VL2: A Scalable and Flexible Data Center Network

Paper Note
Junzhi Gong

September 2018

1 Goals

- To be profitable, these data centers must achieve high utilization, and key to this is the property of agility the capacity to assign any server to any service. Agility promises improved risk management and cost savings.
- **Uniform high capacity.** The maximum rate of a server-to-server traffic flow should be limited only by the available capacity on the network-interface cards of the sending and receiving servers, and assigning servers to a service should be independent of network topology.
- **Performance isolation.** Traffic of one service should not be affected by the traffic of any other service, just as if each service was connected by a separate physical switch.
- **Layer-2 semantics.** Just as if the servers were on a LAN, data-center management software should be able to easily assign any server to any service and configure that server with whatever IP address the service expects.

2 Challenges

- Existing architectures do not provide enough capacity between the servers they interconnect.
- While data centers host multiple services, the network does little to prevent a traffic flood in one service from affecting the other services around it.
- The routing design in conventional networks achieves scale by assigning servers topologically significant IP addresses and dividing servers among VLANs.
- The fragmentation of address space creates an enormous configuration burden when servers must be reassigned among services, and the human involvement typically required in these reconfigurations limits the speed of deployment.

3 Intuition

- Give each service the illusion that all the servers assigned to it, and only those servers, are connected by a single non-interfering Ethernet switch - a Virtual Layer 2 - and maintain this illusion even as the size of each service varies from 1 server to 100000.

4 The Data Center Environment

- **Data center traffic analysis.**
  - The ratio of traffic volume between servers in our data centers to traffic entering/leaving our data centers is currently around 4:1 (excluding CDN applications).
  - Data-center computation is focused where high speed access to data on memory or disk is fast and cheap.
  - The demand for bandwidth between servers inside a data center is growing faster than the demand for bandwidth to external hosts.
  - The network is a bottleneck to computation. We frequently see ToR switches whose uplinks are above 80% utilization.
- **Flow distribution analysis.**
  - The flow size statistics show that the majority of flows are small (a few KB). However, almost all the bytes in the data center are transported in flows whose lengths vary from about 100 MB to about 1 GB. Flows over a few GB are rare.
  - More than 50% of the time, an average machine has about ten concurrent flows, but at least 5% of the time it has greater than 80 concurrent flows. We almost never see more than 100 concurrent flows.
- **Traffix matrix analysis.**
  - *Poor summarizability of traffic patterns.* The number of representative traffic matrices in our data center is quite large. This indicates that the variability in datacenter traffic is not amenable to concise summarization and hence engineering routes for just a few traffic matrices is unlikely to work well for the traffic encountered in practice.
  - *Instability of traffic pattern.* The traffic pattern changes nearly constantly, with no periodicity that could help predict the future. The lack of predictability stems from the use of randomness to improve the performance of data-center applications.
- **Failure characeristics.**
  - *The pattern of networking equipment failures.* Most failures are small in size, while large correlated failures are rare. However, downtimes can be significant.

  - *The impact of networking equipment failure.* Despite the redundancy techniques, we find that in 0.3% of failures all redundant components in a network device group became unavailable. In one accident, the failure of a core switch affected ten million users for about four hours. The main causes of these downtimes are network misconfigurations, firmware bugs, and faulty components (e.g., ports).

5 Design Principles

- **Randomizing to cope with volatility.** VL2 copes with the high divergence and unpredictability of data-center traffic matrices by using Valiant Load Balancing to do destination-independent (e.g., random) traffic spreading across multiple intermediate nodes.

- **Building on proven networking technology.** VL2 is based on IP routing and forwarding technologies that are already available in commodity switches: link-state routing, equal-cost multi-path (ECMP) forwarding, IP anycasting, and IP multicasting. VL2 uses a link-state routing protocol to maintain the switch-level topology, but not to disseminate end hosts' information.

- **Separating names from locators.** VL2s addressing scheme sep- arates server names, termed application-specific addresses (AAs), from their locations, termed location-specific addresses (LAs). VL2 uses a scalable, reliable directory system to maintain the mappings between names and locators.

- **Embracing end systems.** The rich and homogeneous pro- grammability available at data-center hosts provides a mechanism to rapidly realize new functionality.

6 Methods

- **Scale-out topologies.** Rather than scale up individual network devices with more capacity and features, we scale out the devices build a broad network offering huge aggregate capacity using a large number of simple, inexpensive devices. The Clos topology is exceptionally well suited for VLB.

- **VL2 addressing and routing.**

  - *Address resolution and packet forwarding.* VL2 uses two different IP-address families. The network infrastructure operates using location-specific IP addresses (LAs); all switches and interfaces are assigned LAs, and switches run an IP-based (layer-3) link-state routing protocol that disseminates only these LAs. On the other hand, applications use application-specific IP addresses (AAs), which remain unaltered no matter how servers' locations change due to virtual-machine migration or re-provisioning.
  - *Random traffic spreading over multiple paths.* To offer hot-spot-free performance for arbitrary traffic matrices, VL2 uses two related mechanisms: VLB and ECMP.

    VLB distributes traffic across a set of intermediate nodes and ECMP distributes across equal-cost paths. VL2 uses flows, rather than packets, as the basic unit of traffic spreading and thus avoids out-of-order delivery.

  - *Backwards compatibility.* Since VL2 employs a layer-3 routing fabric to implement a virtual layer-2 network, the external traffic can directly flow across the high-speed silicon of the switches that make up VL2. Servers that need to be directly reachable from the Internet (e.g., front-end web servers) are assigned two addresses: an LA in addition to the AA used for intra-data-center communication with back- end servers.

- **Maintaining host information using the VL2 directory system.** The VL2 directory provides three key functions: (1) lookups and (2) updates for AA-to-LA mappings; and (3) a reactive cache update mechanism so that latency-sensitive updates (e.g., updating the AA to LA mapping for a virtual machine undergoing live migration) happen quickly. Our design goals are to provide scalability, relia- bility and high performance.