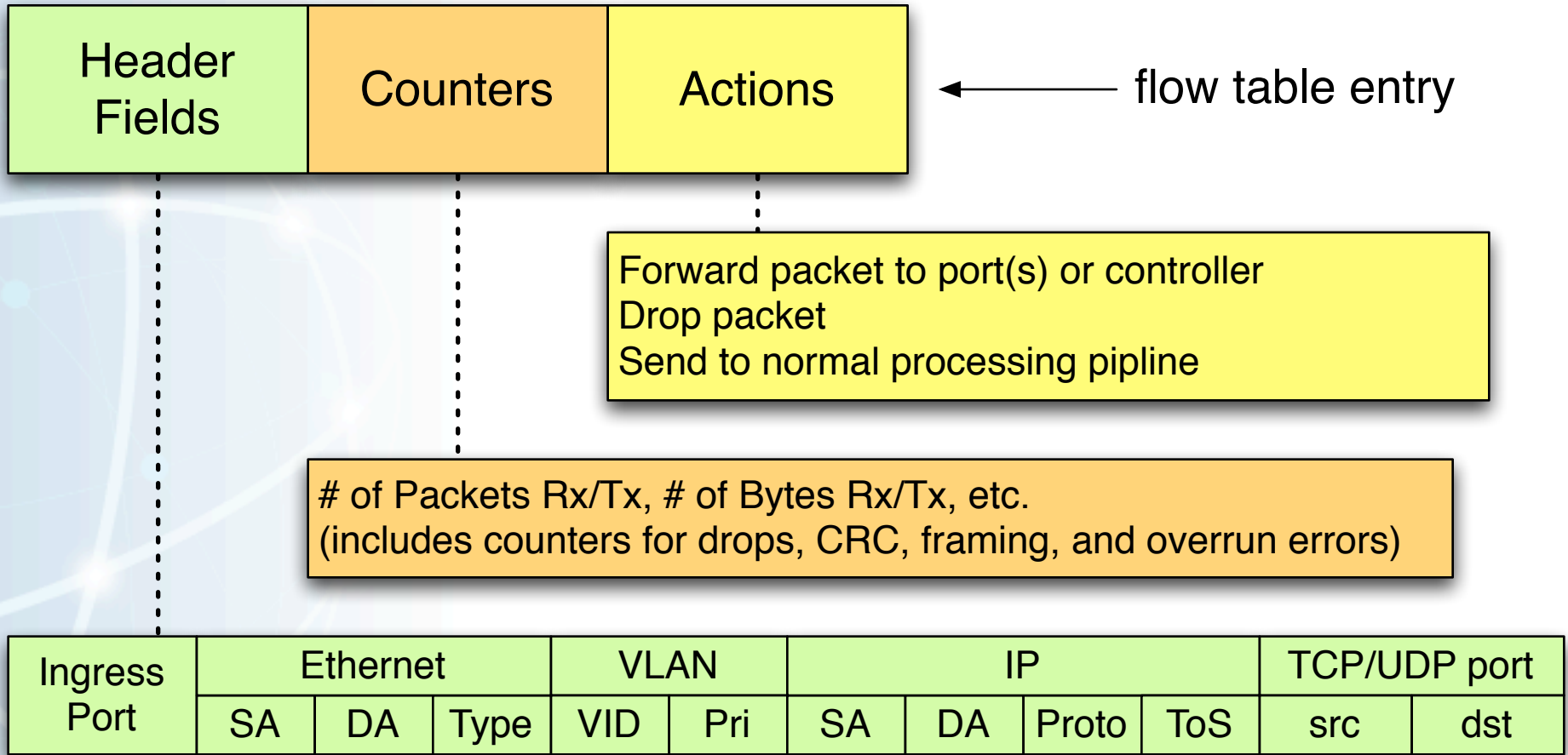
"OpenFlow-only" switch

- supports header formats and actions as defined in OpenFlow Switch Specification [3]

"OpenFlow-enabled" switch

- support everything that an "OpenFlow-only" switch supports
- also supports the "normal" forwarding action
  - process packet using traditional forwarding path of that switch

** NOTE: Earlier version of these slides referenced "Type 0" and "Type 1" in [1]. This classification is no longer used.

What does the Flow Table look like?

Header Fields	Counters	Actions

← flow table entry

Forward packet to port(s) or controller
Drop packet
Send to normal processing pipline

# of Packets Rx/Tx, # of Bytes Rx/Tx, etc.
(includes counters for drops, CRC, framing, and overrun errors)

Ingress Port	Ethernet			VLAN		IP				TCP/UDP port	
	SA	DA	Type	VID	Pri	SA	DA	Proto	ToS	src	dst

Source: concept from [5] updated with information from [3]

What is the Secure Channel?

Control channel between switch and controller

- messages are sent according to the OpenFlow protocol
- allows controller to manage and/or inspect the switch
- switches can send unsolicited messages to the controller
  - packet arrival, state change, error, etc.
- switch is configured with an IP address & port number of controller
- "emergency mode" process deals with connection failures
  - process is defined in [3]

TLS session is established between the switch and controller

- initiated by switch, authenticate via certs signed by site-specific key

Full details can be found in [3] sections 4 and 5

The OpenFlow Protocol

Message types (from [3] section 4.1 "OpenFlow Protocol Overview")

- **controller → switch**:
  - inspect and configure switch
  - query for which capabilities are supported
  - manipulate flows via flow table entries
  - gather statistics

- **asynchronous** (currently, only switch → controller is defined):
  - switch sends unsolicited messages in response to certain events
  - packet arrived which does not match any flow entries
  - switch status change (flow removed, port status has changed)

- **symmetric** (either direction):
  - hello, echo request/reply, vendor-specific messages

The OpenFlow Controller

Manipulates the flow table on behalf of users or applications

Two open-source implementations available:

- **NOX** [14] - most basic controller
- **SNAC** [15] - recommended for production environments [16]

Features are not limited to simply implementing the OpenFlow Protocol
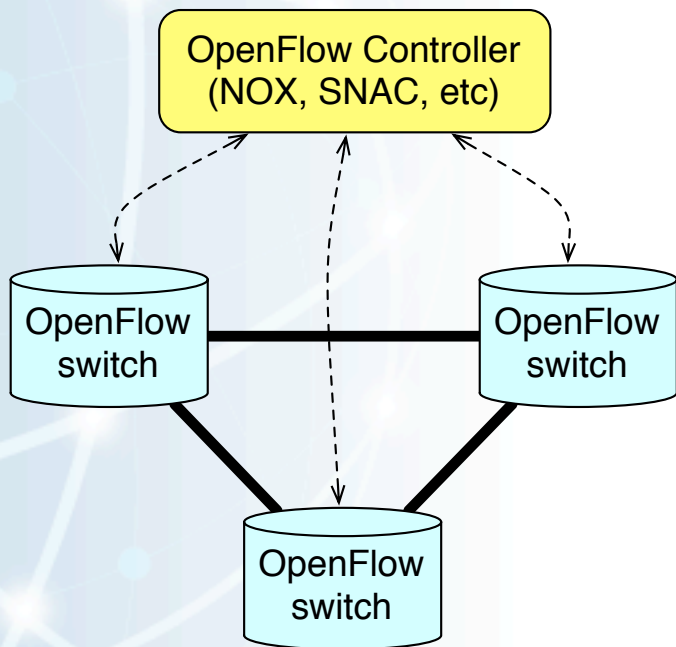
- support for higher-level features
  - admission control
  - full policy engine, flexible policy definition language
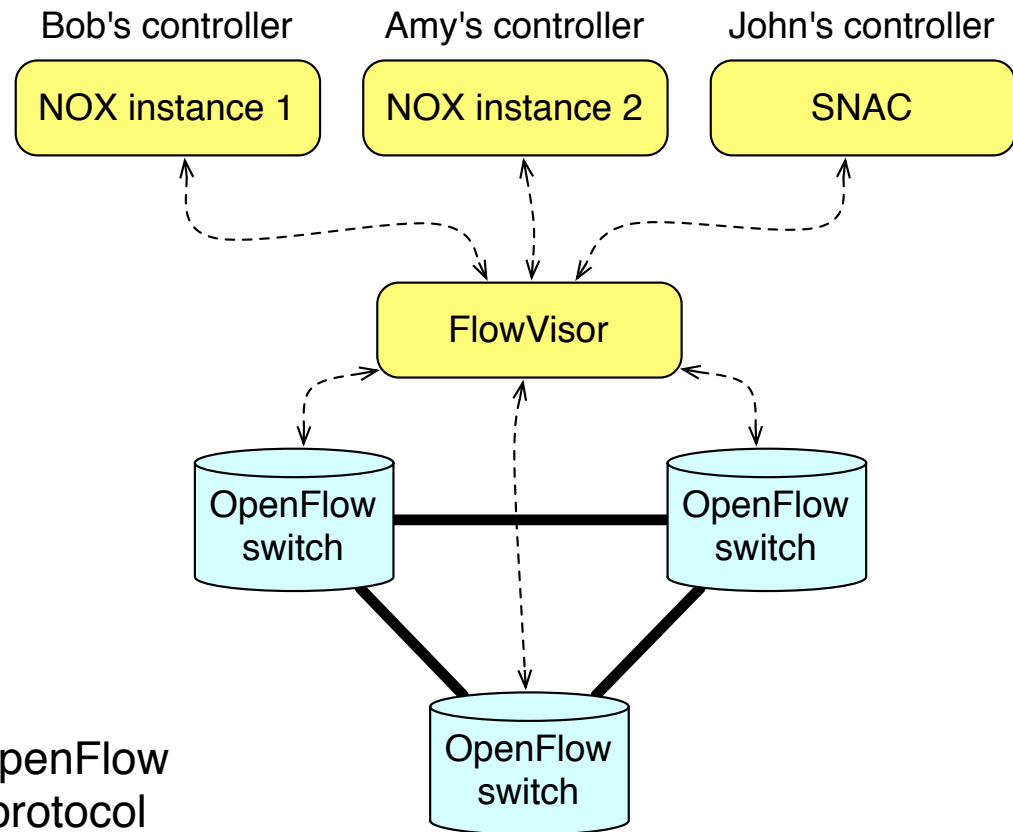
FlowVisor - a special-purpose controller

- transparent proxy between OpenFlow switches and *multiple* controllers
- "slices" network resources, delegates control to each controller

NOX, SNAC, and FlowVisor

Option 1 - just NOX or SNAC:

Option 2 - FlowVisor with *multiple* controllers:

Bob's controller

NOX instance 1

Amy's controller

NOX instance 2

John's controller

SNAC

FlowVisor

OpenFlow Controller (NOX, SNAC, etc)

OpenFlow switch

OpenFlow switch

OpenFlow switch

OpenFlow switch

OpenFlow switch

OpenFlow switch

◄- - - -► OpenFlow protocol

Hardware Options for Deploying OpenFlow

How can I get an OpenFlow switch?

Reference Implementations ("build your own")

- Linux PC: software, using VMs or physical NICs [6] [7]
  - NetFPGA: hardware, 4x1000Base-T per card, line-rate processing
- OpenWRT: software, for wireless access points [8]

Some Commercial Implementations [4] [9] **

- NEC IP8800
- HP ProCurve 3500, 5400, 6600, ... [11]
- Quanta LB4G, Pronto 3240 & 3290
- Toroki Lightswitch [12] (no longer sold)

** **NOTE:** Contact vendor or OpenFlow team for details, <u>don't assume anything</u>!

OpenFlow-enabled Switch

Commercial vendors have taken an interest in supporting OpenFlow

- Level of support currently varies from vendor to vendor

- Some implementations are more experimental
  - Not supported in GA releases
  - Code build not necessarily available through normal channels
  - May be difficult to track down people who knows the details
  - May not implement the latest revision of the OpenFlow Switch Specification

- Other vendors have been willing to offer more support
  - Supported in production quality code builds
  - Features and limitations are documented
  - Can get help if you run into a problem
  - More likely to implement a newer OpenFlow spec

10/04/2010

OpenFlow-enabled Switch

Guidelines regarding commercial OpenFlow-enabled switches

- Still in the early stages of implementation
- Determine which features you are most interested in
  - e.g., what header fields do you want to match on?
- Thoroughly research which vendor/platform combination contains the desired features
  - Communicate with your vendor and/or the OpenFlow team
- Make sure you are comfortable with the level of support offered by the vendor
  - will vendor ship with OpenFlow-enabled firmware pre-installed?
  - or do you have to track it down and install it yourself?

If you like what you are seeing, please <u>ask your vendor to support it!</u>

ESnet OpenFlow Deployment

ESnet OpenFlow Deployment

Configuration Details

- Purchased NEC IP8800 OpenFlow-enabled switches
- Currently running FlowVisor and SNAC controllers
- Other users are running their own controllers
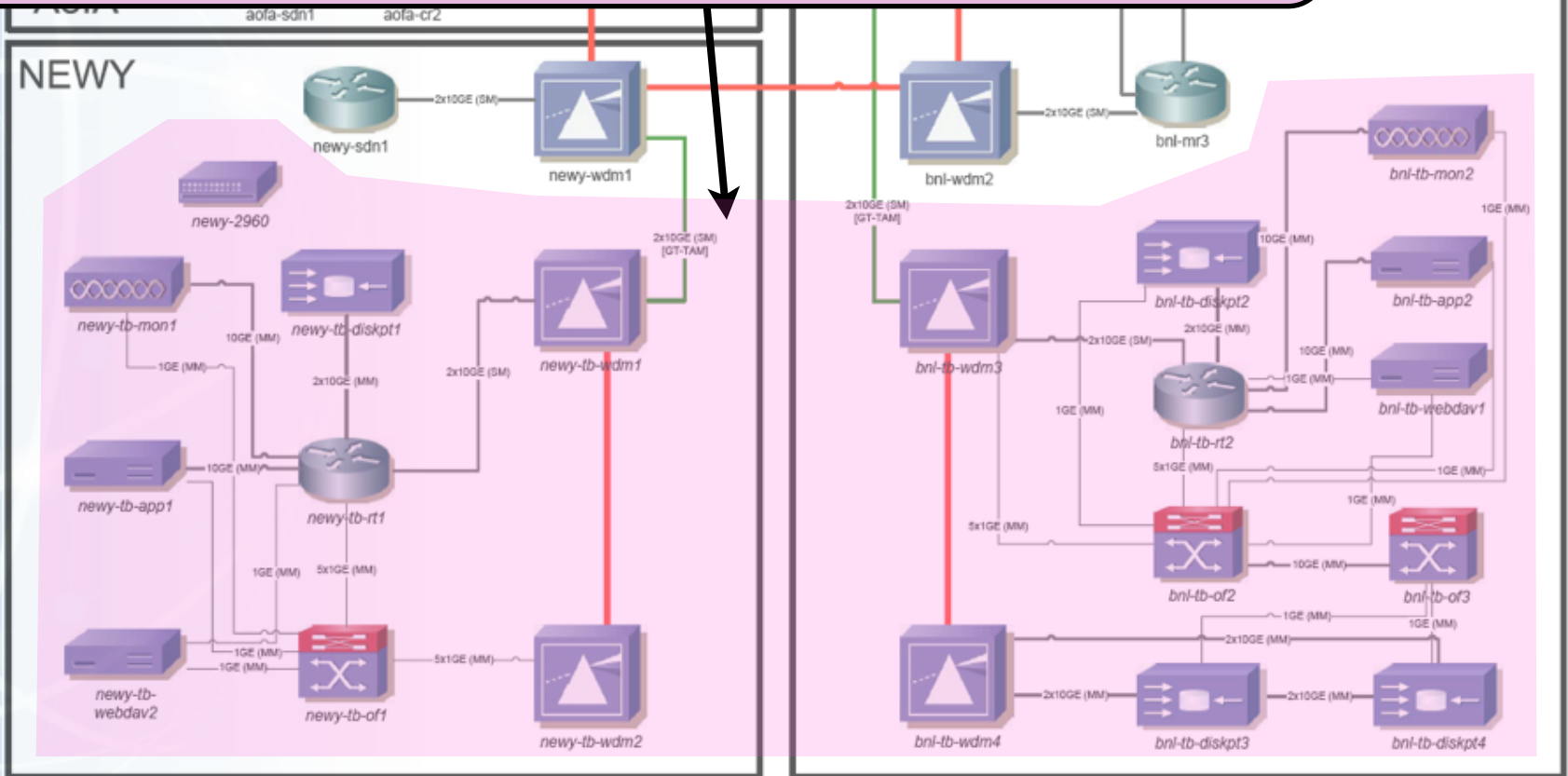 - e.g. NOX talking to FlowVisor

Status

- Currently setup in Berkeley at LBL
- Long Island deployment scheduled for end of 2010
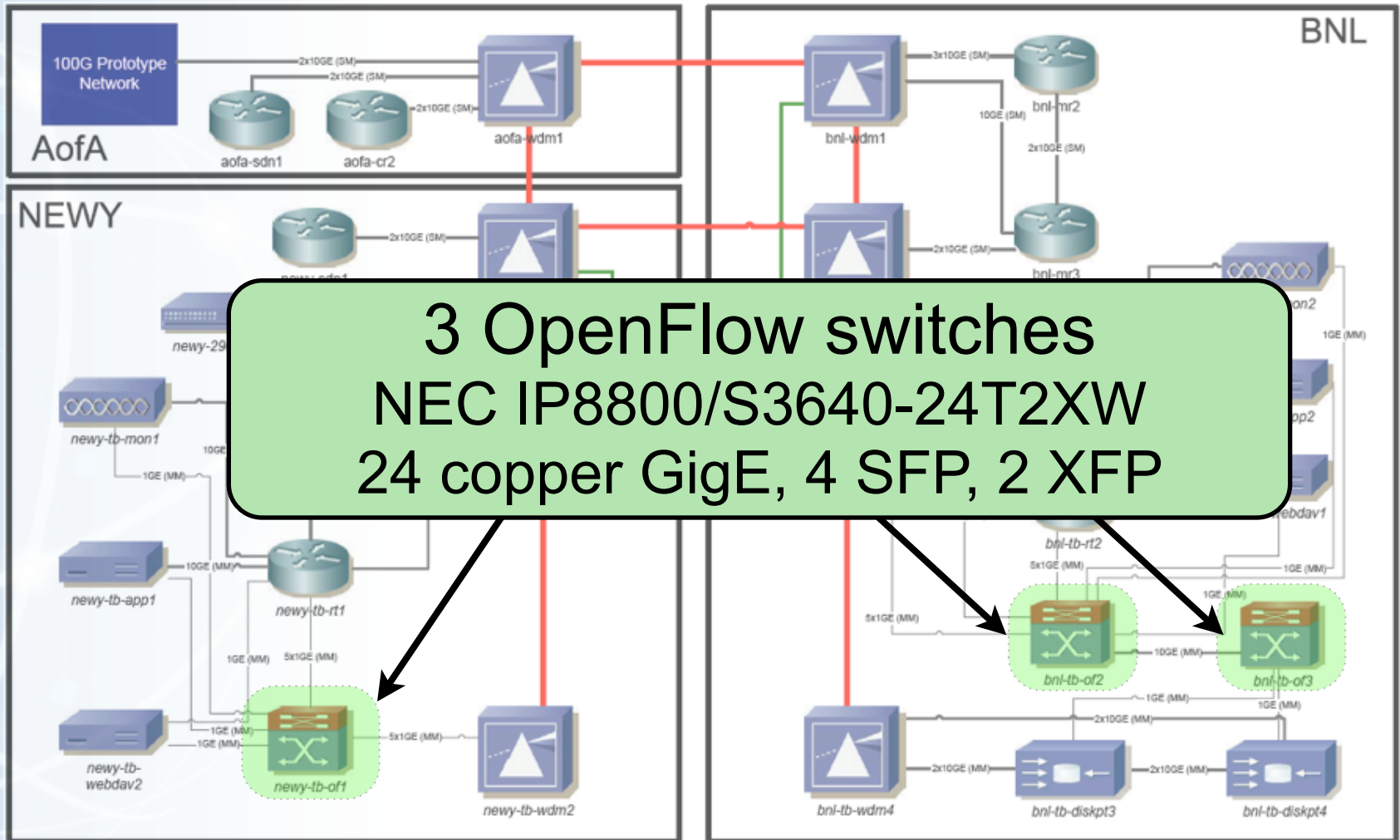
ESnet OpenFlow Deployment

Production DWDM core

Production routers
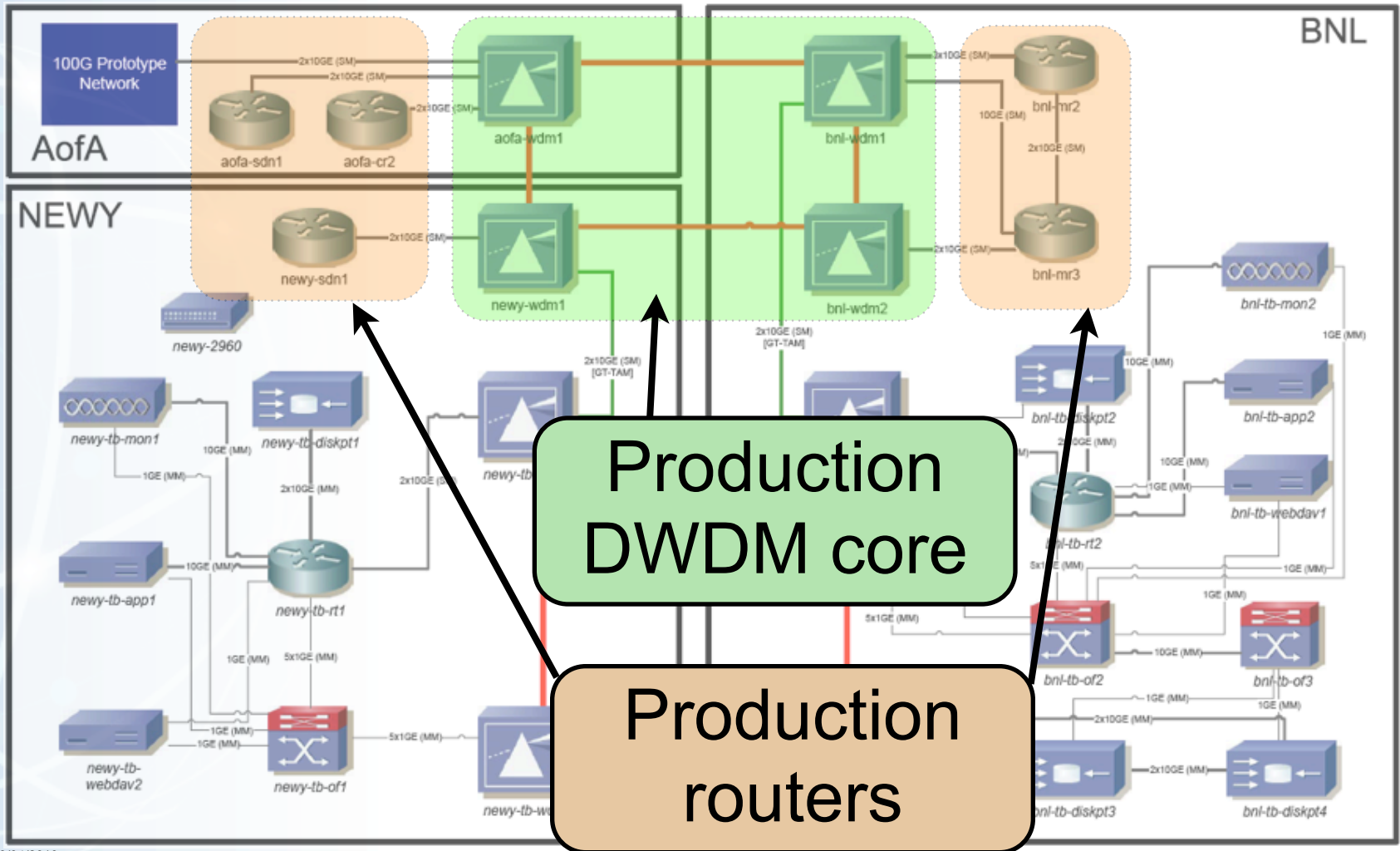
ESnet OpenFlow Deployment

Advanced Networking Initiative (ANI) Testbed

ESnet OpenFlow Deployment

3 OpenFlow switches
NEC IP8800/S3640-24T2XW
24 copper GigE, 4 SFP, 2 XFP

ESnet OpenFlow Deployment

OpenFlow controllers running in VMs

NEC IP8800: virtual switch details

```
openflow-1# showswitch 101 detail
Virtual switch 101
  Datapath ID  : 81117330960(0x12e2f83e10)
  Datapath name: openflow-1
  Port         : gigabitethernet 0/1.101 (link up, 1G full)
                 gigabitethernet 0/2.101 (link down)
                 gigabitethernet 0/6 (link up, 1G full)
                 gigabitethernet 0/7 (link up, 1G full)
                 gigabitethernet 0/8 (link up, 1G full)
                 gigabitethernet 0/9.101 (link down)
                 gigabitethernet 0/11 (link down)
                 tengigabitethernet 0/25 (link down)
                 tengigabitethernet 0/26.101 (link up, 10G full)
  Conn mode    : tcp
  Controller   : 172.31.255.203:7733 (connected)
  Exact match  : hw 0 / hw max 1260
  Wildcard     : hw 0 / hw max 50
  Exact match  : sw 0 / sw max 131072
  Wildcard     : sw 1 / sw max 50
  Emerg        : sw 0 / sw max 50
  Packet buff  : 256 packets / 4294967295 MB
  Miss sendlen : 65535 bytes
  Wcard mode   : Fix
  Counter type : Packet Counter
  MAC learning : On
  Min EQ       : 0
  Max EQ       : 7
  Default EQ   : 4
  VSI Enable   : yes
```

ports in this virtual switch

IP/port of FlowVisor instance

NEC IP8800: flow table entry

```
openflow-1# showflow detail
-----------------------------------------------------------------------
Matching key
  Input port  : *              VLAN ID       : *
  VLAN pri    : *
  MAC SA      : *              MAC DA        : *
  TYPE        : 0x800[IP]      IP protocol : 6[TCP]
  IP SA       : *              IP DA         : *
  ToS         : *
  Source port : *              Dest port   : 80(0x50)
Information
  Idle timeout: 0 sec          Hard timeout: 0 sec
  Priority   : 10
  IFP slice no: -              IFP entry no: -
  Packet count: 0              Byte count  : 0
  SendFlowRem : 0              Emerg Flow  : 0
  Created    : Tue Sep 21 21:25:41.157000000 2010
  Expires    : -
  Cookie     : 16572993974377222320
  Action type : 0[OUTPUT]       Max length  : 65535
                                Output port : 65533(0xfffd)[CONTROLLER]
                                Egress Cos  : 6
-----------------------------------------------------------------------
```
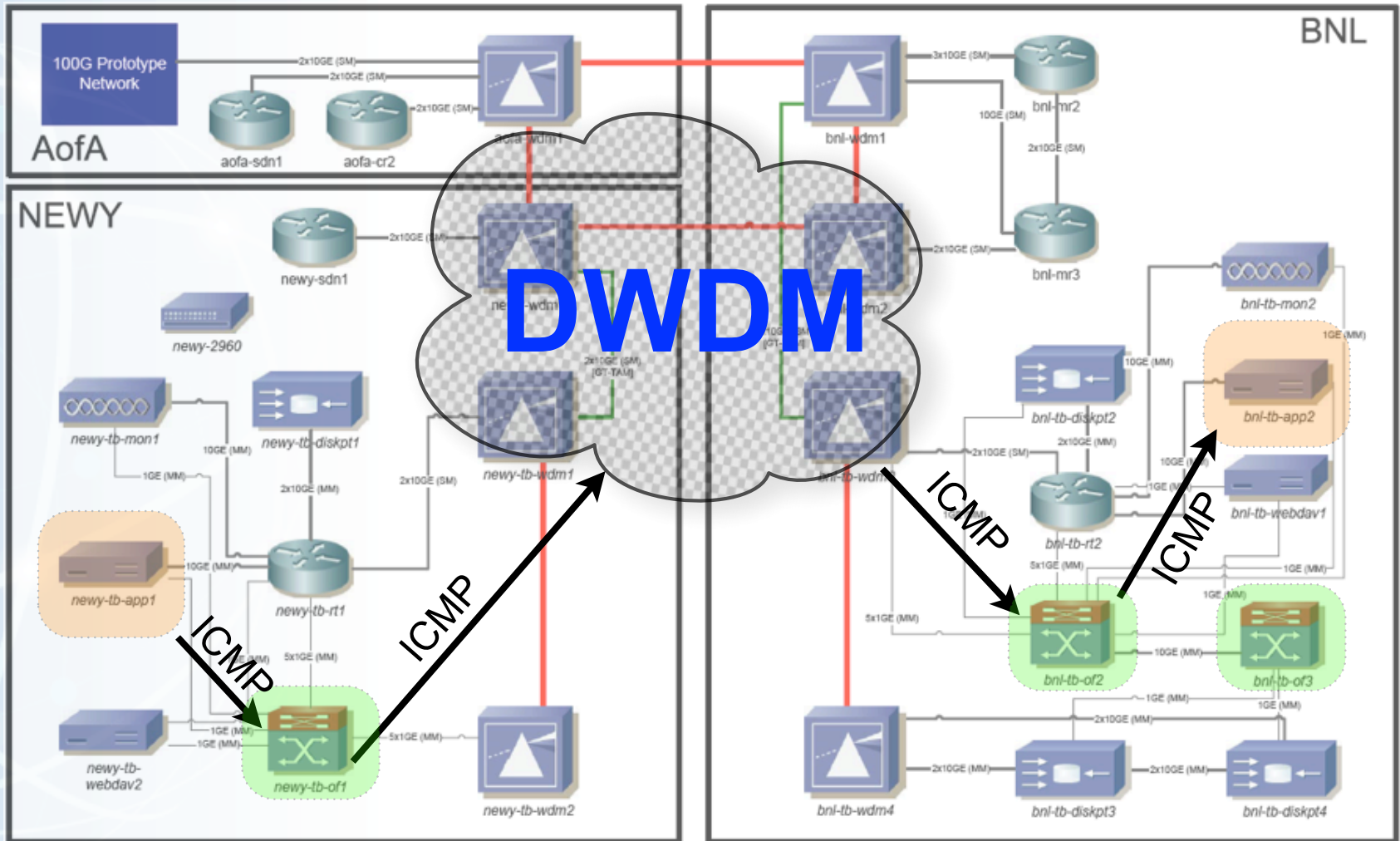
match on header fields

stats

action

SNAC Web GUI (talking to FlowVisor)

OpenFlow Data Plane Demo

OpenFlow Data Plane Demo

```
PING 192.168.255.100 (192.168.255.100) 56(84) bytes of data.
64 bytes from 192.168.255.100: icmp_seq=1 ttl=64 time=75.8 ms
64 bytes from 192.168.255.100: icmp_seq=2 ttl=64 time=0.823 ms
64 bytes from 192.168.255.100: icmp_seq=3 ttl=64 time=0.753 ms
```

DWDM

ping dst (VM)
192.168.255.100

ping src (VM)
192.168.255.195

NEC IP8800

NEC IP8800

ICMP

10/04/2010
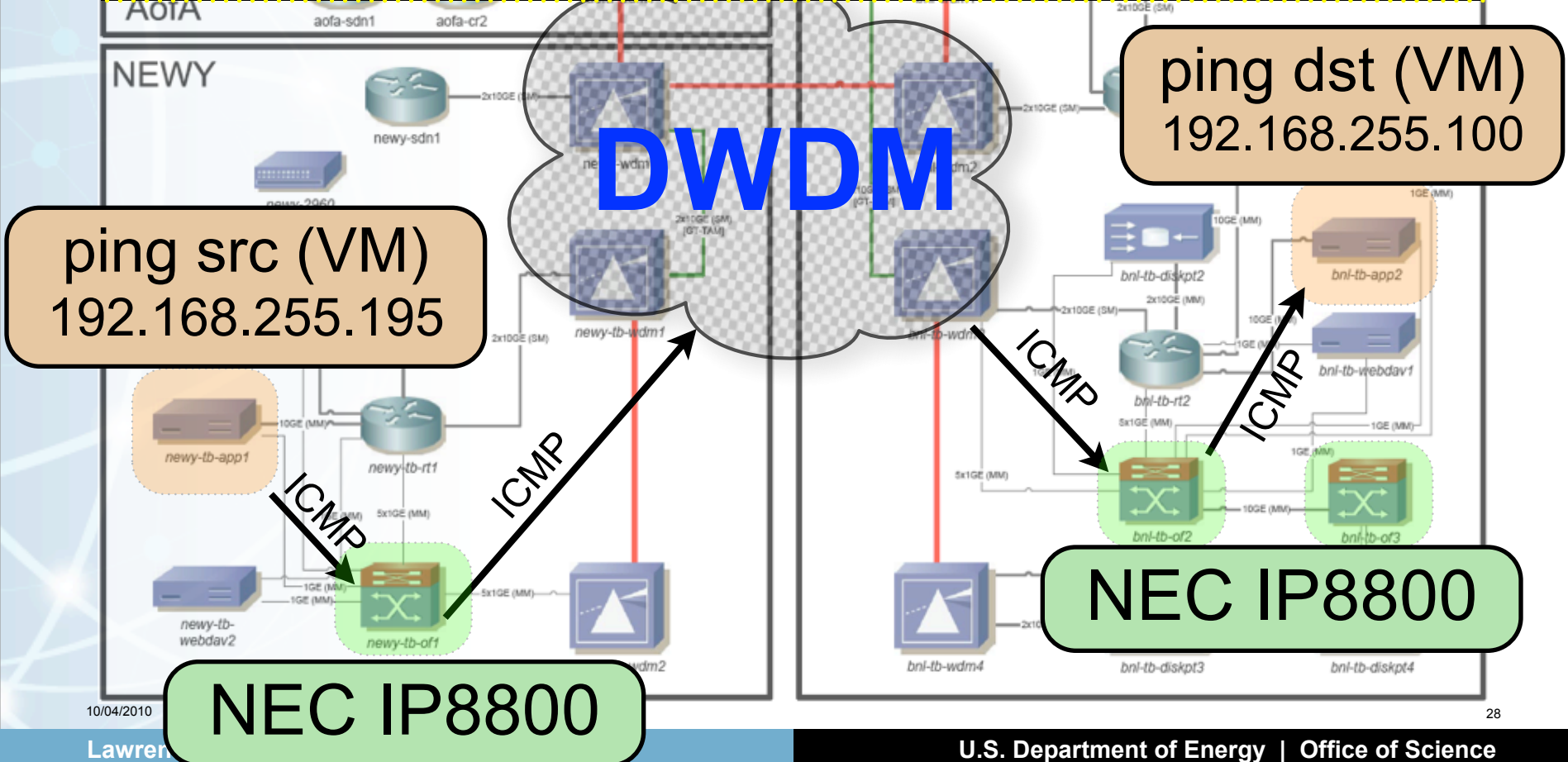
Flow Table - Echo Request

match on header fields

```
openflow-1> showflow detail
--------------------------------------------------------------
Matching key
  Input port  : 1[1]                VLAN ID       : 65535(0xffff)
  VLAN pri    : 0
  MAC SA      : 00:16:36:3a:6b:93   MAC DA        : 00:16:36:6e:78:97
  TYPE        : 0x800[IP]           IP protocol   : 1[ICMP]
  IP SA       : 192.168.255.195/32  IP DA         : 192.168.255.100/32
  ToS         : 0(0)
  Source port : 8(0x8)              Dest port     : 0(0)
Information
  Idle timeout: 5 sec               Hard timeout: 0 sec
  Priority    : 65535
  IFP slice no: 15                  IFP entry no: 2
  Packet count: 121                 Byte count   : 0
  SendFlowRem : 1                   Emerg Flow   : 0
  Created     : Tue Sep 28 19:54:08.857000000 2010
  Expires     : Tue Sep 28 19:56:14.383000000 2010
  Cookie      : 106692613688670231
  Action type : 0[OUTPUT]           Max length   : 0
                                    Output port : 3[6(0x6)]
                                    Egress Cos   : 6
```

stats

action

Flow Table - Echo Reply

match on header fields

```
Matching key
  Input port  : 3[6]            VLAN ID       : 65535(0xffff)
  VLAN pri    : 0
  MAC SA      : 00:16:36:6e:78:97   MAC DA      : 00:16:36:3a:6b:93
  TYPE        : 0x800[IP]           IP protocol : 1[ICMP]
  IP SA       : 192.168.255.100/32  IP DA       : 192.168.255.195/32
  ToS         : 0(0)
  Source port : 0(0)             Dest port    : 0(0)
Information
  Idle timeout: 5 sec           Hard timeout: 0 sec
  Priority   : 65535
  IFP slice no: 15              IFP entry no: 3
  Packet count: 122            Byte count  : 102
  SendFlowRem : 1              Emerg Flow  : 0
  Created    : Tue Sep 28 19:54:08.866000000 2010
  Expires    : Tue Sep 28 19:56:15.243000000 2010
  Cookie     : 83133918480177051O
  Action type : 0[OUTPUT]          Max length  : 0
                                   Output port : 1[1(0x1)]
                                   Egress Cos  : 6
```

stats

action

Discussion Topics

Discussion Topics

IPv6 support

- Currently not supported, but heard that some work was being done

MPLS support

- "An Open-Source Interoperable MPLS LSR" being presented here at NANOG50 by Scott Whyte from Google
  - utilizes NetFPGA card for forwarding
- OpenFlowMPLS project at Ericsson Research [13]

NetFPGA 10G cards - current cards are 4 ports of 1000BASE-T

- Release planned for 2010, but could not find many more details
- liberouter project has 10GigE FPGA-based cards [10]
  - "COMBOv2" series
  - no OpenFlow implementation

References

[1] OpenFlow: Enabling Innovation in Campus Networks

    http://www.openflowswitch.org/documents/openflow-wp-latest.pdf

[2] RFC4741 - NETCONF Configuration Protocol

    http://www.faqs.org/rfcs/rfc4741.html

[3] OpenFlow Switch Specification, Version 1.0.0

    http://www.openflowswitch.org/documents/openflow-spec-v1.0.0.pdf

[4] Configuring OpenFlow switches

    http://www.openflowswitch.org/foswiki/bin/view/OpenFlow/Deployment/HOWTO/Configuring

[5] OpenFlow presentation

    http://www.openflowswitch.org/documents/OpenFlow.ppt

[6] Create an OpenFlow Network within a single PC

    http://www.openflowswitch.org/foswiki/bin/view/OpenFlow/Deployment/HOWTO/Virtual

Lawrence Berkeley National Laboratory   **U.S. Department of Energy | Office of Science**

References

[7] Create OpenFlow network with multiple PCs/NetFPGAs

http://www.openflowswitch.org/foswiki/bin/view/OpenFlow/Deployment/HOWTO/LabSetup

[8] OpenFlow 1.0 for OpenWRT

http://www.openflowswitch.org/wk/index.php/OpenFlow_1.0_for_OpenWRT

[9] Production Setup

http://www.openflowswitch.org/foswiki/bin/view/OpenFlow/Deployment/HOWTO/ProductionSetup

[10] liberouter Programmable hardware

http://www.liberouter.org/hardware.php

[11] Orphal: API Design Challenges for Open Router Platforms on Proprietary Hardware

http://www.hpl.hp.com/techreports/2008/HPL-2008-108.pdf

[12] Toroki: LS4810

http://www.toroki.com/prd_toroki_ls4810.php

Lawrence Berkeley National Laboratory

U.S. Department of Energy | Office of Science

References

[13] OpenFlowMPLS

http://www.openflowswitch.org/wk/index.php/OpenFlowMPLS

[14] NOX: An OpenFlow Controller

http://noxrepo.org/wp/

[15] SNAC: Simple Network Access Control (SNAC)

http://www.openflowswitch.org/wp/SNAC/

[16] Controller Setup

http://www.openflowswitch.org/foswiki/bin/view/OpenFlow/Deployment/HOWTO/ProductionSetup/ControllerSetup

[17] FlowVisor

http://www.openflowswitch.org/wk/index.php/FlowVisor

Lawrence Berkeley National Laboratory

U.S. Department of Energy | Office of Science

Questions?

Thanks!