MIT 6.829: Computer Networks Fall 2017

Lecture 16: Data Center Network Architectures

Scribe: Alex Lombardi, Danielle Olson, Nicholas Selby

1 Background on Data Centers

- Computing, storage, and networking combined
- "Warehouse-Scale Computers"
- Huge (monetary) investment
- Very complicated
- 45% of Data Center costs are due to operating the servers. Rest of the breakdown: 25% power infrastructure, 15% power draw, 15% network
- However, utilization of data center servers used to be only 30%, a huge waste of resources.
- Main issue: specific servers were dedicated to specific applications before their demand is known, so when applications were not utilized at full capacity, the servers went unused.

2 "Agility": Any service, any server

- Turn the servers into a single large fungible pool.
- Allows dynamic expansion and contraction of service footprint as needed.
- Achieving this requires workload management: virtual machines are crucial here.
- Also requires storage management: (possibly moving) servers need to access persistent data.
- In particular, there should be a mechanism for connecting to servers, even in the event of outages.
- Solution provides the abstraction of a Datacenter network as "one big switch".
- However, server traffic is drastically increasing over time *and* is quite unpredictable.
- Example: Microsoft datacenters. The traffic generated in Microsoft datacenters has increased 50% in 6 years! The amount of information exchanged within Microsoft datacenter is MUCH LARGER than all traffic between U.S. and China!

3 Conventional DC Network Problems

- First idea: why not built datacenter networks like we built the internet?
  - Rack of application servers
  - Multiple switches
  - DC layer 3, DC layer 2
  - Router, then connect to the internet
- Problems include: latency (rediscovering lost servers), bottlenecks near the top of the tree (exponential growth in traffic), resulting waste of resources (which are preparing for worst-case traffic).
- Example: *L2*, which uses "Ethernet switching", in which all servers are connected to each other.
  - Pros: flat addressing, seamless mobility and migration.
  - Cons: (1) when one server goes down, you have to reconfigure all other machines (broadcast updates; this limits scale), and (2) it relies on a spanning tree protocol, which forces traffic through a root, causing congestion.

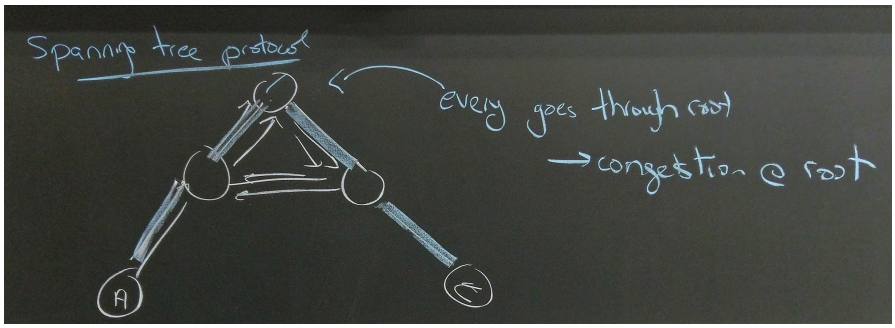

Figure 1: **Spanning Tree Protocol.** Creating a hierarchy for routing ensures that messages will not ring by ignoring paths which complete loops. However, eliminating some paths causes congestion in others, overloading the root of the spanning tree while completely underutilizing branches.

- Example: *L3* (routing hierarchy).

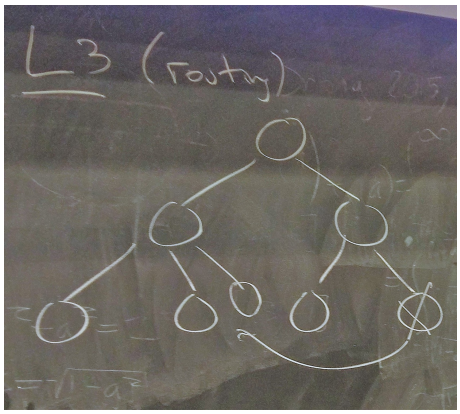

Figure 2: **L3 Routing** (routing with hierarchy)

  - Pros: (1) scalability, (2) ECMP (equal cost multipath routing), in which multiple equal cost paths from a source to a destination are simultaneously utilized.
  - Cons: (1) huge reconfiguration problem, (2) hard to migrate IP addresses.

- Elaboration on ECMP: forwarding is done on a per-flow basis; load balancing is achieved by sending different flows on different paths. ECMP is already a part of most routers.

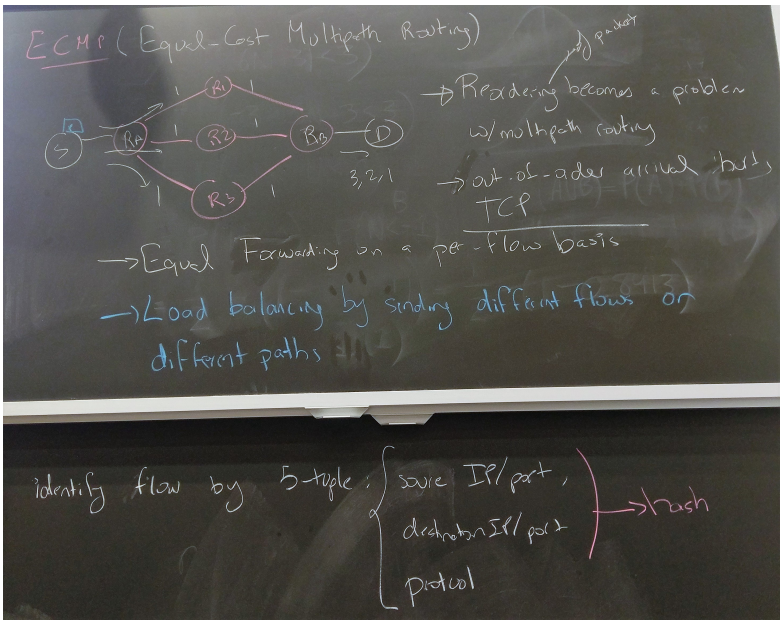

Figure 3: **Equal Cost Multipath Routing (ECMP).**

4 VL2 Paper [1] (Introduction)

- First made measurements in a datacenter.

- Instrumented a large cluster used for data mining and identified distinctive traffic patterns. Results: traffic patterns are highly volatile and unpredictable (only weak correlations).
- Traffic-aware optimization needs to be done frequently.
- DC opportunities: DC controller knows everything about the hosts, the host OSs are easily customizable, and probabilistic flow distribution works well enough (specifically because the flows are all roughly the same size).

5 VL2 Goals

- The illusion of a huge L2 switch. Now, if you move anything around, theyre all connected so you dont need to reconfigure your routers.
- L2 semantics: ability to move any server from one place to another.
- Uniform high capacity
- Performance isolation: if one application is consuming lots of CPU, for example, the other applications are unaffected.

6 Design Principles

- Clos topology: offer huge capacity via multiple paths.
- Randomizing to cope with volatility
- **Separating names from locations**: this allows the "any server, any service" property. This is specific to a datacenter because it requires full information about the network at all times.
- Embracing end systems: leverages the programmability and resources of servers.
- Building on proven network technology: prevents us from having to use expensive routers.

7 VL2 Solution Details

- Addressing and routing: name-location separation.
- VL2 switches run link-state routing and maintains only switch-level topology. This enables us to significantly decrease our forwarding table while still knowing what the switch interaction should be.
- Namely, it incorporates a dirctory service, i.e. a lookup table between server locations and ToR locations.
- Each ToR is connected to two aggregation switches, and each aggregation switch is connected to each intermediary router. If a server fails, we just have to update the directory of services to say that the server is reachable via another ToR.

- Why use hashing in VL2 agent workflow?
  - ECMP may not be able to look into the entire 5-tuple but can always look at source and destination IPs since it's running at the router level.
  - So, we can take 5-tuple and hash it to a specific src IP so that it can keep track of what is coming from where; can only do this on a per-flow basis.
- Why do we anycast and do double encapsulation? Load-balancing subroutine of ECMP dictates which path should used, which means that a priori we need to be able to forward to any intermediary router.
- How does L2 broadcast work? An ARP (Address Resolution Protocol) request is displaced by the directory server (intercepted by VL2 agent).
- How does internet communcation work? Servers are given *two addresses*: a LA (for the internet) and an AA (for inter-datacenter communication with backend servers). The LAs can be dynamically allocated. A server can broadcast its LA via BGP to externally reach the internet.
- VL2 Directory System implementation: read-optimized directory servers are used for lookups (uses first lookup response), and write-optimized replicated state machines are used for updates. Agents send updates to a random DS, who forwards the update to a RSM server. After the RSM acknowledges, the DS disseminates the update to the other servers.

8 Ramifications of VL2

- VL2 $\rightarrow$ Azure, Microsoft's cloud network. This was a 15 billion dollar bet on VL2!
- It was a huge success: from 2010 to 2015, there was a drastic increase in the number of compute instaces, exabytes of data are now stored in Azure, and the datacenter network operates in the Pbps.

References

[1] A. Greenberg, J.R. Hamilton, N. Jain, S. Kandula, C. Kim, P. Lahiri, D.A. Maltz, P. Patel, S. Sengupta. 2009, VL2: A Scalable and Flexible Data Center Network. SIGCOMM 2009.