Network Defragmentation in Virtualized Data Centers

Oliver Michel*, Eric Keller*, Fernando Ramos^  
*University of Colorado Boulder, ^University of Lisbon
Virtualized Data Centers

- Services / Applications
- Virtual Machine
- Hypervisor
- Hardware
  - Servers, Networking Equipment, Cabling, etc.
Virtualized Data Centers

- Cornerstone of **cloud computing**
- Coexistence of tenants on **shared** infrastructure
- **Scaling** of resources based on demand

**Elasticity is one of its key drivers**
New requirement: network virtualization

- **Different** cloud applications require **different** topologies and **different** network guarantees

A reality today: VMware NVP [NSDI’14], Microsoft AccelNet [NSDI’18], Google Andromeda [NSDI’18]
Limitations of existing approaches

- Core algorithmic challenge: *Virtual Network Embedding (VNE)*
- Map the virtual network requests onto the substrate infrastructure

Existing VNE approaches lack a fundamental requirement: **elasticity**
Contribution #1: new scaling primitives to VNE

Expand VN / Contract VN
Problem: network fragmentation

Virtual Network Lifecycle

Embed VN → Expand VN → Contract VN → Remove VN
Problem: network fragmentation

- Fragmentation introduces several **problems**
  - Longer path lengths, higher latencies
  - Harder to map new VNs, lower acceptance ratios

```
expand()
```
Implementation

- Extended an existing VNE algorithm [Ballani11] with expand and contract primitives
- Cloud simulator
  - Leaf-Spine physical topology
  - Different virtual topologies

Virtual Network Topologies: VC, VOC, 3T
Costs of fragmentation

- **Simulation**: leaf-spine topology; 28 switches and 384 servers; adding, deleting, expanding, or contracting virtual networks 1000 times (averaged over 10 runs; randomized workloads)

- **Longer path lengths** -> higher latencies and cost, poorer application performance, prolonged job completion times

- **Lower acceptance ratios** -> reduced revenue
Contribution #2: new management primitive

Expand VN / Contract VN

+ Network defragmentation
Network Defragmentation

- Optimize VNE algorithms to avoid fragmentation?
  - Hard: no prior knowledge of upcoming request

We explore a different approach: explicit defragmentation by periodically running a network migration algorithm (e.g., LIME [SOCC14])

![Diagram showing defragmentation process](chart.png)
Network defragmentation heuristic

- (Naive) remap of all virtual networks
  1. Unmap all VNs
  2. Sort by VN size (biggest to smallest)
  3. Re-embed each VN in order
Evaluation: mean path length stretch
Evaluation: fraction of paths by path length

Lower path lengths reduce latencies and cost, improve application performance, and by improving efficiency increase provider profit.
Summary

- Future work
  - better defragmentation heuristics that minimize migration cost
  - integration with network migration
THANKS.
QUESTIONS?

Oliver Michel
oliver.michel@colorado.edu

Fernando Ramos
fvramos@ciencias.ulisboa.pt