NETWORK VIRTUALIZATION IN MULTI-TENANT DATACENTERS

Teemu Koponen

with

Keith Amidon, Peter Balland, Martín Casado, Anupam Chanda, Bryan Fulton, Igor Ganichev, Jesse Gross, Natasha Gude, Paul Ingram, Ethan Jackson, Andrew Lambeth, Romain Lenglet, Shih-Hao Li, Amar Padmanabhan, Justin Pettit, Ben Pfaff, Rajiv Ramanathan, Scott Shenker, Alan Shieh, Jeremy Stribling, Pankaj Thakkar, Dan Wendlandt, Alexander Yip, and Ronghua Zhang
Plenty of primitives but **no** network virtualization per se.
MULTI-TENANT DATACENTERS

Result with the aforementioned primitives:

- Slow provisioning
- Mobility is limited
- Limited VM placement
- Hardware dependent
- Operationally intensive
- …
NETWORK HYPERVERSOR

Server Hypervisor

Decoupled

Network Hypervisor

Standard x86

VMs

L2, L3, L4-L7 Services

Logical Networks

Workload

Standard IP connectivity
AGENDA

• Overall design of NVP network hypervisor.
• Design challenges.
• Hard lessons learnt.
• What’s next in network virtualization?
WHAT IS A NETWORK HYPERVISOR?

Packet Abstraction + Control Abstraction = Network Hypervisor
WHAT ARE THE ABSTRACTIONS?

Packet abstraction
• Compliance with standard TCP/IP stack is a necessity:
  • L2, L3 semantics (unicast, ARP, …)

Control abstraction
• Networking has no single high level control interface.
  • There's a low-level one though!

Tenant’s Control Plane

Logical Datapath

Packet In

ACL

L2

L3

ACL

Packet Out
GENERALITY OF DATAPATH

Switch CP

Datapath
ACL  L2  ACL

Router CP

Datapath
ACL  L2  L3  ACL

Switch CP

Datapath
ACL  L2  ACL

One logical switch

2-tier logical topology

Arbitrary logical topology

Faithful reproduction of physical network service model.
WHERE TO IMPLEMENT?

- Independence from physical hardware.
- Programmatic control.
- Operational model of compute virtualization.

No extra x86 hops: just the source and destination hypervisor!
INSIDE THE VIRTUAL SWITCH

Logical Topology

First-hop vSwitch

- Identify logical ingress port
- Execute 1st logical datapath
- Determine the next logical datapath
- Determine the next logical datapath
- Send to tunnel
1. Controllers learn the location of VMs.

2. Controllers proactively compute & push all forwarding state required to connect each VM.

\[
\text{Forwarding State} = F(\text{configuration, VM locations})
\]

Repeat above as logical configuration or physical configuration (VM placement) changes.

**Challenge**: How to compute \(O(N^2)\) volume of low-level OpenFlow and OVSDB state, when inputs change all the time.
STATE COMPUTATION

Forwarding State = F(configuration, VM locations)

1. How to Scale Computation
   - **Incremental** computation and pushing for quick updates.

2. How to Guarantee Correctness
   - Avoid all handwritten finite state machines, **machine generated** instead.

   ➡️ Datalog based declarative language to program F.

   ➡️ Shard the computation across controller cluster.

Declarative RT
CONCLUSION: WHAT’S NEXT

Without Network Virtualization

- Workload may run on a topology where addresses provide little information.
- For instance, firewall rules defined over exact /32 addresses!

With Network Virtualization

- New “out-of-band” header fields without breaking legacy TCP/IP stacks.
- **Huge** implications to enforcing security policies: groups, users in packet…
THANK YOU! QUESTIONS?