Flexible Building Blocks for Software Defined Network Function Virtualization (Tenant-Programmable Virtual Networks)

Aryan TaheriMonfared
Chunming Rong

Department of Electrical Engineering and Computer Science
University of Stavanger

QShine, 2014
Outline

1. Introduction
   - Problem? & Solution
   - IaaS Cloud Networking
   - Software Defined Networking & OpenFlow
   - Network Function Virtualization

2. Solution
   - Tenant Controlled Virtual Networks

3. Evaluation
   - Overview
   - Reachability Time
   - Throughput

4. Summary
Outline

1. Introduction
   - Problem? & Solution
   - IaaS Cloud Networking
   - Software Defined Networking & OpenFlow
   - Network Function Virtualization

2. Solution
   - Tenant Controlled Virtual Networks

3. Evaluation
   - Overview
   - Reachability Time
   - Throughput

4. Summary
What is wrong with Virtual Networks (VN) in IaaS?

- Not flexible
- Lack of control
- Limited functionality
- Middle Box placement
- Proprietary APIs
Contributions

- New approach for network virtualization
- Taking advantage of SDN
- Dedicated networking components for each tenant
- Direct & Full control over provisioned VNs
- Standard/Open protocols (OpenFlow, OVSDB)
Outline

1. Introduction
   - Problem? & Solution
   - IaaS Cloud Networking
     - Software Defined Networking & OpenFlow
     - Network Function Virtualization

2. Solution
   - Tenant Controlled Virtual Networks

3. Evaluation
   - Overview
   - Reachability Time
   - Throughput

4. Summary
Virtual Networks in Cloud

- VNs connect VMs and higher level services
- VNs are overlays on top of providers’ infrastructure
- Providers establish and maintain VNs

Challenges

- VNs are not as flexible as VMs
- Functionality is limited by providers’ offering
- Services have limited knowledge/control over the network
- e.g. Basic CIDR, QoS configurations
Outline

1. Introduction
   - Problem? & Solution
   - IaaS Cloud Networking
   - **Software Defined Networking & OpenFlow**
   - Network Function Virtualization

2. Solution
   - Tenant Controlled Virtual Networks

3. Evaluation
   - Overview
   - Reachability Time
   - Throughput

4. Summary
Software Defined Networking & OpenFlow

## Software Defined Networking

**SDN**

- New methods for network management and configuration
- Abstractions between different layers of networking mechanisms: *distributed state, specification, forwarding*
Software Defined Networking & OpenFlow

OpenFlow

- An approach for forwarding abstraction
- Separate forwarding plane from control plane physically
- One control plane can manage multiple forwarding planes

OpenFlow Spec

- OF switch has a set of flow tables, and a group table
- OF controller add/update/delete flow entries
- Flow entry has a matching pattern, ordered actions, priority, counters
Outline

1 Introduction
   - Problem? & Solution
   - IaaS Cloud Networking
   - Software Defined Networking & OpenFlow
   - Network Function Virtualization

2 Solution
   - Tenant Controlled Virtual Networks

3 Evaluation
   - Overview
   - Reachability Time
   - Throughput

4 Summary
Network Function Virtualization

NFV

- Network architecture
- Utilizes virtualization for delivering network functions
- Functions realized in software
- Deployed on standard hardware
- Decoupled from proprietary hardware
- Evolve beyond HW lifecycles
Outline

1. Introduction
   - Problem? & Solution
   - IaaS Cloud Networking
   - Software Defined Networking & OpenFlow
   - Network Function Virtualization

2. Solution
   - Tenant Controlled Virtual Networks

3. Evaluation
   - Overview
   - Reachability Time
   - Throughput

4. Summary
Tenant Controlled Virtual Networks

Overview

- A dedicated set of virtual network devices for each tenant
- Virtual devices are isolated
- Directly controlled and programmed by tenant’s controller
- No redirection layer (e.g. Provider’s controller)
- Decoupled tenants’ controllers from provider’s one (i.e. independent failure domain)
Components

- A pair of dedicated bridges for each tenant per host
- A Tunnel End-Point interface for each tenant per host
- Isolated transport network per tenant

Connectivity

- Tenant’s Local VMs: virtual ToR bridge
- Tenant’s Remote VMs: TEP bridge

Tunnels

- A tenant has a dedicated set of tunnels
- Established on-demand
- Between nodes which are hosting tenant’s VMs
Flow Programming

- Proactive flow programming

- Four types of flow rules:
  - Local Ingress, Local Egress, Local Flood, Remote Egress

- $O(N)$ flow entries in each OVS instance, where $N=$ total number of instances on a host
Tenant Controlled Virtual Networks

Architecture

```
+------------+------------+------------+
|            |            |            |
| brint-xy   | brtun-xy   | br-int     |
|            |            |            |
| VM1        | VM2        | VM3        |
| GRE Tunnels|            |            |
| brint-ab   | brtun-ab   | br-tun     |
|            |            |            |
| VM4        | VM5        | VM6        |
| GRE Tunnels|            |            |
```

eth1

`{ex-xy! ex-ab!}`
Tenant Controlled Virtual Networks

Networks

Compute Node i

br-int

br-tun

eth1

Tenant Transport Network

br-tun

eth1

br-int

Compute Node j

brtun-xy

ex-xy

VM1 VM2 VM3

Tenant Tunnels

Tenant Virtual Network

GRE

Physical Transport Network

Tenant Tunnels

GRE

VM1 VM2 VM3
Packet Flow

**Compute Node i**
- **br-int**
- **br-tun**
- **brtun-xy**
- **brtun-xy**
- **brtun-xy**
- **GRE Tunnels**
- VM1, VM2, VM3

**Compute Node j**
- **br-int**
- **br-tun**
- **brtun-xy**
- **brtun-xy**
- **brtun-xy**
- **GRE Tunnels**
- VM5, VM6, VM7
Tenant Controlled Virtual Networks

Tenant’s Controller
Advantages/Disadvantages

**Advantages**

- Direct access to management and control planes
- Dedicated set of virtual components (e.g. switches, tunnels, interfaces)
- Facilitates virtual network functions (e.g. MB functions)
- Standard/Open protocols
- Layer 2 isolation
- Unified management of {on,off}-premises resources
- Decoupled VN topology and architecture from underlay
- Transparent modification of physical/virtual infrastructure
Advantages/Disadvantages

Disadvantages

- Performance hit
- Start-up time overhead
- Complex implementation
Outline

1. Introduction
   - Problem? & Solution
   - IaaS Cloud Networking
   - Software Defined Networking & OpenFlow
   - Network Function Virtualization

2. Solution
   - Tenant Controlled Virtual Networks

3. Evaluation
   - Overview
   - Reachability Time
   - Throughput

4. Summary
Evaluation

- Must scale in a large infrastructure
- Metrics: reachability time, available bw
- Carried out for different number of VMs, VNs
  ⇒ variety of VMs distribution over hosts, VNs
- Traditional (CNB) vs. tenant-controlled VNs (DNB)

# scenarios: 2 (DNB, CNB)
# runs: 5
# experiments: # tenants’ network (1, 2, 5, 10, 20, 40, 80)
# subexperiments: # VMs (1, 2, 5, 10, 20, 40, 80, 120)
Average run time: ~ 25h
Outline

1 Introduction
   - Problem? & Solution
   - IaaS Cloud Networking
   - Software Defined Networking & OpenFlow
   - Network Function Virtualization

2 Solution
   - Tenant Controlled Virtual Networks

3 Evaluation
   - Overview
   - Reachability Time
   - Throughput

4 Summary
Reachability Time

$t_{rq}$: Instance spawn up request time

$t_{ier}$: First echo reply time

$t_r$: Instance reachability time (start-up time)

$t_r = t_{ier} - t_{rq}$

Total overhead of not-networking processes are uniformly reflected
Average Reachability Time for DNB

The graph shows the average reachability time (in ms) for different network sizes. The x-axis represents the number of instances, while the y-axis shows the reachability time. The lines represent different network sizes:

- 1 Network average-rt
- 2 Networks average-rt
- 5 Networks average-rt
- 10 Networks average-rt
- 20 Networks average-rt
- 40 Networks average-rt
- 80 Networks average-rt

As the number of instances increases, the reachability time also increases significantly.
Average Reachability Time Comparison (DNB/CNB)
Observations

- CNB performs slightly better than DNB
- DNB overhead is less significant when a large number of instances is requested (e.g. 80)
- First $|cns|$ instances require bridge/tunnel establishment
- Last $n - |cns|$ instances have similar start-up time
- $n$: Request instances
- $|cns|$: Compute node cluster size
Outline

1 Introduction
   - Problem? & Solution
   - IaaS Cloud Networking
   - Software Defined Networking & OpenFlow
   - Network Function Virtualization

2 Solution
   - Tenant Controlled Virtual Networks

3 Evaluation
   - Overview
   - Reachability Time
   - Throughput

4 Summary
Throughput

- TCP and UDP performance
- Physical network controller ↔ VMs
- Each direction individually
Average Bidirectional TCP Bandwidth for DNB

Average Bidirectional TCP Bandwidth between Controller and VMs (DNB)

Throughput

- 1 Network total-avg-bw
- 2 Networks total-avg-bw
- 5 Networks total-avg-bw
- 10 Networks total-avg-bw
- 20 Networks total-avg-bw
- 40 Networks total-avg-bw
- 80 Networks total-avg-bw

Number of Instances

Bandwidth (bps)

- 6.0E+8
- 6.5E+8
- 7.0E+8
- 7.5E+8
- 8.0E+8
- 8.5E+8

Introduction

Solution

Evaluation

Summary
Physical $\leftrightarrow$ VM TCP Bandwidth for DNB (breakdown)
Throughput

Bidirectional TCP Bandwidth Comparison (DNB/CNB)
Throughput

Observations

- TCP bandwidth decreases by an increase in the number of instances
- No significant change in bandwidth by increasing number of networks
- Virtual to physical bandwidth is higher than the opposite one
  → VM’s processing power, Rx are more compute intensive
- DNB vs CNB: ~ 12% degraded performance
Summary

- New architecture for DC network virtualization
- Full access to provisioned networking resources
- Standard/Open protocols
- Facilitates virtual network function development
- Early results are promising
Thank you!

- Code: https://github.com/aryantaheri/ovsdb/wiki