Enterprise
GENI

Nick McKeown
nickm@stanford.edu

Stanford OpenFlow team: Guido Appenzeller, Glen Gibb, David Underhill, David Erickson, Adam Covington, Brandon Heller, Rob Sherwood, Masayoshi Kobayashi, Guru Parulkar, Srinivasan Seetharaman, Yiannis Yiakoumis
College Campus Network

Enterprise GENI

GENI Clearing House

Enterprise GENI

Enterprise GENI
Staged Approach

1. Define OpenFlow feature
2. Add OpenFlow to commercial switches and APs
3. Deploy at Stanford
4. Integrate with GENI Clearing House
5. Deploy on many college campus networks
6. We all create lots of open-source software so researchers can build on each other’s work

(We’re part-way into Stage 2)
OpenFlow Basics
OpenFlow Basics (1)

Exploit the flow table in switches, routers, and chipsets

Flow 1.

<table>
<thead>
<tr>
<th>Rule (exact &amp; wildcard)</th>
<th>Action</th>
<th>Statistics</th>
</tr>
</thead>
</table>

Flow 2.

<table>
<thead>
<tr>
<th>Rule (exact &amp; wildcard)</th>
<th>Action</th>
<th>Statistics</th>
</tr>
</thead>
</table>

Flow 3.

<table>
<thead>
<tr>
<th>Rule (exact &amp; wildcard)</th>
<th>Action</th>
<th>Statistics</th>
</tr>
</thead>
</table>

Flow N.

<table>
<thead>
<tr>
<th>Rule (exact &amp; wildcard)</th>
<th>Default Action</th>
<th>Statistics</th>
</tr>
</thead>
</table>
OpenFlow Basics (2)

Rule (exact & wildcard)
- As **general** as possible
  - e.g. Port, VLAN ID, L2, L3, L4, ...
- As **wide** as possible

Action

Statistics
- Count packets & bytes
- Expiration time/count

Small number of fixed actions
- e.g. unicast, mcast, map-to-queue, drop
  - Extended via virtual ports
  - e.g. tunnels, encapsulate, encrypt
OpenFlow Basics (3)

- OpenFlow Switch specification
- Secure Channel
- Flow Table
- PC

Controller

Add/delete flow entries
Encapsulated packets
Controller discovery
OpenFlow Usage

Dedicated OpenFlow Network

Controller

Chip’s code

OpenFlow Protocol

Chip
Usage examples

Chip’s code:
- Static “VLANs”
- His own new routing protocol: unicast, multicast, multipath, load-balancing
- Network access control
- Home network manager
- Mobility manager
- Energy manager
- Packet processor (in controller)
- IPvChip
- Network measurement and visualization
- ...
Enable Innovation in your Network.

OpenFlow is an open standard that allows you to run experimental protocols in production networks. It is in the process of being implemented by major switch vendors and used today by universities to deploy innovative networking technology in their campus networks.
Let’s see an example...
- Experimental feature on ProCurve 5400-series
- 144-ports of 1GE, hardware forwarding
- OpenFlow added by HP Labs and ProCurve group
- In 23 wiring closets in CS Building at Stanford
- Experimental feature on IP8800 series router
- 24-ports of 1GE, 2-ports of 10GE, hardware forwarding
- OpenFlow added by NEC team in Japan
- NEC announced plans for OpenFlow products
- Deployed at Stanford and in JGN2plus in Tokyo
Juniper

- OpenFlow added to Junos SDK
- First platform: MX-480 carrier class Ethernet
- 24-ports 10GE or 240-ports 1GE
- Hardware forwarding
- Deployed in Internet2 in NY and at Stanford
Cisco

- Experimental feature on Catalyst 6509
- Software forwarding
- Deployed at Stanford

Pere Monclus  Sailesh Kumar  Flavio Bonomi
Nicira

- Created NOX controller
- Available at [http://NOXrepo.org](http://NOXrepo.org) (GPL)
- Deployed at Stanford

Martin Casado  
Scott Shenker  
Teemu Koponen  
Natasha Gude  
Justin Pettit
Using Enterprise GENI for research
Step 1: Separate VLANs for Production and Research Traffic

Normal L2/L3 Processing

Flow Table

Research VLANs

Production VLANs

Controller
Step 2: Virtualize OpenFlow Switch
Virtualizing Control

Netadmin's experiments

Multicast

http Load-balancer

OpenFlow Protocol

OpenFlow Hypervisor & Policy Control

OpenFlow Switch

Netadmin's experiments

Multicast

http Load-balancer

OpenFlow Protocol

OpenFlow Hypervisor & Policy Control
Enterprise GENI

OpenFlow Hypervisor & Policy Control

GENI Aggregate Manager

GENI Clearing House

OpenFlow Protocol

OpenFlow Protocol

OpenFlow Protocol
Many Open Questions!

- Scalability of a controller
- Load-balancing over redundant controllers
- Federation, hierarchy and aggregation
- Protecting the controller against DDOS

Our goal is to enable the research community to explore all these questions
Thanks…

(It takes a village)
Internet2 Team

Chris Small

Matt Zekauskas
Installing Juniper MX-480 in NY
We plan trials in early 2009 5-6 college campuses

Contact us if you would like to take part
nickm@stanford.edu