OpenFlow: What is it and Where is it going?

Rob Sherwood
NANOG 53
Talk Outline

• Background and Problem
• What is OpenFlow?
• Use Cases
• Standardization:
  – Open Networking Foundation (ONF)
  – Interoperability
• Deployments and Adoption
BACKGROUND AND PROBLEM
Many People’s Ideas in This Talk

• OpenFlow is a much larger movement
  – Scott Shenkar, Nick McKeown, Guru Parulkar
  – Martin Cassado, Guido Appenzeller
  – Jean Tourrilhes, Dan Pitt
    • Many more in the ONF WGs

• About Me:
  – Three years working “in the trenches” on OpenFlow
  – Currently at OpenFlow startup, Big Switch Networks
  – Wearing my “OpenFlow Evangelist” hat
Don’t All of Our Problems Have Solutions?

<table>
<thead>
<tr>
<th>Problem</th>
<th>Optical</th>
<th>L2</th>
<th>L3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Virtualization</td>
<td>WDM</td>
<td>Vlans, QinQ</td>
<td>NAT, MPLS, VRF</td>
</tr>
<tr>
<td>Load balancing</td>
<td>proprietary</td>
<td>TRILL, LAG, VM placement</td>
<td>MPLS-TE, ECMP, BGP prepending</td>
</tr>
<tr>
<td>Reservations</td>
<td>Manual provisioning</td>
<td>Vlan pcp, FCoE flow control</td>
<td>DiffServ, MPLS AutoBandwidth</td>
</tr>
<tr>
<td>.... More problems</td>
<td>.... Litany of RFCs</td>
<td>.... More standards</td>
<td>.... Alphabet soup</td>
</tr>
</tbody>
</table>

• but what is the *solution* to the solutions?
Complexity *is* the Problem

- A new solution for each problem doesn’t scale
  - We’ve had 20+ years of problems + solutions
  - Probably can’t handle another 20 more years

- Complexity compounds:
  - Feature * Layer interaction = more headaches

- Complexity costs $$$
  - Longer to qualify/deploy new features
  - Longer to debug problems
How Do We Reduce Complexity?

• Find better abstractions
  – Make individual config changes less complex
  – Stop solving the same problems at different layers
  – Extract commonality among similar solutions

• Reduce # of management “touch points”
  – Make config changes in fewer places
  – You pay $$$ for multi-chassis systems for a reason
  – Decouple control from forwarding
What Can We Abstract?

Exact Same Process for:

- BGP
- OSPF/ISIS
- L2 MAC Learning
- 128.8.0.0/16
WHAT IS OPENFLOW?
OpenFlow is an Abstraction and API

Route Engine

Line Card
Line Card
Line Card

OpenFlow Controller

OpenFlow Protocol
Over the Network

OpenFlow Datapath
Flow Table
OpenFlow in Practice

OpenFlow Protocol on SSL

Server

OpenFlow Controller

Linux

Management Network

Line Card

Line Card

Line Card

Line Card

OF Agent
OpenFlow in Practice

- Controller is independent from datapath
  - E.g., on an external server, blade, etc.
- Datapath = existing box + OpenFlow firmware
  - Requires vendor to ship OpenFlow firmware
- Communication over network to datapaths
  - Use TLS/SSL for mutual authentication
  - Out-of-band management network simpler
  - In-band schemes exist
- Open Standard
  - Could write your own controller! (IF inclined...)
  - Growing open source controller ecosystem
OpenFlow API Highlights

1. Punt packets up to controller
2. Send packets down to datapath
3. Add/Del/Mod forwarding entries in datapath
   - Capabilities of forwarding table next slide
4. Query stats
   1. Interface counters
   2. Flow counters
   3. Forwarding table usage
## Flow Table Abstraction

<table>
<thead>
<tr>
<th>Priority</th>
<th>Match</th>
<th>Action List</th>
</tr>
</thead>
<tbody>
<tr>
<td>500</td>
<td>TCP.dst=22</td>
<td>TTL--, Fwd:port 3</td>
</tr>
<tr>
<td>200</td>
<td>IP.dst=128.8/16</td>
<td>Queue: 4</td>
</tr>
<tr>
<td>100</td>
<td>*</td>
<td>DROP</td>
</tr>
</tbody>
</table>

**OpenFlow Datapath**

**Flow Table**
Flow Table Abstraction

• Simplifies cross-layer and feature interaction
  – Switching: match L2.dst, forward out port
  – Routing: match L3.dst, dec TTL, forward port
  – NAC: match ACL, DROP

• Multiple tables for more complex features
  – VRF, PseudoWire, Policy routing

• Match on most packet fields: L1, L2, L3, L4

• Lots of action types: Vlan, Mpls, IP, QoS, etc.
Decouple Control from Forwarding

- OpenFlow permits fewer controllers than datapaths
- Reduce number of management touchpoints
- Mapping from datapaths to controllers is a crucial network design question

OpenFlow does not imply centralized control!

Allows load balancing and failover
New Network Design Questions

• Traditional networks assume one-to-one forwarding to control plane mapping
  – Is this right for everyone?
• How many controllers do I need?
  – Balance between touchpoints and control traffic load
  – How many backup controllers? Hierarchy?
• Where do I put controllers?
  – Controllers per: POP, region, continent?
• Similar questions to BGP Route Reflectors
• Likely no “one-size-fits-all” solution
Protocol Between Controllers?

• By design, not specified by OpenFlow
  – Controller is software: evolve independently
  – Likely no “one-size-fits-all” solution

• OpenFlow: building block to a larger solution

• Depends on each network’s requirements
  – Failure recovery time
  – Management network
  – Number of forwarding nodes
IMHO, too early to standardize
OpenFlow is just the same as XXX!

• ‘XXX’ = LISP, MPLS-TE, policy routing, etc.
• Broad answer:
  – OpenFlow is a very-low level abstraction/API
  – Could probably implement XXX using OpenFlow
  – Could not implement OpenFlow using XXX
• If XXX meets my needs, why use OpenFlow?
  – More holistic network view
  – Reduced complexity from feature interaction
More Information

- [www.openflow.org](http://www.openflow.org)
  - Public specifications and white papers
  - On-line tutorials
- [http://openflow.stanford.edu](http://openflow.stanford.edu)
  - FlowVisor: OpenFlow hypervisor tool (mine :-)
  - Beacon: Open Source Java-based Controller
  - Mininet: OpenFlow network emulator in a box
- [http://noxrepo.org](http://noxrepo.org)
  - Open Source C++/Python Controller
OPENFLOW USE CASES
Use Cases Outline

• OpenFlow is an enabler – the sky is the limit
• To spark your imagination, this talk:
  – Cherry-picked use cases for the NANOG crowd
  – Emphasis on service provider networks
  – Demonstrations from Stanford University
• Online:
  – Lots more use cases, demos, and videos
  – http://openflow.org/videos
Virtualized Control Plane

Each use case/demo presented here runs in an isolated slice of Stanford’s production network.

FlowVisor slices OpenFlow networks, creating multiple isolated and programmable logical networks on the same physical topology.
Moved a VM from Stanford to Japan without changing its IP.

VM hosted a video game server with active network connections.
Mobility: n-casting with OpenFlow

- Demonstrate what flexibility of routing enables in mobile networks
- Show how technology agnostic handover can be easily achieved
- Customized network services for applications, devices and technologies
- Simplify control and services
- Unified control for wireline and wireless networking equipments

- Demonstration: n-casting
  - Reroute flows between WiFi and WiMAX without additional logic
  - n-casting provided over for video streaming where application handles duplication well
  - coded in 227 lines of C/C++
Reducing Energy in Data Center Networks

- Shuts off links and switches to reduce data center power
- Choice of optimizers to balance power, fault tolerance, and BW
- OpenFlow provides network routes and port statistics

The demo:
- Hardware-based 16-node Fat Tree
- Your choice of traffic pattern, bandwidth, optimization strategy
- Graph shows live power and latency variation

demo credits: Brandon Heller, Srini Seetharaman, Yiannis Yiakoumis, David Underhill
STANDARDIZATION
Open Networking Foundation

- ONF now maintains OpenFlow
  - Plus related protocols
  - Stanford not setup to be a standards body
- Composed of “Promoters” and “Adopters”
- Cross-license all IPR, royalty free
- Non-profit industry consortium [501(c)(6)]
- Founded March 22, 2011
  - NY Times + other press releases
ONF Board Composition

8 Board members/6 “promoter” member companies

- Urs Hölzle (Sr. VP, Engineering, Google), chairman, president
- Jonathan Heiliger (VP, Technical Operations, Facebook), secretary
- Adam Bechtel (VP, Infrastructure Group, Yahoo)
- Stuart Elby (VP, Network Architecture, Verizon)
- Arne Josefsberg (GM, Windows Azure Infrastructure, Microsoft)
- Bruno Orth (VP, Strategy and Architecture, Deutsche Telekom)
- Nick McKeown (Professor, EE and CS, Stanford)
- Scott Shenker (Professor, EECS, UC Berkeley and ICSI)
36 “Adopter” Member Companies

- Big Switch Networks
- Broadcom
- Brocade
- Ciena
- Cisco
- Citrix
- Comcast
- CompTIA
- Dell
  - IP Infusion
  - Ixia
  - Juniper Networks
  - Marvell
  - Mellanox
  - Metaswitch Networks
  - Midokura
  - NEC
  - Netgear
- Ericsson
- Extreme Networks
- Force10 Networks
- Fujitsu
- HP
- Huawei
- IBM
- Infoblox
- Intel
- Netronome
- Nicira Networks
- Nokia Siemens Networks
- NTT
- Plexxi Inc.
- Pronto Systems
- Riverbed Technology
- Vello Systems
- VMware
OpenFlow Interop-fest at Interop

- Interop: 13,000 attendees, 350 vendors
- 30’ x 40’ OpenFlow “Interop Labs” booth
- 14 vendors with OpenFlow datapaths
  - Various stages of prototype
  - Few commercial products
- Inter-operated under a single FlowVisor
  - … for the most part, eventually
<table>
<thead>
<tr>
<th>Rack 1 (controller):</th>
<th>Rack 2 (switches):</th>
</tr>
</thead>
<tbody>
<tr>
<td>Big Switch Controller</td>
<td>Brocade</td>
</tr>
<tr>
<td></td>
<td>IBM (Blade)</td>
</tr>
<tr>
<td></td>
<td>Dell</td>
</tr>
<tr>
<td></td>
<td>Quanta (Pronto)</td>
</tr>
<tr>
<td></td>
<td>Broadcom (Reference)</td>
</tr>
<tr>
<td></td>
<td>Netgear</td>
</tr>
<tr>
<td></td>
<td>Extreme</td>
</tr>
<tr>
<td></td>
<td>HP Procurve</td>
</tr>
<tr>
<td></td>
<td>Juniper (MX240,480)</td>
</tr>
<tr>
<td></td>
<td>NECC</td>
</tr>
<tr>
<td></td>
<td>Fulcrum (Reference)</td>
</tr>
<tr>
<td></td>
<td>NEC</td>
</tr>
<tr>
<td>Rack 3/4 (switches):</td>
<td>Upstairs:</td>
</tr>
<tr>
<td></td>
<td>Citrix (OVS)</td>
</tr>
<tr>
<td></td>
<td>Fulcrum (Reference)</td>
</tr>
<tr>
<td></td>
<td>Marvell (Reference)</td>
</tr>
</tbody>
</table>
OpenFlow Demos @ Interop

- NEC (Programmable Flow Demo)
  - Winner of “The Best of Interop 2011”
  - Category: infrastructure
- HP: Per-flow QoS demo
- Pronto: OpenFlow-enabled switches:
  - 48 X10GE, 48X10GE+4X40GE, 16X40GE 1U models
- Brocade - Service provider group
- Big Switch Networks: big virtual switch controller demo
- Juniper Networks: bandwidth calendar application
- ... plus standard Stanford demos
OPENFLOW DEPLOYMENTS
**Deployment Overview**

- Most *public* deployments are R&E Networks
  - OpenFlow did start in the R&E community
- Many interesting private deployments
  - Details are scarce and non-public
  - Best bet is to talk to ONF member operators
70+ World-wide Public Deployments
NSF GENI: 9+ Sites

Washington

Stanford

Wisconsin

Indiana

Internet2

NLR

Princeton

Rutgers

Clemson

GATech
Planned: 34 POPs in Internet2
OFELIA - Aim and Partners.

Federation of five islands

- 5 OpenFlow-enabled islands at academic institutions:
  - Berlin (TUB) – partial replacement of existing campus network with OF-switches
  - Ghent (IBBT) – central hub, large-scale emulation wall
  - Zürich (ETH) – connection to OneLab and GpENI
  - Barcelona (i2CAT) – experience with facility projects (IaaS, FEDERICA)
  - Essex (UEssex) – national hub for UK optical community; L2 (Extreme) switches, FPGA testbed

- NEC provides homogeneous L2 hardware platform (OF-enabled Ethernet switches)
- ADVA as major vendor of optical access and data center equipment
- Different external vendors (HP, Extreme, Juniper)

Partners with complementary technological strengths and user groups from five countries with strong research communities in networking.

<table>
<thead>
<tr>
<th>partner</th>
<th>L2</th>
<th>L1/optics</th>
<th>L3</th>
<th>Wireless</th>
<th>emulation</th>
<th>Control SW</th>
<th>processing</th>
<th>US connections</th>
<th>MIM source</th>
</tr>
</thead>
<tbody>
<tr>
<td>iBBT</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TUB</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>I2cat</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>UEssex</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ETH</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Conclusion

- Networking needs better abstractions to reduce complexity
- OpenFlow is an abstraction and API
  - Time will tell if it is the right answer
  - ... but is probably asking the right questions
- Lots of use cases for operator community
- It's on its way to wide-spread adoption
  - Newly formed ONF
  - More deployments than I can count

http://www.openflow.org

Thanks you!