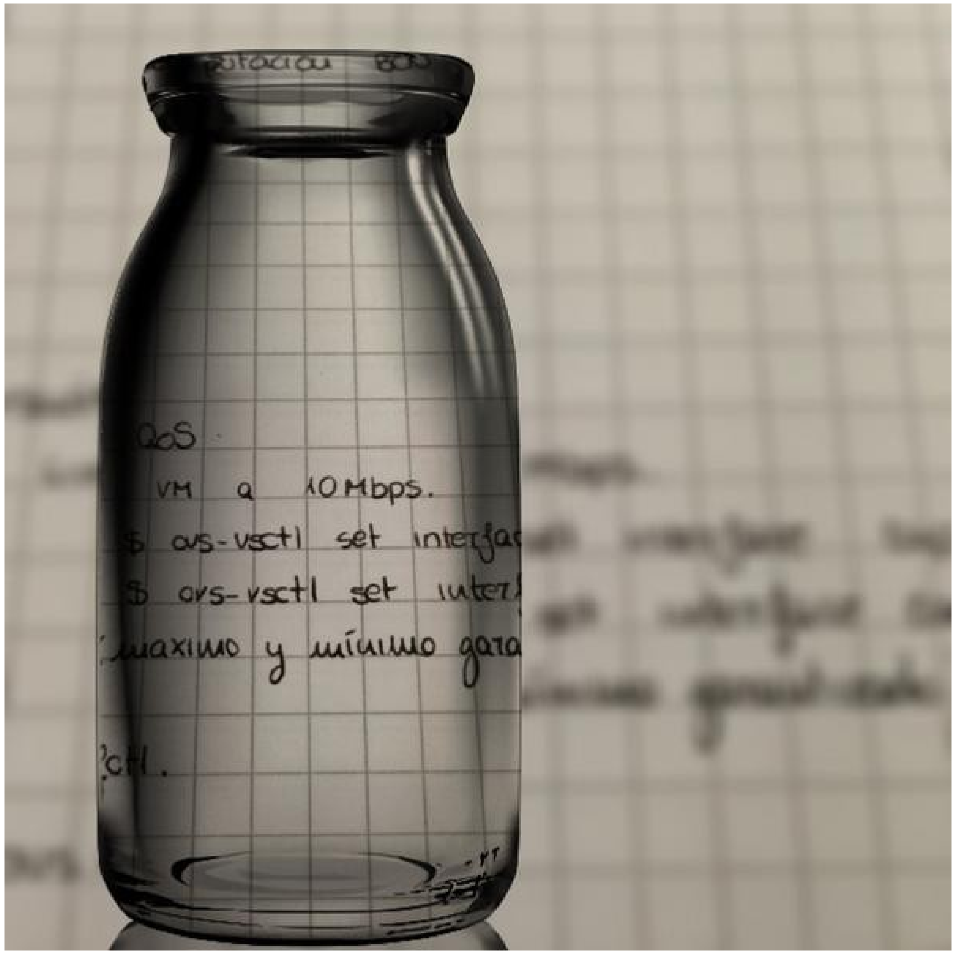
Open vSwitch: Past, Present, and Future



Ben Pfaff
VMware NSBU

Preview

- What is Open vSwitch?
- Past
- Present **(Demo!)**
- Future
- **Questions!**

What is Open vSwitch?

Semi-official description:

> Open vSwitch is a production quality, multilayer virtual switch licensed under the open source Apache 2.0 license. It is designed to enable massive network automation through programmatic extension, while still supporting standard management interfaces and protocols (e.g. NetFlow, sFlow, SPAN, RSPAN, CLI, LACP, 802.1ag).

The exciting parts:

- Write a program to control your network.
- Fast! (and getting faster)
- Everywhere! (and getting everyerwhere)

Past: Ethane (2006-2007)

“...all complex functionality, including routing, naming, policy declaration and security checks are performed by a central controller (rather than in the switches as is done today). Each flow on the network must first get permission from the controller which verifies that the communicate is permissible by the network policy. If the controller allows a flow, it computes a route for the flow to take, and adds an entry for that flow in each of the switches along the path.”

Past: Vigilante (Repo #1)

- Aug 14, 2007: First commit, by Martin.
- Sep 12, 2007: Convert codebase to C++.
- Sep 19, 2007: First commit by Natasha.
- Sep 25, 2007: First commit by Justin.
- Sep 26, 2007: Commit with highest proportion of swearing in commit message: “Add crap needed by crappy UML crap.”
- Oct 10, 2007: NOX-like core of earliest controller framework.

Past: OpenFlow

(Early names: Firefly, Enso, Viros.)

- Nov 27, 2007: First mention of OpenFlow.
- Nov 30, 2007: OpenFlow consortium starts.
- March 4, 2008: New OpenFlow repo (repo #2).
- Dec 2, 2008: Last merge from Nicira repo to OpenFlow repo.

Past: vswitchd

- Dec 2, 2008: Defined vswitchd features (VLANs, bonding, mirroring, STP, NetFlow, ...). Brcompat.
- Jan 2, 2009: Feature complete.
- Jan 8, 2009: Last commit by Martin.
- Mar 10, 2009: Rewrote whole thing because I misunderstood intent.
- March 2009: Whirlwind trip to Citrix in Cambridge.

Past: Open vSwitch (Repo #3)

- May 20, 2009: First use of Open vSwitch name.
- Jul 8, 2009: Start of public repo.
- Aug 26, 2009: Open vSwitch mailing lists created.
- Nov 11, 2009: OVSDB implementation checked in (soon an RFC). OF-CONFIG didn't exist even as an outline before Nov 2011.
- Jan 21, 2010: OpenFlow 1.0 support.
- May 15, 2010: Open vSwitch 1.0 released.

  …
- Jul 23, 2012: VMware acquisition announced.

The Ghost of Open vSwitch Present

Contributing Organizations (based on emails)

Cisco
Citrix
Fujitsu
Google
HP
Intel
Nokia-Siemens
Oracle
VMware

ALT Linux
Debian
FreeBSD
Red Hat
Ubuntu
VA Linux

big switch networks
Nicira
Toroki

Clemson
Stanford
Wisconsin-Madison
Pisa

Open vSwitch Hall of Fame

All contributors with 10 or more commits in current or previous repo

4707	Ben Pfaff
827	Justin Pettit
615	Ethan Jackson
503	Jesse Gross
197	Simon Horman
88	Pravin B Shelar
53	Gurucharan Shetty
47	Ian Campbell
46	Andrew Evans
37	Ansis Atteka
35	Isaku Yamahata
31	Keith Amidon
27	Ed Maste
24	Mehak Mahajan
22	Jarno Rajahalme
15	Chris Wright
13	Kyle Mestery
12	Pravin Shelar
11	Joe Stringer
10	Arun Sharma
10	Sajjad Lateef
10	Thomas Goirand

Lessons Learned

- The best real packet classifiers are not found in papers.
- "I've rewritten the fucking datapath N times, I'm not going to let iteration N+1 stop me."
- Standardization does not fix warts.
- Reactive programming is seductive, but doesn't scale.
- In-band control is hard.
- Guessing about performance is risky.

Introduction to Programming Open vSwitch (Demo!)

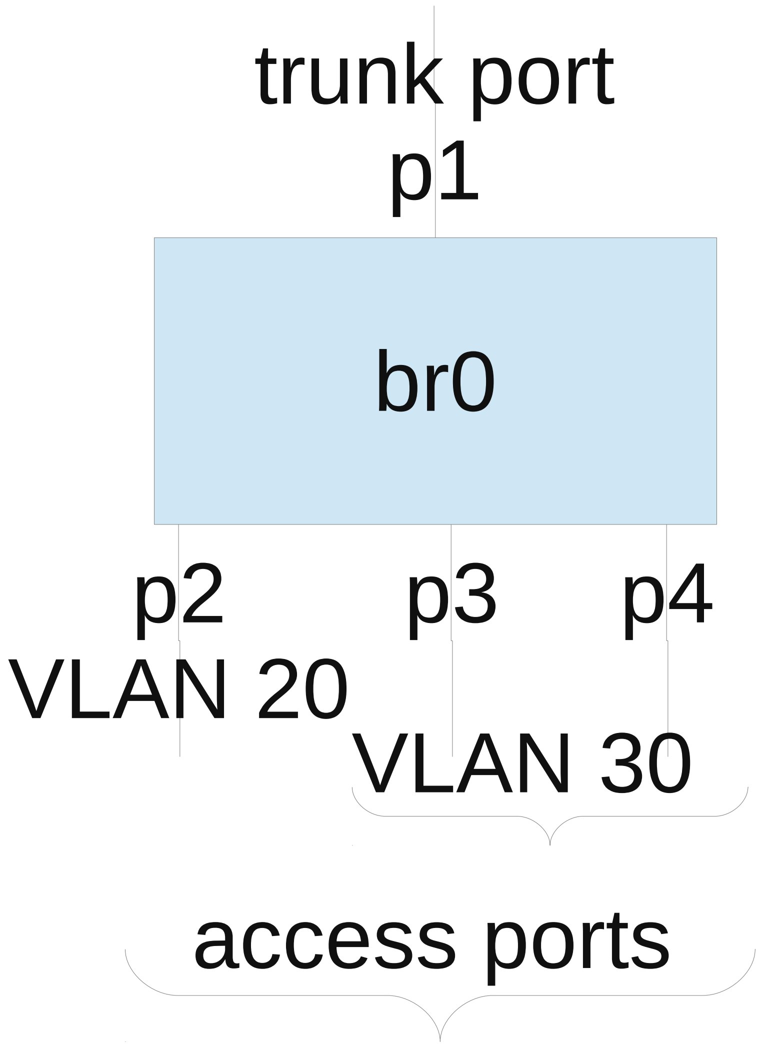
- Not another stupid networking demo...
- The curse of “normal”.
- Normal pipeline:

    0. Admission control (reserved multicast, …).
    1. Input VLAN processing
    2. Learn source MAC+VLAN on ingress port.
    3. Look up output port for dest MAC+VLAN.
    4. Output VLAN processing.

Demo Features

- resubmit action
  - Cartesian product
- learn action
- registers
- ofproto/trace

Demo: Test Case



Future: Predictable Stuff

- OpenFlow 1.1 and later
- MPLS
- ESX
- Hyper-V

Future: Performance

- Not a problem:
  - Small number of long flows
  - Large number of medium-length flows.
- Tuning solves some problems.
- Real problem:
  - Large number of short flows.

Performance Improvements

- Might help:
  - Patch port optimization.
  - Classification batching.
  - Threading?
- Unlikely to help:
  - Buffering packet data in kernel.
  - Faster user<->kernel interface.

Performance: Fairness

- Which flows get dropped?
  - Random is bad.
  - Per-tenant fairness is better.
- Per-port fairness.
- Per-destination fairness.

Performance: Solutions

Two independent directions (why?):

1. Every packet to userspace
   - DPDK, PF_RING, netmap.
2. Megaflows:
   - Use: clusters, fairness (+hashing).
   - History (why only now?).

Thanks

Nick McKeown, Scott Shenker, Martin Casado, Natasha Gude, Justin Pettit, Teemu Koponen, Dan Wendlandt, Peter Balland, Reid Price, Henrik Amren, Keith Amidon, David Erickson, Murphy McCauley, Jean Tourrilhes, Ian Campbell, Jesse Gross, Steve Mullaney, David Tsai, Sujatha Sumanth, Ethan Jackson, Hao Zheng, Andrew Lambeth, Rajiv Ramanathan, Rob Enns, Rob Hoes, Simon Horman, James Page, Joe Stringer, Kyle Mestery, Ed Maste, and many others I'm sure I'll be embarrassed to have pointed out that I've missed.

Questions?