



# DATACENTER NETWORKING



1. Portland: A scalable fault-tolerant layer 2 data center network fabric: SIGCOMM 2009
2. VL2: A scalable and flexible data center network SIGCOMM 2009
3. Networking the cloud, Albert Greenberg, ICDCS 2009 Keynote talk
4. The cost of cloud: research problems in data center networking, Albert Greenberg, ACM CCR Review, January 2009

1

## Data Center Costs



| Amortized Cost* | Component            | Sub-Components                   |
|-----------------|----------------------|----------------------------------|
| ~45%            | Servers              | CPU, memory, disk                |
| ~25%            | Power infrastructure | UPS, cooling, power distribution |
| ~15%            | Power draw           | Electrical utility costs         |
| ~15%            | Network              | Switches, links, transit         |


- Upwards of \$1 to \$4 B for mega data center
- Server costs dominate
- Network costs significant

**The Cost of a Cloud: Research Problems in Data Center Networks.**  
 Sigcomm CCR 2009. Greenberg, et.al.

\*3 yr amortization for servers, 15 yr for infrastructure. 5% cost of money

2


## Data center vs Enterprise



- Enterprise: IT cost dominates
  - 1 Human: 100 servers
  - Automation is partial, configuration, monitoring not fully automated
- Data center: Other costs
  - 1 Human: 1000 servers
  - Automation is mandatory, scale

3

## Data center vs Enterprise



- Enterprise: Scale not present
  - Limited Shared resources
  - Isolation
- Data center: Scale out
  - 100000 servers
  - Upfront cost is high, leverage shared resources
- Scale up vs scale out
- Enterprise: a few high priced servers— scale up
- Datacenter: scale out, distributed workload, spread out a number of commodity servers

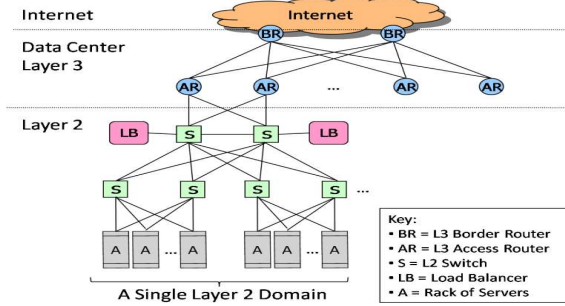
4

## Data center vs Enterprise

- Enterprise: CAPEX
  - Capital expenditure borne by the enterprise
  - License and maintenance
  - Utilization not important
- Data center: OPEX
  - Pay per use for customers
  - Upfront cost is high, amortized over time and use
  - Utilization is very important

5

## Architecture of Data Center Networks (DCN)



6

## PortLand: A Scalable Fault-Tolerant Layer 2 Data Center Networks Fabric

Radhika Niranjana Mysore, et.al,  
 Department of Computer Science and Engineering  
 University of California San Diego

SIGCOMM 2009

7

## Motivation

- **Requirements for Data Center Networks (DCN):**
  - **R1:** Any VM may migrate to any physical machine without change their IP addresses
  - **R2:** An administrator should not need to configure any switch before deployment
  - **R3:** Any end host should efficiently communicate with any other end hosts through any available paths
  - **R4:** No forwarding loops
  - **R5:** Failure detection should be rapid and efficient
- **Implication on network protocols:**
  - A single layer2 fabric for entire data center (R1&R2)
  - Mac forwarding tables with hundreds of thousands entries (R3)
  - Efficient routing protocols which disseminate topology changes quickly to all points (R5)

8

## Recall: SEATTLE

| Layer         | PlugNPlay | Scalability | Switch state | VM migration |
|---------------|-----------|-------------|--------------|--------------|
| Layer 2 (MAC) | +         | -           | -            | +            |
| Layer 3 (IP)  | -         | +           | +            | -            |

- OSPF among Switches
  - Links state broadcast to all switches
- Switch stores O(N) state
- Datacenter: Virtualization
- Each end host can have 10 to 20 virtual endpoints
- 100000 servers → 2 M endpoints

9

## SEATTLE vs Portland

- SEATTLE: 1-hop DHT; directory stores IP,MAC,location(Switch\_ID) mappings
- Portland: Consider 1 fixed tree structure
- Use MAC address that encodes location!

10

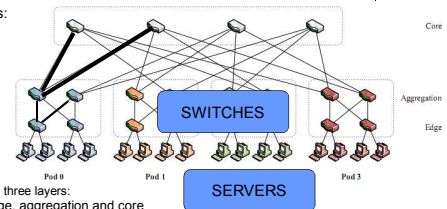
## Datacenter considerations

- Layer 2 approach:
  - Forwarding on flat MAC addresses
  - Less administrative overhead
  - Bad scalability
- Combine of layer 2 and layer 3:
  - VLAN
  - Resource partition problem
- End host visualization:
  - Needs to support large addresses and VM migrations
  - In layer 3 fabric, migrating the VM to a different switch changes VM's IP address
  - In layer 2 fabric, migrating VM incurs scaling ARP and performing routing/forwarding on millions of flat MAC addresses.

11

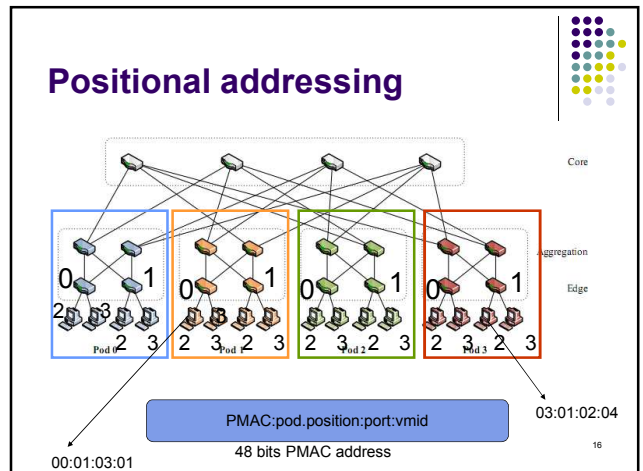
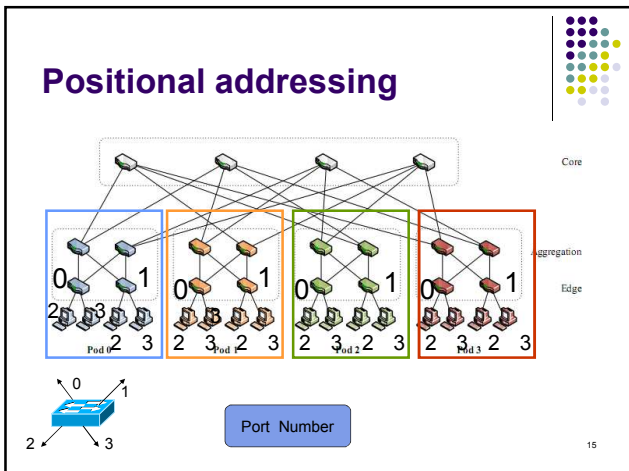
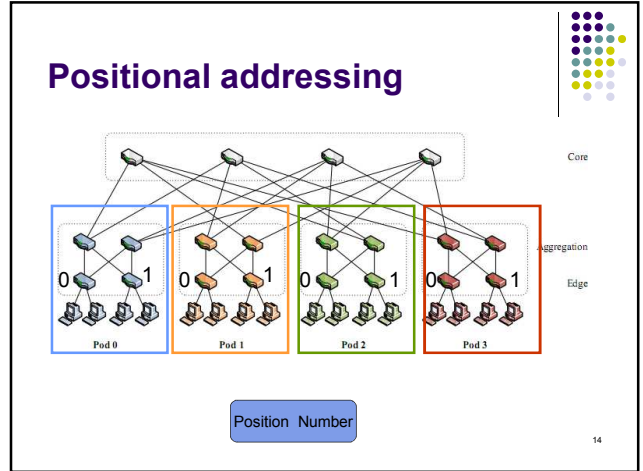
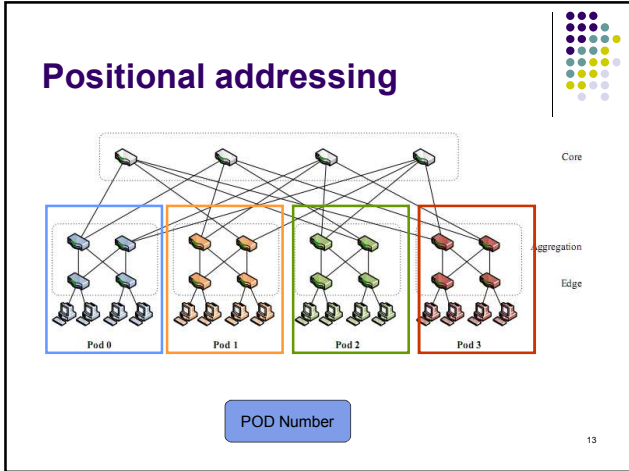
## Fat Tree topology

- Fat Tree Networks:



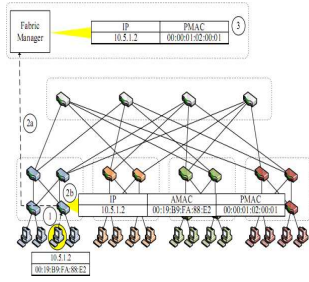
- Split fat tree into three layers:
  - Labeled edge, aggregation and core
- Split fat tree into k pods (k=4)
- Each pod with  $k^2 / 4$  hosts
- Each source and destination has  $k^2 / 4$  paths
- B/W or capacity progressively increases higher towards the root

12



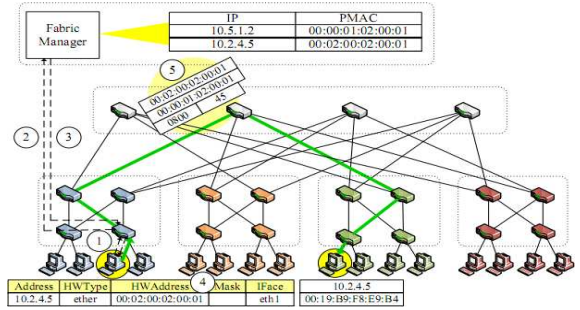
## Positional Pseudo MAC Addresses

- Pseudo MAC (PMAC) addresses encodes the location of the host
  - 48-bit: pod.position.port.vmid
  - Pod (16 bit): pod number of the edge switch
  - Position (8 bit): position in the pod
  - Port (8 bit): the port number it connects to
  - Vmid (16 bit): VM id of the host
  - Edge switches assign vmids to MAC addresses seen on its ports



17

## Proxy-based ARP



18

## Fabric Manager

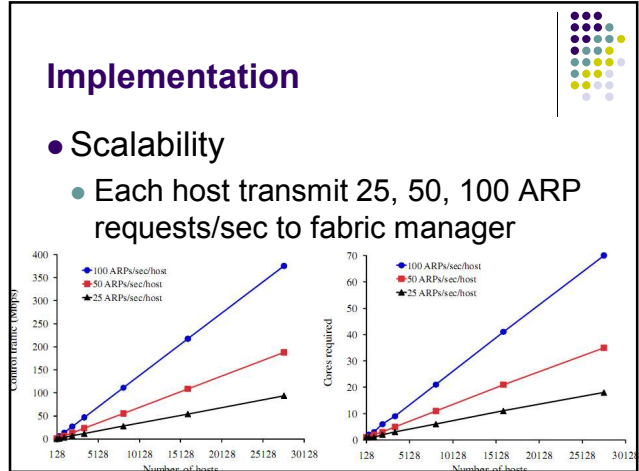
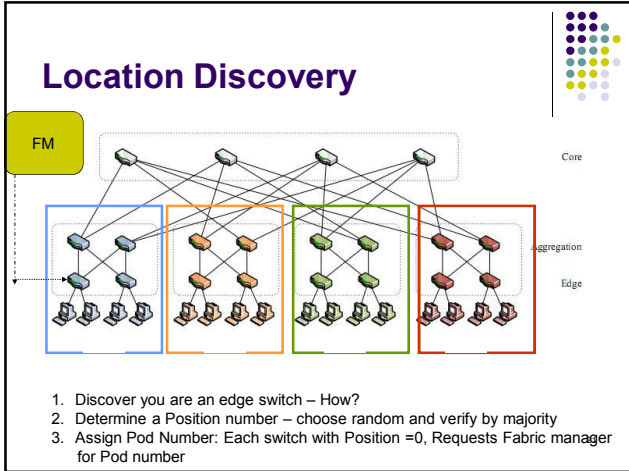
- Characteristics:
  - Logically centralized user process running on a dedicated machine
  - Maintains soft state about network configuration information
  - Responsible for assisting with ARP resolution, fault tolerance and multi cast
- Why centralized?
  - Eliminate the need for administrator configuration

19

## Distributed Location Discovery

- Switches periodically send Location Discovery Message (LDM) out all of their ports to set their positions and to monitor liveness
- LDM contains: switch identifier, pod number, position, tree level, up/down
- Find position number for edge switch:
  - Edge switch randomly proposes a value in [0, k/2-1] to all aggregation switch in the same pod
  - The unused and not tentatively reserved ones are verified
- Find tree level and up/down state:
  - Port states: disconnected, connected to end host, connected to another switch
  - A switch with at least half of ports connects to end hosts is an edge switch, ports connect to other switches are upward.
  - A switch get LDM from edge switch is aggregation switch, ports connect to edge switch are downward, ports connect to core switches are upward.
  - A switch with all ports connect to aggregation switch is core switch, all ports are downward.

20



- ### Conclusions
- A scalable, fault tolerant layer 2 routing and forwarding protocol for DCN
  - Based on fat tree network topology
  - PMAC used to encode the location of the end host
  - AMAC to PMAC translation needed
  - Header rewriting

Virtual Layer 2:  
 A Scalable and Flexible  
 Data-Center Network  
 Albert Greenberg et.al., SIGCOMM 2009  
**Microsoft Research**

Slides from ICDCS 2009 Keynote talk and SIGCOMM 09 presentation

## Tenets of Cloud-Service Data Center

- **Agility:** Assign any servers to any services
  - Boosts cloud utilization
- **Scaling out:** Use large pools of commodities
  - Achieves reliability, performance, low cost

Statistical  
Multiplexing  
Gain



Economies  
of Scale

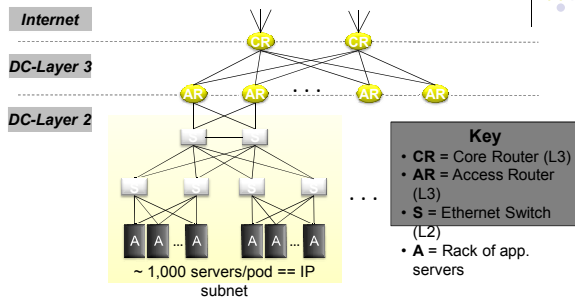
25

## VL2: basic idea

- Configure servers in such a way that they appear to be in one big IP subnet
- Avoid Broadcast (ARP) by using a directory service to convert IP to MAC address of RAC containing the server
- Use encapsulation to forward packet to ToR switch
- Aggregate switches uses IP anycast to do load balancing among many upward paths to Intermediate switch

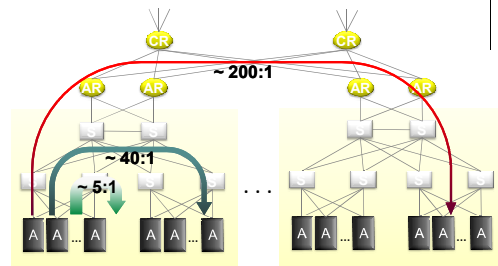
26

## Status Quo: Conventional DC Network



27

## Conventional DC Network Problems



- Dependence on high-cost proprietary routers
- Extremely limited server-to-server capacity

28

### And More Problems ...

~ 200:1

IP subnet (VLAN) #1

IP subnet (VLAN) #2

- Resource fragmentation, significantly lowering cloud utilization (and cost-efficiency)

29

### And More Problems ...

~ 200:1

Complicated manual L2/L3 re-configuration

IP subnet (VLAN) #1

IP subnet (VLAN) #2

- Resource fragmentation, significantly lowering cloud utilization (and cost-efficiency)

30

### And More Problems ...

Revenue lost

Expense wasted

- Resource fragmentation, significantly lowering cloud utilization (and cost-efficiency)

31

### An Example VL2 Topology: Clos Network

| Node degree (D) of available switches & # servers supported |                   |
|---|-------------------|
| D   | # Servers in pool |
| 4   | 80                |
| 24  | 2,880             |
| 48  | 11,520            |
| 144   | 103,680           |

Top Of Rack switch

20 ports

$[D^2/4] * 20$  Servers

- A scale-out design with broad layers
  - Same bisection capacity at each layer → no oversubscription
  - Extensive path diversity → Graceful degradation under failure

32

### Addressing and Routing: Name-Location Separation

**Cope with host churns with very little overhead**

**VL2 Switches run link-state routing and**

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

Directory Service

Lookup & Response

Servers use flat names

33

### Separating Names from Locations: How Smart Servers Use Dumb Switches

- Encapsulation used to transfer complexity to servers
  - Commodity switches have simple forwarding primitives
  - Complexity moved to computing the headers
- Many types of encapsulation available
  - IEEE 802.1ah defines MAC-in-MAC encapsulation; VLANs; etc.

34

### Embracing End Systems

- Data center OSes already heavily modified for VMs, storage clouds, etc.
  - A thin shim for network support is no big deal
- No change to applications or clients outside DC

35

### Use Randomization to Cope with Volatility

| D   | # Servers in pool |
|-----|-------------------|
| 4   | 80                |
| 24  | 2,880             |
| 48  | 11,520            |
| 144 | 103,680           |

- Valiant Load Balancing
  - Every flow "bounced" off a random intermediate switch
  - Provably hotspot free for any admissible traffic matrix
  - Servers could randomize flow-lets if needed

36

### Traffic Forwarding: Random Indirection

**Cope with arbitrary TMs with very little overhead**

Links used for up paths (blue)  
Links used for down paths (red)

$I_{ANY}$   $T_1$   $T_2$   $T_3$   $T_4$   $T_5$   $T_6$   
x y z

37

### Traffic Forwarding: Random Indirection

**Cope with arbitrary traffic**

[ESim File Anycast]

- Harness huge bisection bandwidth
- Obviate esoteric traffic engineering or optimization
- Ensure robustness to failures
- Work with switch mechanisms available

Links used for up paths (blue)  
Links used for down paths (red)

$I_{ANY}$   $T_1$   $T_2$   $T_3$   $T_4$   $T_5$   $T_6$   
x y z

38

### Does VL2 Ensure Uniform High Capacity?

- How “high” and “uniform” can it get?
  - Performed all-to-all data shuffle tests, then measured aggregate and per-flow goodput

|                                     |       |
|-------------------------------------|-------|
| Goodput efficiency                  | 94%   |
| Fairness <sup>§</sup> between flows | 0.995 |

§ Jain's fairness index defined as  $(\sum x_i)^2 / (n \sum x_i^2)$

- The cost for flow-based random spreading

Fairness Index<sup>§</sup>

Time (s)

39

### VL2 Conclusion

- VL2 achieves **agility at scale** via
  1. L2 semantics
  2. Uniform high capacity between servers
  3. Performance isolation between services

**Lessons**

- Randomization can tame volatility
- Add functionality where you have control

40

