

VL2: A Scalable and Flexible Data Center Network

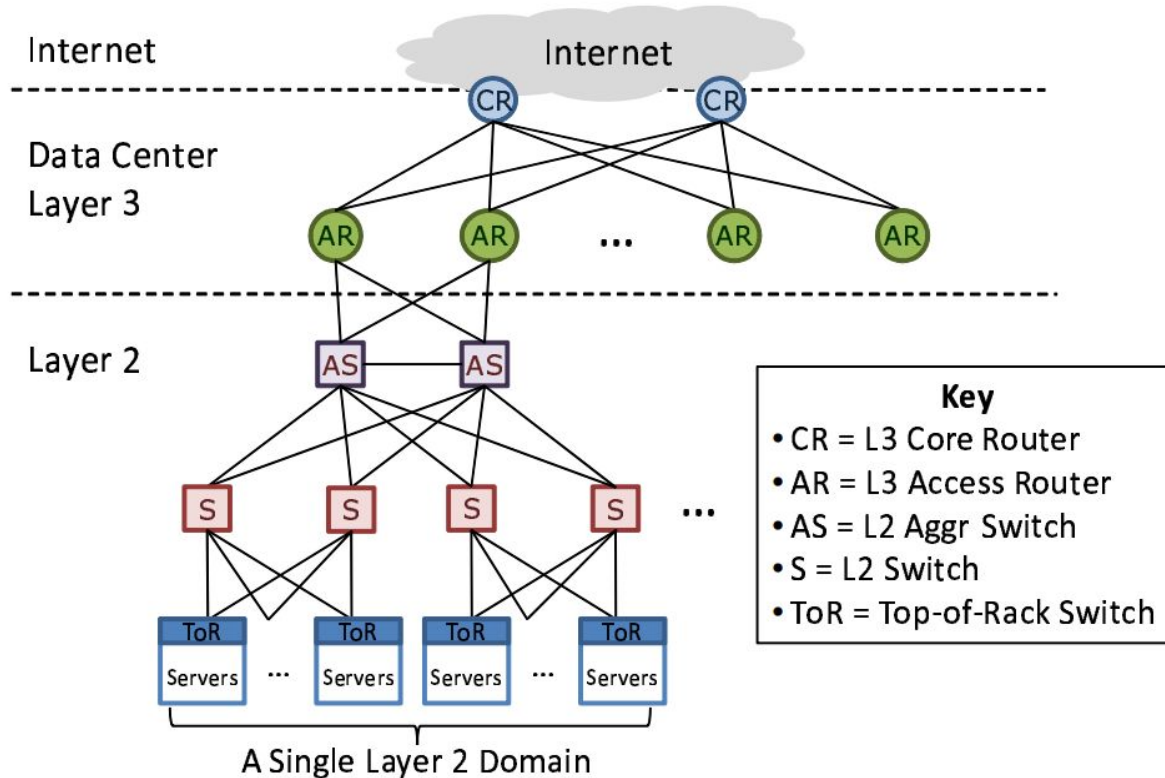
Data Center Requirements

Agility

Performance

Isolation

Traditional Hierarchical Topology



Tree-like layered design

Limited bandwidth, high oversubscription near core

VLAN & ARP scaling issues

Single point failures at higher layers

Measurements

Measurements: Flow Distribution

Majority of flows are small

Almost all bytes carried in large flows

Flows over a few GB are rare

Simpler & more uniform than Internet traffic

PDF: Probability Density Function
CDF: Cumulative Distribution Function

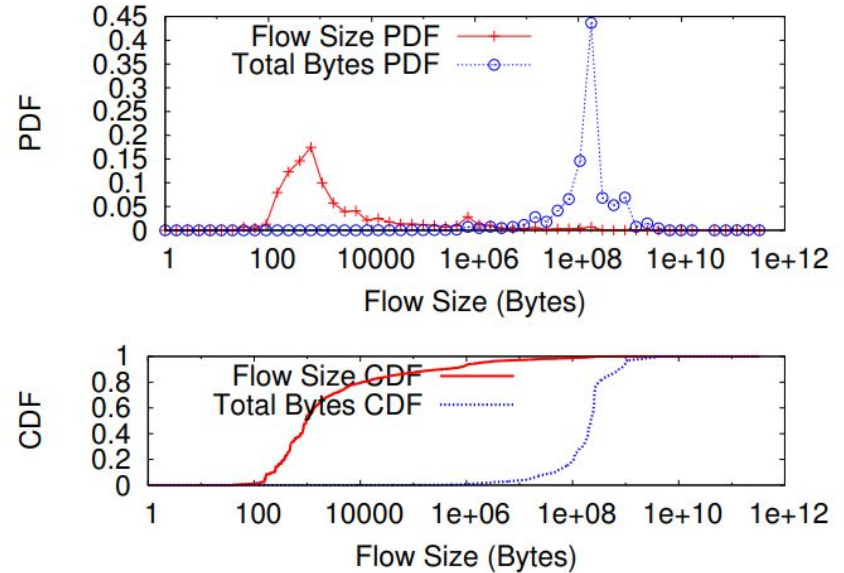


Figure 2: Mice are numerous; 99% of flows are smaller than 100 MB. However, more than 90% of bytes are in flows between 100 MB and 1 GB.

Measurements: Concurrent Flows

Two patterns:

10 flows per node (>50% of the time)

80 flows per node ($\geq 5\%$ of the time)

Rarely >100 concurrent flows

Implies: VLB works well at flow level

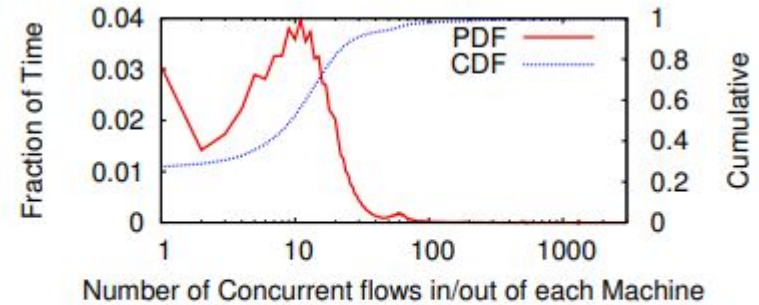


Figure 3: Number of concurrent connections has two modes: (1) 10 flows per node more than 50% of the time and (2) 80 flows per node for at least 5% of the time.

Measurements: Traffic Matrix

Tried clustering traffic matrices (ToR-to-ToR) over time

Large number of clusters → poor summarization

Traffic patterns change rapidly, no periodicity

Traffic Engineering infeasible → need VLB

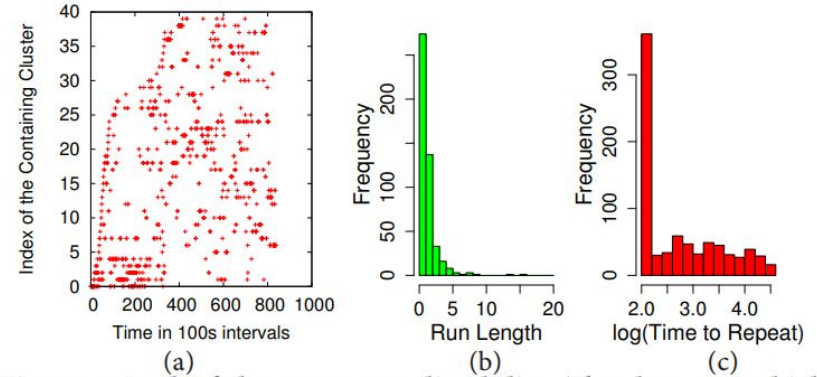


Figure 4: Lack of short-term predictability: The cluster to which a traffic matrix belongs, i.e., the type of traffic mix in the TM, changes quickly and randomly.

Figure 4(a): Traffic matrix shifts almost constantly

Figure 4(b): Run length short

Figure 4(c): No clear repeat pattern

Clos Topology

Scale-out Clos network between Aggr and Int switches

Built with commodity switches (cheap, uniform)

Richly connected: many equal-cost paths

Core bottleneck eliminated; resilient to failures

Supports VLB with LA (Locator Addresses) & AA (Application Addresses)

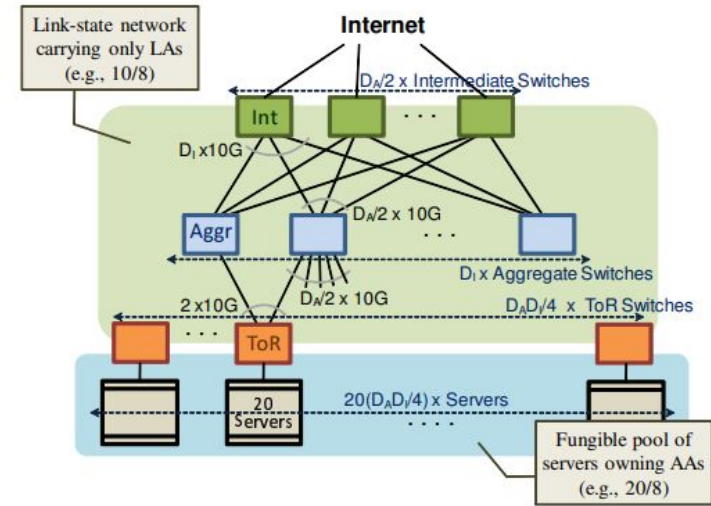


Figure 5: An example Clos network between Aggregation and Intermediate switches provides a richly-connected backbone well-suited for VLB. The network is built with two separate address families — topologically significant Locator Addresses (LAs) and flat Application Addresses (AAs).

Valiant Load Balancing (VLB)

Traffic unpredictable

Two-phase routing:

1. Send packet to a random intermediate switch
2. Forward to destination ToR

Realized with Anycast + ECMP

Ensures balanced load across all paths

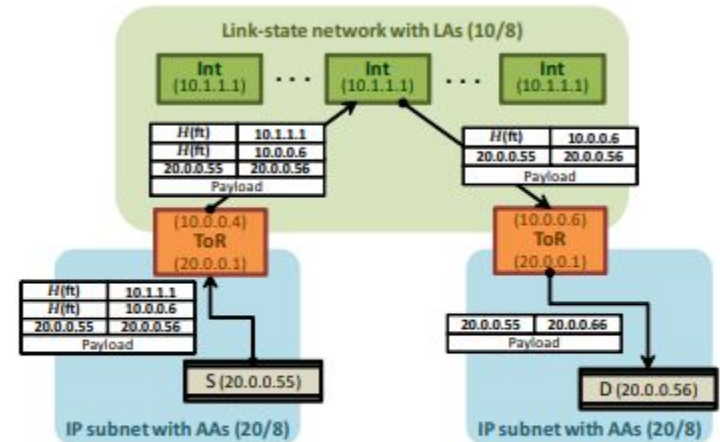


Figure 6: VLB in an example VL2 network. Sender S sends packets to destination D via a randomly-chosen intermediate switch using IP-in-IP encapsulation. AAs are from 20/8, and LAs are from 10/8. $H(f_t)$ denotes a hash of the five tuple.

Address Resolution & Packet Forwarding

Two IP families:

Locator Address (LA) : used by switches for routing

Application Address (AA) : stable ID for servers/services

Each AA mapped to a ToR's LA via Directory System

Packet forwarding:

VL2 agent encapsulates packet with LA (destination ToR)

Intermediate switch decapsulates and forwards to AA

Address resolution: ARP intercepted => resolved by Directory

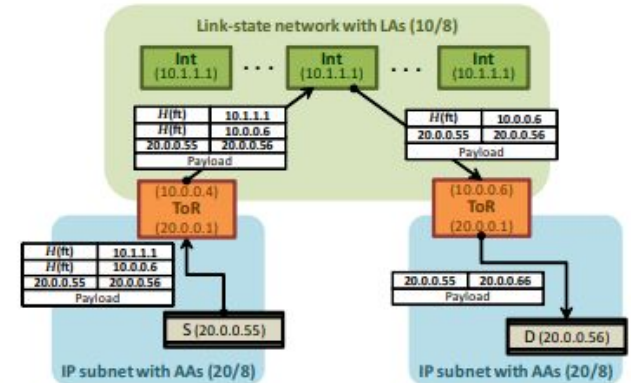


Figure 6: VLB in an example VL2 network. Sender S sends packets to destination D via a randomly-chosen intermediate switch using IP-in-IP encapsulation. AAs are from 20/8, and LAs are from 10/8. $H(f, t)$ denotes a hash of the five tuple.

VL2 Directory System

Directory Servers maintain AA → LA mappings

Agents on each host:

- Intercept ARP, send queries to Directory
- Cache mappings locally

RSM (Replication State Machines): ensure consistency

Provides access control & service isolation

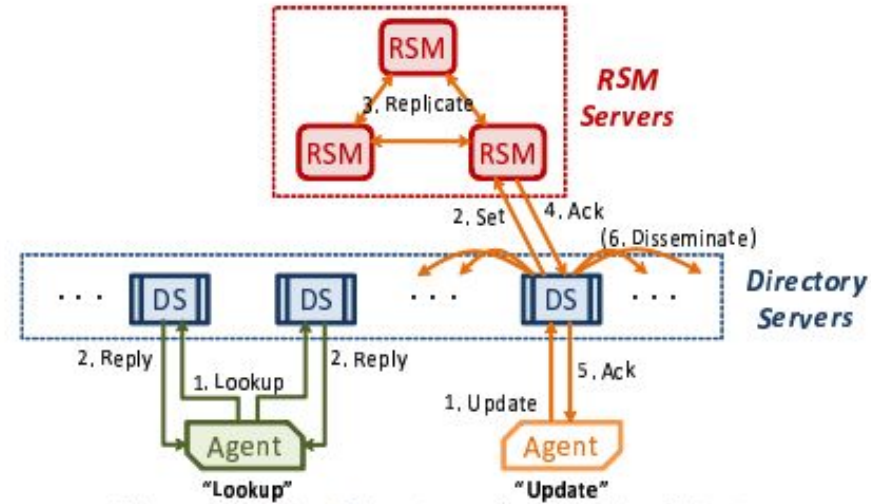


Figure 7: VL2 Directory System Architecture

VL2 Design

Clos Topology: scale-out, high bandwidth, fault tolerant

Valiant Load Balancing (VLB): traffic spreading, robust to unpredictable workloads

Addressing Separation:

- Locator Addresses (LAs) for routing

- Application Addresses (AAs) for stable service IDs

Directory System: AA→LA mapping, access control, service isolation

Evaluation – Prototype

VL2 prototype built and tested

Scale: 100 servers, commodity switches

Components:

- VL2 Agents on each host

- Directory System implementation

- Clos topology with Aggr + Intermediate switches

Goal: validate performance, agility, isolation in practice

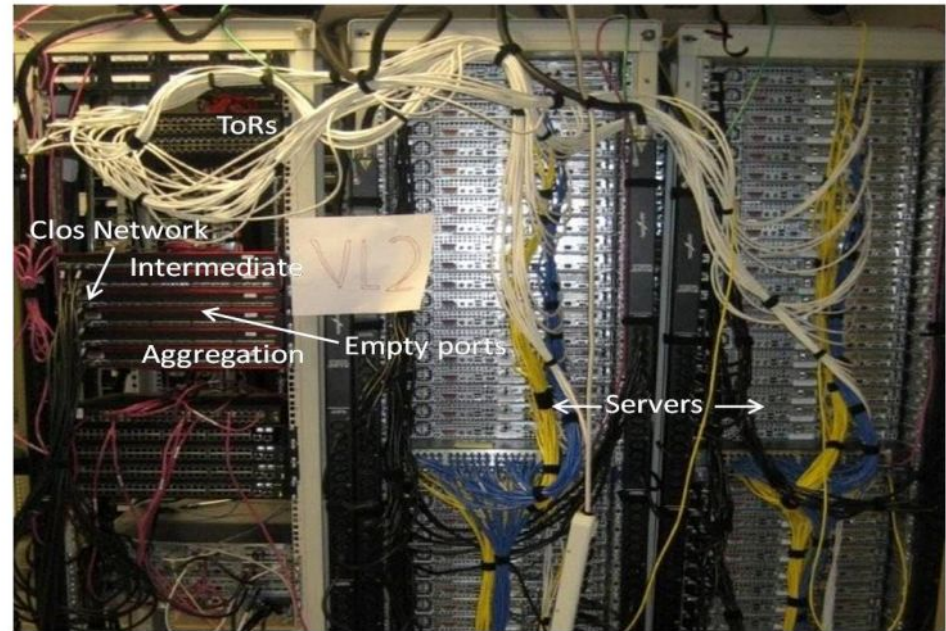


Figure 8: VL2 testbed comprising 80 servers and 10 switches.

Evaluation – Goodput

Prototype built with 100 servers

Shuffle test: 2.7 TB data exchange among 75 servers

Aggregate goodput sustained ≈ 60 Gbps

Traffic spread evenly across network paths

VL2 design works in practice

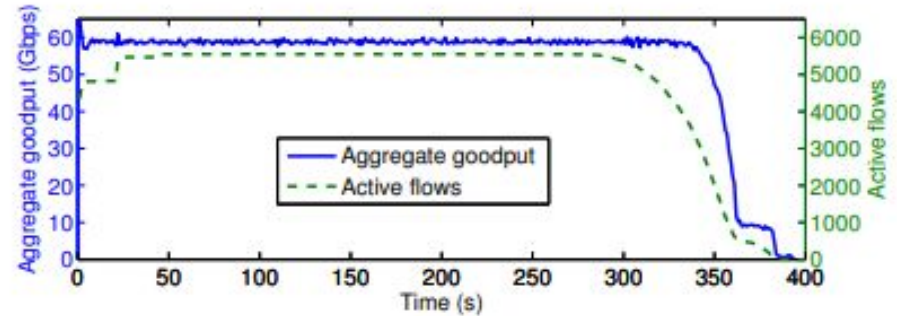


Figure 9: Aggregate goodput during a 2.7TB shuffle among 75 servers.

Evaluation – VLB Fairness

VL2 uses Anycast + ECMP to spread flows

Concern: ECMP does flow-level splitting, may be uneven

Experiment: 75-node testbed, traffic from real DC workload

Result: Jain's fairness index ≈ 0.98 across all Agg switches
VLB effectively balances load, prevents hotspots

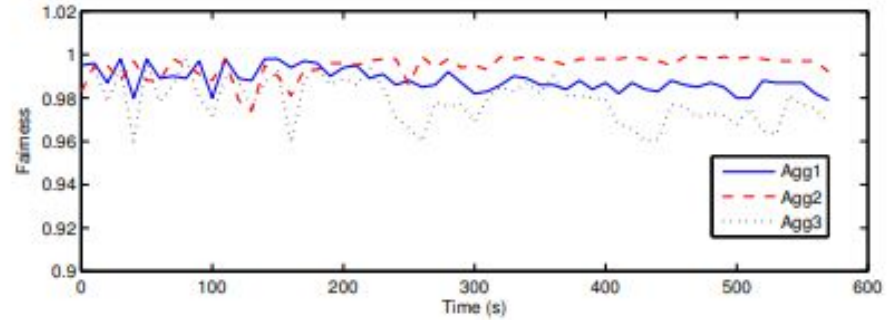


Figure 10: Fairness measures how evenly flows are split to intermediate switches from aggregation switches.

Evaluation – Performance Isolation

Goal: Check if one service affects another

Experiment 1: Long-lived TCP flows

Service 1's goodput remains stable despite Service 2 traffic

Experiment 2: Short TCP bursts (mice flows)

Only minor, brief fluctuations observed

Result: VLB + TCP ensures isolation between services

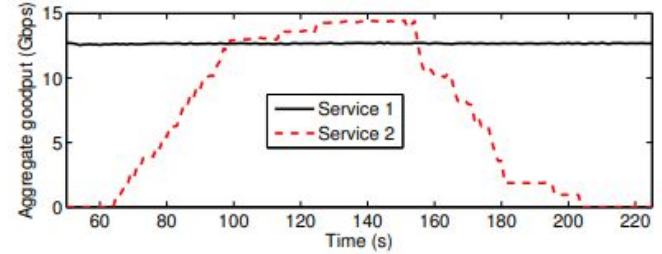


Figure 11: Aggregate goodput of two services with servers intermingled on the ToRs. Service one's goodput is unaffected as service two ramps traffic up and down.

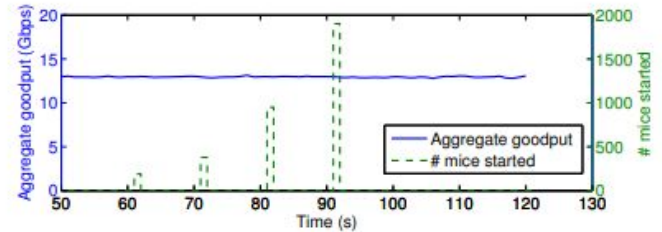


Figure 12: Aggregate goodput of service one as service two creates bursts containing successively more short TCP connections.

Evaluation – Directory System Performance

Lookup latency: most queries < 1 ms

Convergence latency: system
consistent within ~100 ms

Update latency: majority < 100 ms

Directory fast and reliable, not a
bottleneck

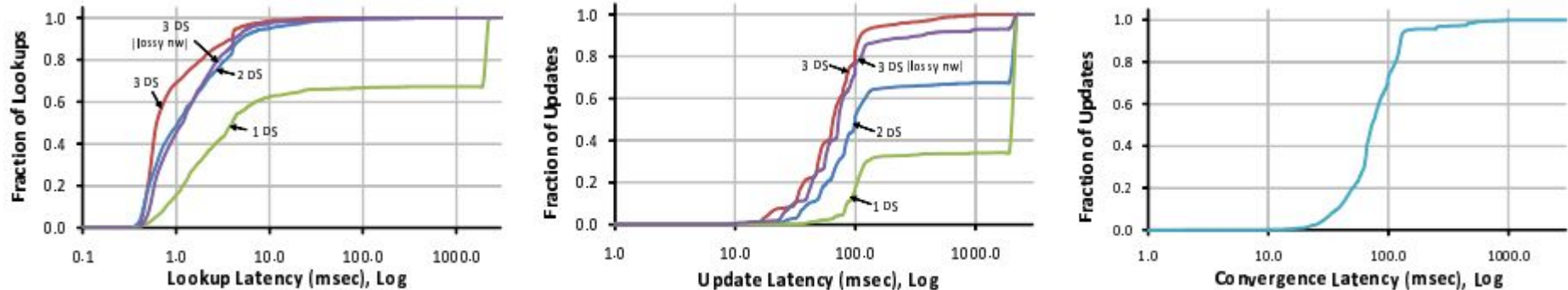


Figure 14: The directory system provides high throughput and fast response time for lookups and updates

Conclusion

Traditional tree topology: bottlenecks, poor agility, weak isolation

VL2 key ideas:

- Clos topology for scalability & high bandwidth

- Valiant Load Balancing (VLB) for even traffic distribution

- Address separation (AA & LA) for agility & transparency

- Directory system for scalable lookup & isolation

Result: Meets data center needs of Agility, Performance, and Isolation

Q&A