When Protocols Collide...

Stephen Stuart, Google
ONS 2011-10-19

Google Confidential and Proprietary

"Why don't more people..."

Google

More people would "try out" SDN in general, and Openflow specifically, if they could connect an SDN-enabled network to their existing network.

- Q: Why don't more people use this stuff?
  A: It doesn't speak the standard protocols.
  (actual Q&A from 2009)

Led to research project for Google, other like-minded folk:

- What happens when SDN and "standard protocols" collide?
- Goal: build an open-source LSR with SDN inside

NB: this is not a talk about Google production anything

Google Confidential and Proprietary

What does "collide" mean?

Google

Current SDN implementations don't speak the standard protocols:

- IP and Ethernet are nice, but...
- ...need IPv6, MPLS, ISIS with TE, RSVP-TE, BGP
- Not that interested in OSPF, except that it's free

Background - multiple ways to collide:

- Really high policy boundary: BGP
- Really low policy boundary: ISIS, RSVP-TE

How to get there from here:

- Incremental deployment vs. "all in"

Google Confidential and Proprietary

What does SDN provide?

Google™

Manage ephemeral state ephemerally:

- Configuration vs. programming
- Lots of state doesn't need to survive a reboot

Centralize decision-making based on global state:

- Existing mechanisms make distributed decisions based on imperfect local knowledge of global state
- Centralized decision-making can avoid inefficiencies

Add new services to existing networks:

- Within the context of the existing "routed network"
- Adjacent to the "routed network"

Google Confidential and Proprietary

First challenge, pass packets

Google

Building an open-source LSR with SDN inside

Last year, Open-source LSR (NANOG 50):

- Linux/MPLS, Quagga/LDP
- Custom glue for kernel-NetFPGA LFIB synchronization
- NetFPGA hardware forwarding

This year, Project W (here, now):

- Quagga IP stack, IP/Ethernet and BGP/OSPF
- RouteFlow glue, NOX/OVS-flavored SDN
- Pronto 3290 hardware forwarding

Google Confidential and Proprietary

Next? Up, down, sideways

Google

Up:

- ISIS with TE extensions
- RSVP-TE and LDP with userland LFIB
- LFIB without controller kernel MPLS support

Down:

- LFIB synced to hardware via OF 1.x
- Track multi-table support in OF x.x
- Distributed router (1 protocol engine, N switches)

Sideways:

- Expose controller to provision services "next to" the routed network

Google Confidential and Proprietary

More...

Google

OA&M:

- Counter export (IPFIX?), TE extensions to IGPs
- track progress in configuration,
- MPLS auto-bandwidth

Test, release, support:

- System testing: IGP/BGP/ssh flow expiry, IP fragment handling, &c.
- Regression test, release engineering, support at OSRF

Build real network

- Includes building a real ecosystem
- RFC2702 is still an interesting read, s/traffic trunk/flow

Google Confidential and Proprietary