big switch networks

FlowVisor Engineering Tutorial
Open Networking Summit 2012

Rob Sherwood

Talk Outline

- Glossary: Flavors of Virtualization
- Use Cases
- Case-study: FlowVisor network slicing tool
  - Architecture
  - Opt-In
  - Resource allocation: FlowSpace
  - Deployments
- Virtualization discussion

Flavors of Network Virtualization

Rob's break down for ease of discussion

- Topology Virtualization
  - Who decides the network topology?
  - Virtual links, broadcast isolation, virtual nodes
  - e.g., tunnels, VLANs, "single-touch-management" solutions
- Address Virtualization
  - Who controls 10.0.0.0/8?
  - Prevent MAC address collisions
  - e.g., VRF + MPLS, VLANs
- This talk: *Policy* Virtualization; a.k.a. network *slicing*
  - Who sets ACLs? Who decides forwarding paths?
  - e.g., FlowVisor: *www.flowvisor.org*
- Commercial SDN solutions emerging that combine all three!

Network Virtualization Use Cases

- Multi-tenancy
  - Public/Private Cloud service providers
- Better resource utilization
  - Move traffic around by load – turn off the rest
- Simplified Management
  - Network policies migrate with VMs
  - Decouple policy from topology
  - Multi-layer delgation
- Rapid/Isolated service deployment
  - FlowVisor's initial use case

FlowVisor's Original Problem:

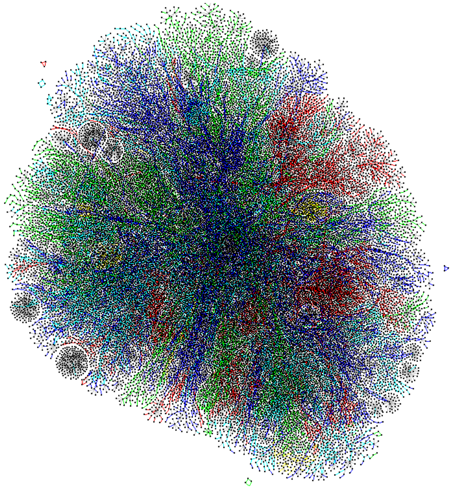
- Realisticly evaluating new network services is *hard*

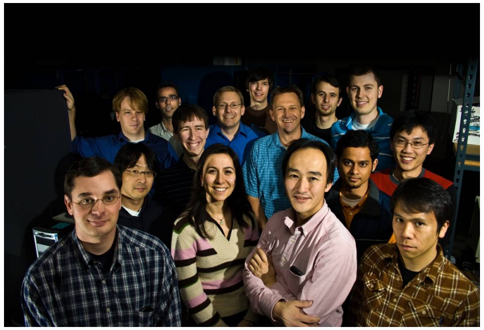
- services that require changes to switches and routers
- e.g.,
  - routing protocols
  - traffic monitoring services
  - IP mobility

Result: Many good ideas don't gets deployed;
Many deployed services still have bugs.

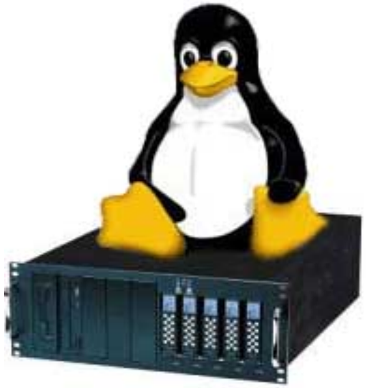
Why is Evaluation Hard?

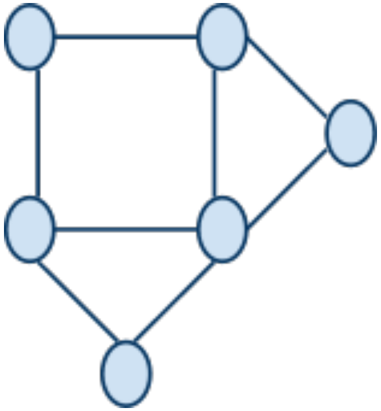
Real Networks

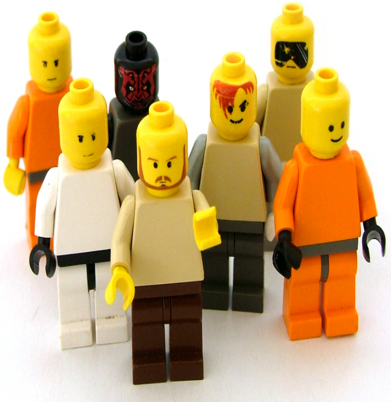




Testbeds







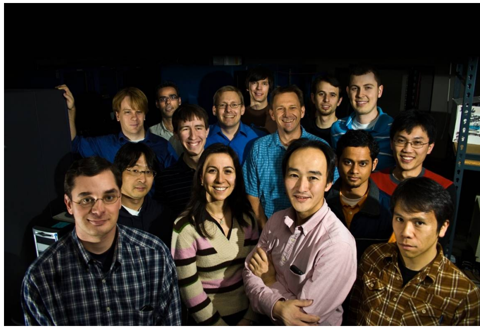
Network Slicing with FlowVisor

- A network slice is a collection of sliced switches/routers

- Data plane is unmodified
  - Packets forwarded with no performance penalty
  - Works with existing hardware/ASICs

- Transparent slicing layer
  - each slice believes it owns the data path
  - enforces isolation between slices
    - i.e., rewrites, drops rules to adhere to slice police
  - forwards exceptions to correct slice(s)

Real User Traffic: Opt-In



- Allow users to Opt-In to services in real-time
  - Users can delegate control of individual flows to Slices
  - Add new FlowSpace to each slice's policy

- Example:
  - *"Slice 1 will handle my HTTP traffic"*
  - *"Slice 2 will handle my VoIP traffic"*
  - *"Slice 3 will handle everything else"*

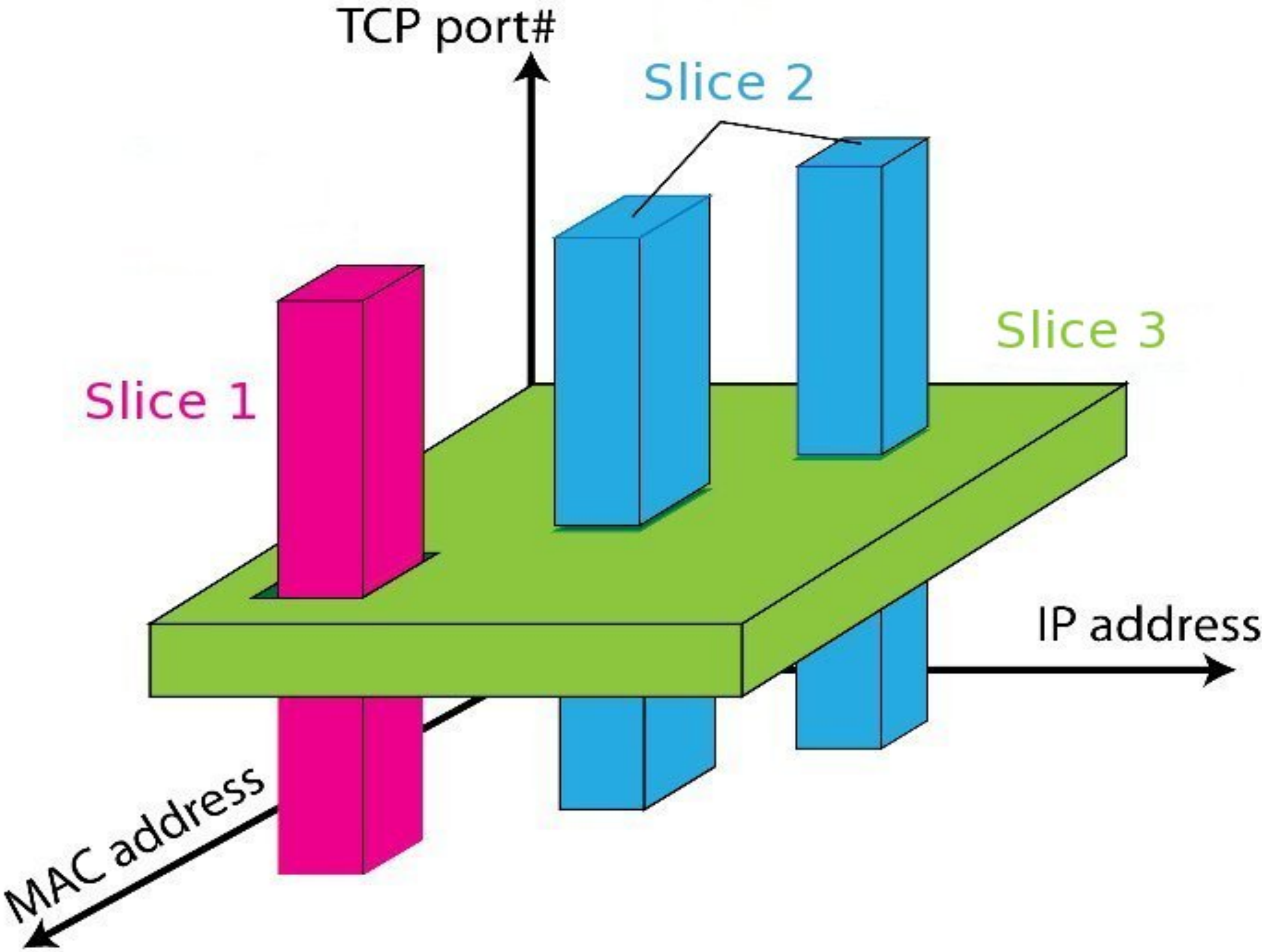
- Creates incentives for building high-quality services

Slicing Policies

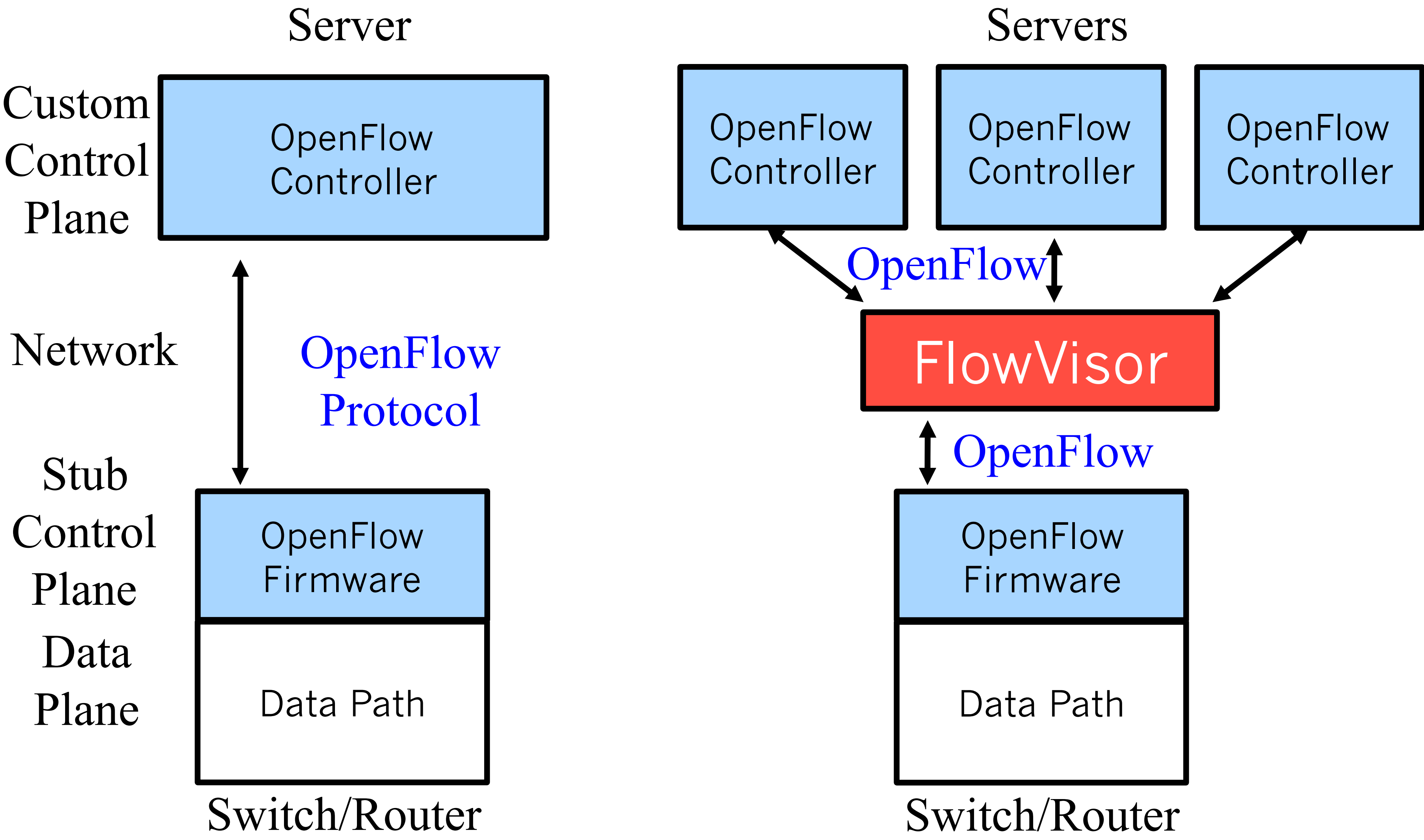
- The policy specifies resource limits for each slice:
  - Link bandwidth
  - Maximum number of forwarding rules
  - Topology
  - Fraction of switch/router CPU

  - *FlowSpace: which packets does the slice control?*

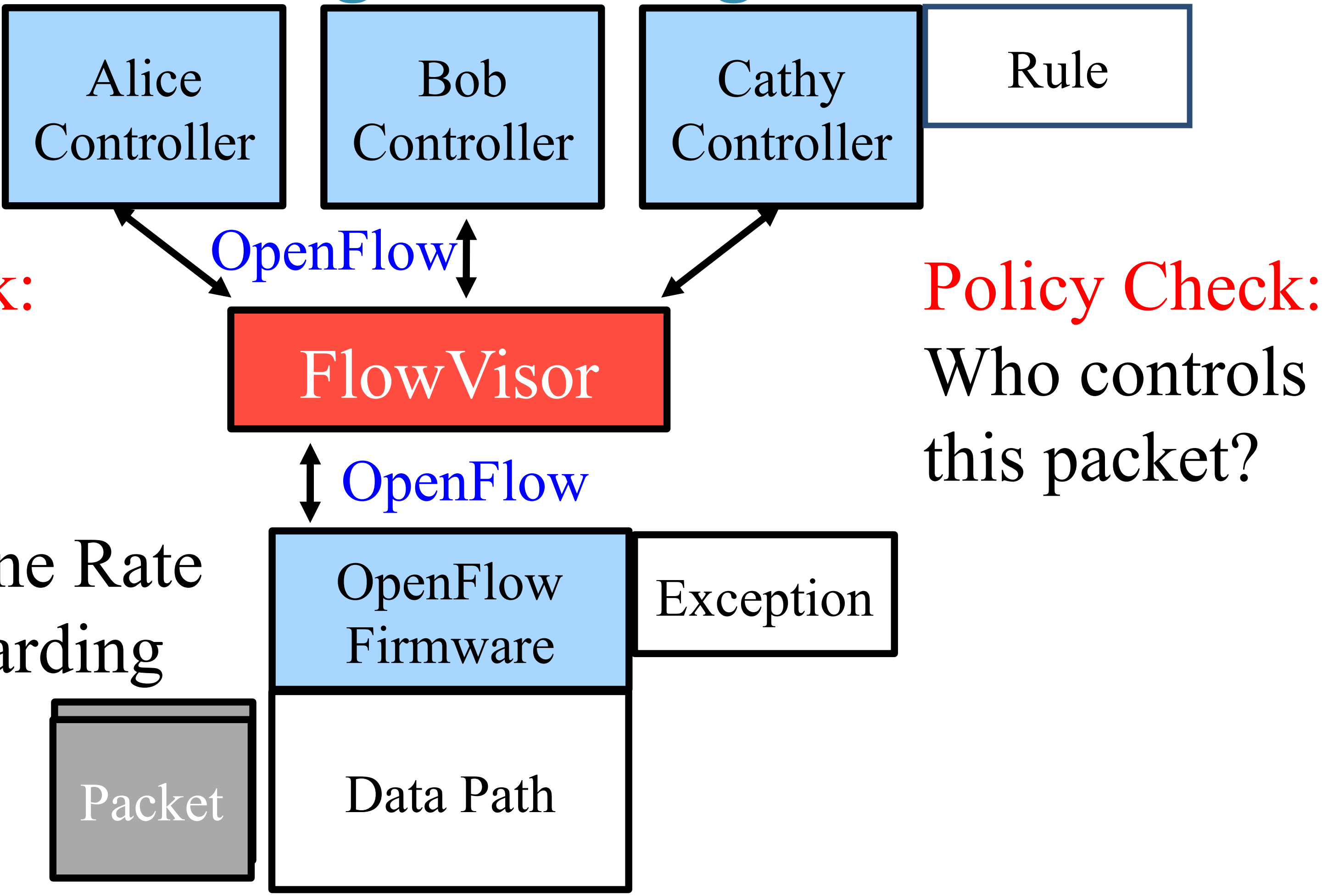
FlowSpace: Maps Packets to Slices



FlowVisor Architecture



FlowVisor Message Handling



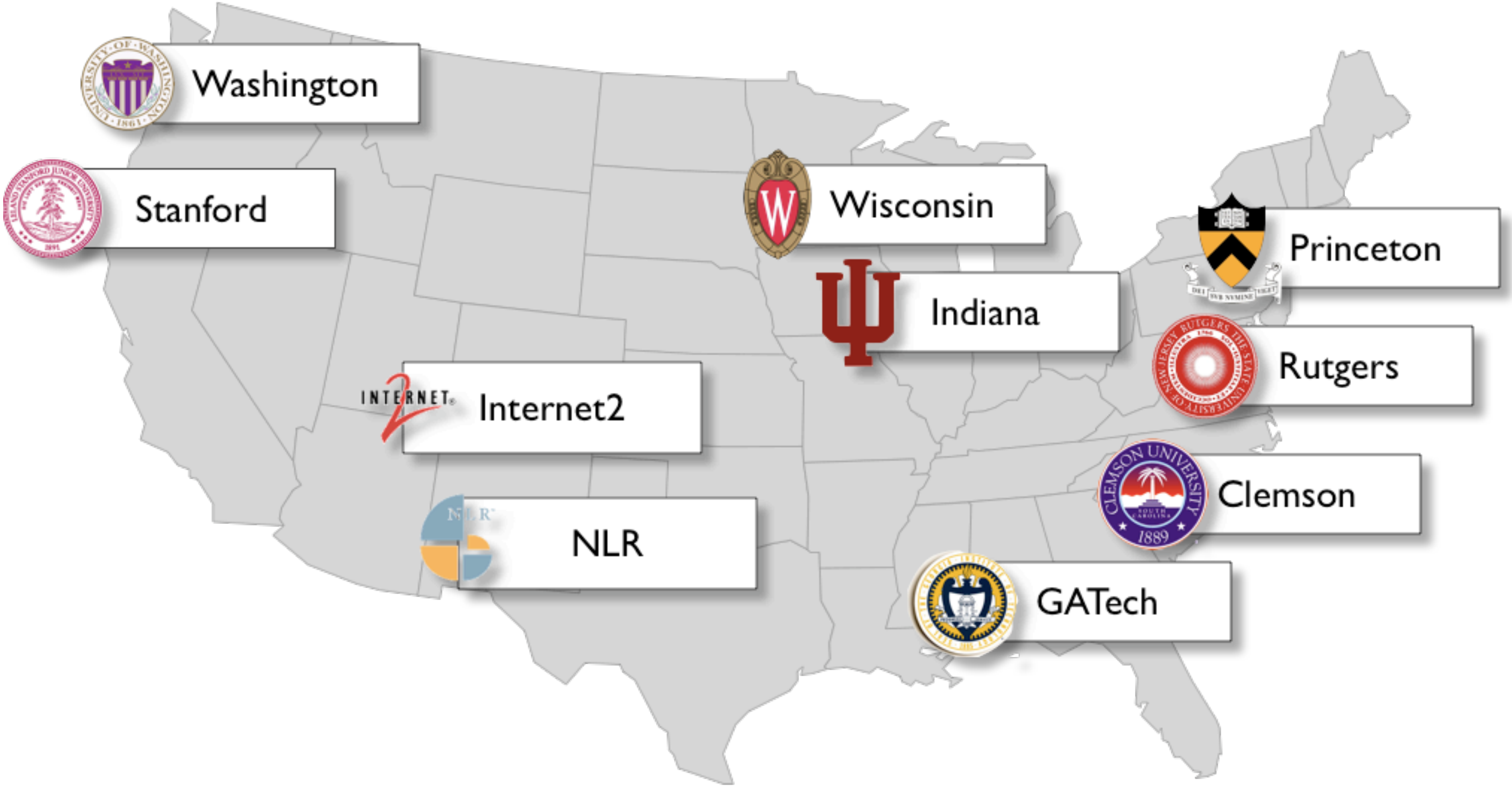
FlowVisor Deployment: Stanford

- Our real, production network
  - 15 switches, 35 APs
  - 25+ users
  - 2+ year of use
  - my personal email and web-traffic!

- Same physical network hosts Stanford demos
  - 7 different demos

FlowVisor Deployments: GENI

big switch networks



FlowVisor Commands

```
> /usr/sbin/flowvisor /usr/etc/flowvisor/config.xml &
> man flowvisor
> man fvconfig
> man fvctl
```

FVCTL

- fvctl is the cli used to control a running instance of flowvisor (over XMLRPC)
- fvctl --passwd-file=/etc/flowvisor/fvpasswd command [args…]
- fvctl command [args…]

> fvctl createSlice

- Specifies a controller/slice
  - fvctl createSlice <slicename> <controller_url> <email>
    - controller_url = tcp:<ip address>:<port #>

  - ***fvctl createSlice FinanceDept tcp: 155.55.5.5:6633 bob@finance.example.edu***

> fvctl addFlowSpace

- Insert a flowspace rule
  - fvctl addFlowSpace <dpid> <priority> <match> <actions>
  - **fvctl addFlowSpace 00:c8:08:17:f4:4b:82:00 100 in_port=22 Slice:ESNet=4**

> fvctl addFlowSpace: match

- Flow packet match fields
  - **in_port**=port_no
  - **dl_vlan**=vlan
  - **dl_src/dl_dst**=mac
  - **nw_src/nw_dst**=ip[/netmask]
  - **nw_proto**=proto
  - **nw_tos**=tos/dscp
  - **tp_src/tp_dst**=port
  - *More in fvctl man page*

> fvctl addFlowSpace: actions

- Comma-separated list of slices with control permissions over matching flowspace
- Permissions
  - Delegate=1
  - Read=2
  - Write=4
- Ex. Slice:IU-LB=4,Monitor=2

> fvctl getSliceInfo

- Dumps information about the slice
  - controller_port=6633
  - controller_hostname=140.221.223.153
  - creator=fvadmin
  - contact_email=bob@example.edu

> fvctl changeSlice

- Edit a slice attributes:
  - controller_port=6633
  - controller_hostname=140.221.223.153
  - creator=fvadmin
  - contact_email=you@example.edu
- Examples:
  - fvctl changeSlice <slicename> <key> <value>
  - **fvctl changeSlice iCAIR controller_port 6644**

> fvctl listSlices ; fvctl deleteSlice

- lists the slices that have been created
  - Slice 0: iCAIR
  - Slice 1: CPQD
  - Slice 2: Clemson
  - Slice 3: I2-NOX
  - Slice 4: IU-100G
  - Slice 5: SARA
  - Slice 6: ESNet
  - Slice 7: fvadmin
- **fvctl deleteSlice ESNet**

> fvctl listDevices

- List DPID of all connected OpenFlow devices

```
Device 0: 00:00:0e:83:40:39:18:58

Device 1: 00:00:0e:83:40:39:1a:57

Device 2: 00:00:0e:83:40:39:19:96

Device 3: 00:00:0e:83:40:39:1b:93

Device 4: 00:00:0e:83:40:39:18:1b

Device 5: 00:00:0e:84:40:39:19:96

Device 6: 00:00:0e:84:40:39:1a:57

Device 7: 00:00:0e:84:40:39:1b:93

Device 8: 00:00:0e:84:40:39:18:1b

Device 9: 00:00:0e:84:40:39:18:58
```

> fvctl getLinks

- List port # and DPID of both ends of each link
- Link 0: Link[srcDPID=00:00:0e:83:40:39:1b:93,srcPort=2,dstDPID=00:00:0e:83:40:39:18:1b,dstPort=2]
- Link 1: Link[srcDPID=00:00:0e:84:40:39:18:1b,srcPort=2,dstDPID=00:00:0e:84:40:39:1b:93,dstPort=2]

Virtualization Discussion

- Why OpenFlow for both Sound and North interface?
  - Why not in the *controller* with a custom North Bound API?
  - Why not in the *switch*, with a custom Sound Bound API
- Decouple virtualization from control
  - Let them evolve independently
  - FlowVisor works with most (all?) publicly available controllers
- Incrementally deployable
  - First: Deploy in slice
  - Then deploy same code directly on real network

Conclusion

- FlowVisor is an OpenFlow-based virtualization tool
- Deployed in lots of R&E networks, Interop 2011-12
- Open source, documentation, tools
- Thanks to Ron Milford/InCntre for slides!

www.flowvisor.org