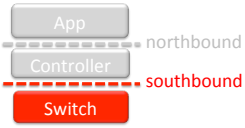
SDN Switches

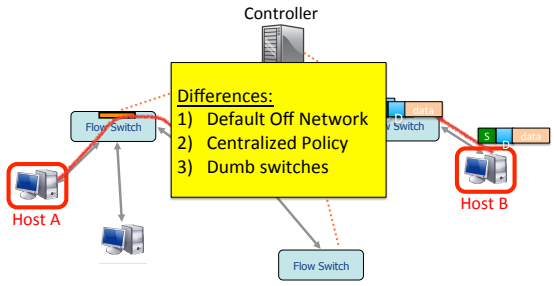


Session Outline

- Brief OpenFlow history
- The SDN Switch design space
- Performance considerations

Ethane, a precursor to OpenFlow

default-off, centralized policy, dumb simple switches



Source:
Ethane: Taking Control of the Enterprise, SIGCOMM 2007, Casado et al.

Ethane's dumb, simple switches might be more broadly useful…

OpenFlow

OpenFlow is a **pragmatic compromise**

+ Vendors don't need to expose implementation
+ Speed & scale of vendor hardware
+ Leverages ACL tables inside most switches today

- Least-common-denominator interface may prevent using all hardware features
- Limited table sizes (today)
- Switches not designed for this (today)
- New failure modes to understand

OpenFlow is **broadly implementable**

Hardware


Chassis:
Juniper MX
ProCurve 54xx


Top-of-Rack:
Netgear 7324
NEC IP8800


Optical:
Ciena CoreDirector


NetFPGA

Software


X86:
Open vSwitch
OpenFlow Reference


WiFi AP:
PC Engines


WiMAX BS:
NEC

this was 2010… more choices now

OpenFlow **defines three things:**

State:
What can software configure to match packets, and how is it represented?

Behavior:
Given a state, how can (and should) the switch forward or modify packets?

Control Interface:
How do I describe desired changes to the switch state?

OpenFlow defines three things:


1.0

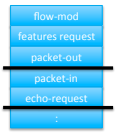
state

single flow table + port table

behavior

match-one / send-to-controller, expire soft flows, modify or forward packets

interface



single message queue w/optional barriers, vendor extensions, C-style structs, asynchronous errors

+ usable by machines: runs on most existing switch hardware
+ usable by people: simple interface to understand

Issues with OpenFlow 1.0

- Table space explosion

Best case: 100 + 100 + 100 = 300 entries
Worst case: 100 x 100 x 100 = **1 million entries!**

Issues with OpenFlow 1.0

- Table space explosion
- Limited match options

Supported:
MAC, VLAN, IP, L4 ports

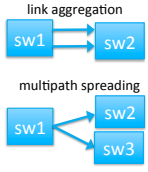
Missing:
IPv6
QinQ
MPLS
Ipv[X]
Optical circuits

Issues with OpenFlow 1.0

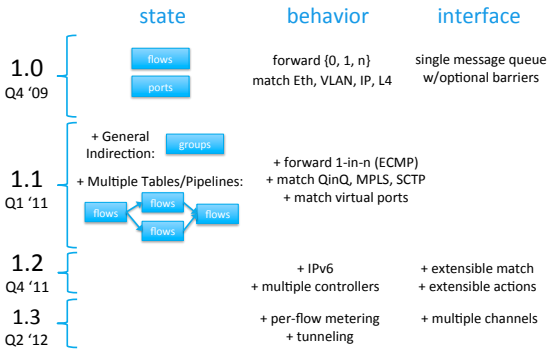
- Table space explosion
- Limited match options
- Limited forwarding options

Supported:
broadcast, multicast, drop

Missing:
packet spreading
forwarding to a virtual port
general byte modifications



OpenFlow has **evolved** towards production readiness.



SDN Switch design space is barely explored: three alternatives to OF1.0

state behavior interface

OF1.1+:
Flexible pipeline of multiple tables

SDN Switch design space is barely explored: three alternatives to OF1.0

state behavior interface

Hypothetical OF2.0:
match and modify arbitrary offsets

Pkt[IP Dst] == IP X : out-port Y, decrement TTL
Pkt[MAC Src] == MAC X : out-port Y, set to 00:11:…

Defined by RFCs

Bytes[14:15] == 0xaaaa : out-port Y, set to 0xbbbb
Bytes[14] == 0xaa : out-port Y, bytes[15] += 1

Defined by Network

SDN Switch design space is barely explored: three alternatives to OF1.0

state behavior interface

Hypothetical transactional interface

Database Transaction Example
- Debit $100 to Groceries Expense Account
- Credit $100 to Checking Account

Network Example
- Remove Port A from broadcast group
- Add Port B to broadcast group

Performance Considerations

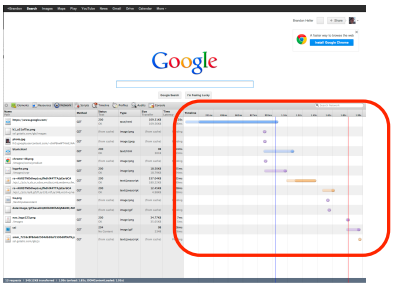
In this section, performance == control plane performance, NOT forwarding rate.

Performance depends primarily on 3 factors: usage, domain, and switch architecture

Key performance indicators

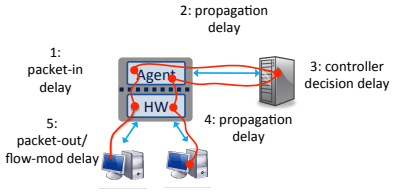
- Flow Setup Delay
- Flow Setup Rate
- Control Channel Bandwidth

Connecting to Google: 13 requests, many sequential



Performance depends on **usage: flow setup delays**

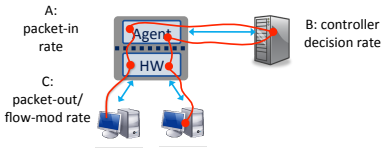
Reactive Network: 1, 2, 3, 4, 5



Proactive Network: none

Performance depends on **usage: flow setup rate**

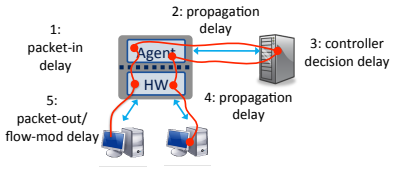
Reactive Network: A, B, C



Proactive Network: none

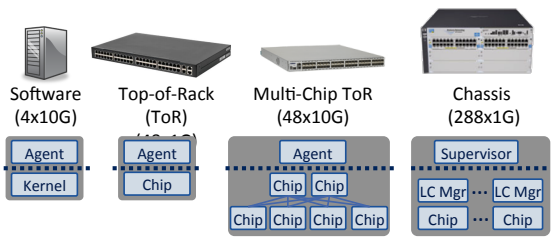
Performance depends on **domain**

Wide-area Network: 1, 2, 3, 4, 5
Enterprise Network: 1, 2, 3, 4, 5

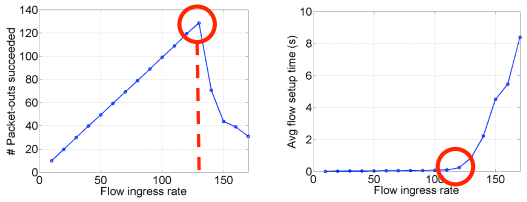


Performance depends on **switch architecture**

Increasing forwarding bandwidth
Decreasing control performance



First-generation OpenFlow switches
have surprisingly low message throughput



HW Switch: ~130 packet-ins/s ~100 flow-mods/s
SW Switch: 10K+/s
Controllers: 20k-13M responses/s

Sources: measurements by David Erickson and
http://yuba.stanford.edu/openflow/documents/openflow_deployment_journal_paper_aug2012.pdf

Hope is not lost.

- In practice, performance can be just fine
- Second-gen switches coming
- Software workarounds at the controller

My suggestions:

- Assume the switch is the bottleneck
- Benchmark vendor switches
- If tons of flows, use virtual switching, or re-consider reactive control

Main performance takeway:

Performance depends primarily on 3 factors: usage, domain, and switch architecture