CSE 461: Computer networks

Spring 2021

Ratul Mahajan
Building Massive Cloud Networks
Microsoft and Facebook just laid a 160-terabits-per-second cable 4,100 miles across the Atlantic

*Enough bandwidth to stream 71 million HD videos at the same time*

By Thuy Ong | @ThuyOng | Sep 25, 2017, 7:56am EDT

HUGE data center networks (DCN)

• Thousands of routers

• Hundreds of thousands of servers
Google’s Oregon DC
Inside a Google DC
DCN topologies

• Big iron → Commodity switches
DCN topologies

• Big iron $\rightarrow$ Commodity switches
• 1 Gbps $\rightarrow$ 10 Gbps $\rightarrow$ 40 Gbps $\rightarrow$ 100 Gbps (soon)
• Copper $\rightarrow$ Fiber
Oversubscription ratio

• Ratio of bisection bandwidth across layers of hierarchy
• Key design parameter that trades-off cost and performance
  • Higher oversubscription = lower cost but higher chance of congestion
DCN routing

• Spanning tree (L2) → OSPF/ISIS → BGP

• Each router acts as its own autonomous system (AS)
Backbone

• Provides global connectivity to DCs
Backbone

• Provides global connectivity to DCs

• May also have two backbones
  • A “public” backbone to connect to the outside world
  • A ”private” backbone for inter-DC connectivity

• Uses transcontinental and transoceanic fiber cables

• Routing: ISIS/OSPF $\rightarrow$ MPLS $\rightarrow$ SDN-based traffic engineering
MPLS – Multi Protocol Label Switching

• Can explicitly program paths -- tunnels
  • Allows taking non-shortest paths

• Auto-bandwidth: Constrained-shortest paths first (CSPF)
  • Fully distributed computation
  • Estimate demand
  • Find shortest path(s) that can fulfill the demand
SDN – Software Defined Networking

Decouple control and data plane
- Control plane populates the data plane entries (routing)
- Data plane forwards traffic (forwarding)

Traditionally, routing and forwarding are in the same device

Control plane separation opens up lots of new opportunities
- Traffic engineering in backbones (next)
- Network virtualization (later)
SDN-based traffic engineering

Centralized computation of forwarding tables

• Compute “optimal” paths outside of the network
• Based on estimated load; also factor in application priorities
Using the cloud

• Use a software service (e.g., email) -- SaaS
• Use application building blocks (e.g., database) -- PaaS
• Launch VMs – IaaS

• Build virtual networks
  • Provides the same abstraction as physical networks but with virtual devices
Connecting to the cloud

• Public Internet
• VPN from your physical resources to the cloud
• BGP peering
  • E.g., Amazon Direct Connect
The last ten years of the cloud

Scale, scale, scale ... (mostly)

Relatively small conceptual shifts
  • Lot of automation – minimize “snowflakes” and “fat fingers”
  • Troubleshooting: Find needles in haystack
    • E.g., Everflow [SIGCOMM ‘15], CorrOpt [SIGCOMM ‘17]
  • Centralized control of resources
    • E.g., SWAN [SIGCOMM ‘13], Footprint [NSDI ‘16]
  • Low-latency technologies, e.g., RDMA
Bigger shifts are coming

Verification
  • E.g., Batfish [NSDI ‘15], Minesweeper [SIGCOMM ‘17]

High-level synthesis
  • E.g., Propane [SIGCOMM ’16, PLDI ’17]

Programmable NICs and switches
  • E.g., ProjecToR [SIGCOMM ‘16], RAIL [NSDI ‘17]

Edge computing
  Tighter coupling with applications
What is in the box?
Router

A computer optimized for routing and forwarding

- Operating system to manage resources
- Routing protocol implementations (e.g., BGP, OSPF)
- Lots of ports (not TCP ports)
- Chip to forward traffic between ports at “line rate”
Router (2)

Traditionally, a hardware-software combo sold by a router vendor
  • Cisco
  • Juniper
  • Arista
  • ....

But moving toward open systems
  • SONiC – open source router OS from Microsoft
  • Running on “commodity” hardware
Configuring the router

Routers are not plug-n-play

• Configure IP addresses
• Configure which protocols to run
• Configure those protocols
• Configure management aspects, e.g., DNS servers, NTP servers

Configuration uses custom syntax:

• Example Cisco file: 
  https://github.com/batfish/pybatfish/blob/master/jupyter_notebooks/networks/example/configs/as1border2.cfg
Configuring the router (2)

Traditionally, configuration has been done manually

- Figure out the change, reason about it manually
- Log in to the router and apply the change
- High risk of logical errors and “fat fingers”

Increasingly, more automation

- Ansible, SaltStack, Nornir
- Batfish
Making a network out of routers

1. Get them connected
Making a network out of routers

1. Get them connected

2. Configure routers
   • Basic initial configuration provides connectivity to the router

3. Monitor, monitor, monitor

4. Configuration changes and maintenance