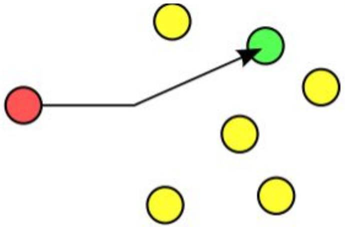
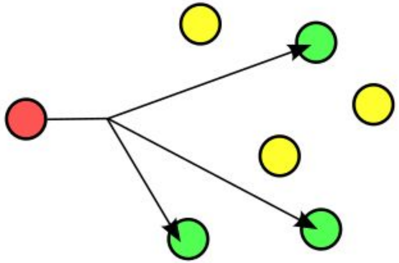
Data Center Architecture

IP Addressing Method

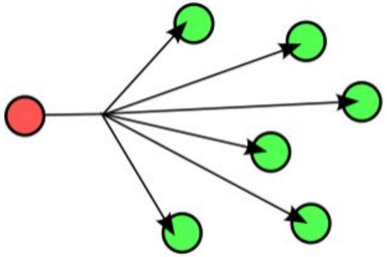
Unicast



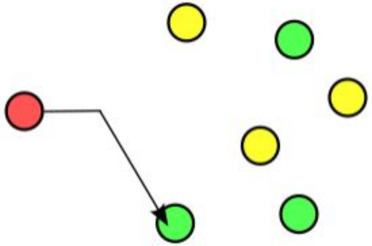
Multicast



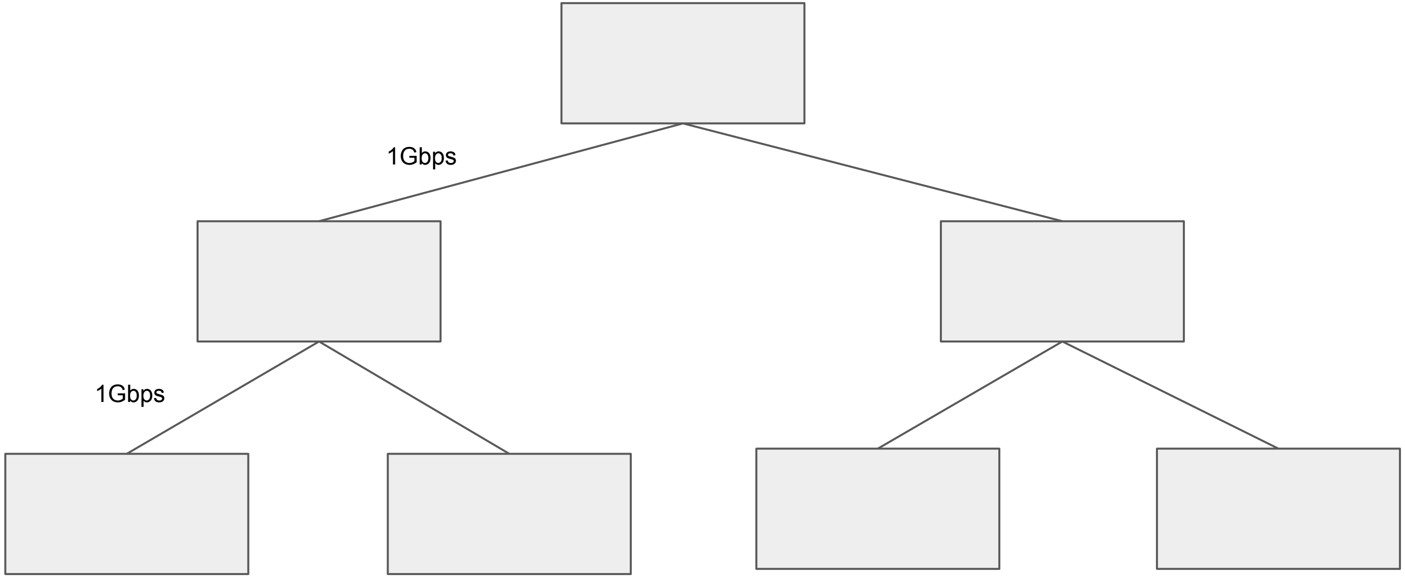
Broadcast



Anycast



Oversubscription Ratio



VL2

Conventional data center architecture

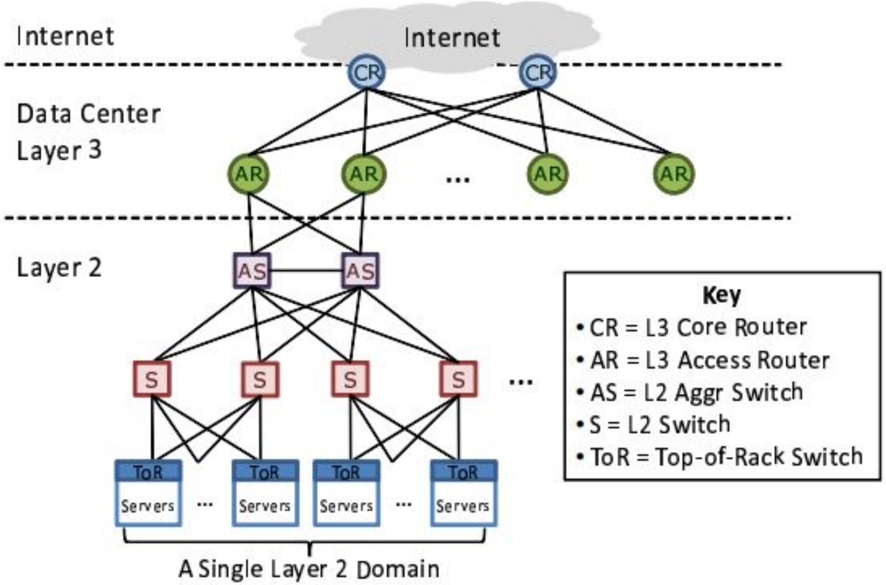
3 layers: Core Layer, Aggregation Layer, Access Layer

Addressing: Receive a request from Internet-> a certain server

Virtual IP-> CR -> AR -> Layer 2 -> AS ->

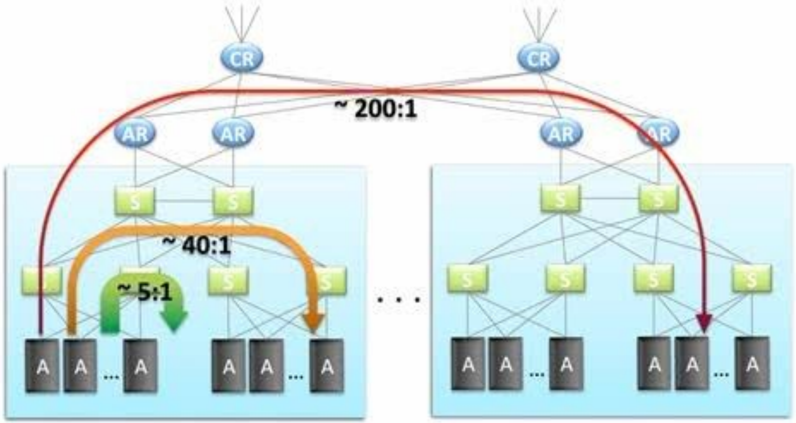
->S(L2 Switch) -> Server(Direct IP)

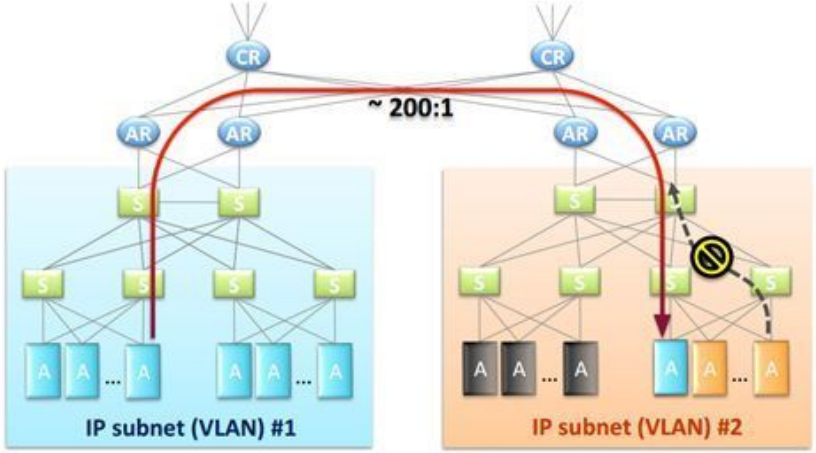
Load Balancer schedule the mapping



Potential Problems

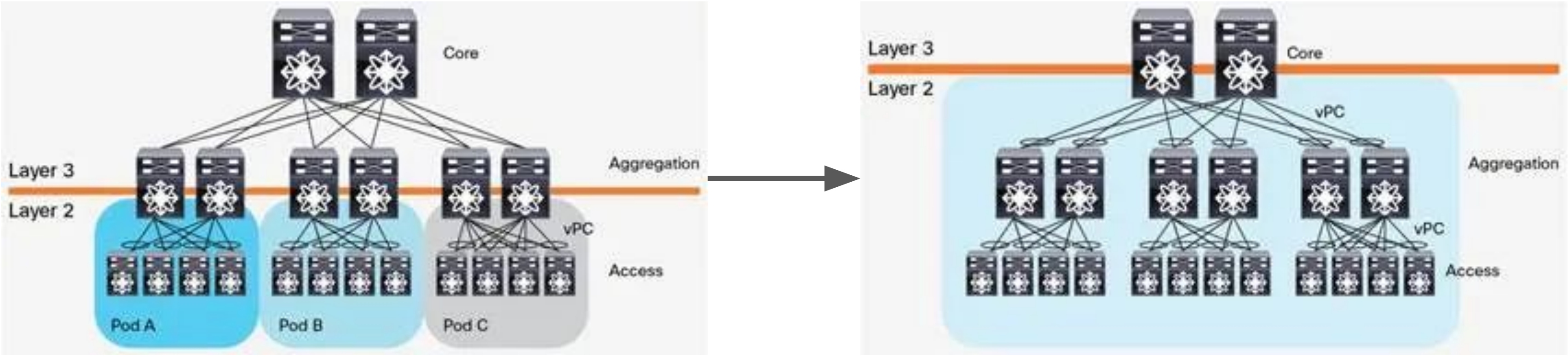
- Bandwidth limit from server to server
- Unbalanced Resource
- Affected by other server





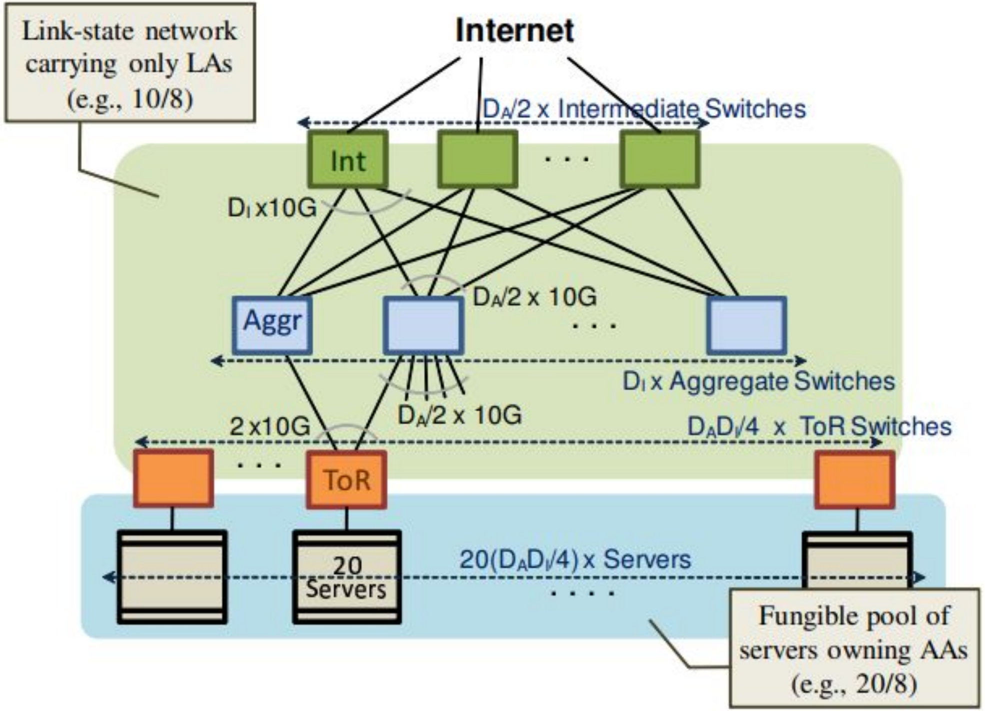
Objectives

- Uniform High Capacity
- Performance Isolation
- Layer-2 semantics



VL2 Network

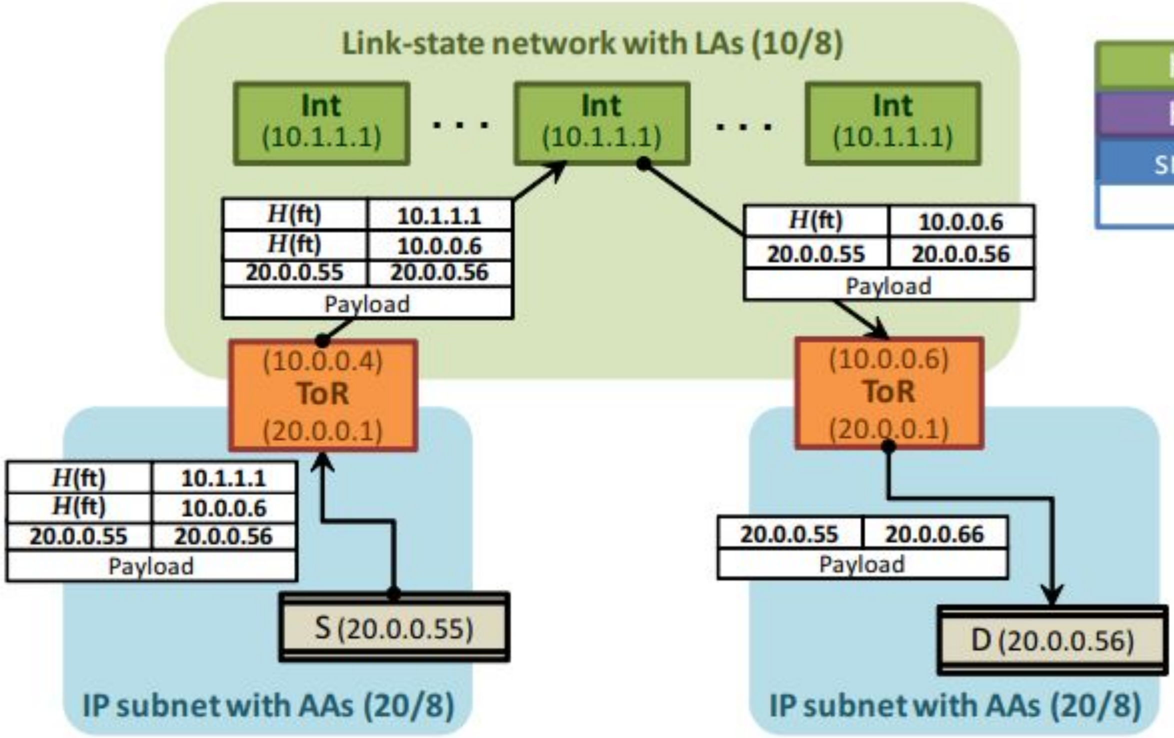
- Application Address(AA)
- Locator Address(LA)


Link-state network carrying only LAs (e.g., 10/8)
Internet
$D_A/2$ x Intermediate Switches
Int
. . .
D_I x10G
$D_A/2$ x 10G
Aggr
. . .
D_I x Aggregate Switches
2 x10G
$D_A/2$ x 10G
$D_AD_I/4$ x ToR Switches
. . .
ToR
20 Servers
$20(D_AD_I/4)$ x Servers
. . . .
Fungible pool of servers owning AAs (e.g., 20/8)

Design Principles

- Randomizing to cope with volatility
- Building on proven networking technology
- Separating names from locators
- Embracing End Systems

Addressing



H(ft)	10.1.1.1
H(ft)	10.0.0.6
20.0.0.55	20.0.0.56
Payload	

H(ft)	10.1.1.1
H(ft)	10.0.0.6
20.0.0.55	20.0.0.56
Payload	

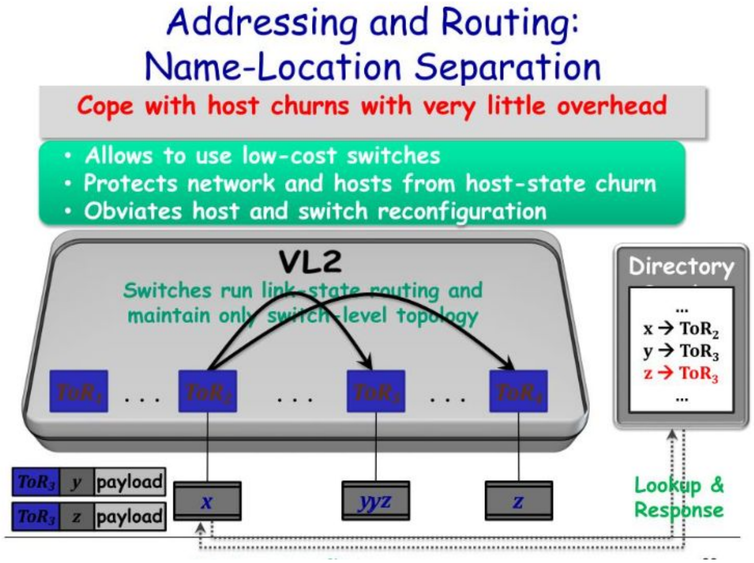
H(ft)	10.0.0.6
20.0.0.55	20.0.0.56
Payload	

20.0.0.55	20.0.0.66
Payload	

H(ft)	Int LA
H(ft)	dstToR LA
src AA	dst AA
payload	

Addressing

- S request(D AA) -> Shim layer-> Directory system
- Directory system(D LA)->Shim layer ->S ToR
- S ToR-> Aggr -> Int -> Aggr -> D TOR -> D



Addressing and Routing: Name-Location Separation

Cope with host churns with very little overhead

- Allows to use low-cost switches
- Protects network and hosts from host-state churn
- Obviates host and switch reconfiguration

Switches run link-state routing and maintain only switch-level topology

Load Balancing and Multi-path Transmission

- Load balancing: VLB
- Multipath Transmission: ECMP

Every intermediate switch use same LA. Every aggregation switch can communicate with servers.

Valiant Load Balancing (VLB)

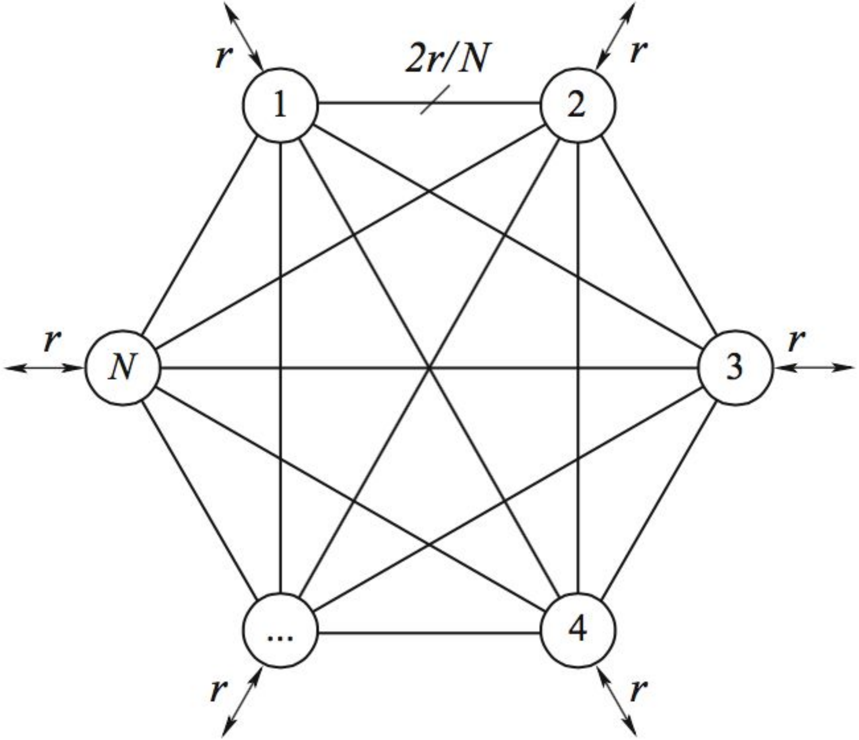
- Random load balancing
- Decentralization
- 2-step routing

- Step1: a flow enters the first node

  and divert to different nodes

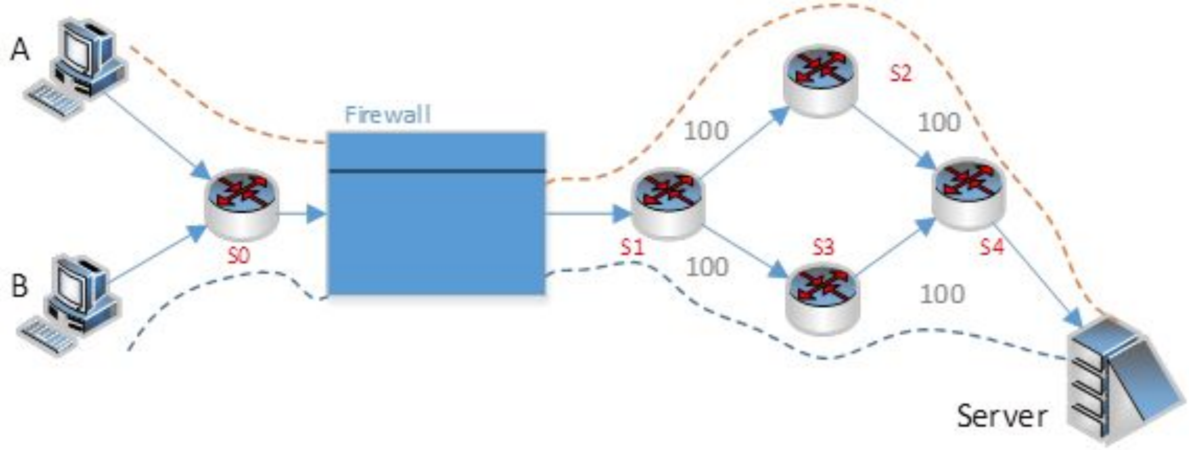
- Step2: reach the terminal node

Reduce the edge load



Equal-cost Multi-path（ECMP)

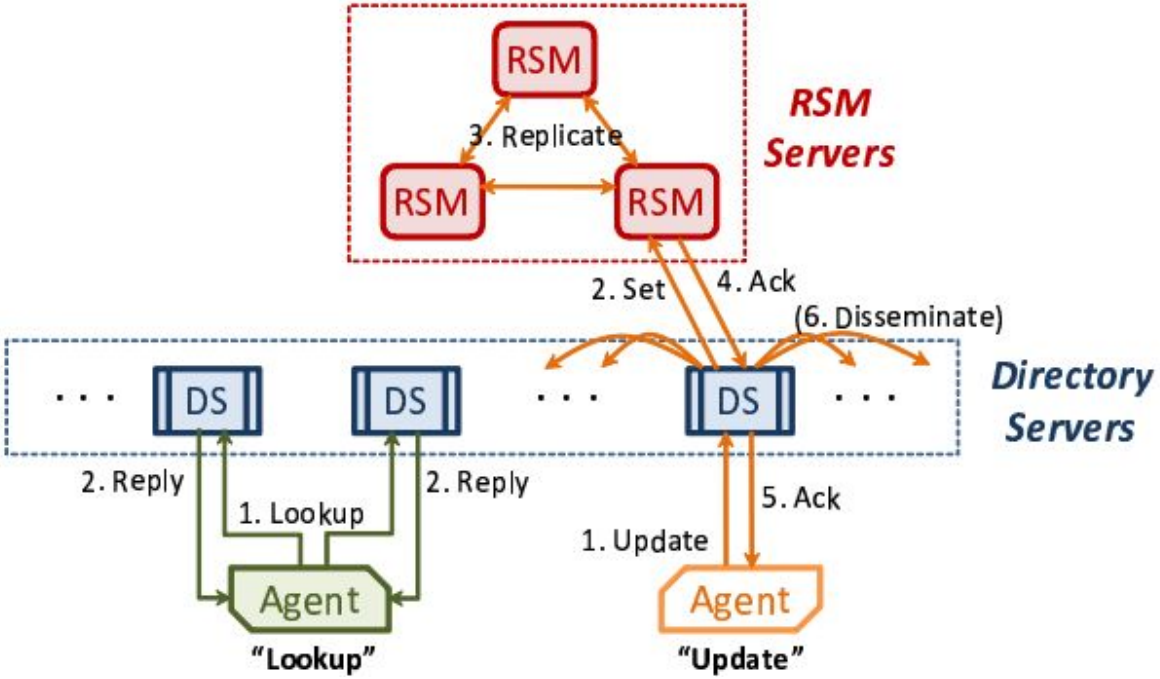
- Flow-level load balancing



Routing

- In a nutshell, take a random path ip to a random intermediate switch and random path down to ToR.
- All Intermediate switch has the same LA address.
- Leverage ECMP to utilize multiple paths from one node to another.

Directory update and lookup



Directory update

- RSM: maintain the consistency between directory servers
- DS:read mapping, process the request
- DS cache all AA-LA mapping from RSM
- Agent(update info)-> DS -> RSM -> all RSM -> ack to DS -> Agent -> all DS
- Passive update: source agent -> DS ->source agent -> source ToR ->...-> target ToR -> DS -> all DS

Disadvantages

- Much more cables. All Intermediate and aggregate switches must be connected to each other.
- Requires a high-performance, low-latency directory system to provide mapping search services, which brings additional overhead to the data center.
- VL2 modifies the hosts and servers protocol stack(remained switch HW unmodified)

Discussion Questions

- How will you change VL2 if traffic patterns were predictable/can be modeled really well by some learning algorithm? How would you implement such a change and how does it compare with the implementation in the paper?
- What optimizations can be performed in VL2 for example in the topology, network devices, etc using new hardware/software available today?

Discussion Questions

- HW: configurable and high performance hardware such as FPGA to configurable ASIC to implement the switches.
- Portland:
- Portland is highly customized

portland-sigcomm09.pdf (ucsd.edu)

Oktopus

Towards Predictable Datacenter Networks - Microsoft Research

Tenant Workload

- Moving workload from on premise to datacenter saw unpredictable performance and mismatch between desired and achieved network performance.
- Network Communication between two nodes from one tenant can be affected by communication from another tenant.

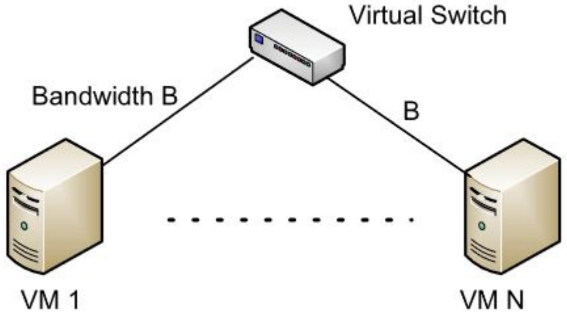
Virtual Network Abstraction

Tenants will request N virtual machines with specified CPU, memory, storage. But it will also need to specify its network performance expectation

Goal:

- Tenant suitability: Allow tenants to reason in an intuitive way about their network performance.
- Provider flexibility: Providers should be able to multiplex many virtual network on their physical network.

Virtual Cluster



Request <N, B>

Each VM can send and receive at rate B

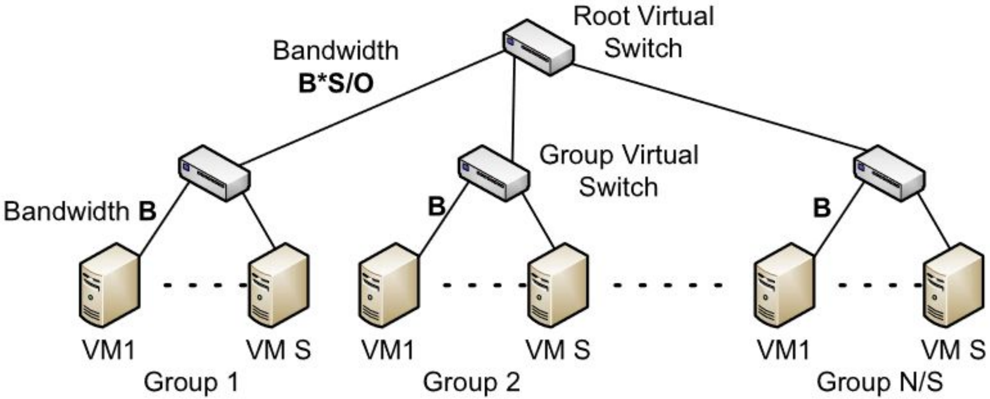
Switch bandwidth needed = N*B

Figure 2: Virtual Cluster abstraction.

Illusion of compute nodes connected through Ethernet switch.
No oversubscription!

Do tenants really need no oversubscription?

Virtual Oversubscribed Cluster



Request <N, S, B, O>

N VMs in groups of size S, Oversubscription factor O

Group switch bandwidth = S*B, Root switch bandwidth = N*B/O

Main components

- Management plane

Where to put the VM in the datacenter.

- Data plane

How to limit the bandwidth of a tenant.

Cluster Allocation

A logically centralized network manager (NM) will maps the tenant VM to physical machine and making sure that the guarantee can be met.

Uses greedy allocation algorithm to allocate VMs in the lowest level.

Main components

- Management plane

Where to put the VM in the datacenter.

- Data plane

How to limit the bandwidth of a tenant.

Rate limiting VMs - Distributed Rate Limiting (DRL)

- Rate-limiting is done at hypervisors to enforce bandwidth available at each VM.
- No need to adds complexity to datacenter switches.
- One VM acts as controller VM that periodically receives information about traffic measurements. Then, it will tell the other VM to adjust their sending rate.
- Tenants without VM is at a lower priority then those with virtual networks.

Hadrian

- Uses minimum bandwidth guarantee to drive up network utilization with still having fairness in terms of proportionality between paying tenants.
- VM Placement algorithm is modeled by a max-flow network problem. Still lacks optimization for memory requirements, fault tolerance, reducing VM migration, energy efficiency based allocation.
- Prior work only guarantees intra-tenant bandwidth. Hadrian can specify inter-tenant communication bandwidth.