OpenFlow Deployment

Vjeko Brajkovic
Arvind Krishnamurthy

University of Washington

Deployment Setting

- 6 floor CSE building
  - 8 subnets available on each floor
  - Aggregated at the basement and connected to a router
- Building has duplicate (mostly unused) network machine rooms in each floor
  - Next to the production wired network run by campus network
  - Vertically stacked across floors
  - Preexisting spare (unused) conduit for research network
    - spare cable ducts and wall sockets

Current Status

- Deployed OpenFlow for students in the networks lab
  - HP Procurve 6600 with OpenFlow 0.89
  - NOX, FlowVisor, SNAC, but no SFA
  - Separate VLANs: production traffic, experimental traffic, management traffic, and public non-OpenFlow
- Demonstrated the Plug-n-Serve system spanning UW and Stanford

Future Plan

- 2-4 Openflow switches on 2 to 3 floors
- Aggregated at the basement
  - Either transition into the campus production network
  - Or use 10GigE dark fiber link to the Pacific GigaPoP
- Opt-in deployment
  - Initially, graduate students and faculty in systems/networking (ten to twenty users by GEC-8)
  - Wireless access point in undergrad CS labs
  - Later expanded to the entire department (transition to being the production network)

Research Testbed

- Augment OpenFlow switches with Intel RouteBricks (software programmable router)
  - RouteBricks can be a stand-in for "middleboxes"
- Enable research on where functionality should be placed in the network:
  - What should be in the lightweight OF switch?
  - What should be in a middlebox?
  - What should be at the edge (end-hosts)?

Our Research Agenda
"End to the Middle"

- Can we eliminate middleboxes altogether?
  - caches, traffic shapers, firewalls, IDSs, NATs, VPNs, proxies, load balancers, and so on
- Can they be mostly implemented in commodity hardware at the edge?
- Built a research prototype where NAT, IDS, QoS functionality is at the edge
  - OpenFlow switches serve as network substrate
  - Trusted Boot stack leverages TPM support
  - RADIUS server support for validating TPM attestations

Why?

- Transparency: open source software on commodity hardware
- Cost: lower barrier for small networks
- Pervasiveness: no longer point solutions
- Scalability: e.g., if you want DPI, have it scale at the edge
- Reliability and failover: replicated state machines at the edge

Why now?

- Multicores on end-hosts: extra resources with performance isolation
- Virtual machines: protection and portability
- TPMs: attestations from commodity PCs

Example: NAT

- Calls to bind/listen invoke distributed coordination
- Agree (using paxos) on port leases
- Packets forwarding:
  - Use OpenFlow or
  - Deliver to any live host, then forwarded