# Network Virtualization

**Omar Baldonado**
**Facebook, Network Infrastructure**
**November 22, 2019**

# What is “virtualization”?

- Creating a virtual version of a common resource
  - **Virtual memory** - process has its own address space
  - **RAID storage** - process thinks its writing to one disk, but many underneath
  - **Virtual machine** - the OS doesn’t know it is running on top of another OS (and not hardware)

- A way to share a common resource

# Progress toward “network virtualization”

- Many different steps/techniques over the years
- Generally, doing something a little different from the typical layer-defined behavior

# Ex 1: Network Address Translation (NAT)

# Ex 1: an Internet debate from the late 80s/early 90s

At Stanford! Steve Deering (PhD 1991, inventor of IPv6)

- “We’re going to run out of IPv4 address space - we need IPv6”
- “But it might take a while to roll out IPv6...”

And thus, network address translation (NAT) was born - from RFC 1918:

**3. Private Address Space**

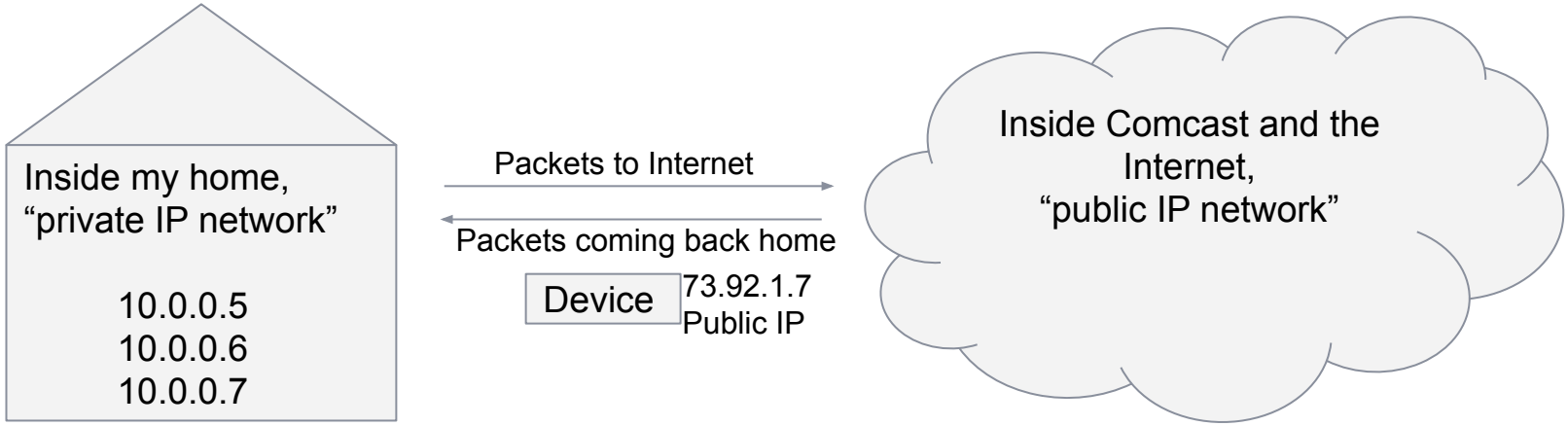
The Internet Assigned Numbers Authority (IANA) has reserved the following three blocks of the IP address space for private internets:

| | | | |
|---|---|---|---|
| 10.0.0.0 | - | 10.255.255.255 | (10/8 prefix) |
| 172.16.0.0 | - | 172.31.255.255 | (172.16/12 prefix) |
| 192.168.0.0 | - | 192.168.255.255 | (192.168/16 prefix) |

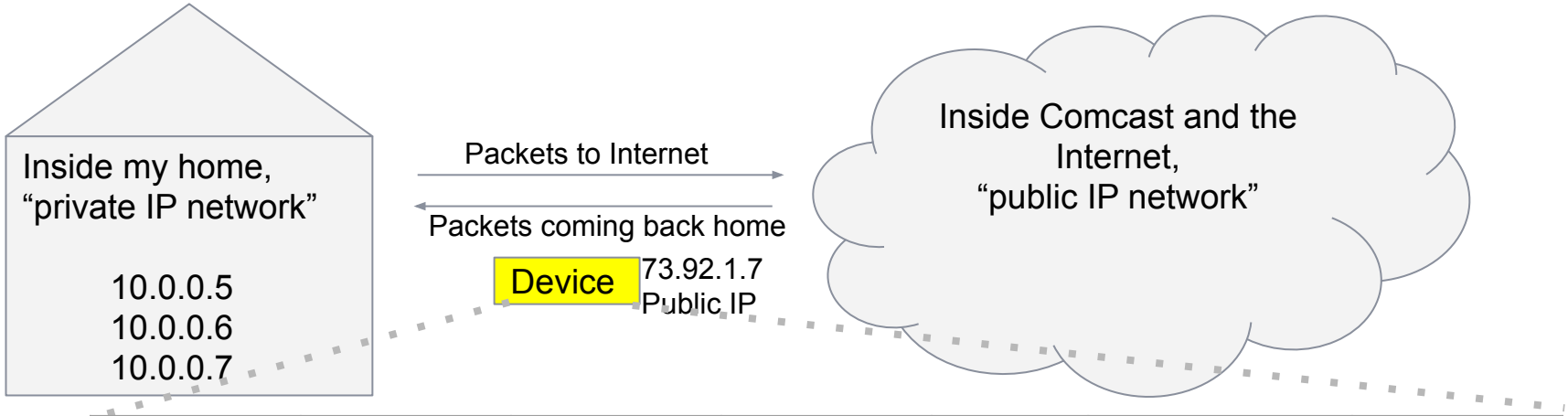
# ifconfig on my laptop at home

```
ocb-mbp:~ ocb$ ifconfig
lo0: flags=8049<UP,LOOPBACK,RUNNING,MULTICAST> mtu 16384
     options=1203<RXCSUM,TXCSUM,TXSTATUS,SW_TIMESTAMP>
     inet 127.0.0.1 netmask 0xff000000
     inet6 ::1 prefixlen 128
     inet6 fe80::1%lo0 prefixlen 64 scopeid 0x1
     nd6 options=201<PERFORMNUD,DAD>

….

en0: flags=8863<UP,BROADCAST,SMART,RUNNING,SIMPLEX,MULTICAST> mtu 1500
     ether 8c:85:90:95:15:4a
     inet6 fe80::14b2:9162:5553:8b72%en0 prefixlen 64 secured scopeid 0x8
     inet 10.0.0.7 netmask 0xffffff00 broadcast 10.0.0.255
     inet6 2601:647:5a00:6510:c0f:3811:351b:5c4d prefixlen 64 autoconf secured
     nd6 options=201<PERFORMNUD,DAD>
     media: autoselect
     status: active
```

# Private in home, public in Internet



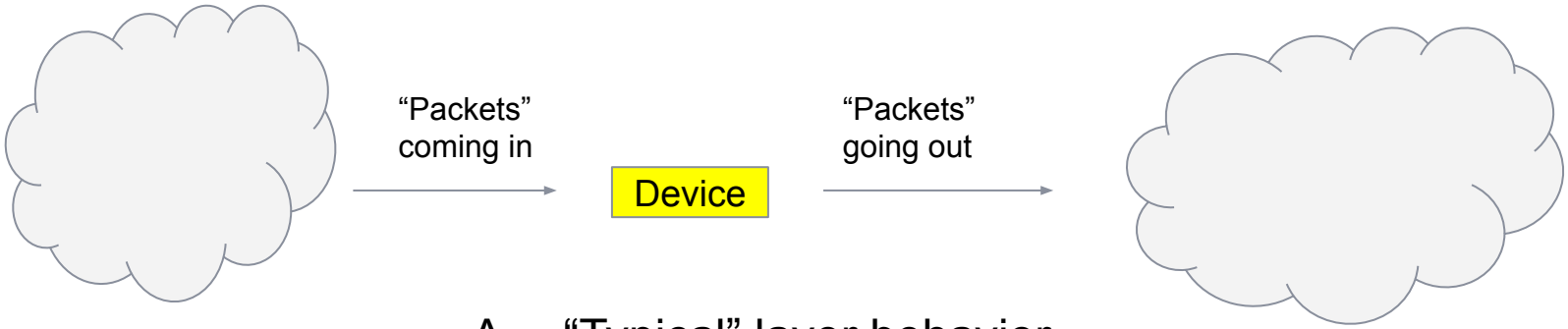
# Translation table between private and public




Address & port **translation** table

| Original Source IP | Original Source Port | New Source IP | New Source Port | Protocol | Destination IP | Destination Port |
|---|---|---|---|---|---|---|
| 10.0.0.5 | 53323 | 73.92.1.7 | 45584 | TCP | 157.240.22.35 | 80 |
| 10.0.0.5 | 43023 | 73.92.1.7 | 9489 | TCP | 157.240.22.174 | 80 |
| 10.0.0.7 | 35803 | 73.92.1.7 | 49348 | TCP | 69.171.250.54 | 80 |

# Changing the packet



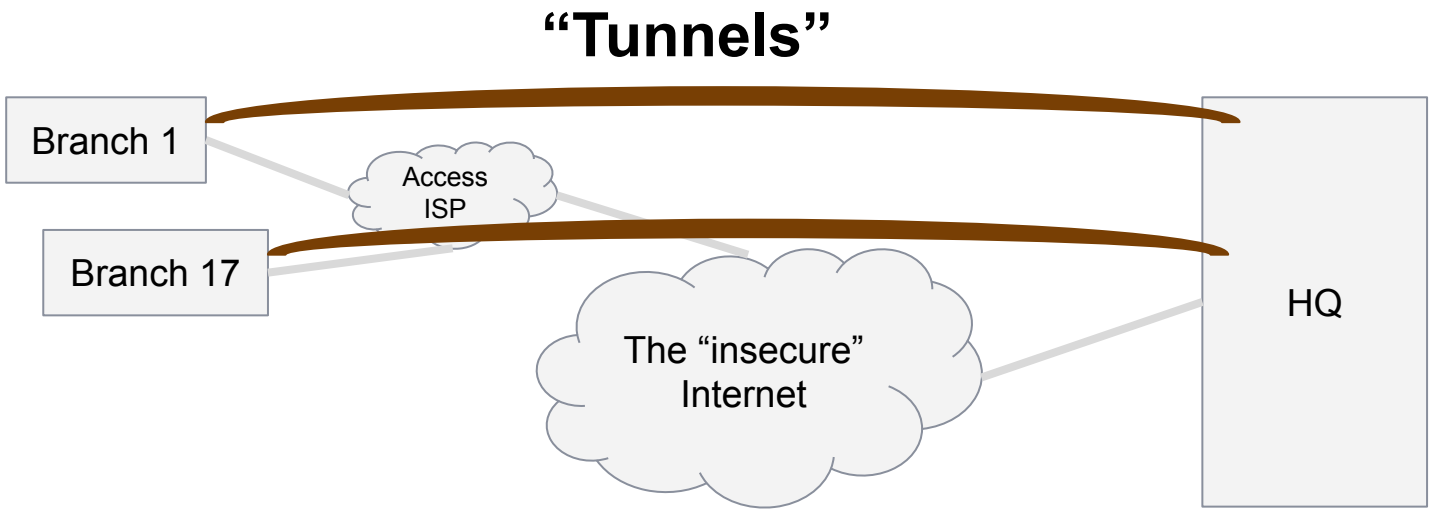

A. “Typical” layer behavior
B. Translation

# Ex 2: Virtual Private Network (VPN)

# Ex 2: Virtual Private Networks (VPNs) in mid 90s

Use case:

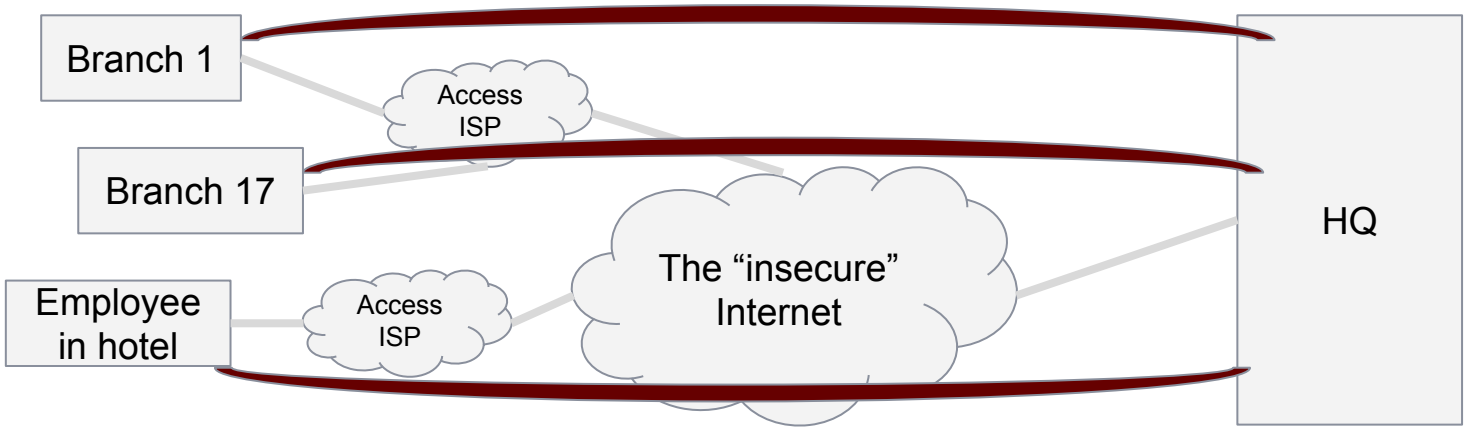
- Companies have “branches” (banks, sales offices) that want to connect to headquarters over Internet



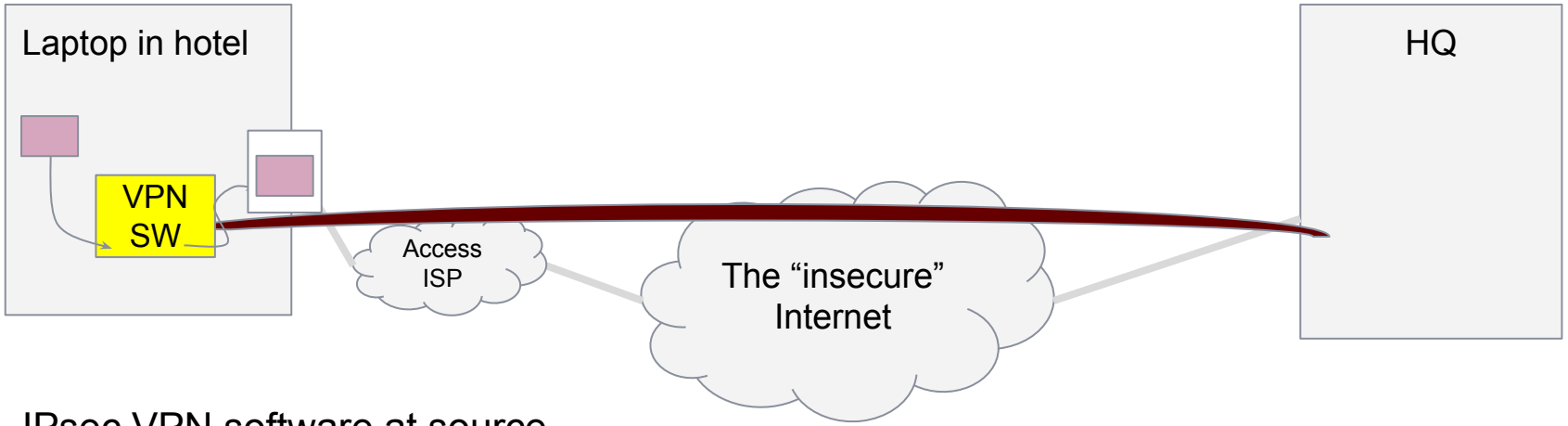
# Ex 2: Virtual Private Networks (VPNs) in mid 90s

Use case:

- Companies have “branches” (banks, sales offices) that want to connect to headquarters over Internet
- Connect from public network (like a hotel)



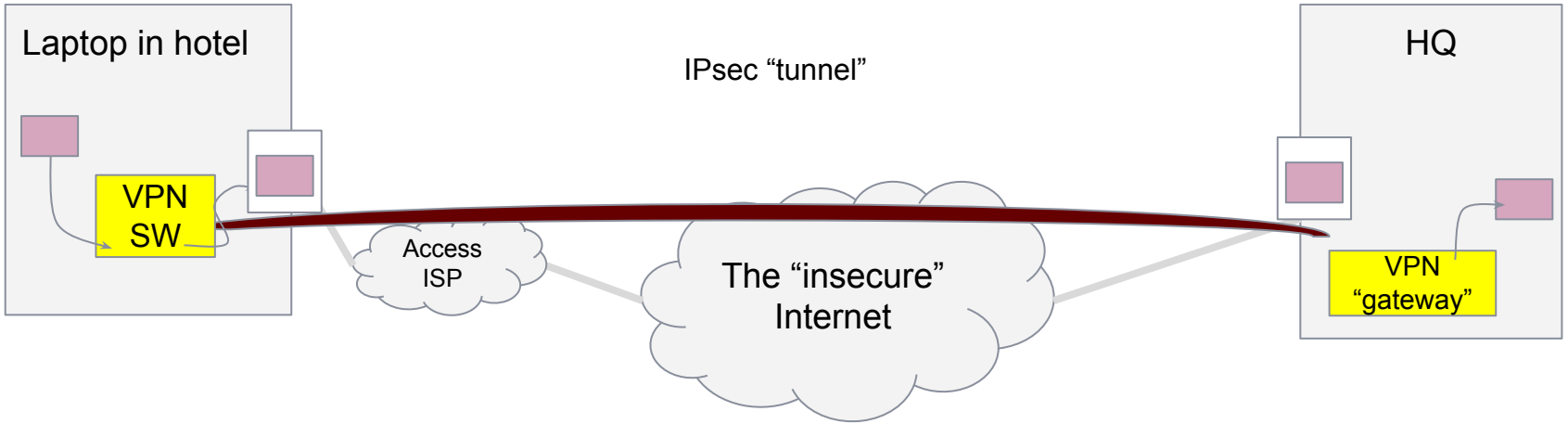
# How a “tunnel” works - encapsulation




IPsec VPN software at source

- Creates new packet with “tunnel” IPs
- **Encapsulates** encrypted original IP packet as payload in new packet
- Sends it out to destination IP tunnel endpoint

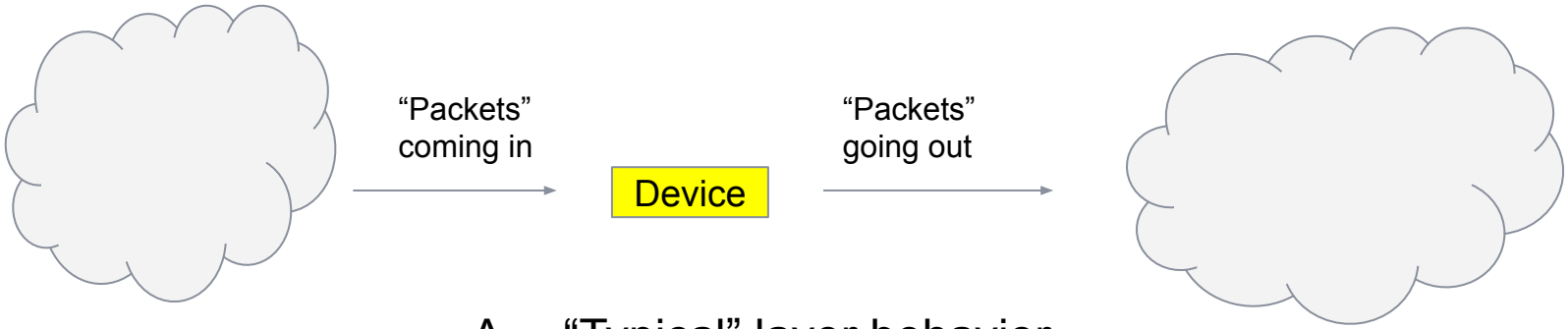
# How a “tunnel” works - de-encapsulation




IPsec VPN gateway at destination

- Receives encapsulated packet
- **Deencapsulates** - removes outer IP header
- Unpacks the payload and decrypts
- Sends it along into HQ

# Changing the packet

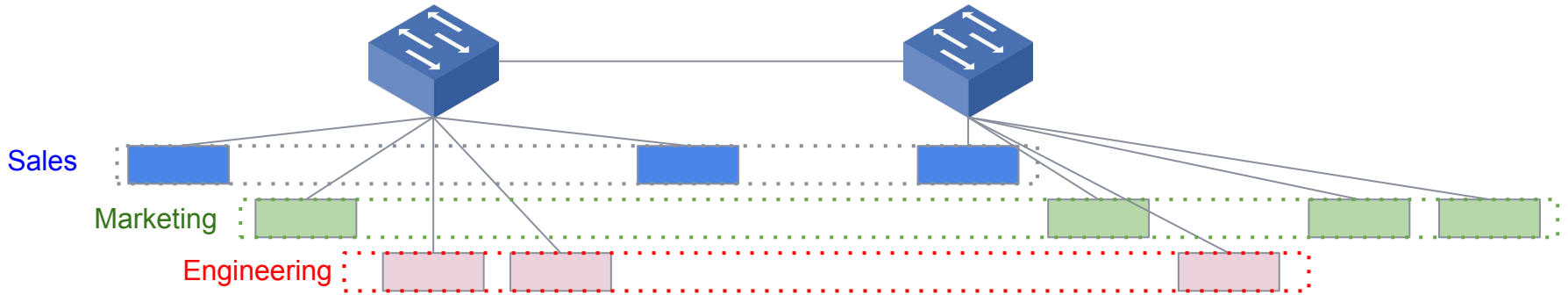



A. “Typical” layer behavior
B. Translation
C. Tunnels

# Ex 3: Virtual LANs (VLANs)

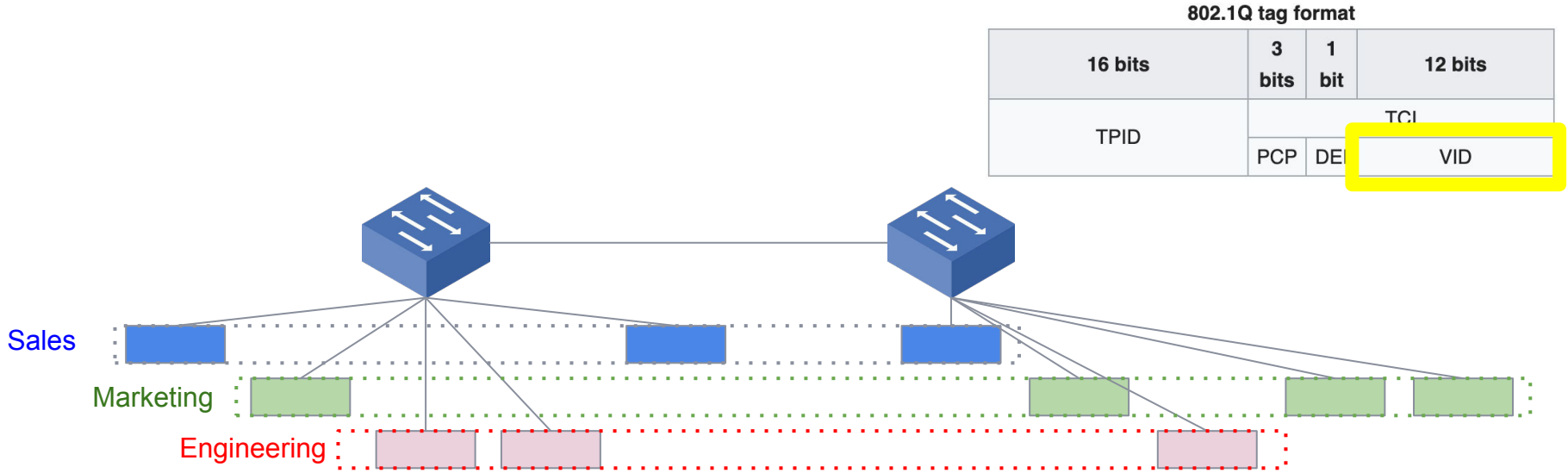
# Ex 3: from late 90s/early 00s

- “Ethernets have a lot of traffic now - wasn’t so bad with just email...”
  - Recall CSMA/CD class
- Too much broadcast in a big IP subnet
  - But without one big IP subnet, how to span multiple devices?
- Introduced a “tag” in header to create a virtual LAN (layer 2)

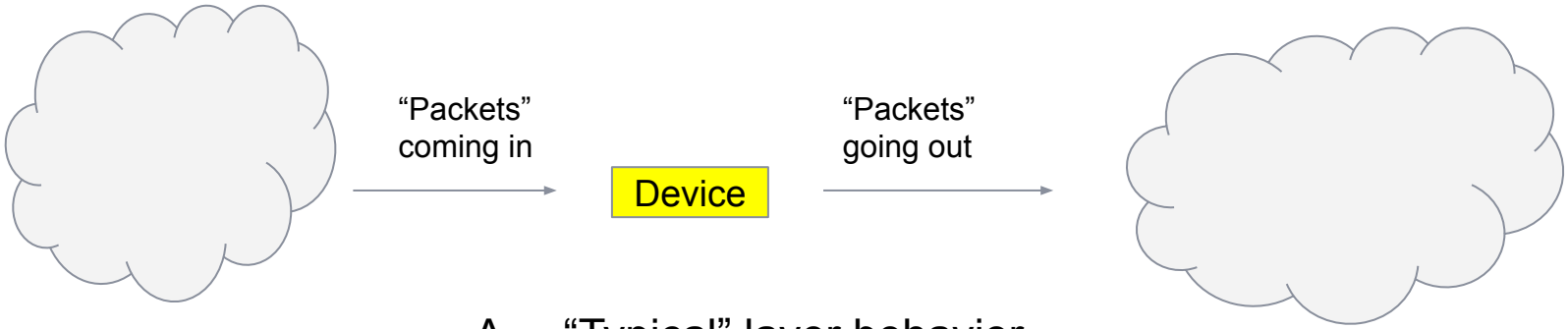


# Pros and cons

- Pros: super-easy to configure (don't worry about subnets, routing, …)
  - Lots of people want L2 data centers
- Cons: 12 bits ~ 4K networks



# Networking device - ins and outs




A. “Typical” layer behavior
B. Translation
C. Tunnels
D. Tagging

# Lessons (from mid 2000s)

- Disparate tools in a toolbox
- Hard to implement compatible standards and technologies
- Hard to build “networks” with thousands of endpoints, and hundreds of thousands of tunnels

```
You are in a maze of little twisty
passages, all different.
```

# Setting the stage - some trends

- Data centers @scale
- Efficient use of resources, even inside a company
- Rise of hosting/cloud providers mid-late 2000s
- Server virtualization (VMware, ...) - orders of magnitude more VMs, containers to address
- SDN - centralized control/mgmt software

# State-of-the-art network virtualization

Allow complete virtual networks (“overlays”)
on top of a shared physical network (“underlay”)

Seen in clouds (AMZN, MSFT, GOOG, BABA, ORCL, …)
and enterprise-solutions from VMware, Citrix, ...

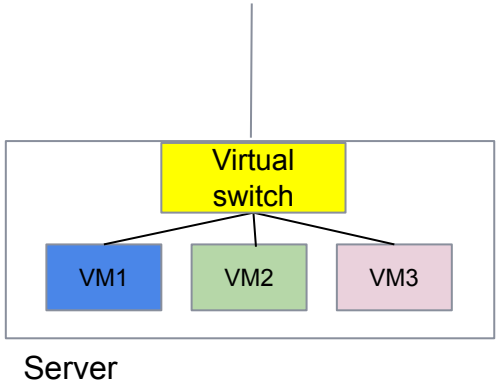
# Network virtualization - basic requirements

- Multi-tenancy - customer's VMs can connect only to their VMs *and no one else's* (isolation)

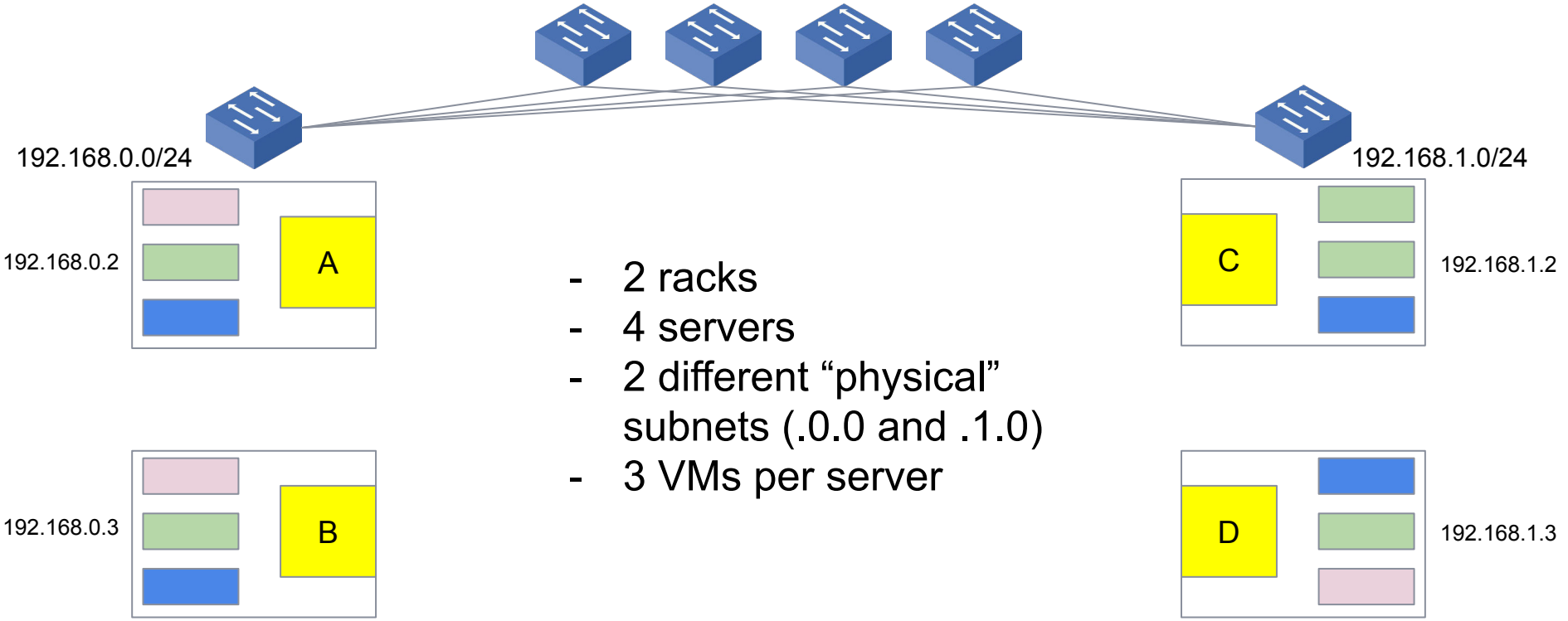
- Both virtual addressing and virtual topologies, independent of physical location/topology

- Operate @scale - easy to turn up, extend, operate, turn down networks of VMs

# Building block: virtual switch on a host

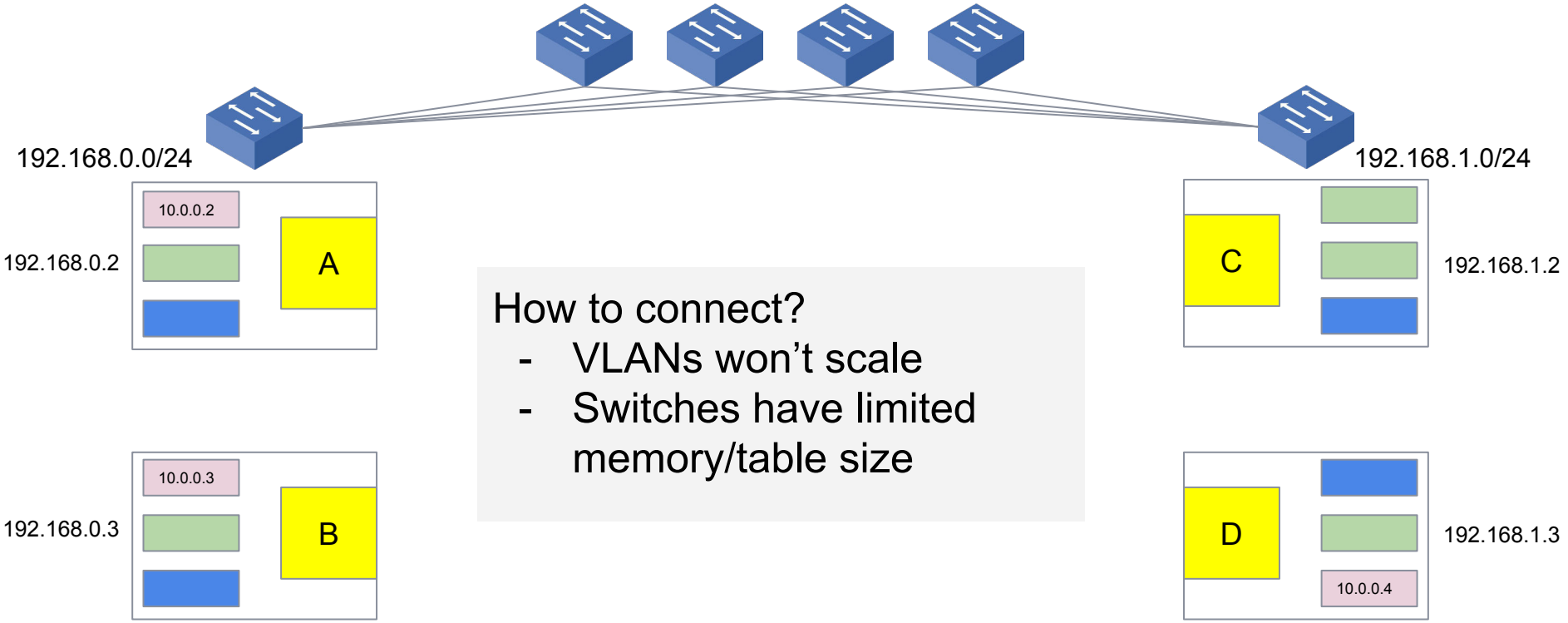


# Physical “underlay” + VMs




- 2 racks
- 4 servers
- 2 different “physical” subnets (.0.0 and .1.0)
- 3 VMs per server

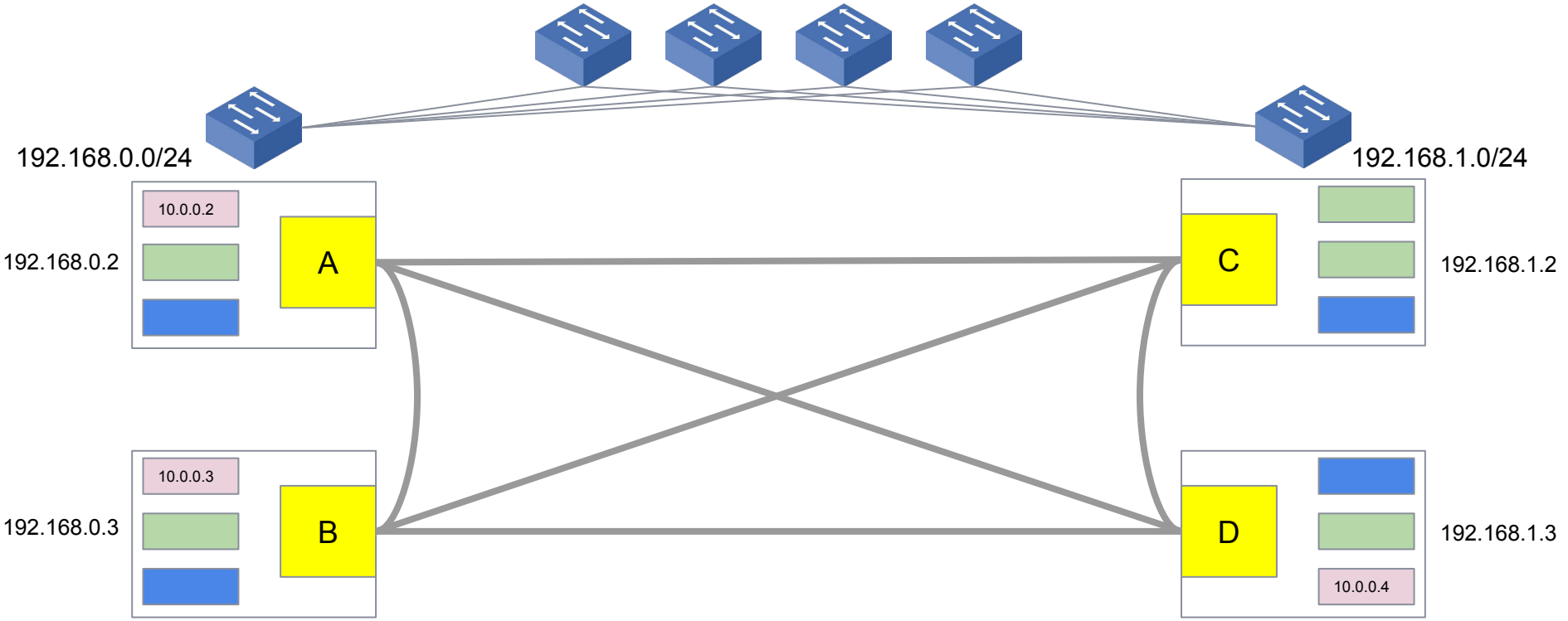
# Ex 1: Red VMs in same subnet



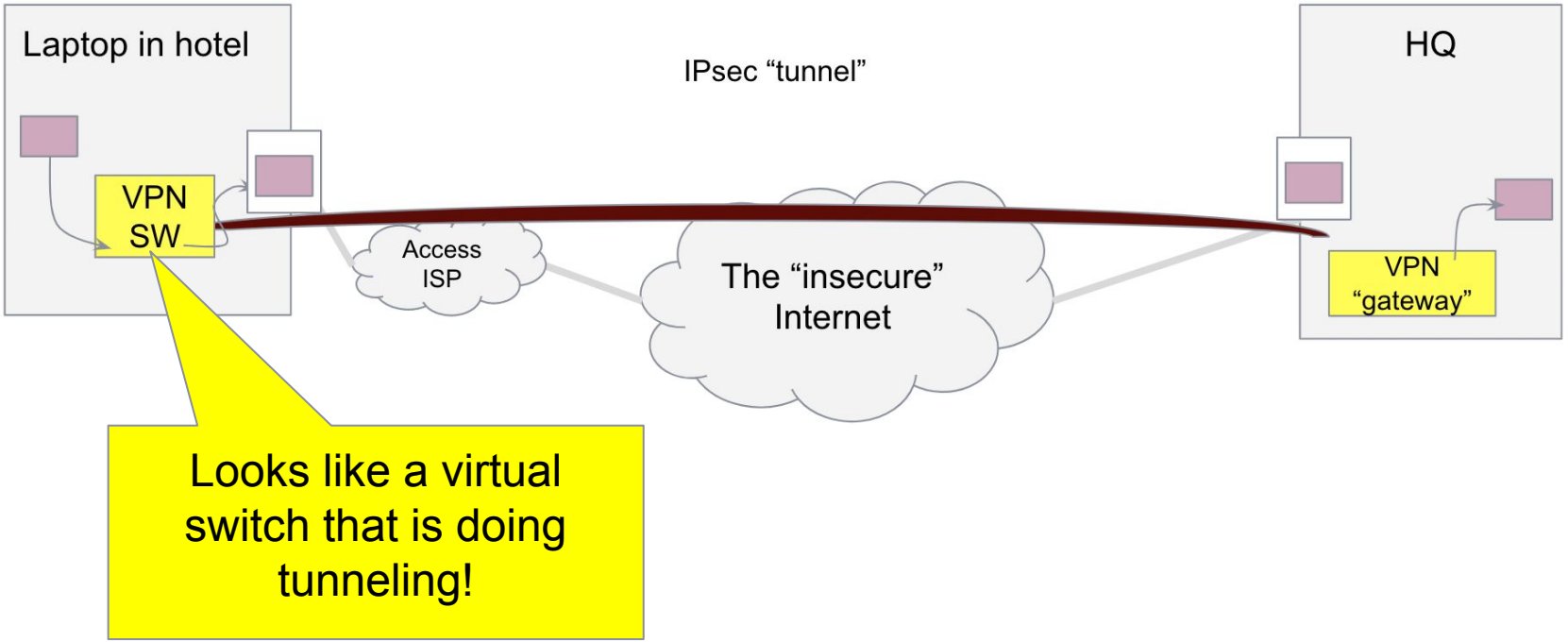

How to connect?

- VLANs won't scale
- Switches have limited memory/table size

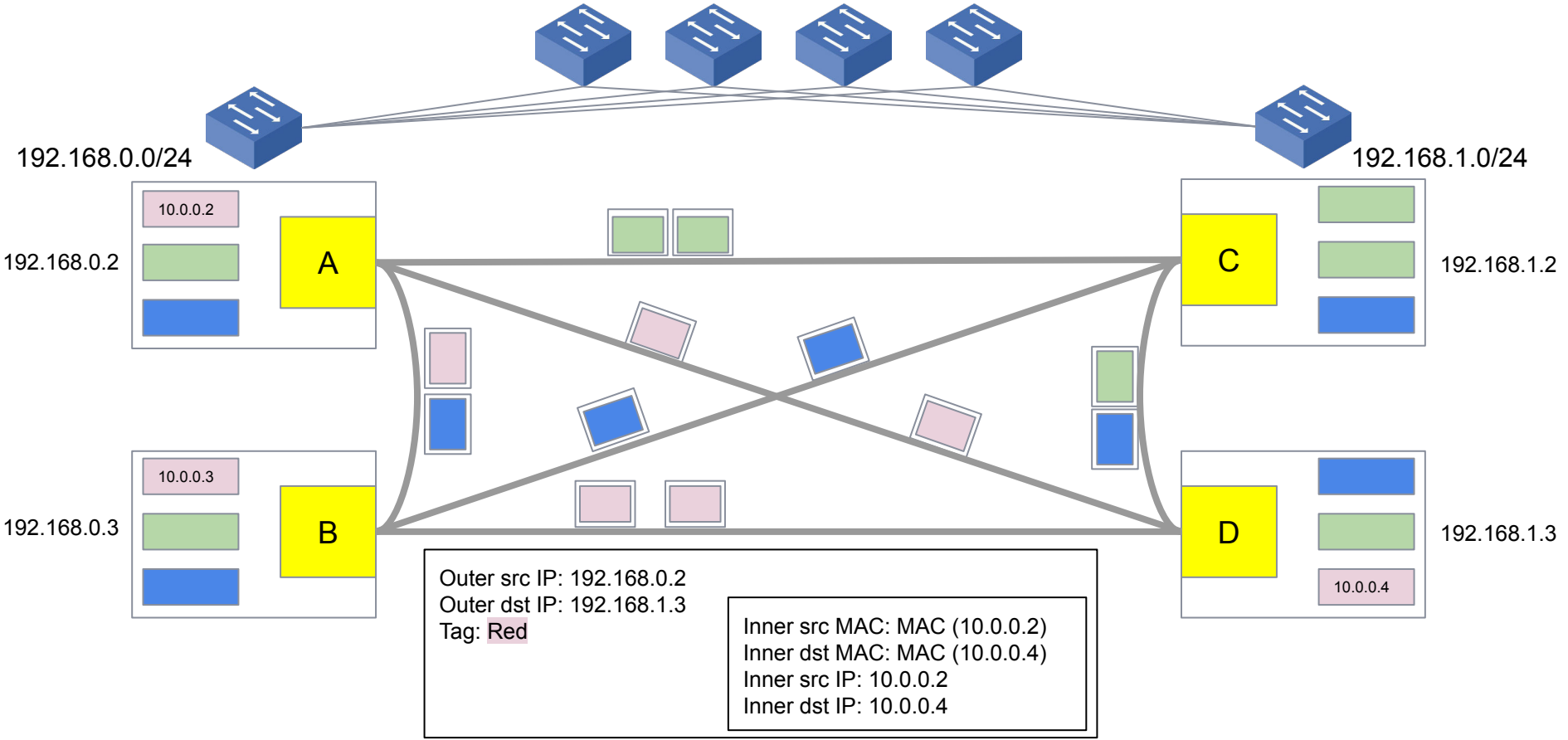
# Tunnels & tags to rescue!



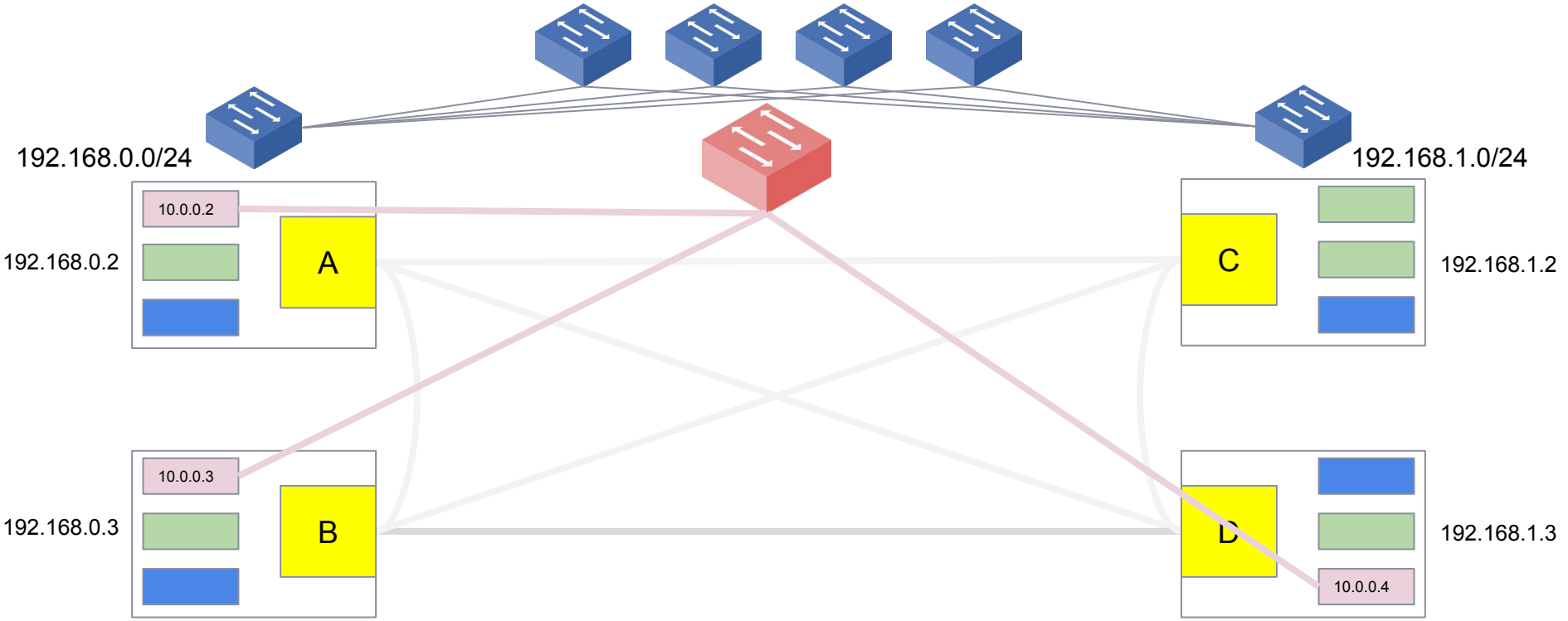
# Remember this picture?



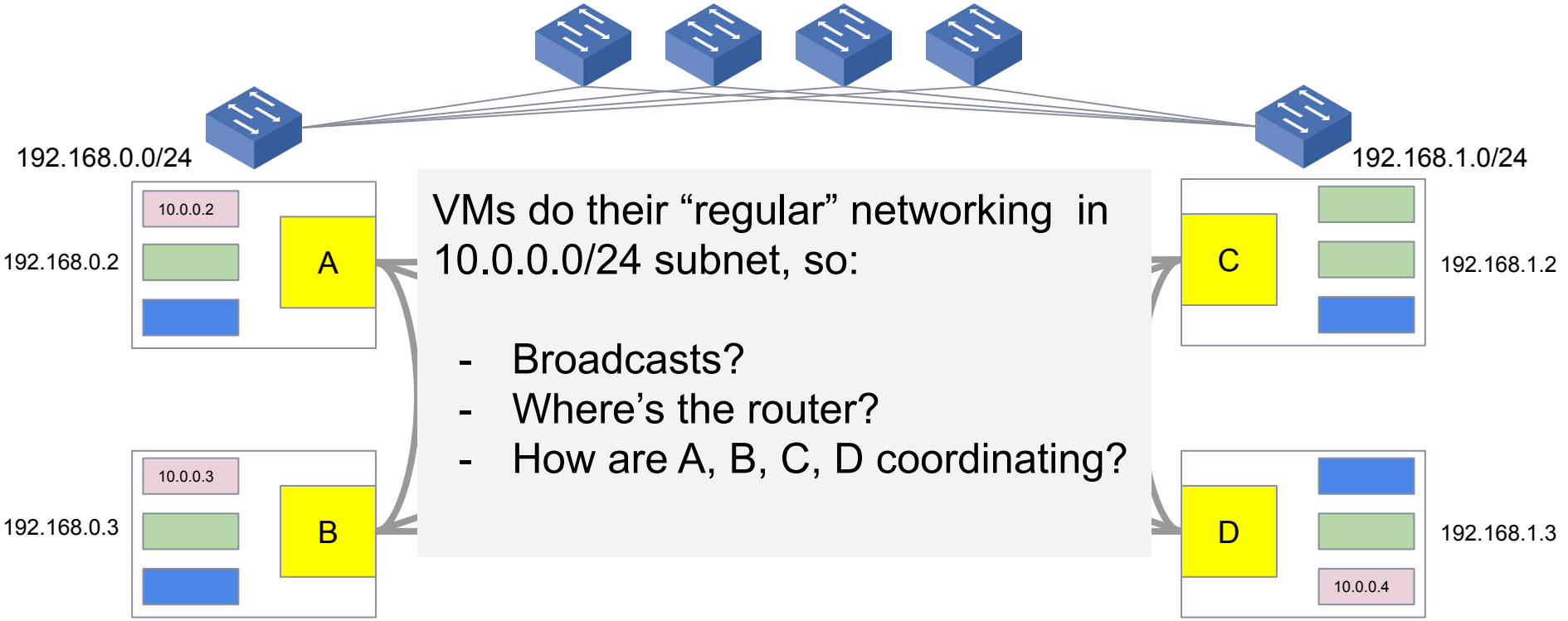
# Tunnels & tags to rescue!



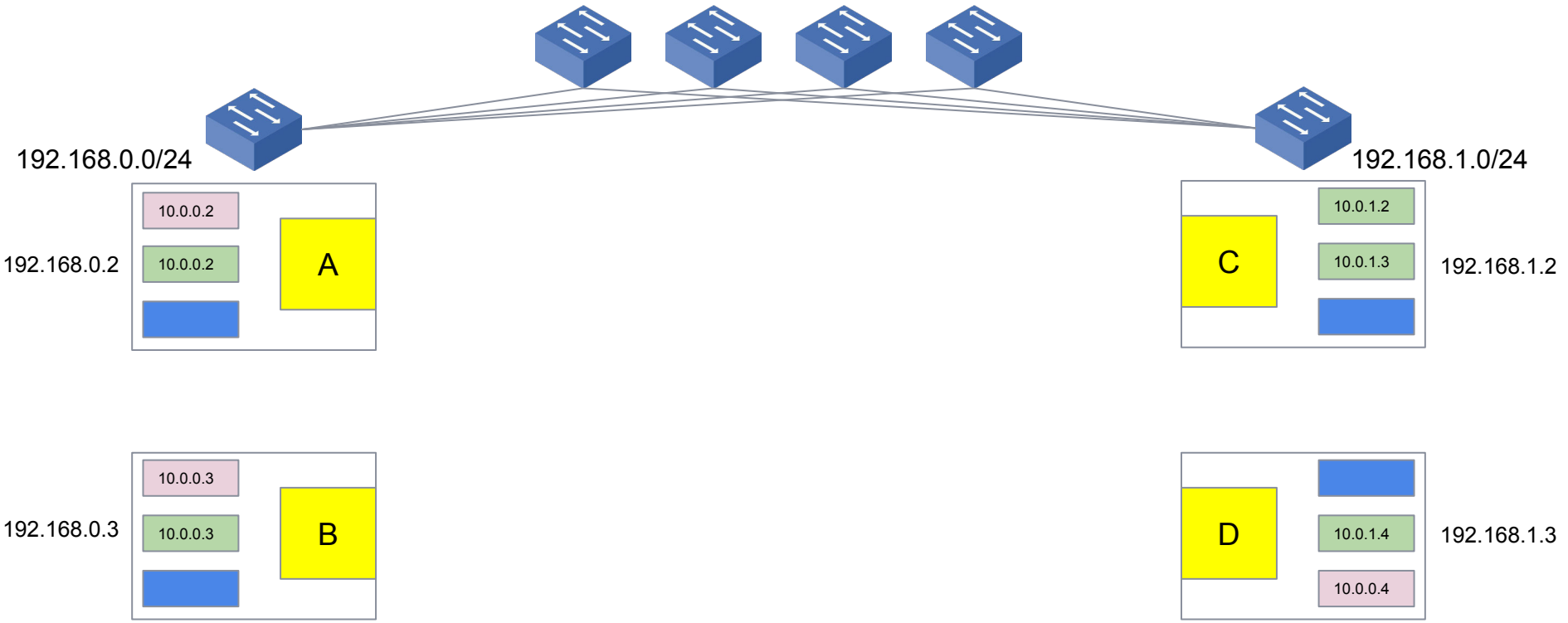
# Red VMs connected to a “logical” switch



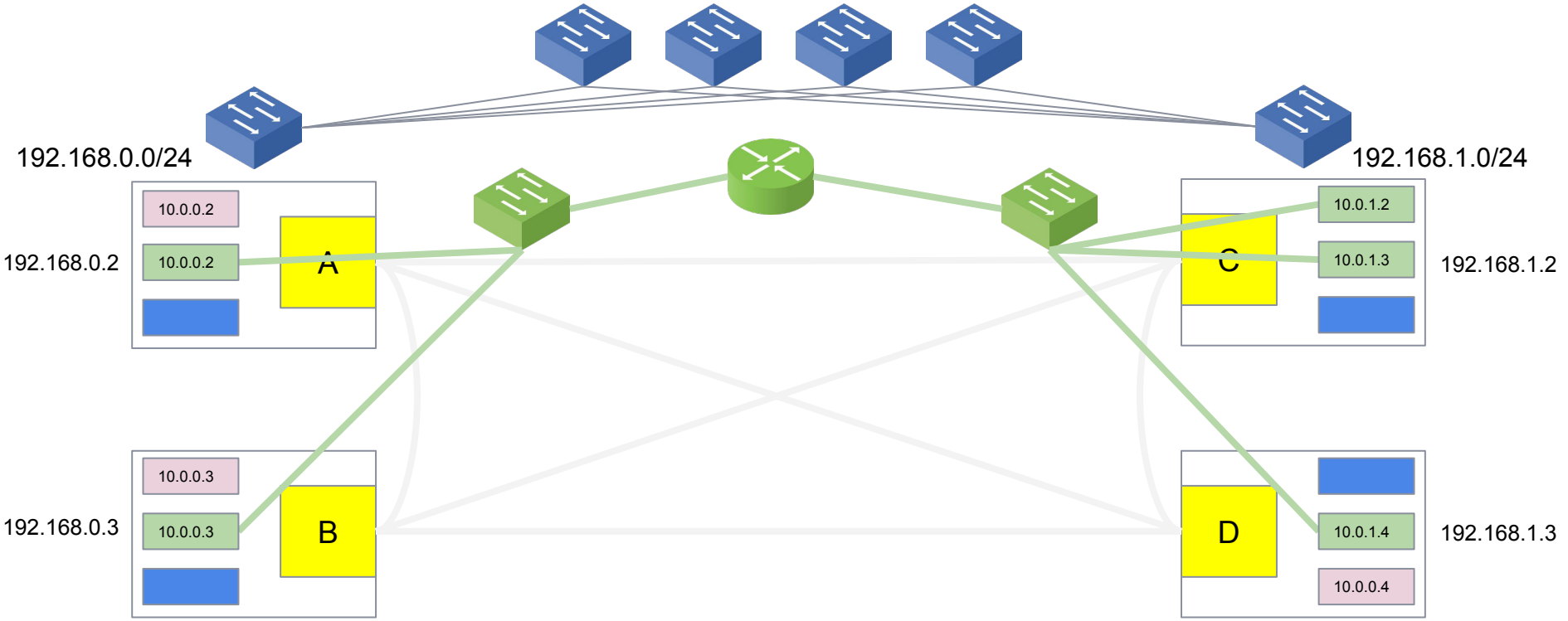
# But some questions...



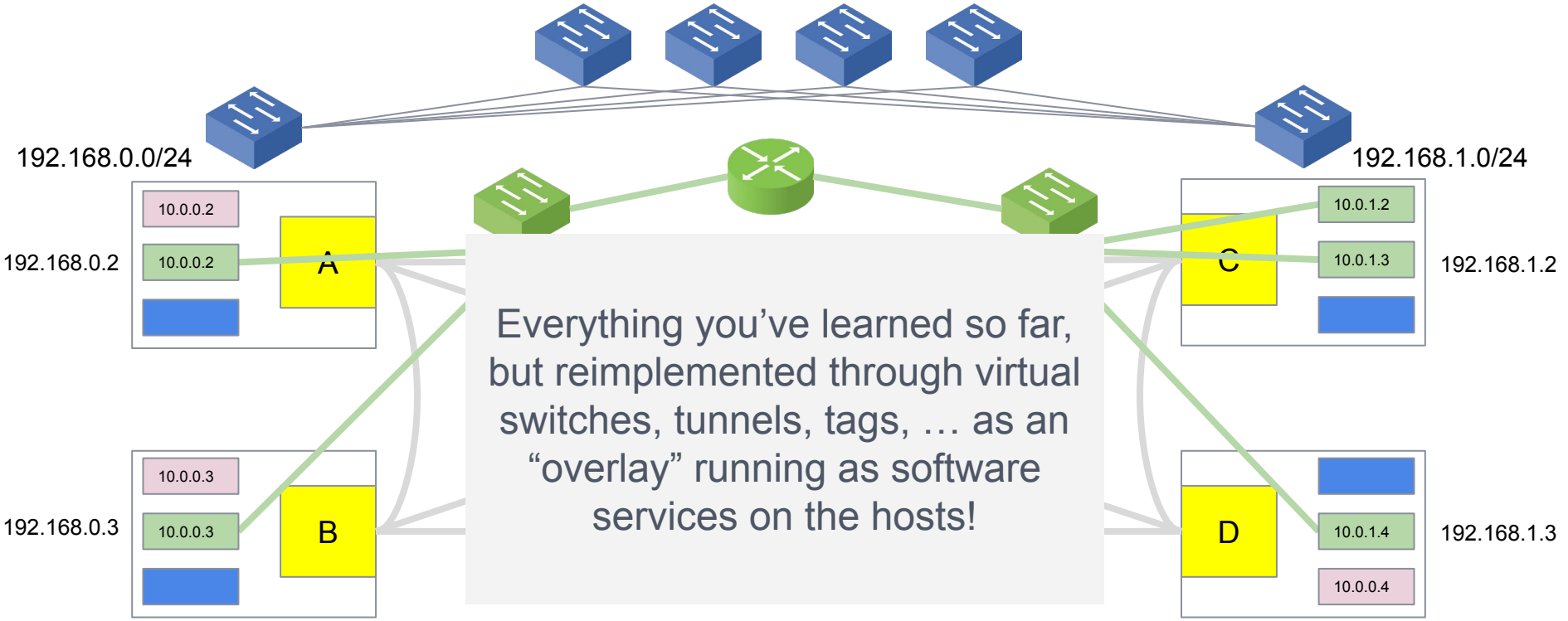
# Ex 2: Green VMs in different subnets



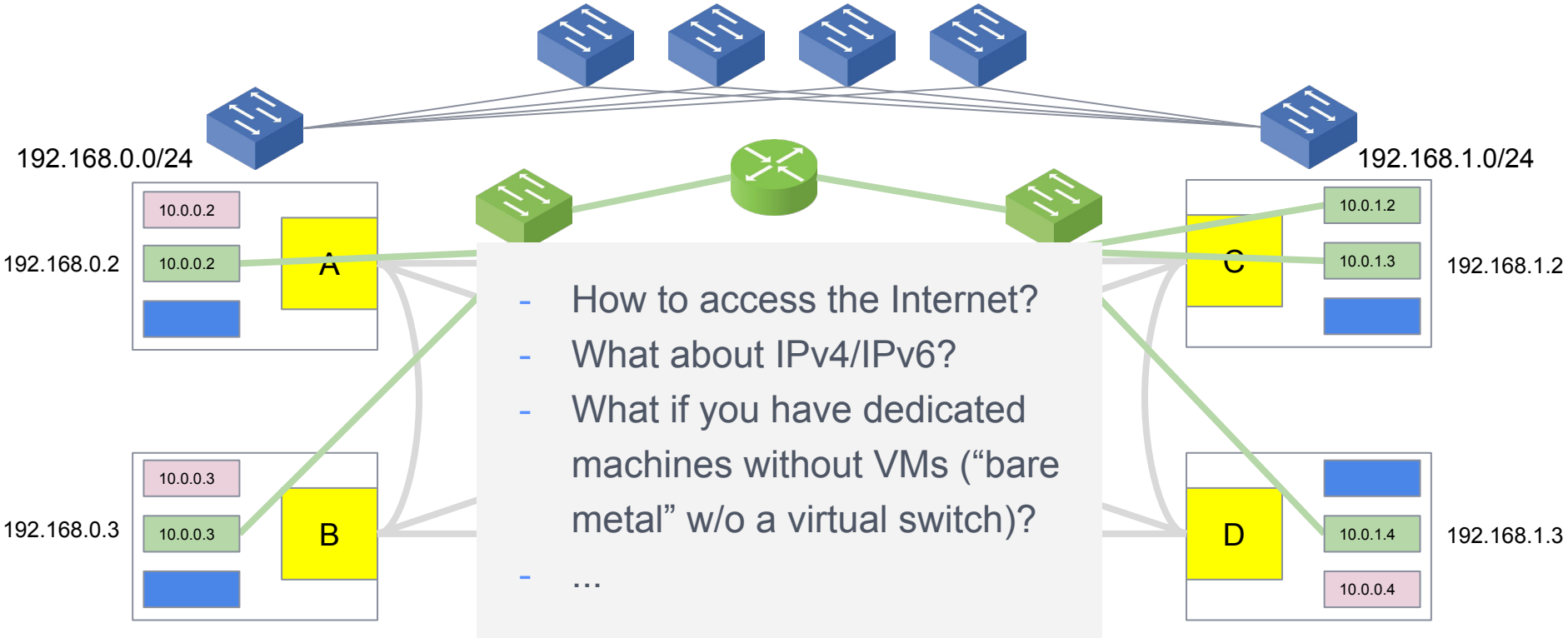
# Ex 2: logically, green switches and green router



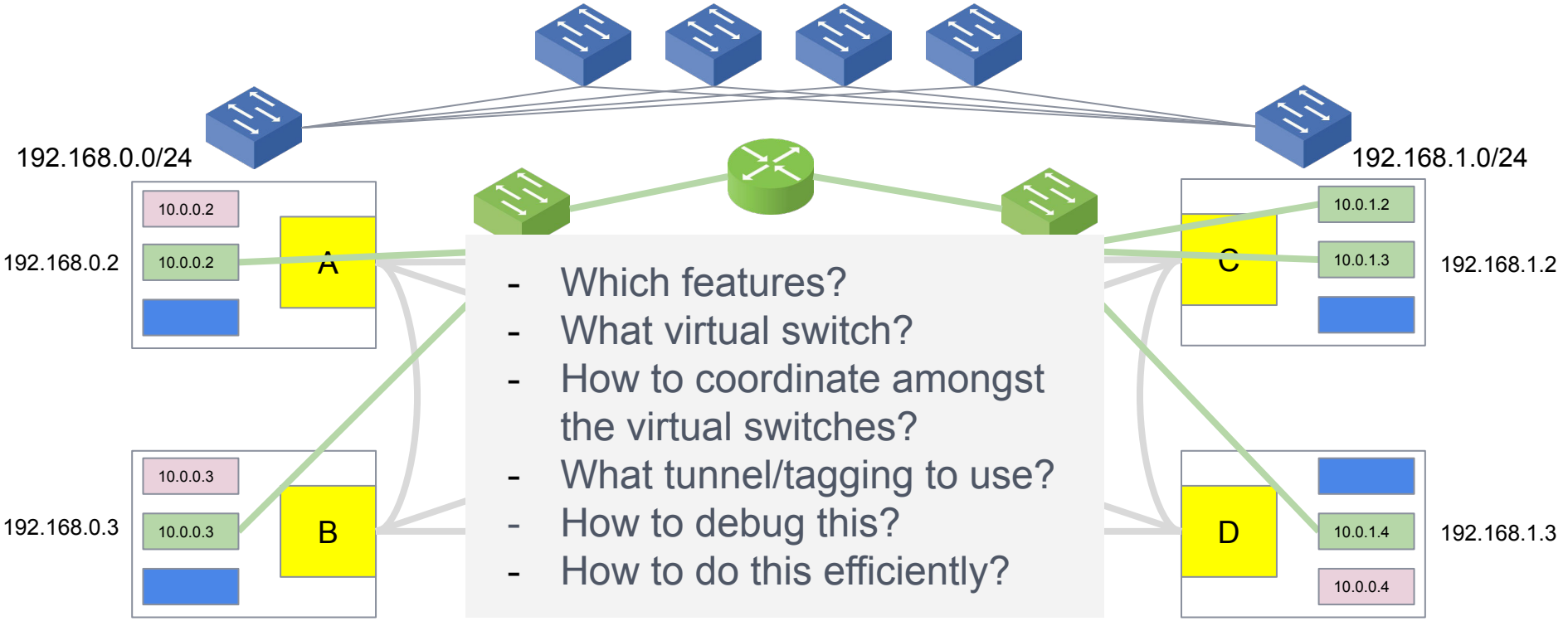
# Ex 2: logically, green switches and green router



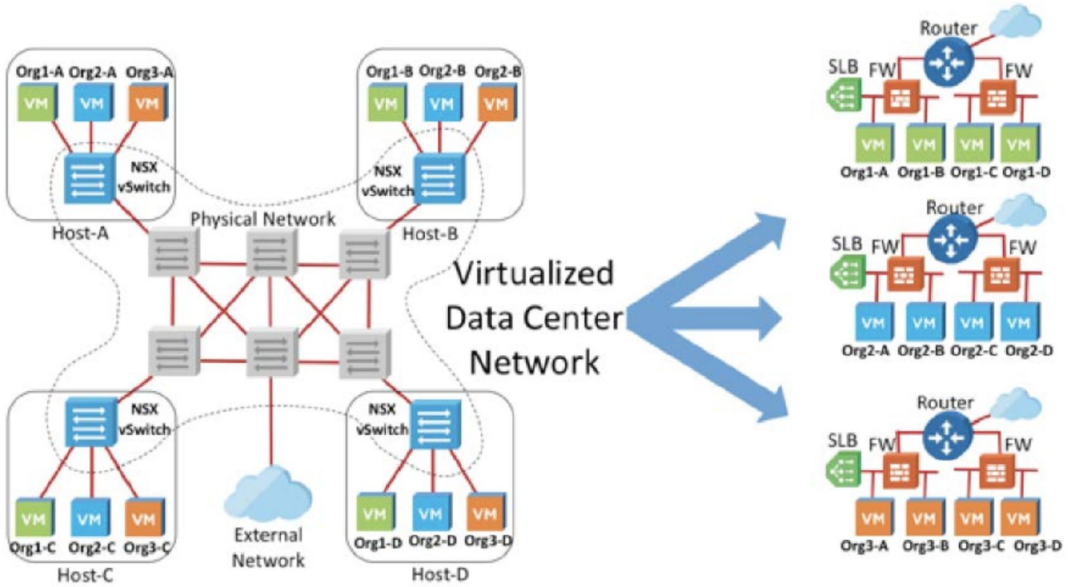
# More features left to provide...



# Every cloud has similar design choices



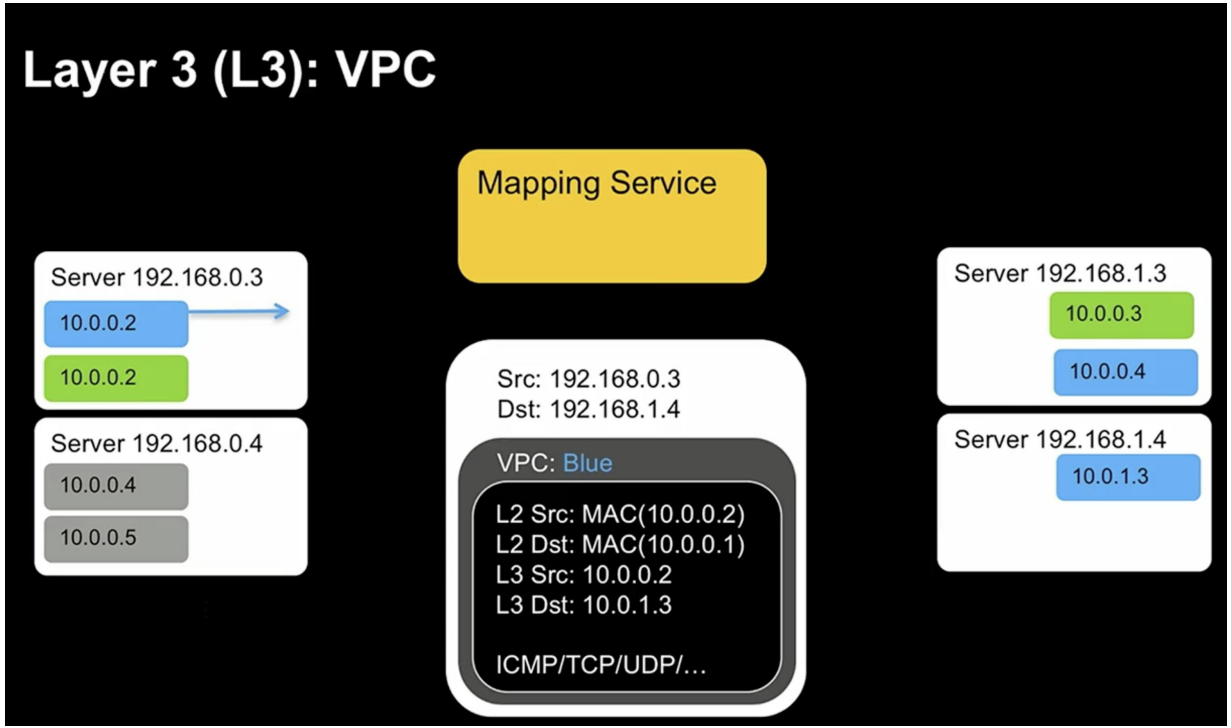
# Ex: VMware/Nicira NSX




Source: VMware NSX Network Virtualization Fundamentals,

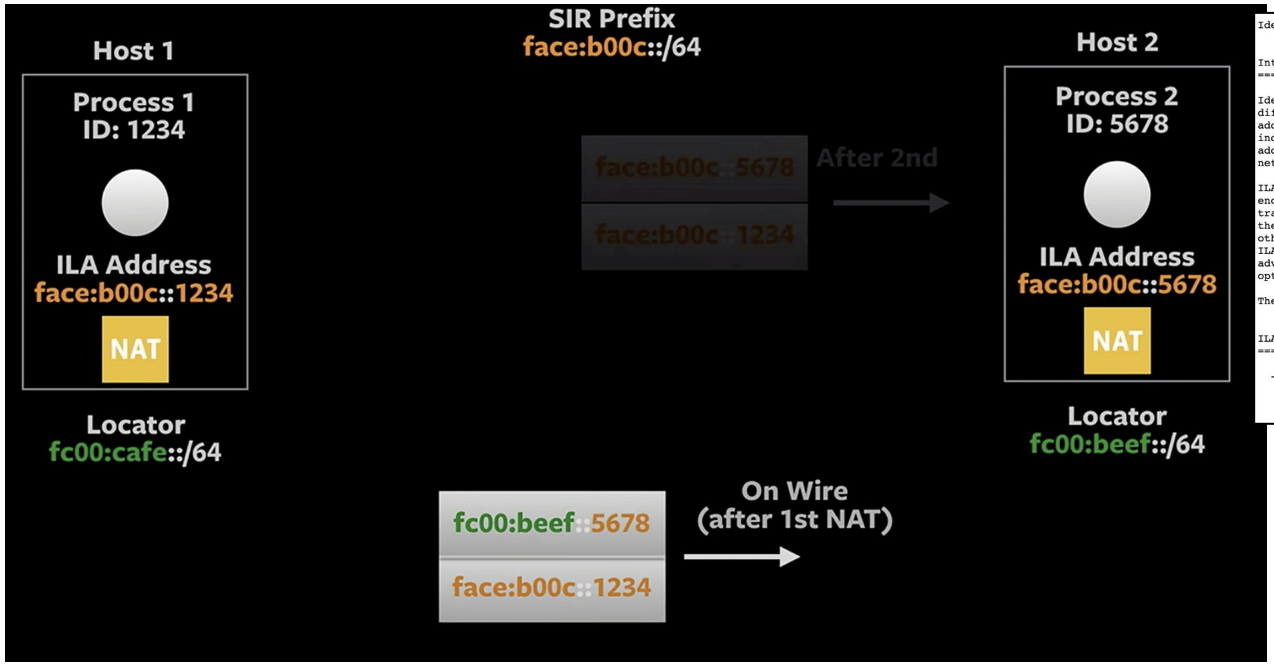
https://www.vmware.com/content/dam/digitalmarketing/vmware/en/pdf/products/nsx/vmware-network-virtualization-fundamentals-guide.pdf

# Ex: Amazon Virtual Private Cloud (VPC)




Source: Networking @Scale 2017 video from Amazon, https://engineering.fb.com/networking-traffic/networking-scale-2017-recap/

# Ex: Facebook & Identifier Locator Addressing (ILA) - containers + translation (instead of VMs & tunnels)



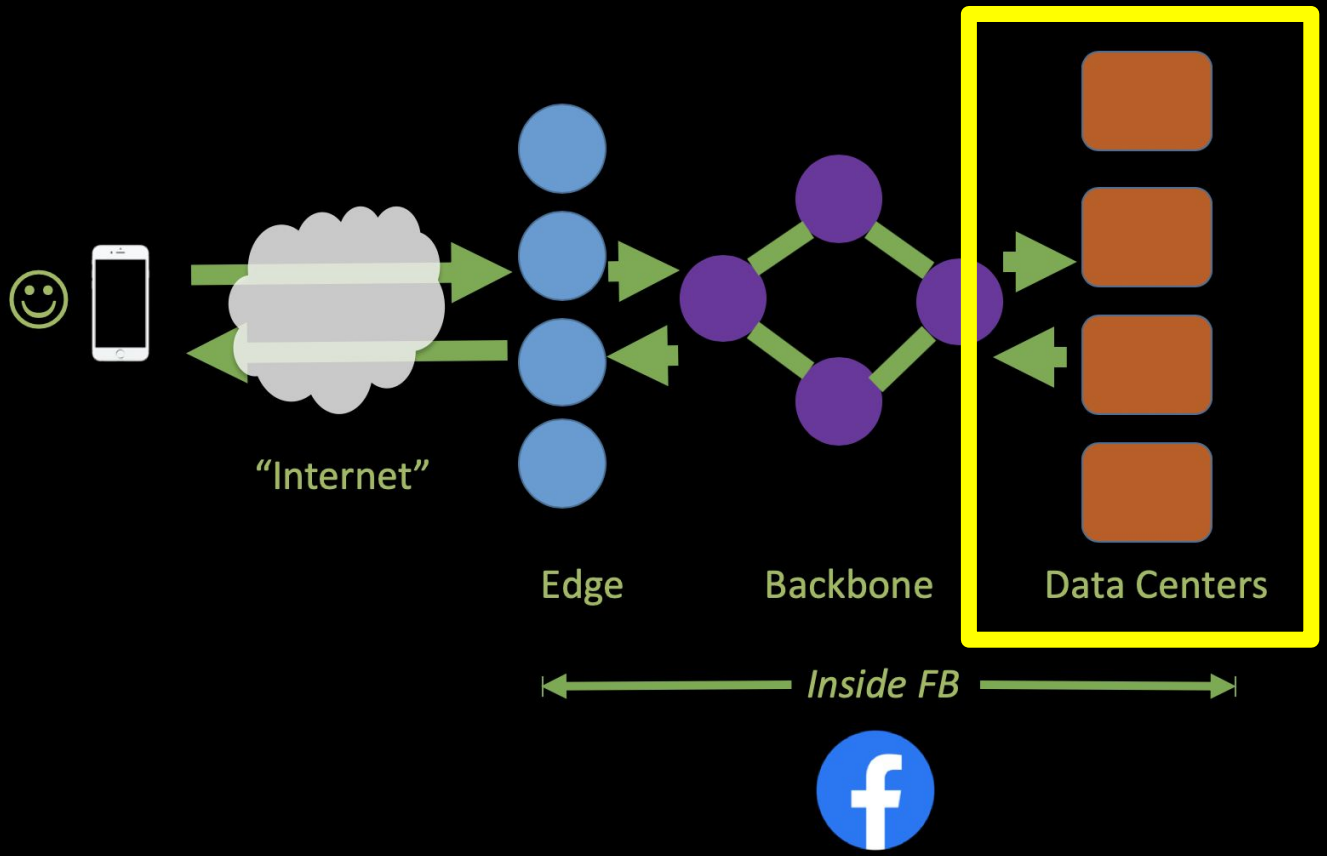

```
Identifier Locator Addressing (ILA)


Introduction
============

Identifier-locator addressing (ILA) is a technique used with IPv6 that
differentiates between location and identity of a network node. Part of an
address expresses the immutable identity of the node, and another part
indicates the location of the node which can be dynamic. Identifier-locator
addressing can be used to efficiently implement overlay networks for
network virtualization as well as solutions for use cases in mobility.

ILA can be thought of as means to implement an overlay network without
encapsulation. This is accomplished by performing network address
translation on destination addresses as a packet traverses a network. To
the network, an ILA translated packet appears to be no different than any
other IPv6 packet. For instance, if the transport protocol is TCP then an
ILA translated packet looks like just another TCP/IPv6 packet. The
advantage of this is that ILA is transparent to the network so that
optimizations in the network, such as ECMP, RSS, GRO, GSO, etc., just work.

The ILA protocol is described in Internet-Draft draft-herbert-intarea-ila.


ILA terminology
===============

  - Identifier  A number that identifies an addressable node in the network
                independent of its location. ILA identifiers are sixty-four
                bit values.
```

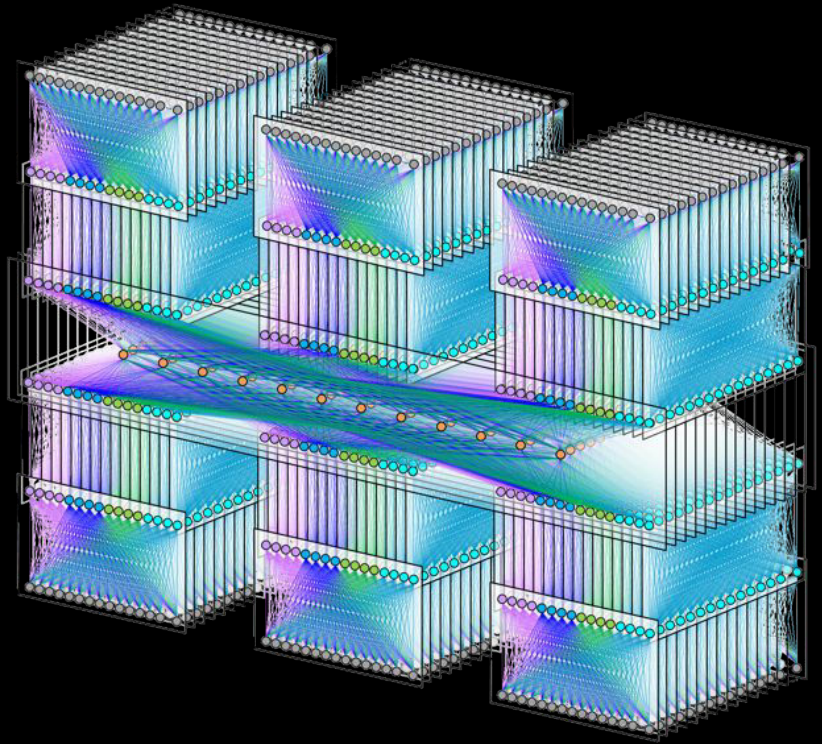
Source: Networking @Scale 2017 video from Facebook, https://engineering.fb.com/networking-traffic/networking-scale-2017-recap/

# Questions?

# Networking at Facebook


NETWORK INFRA


"Internet"
Edge
Backbone
Data Centers
Inside FB



NETWORK INFRA

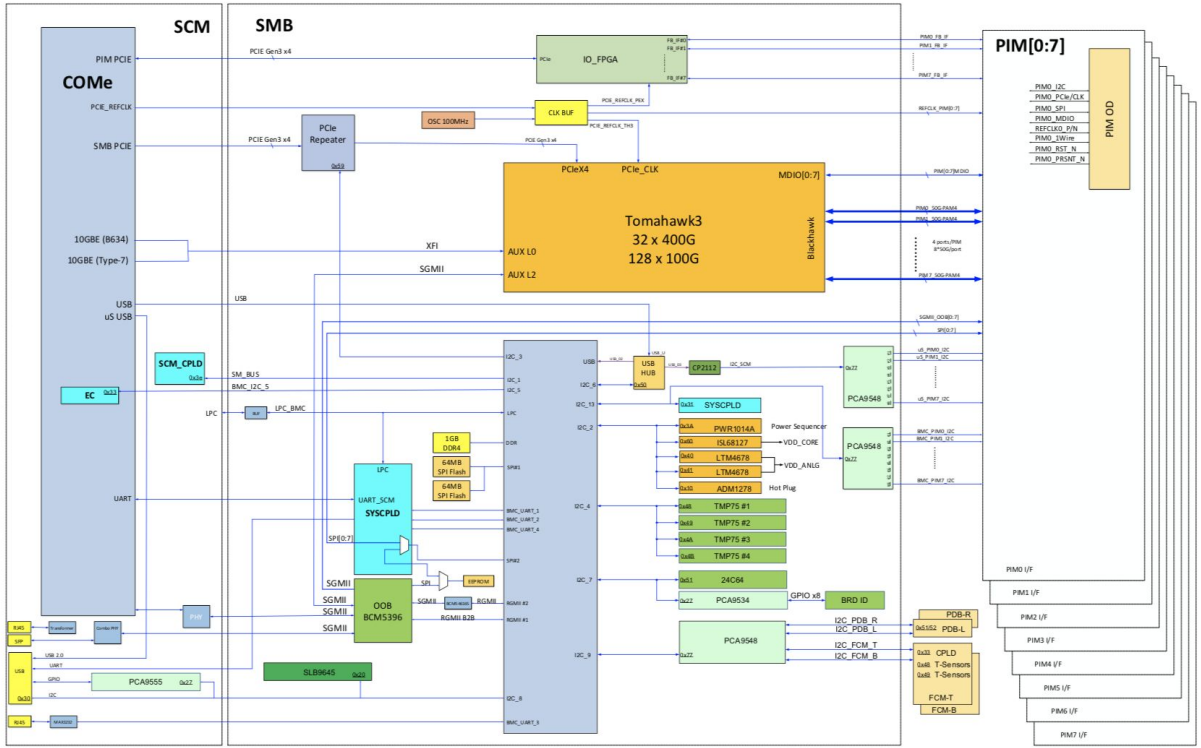
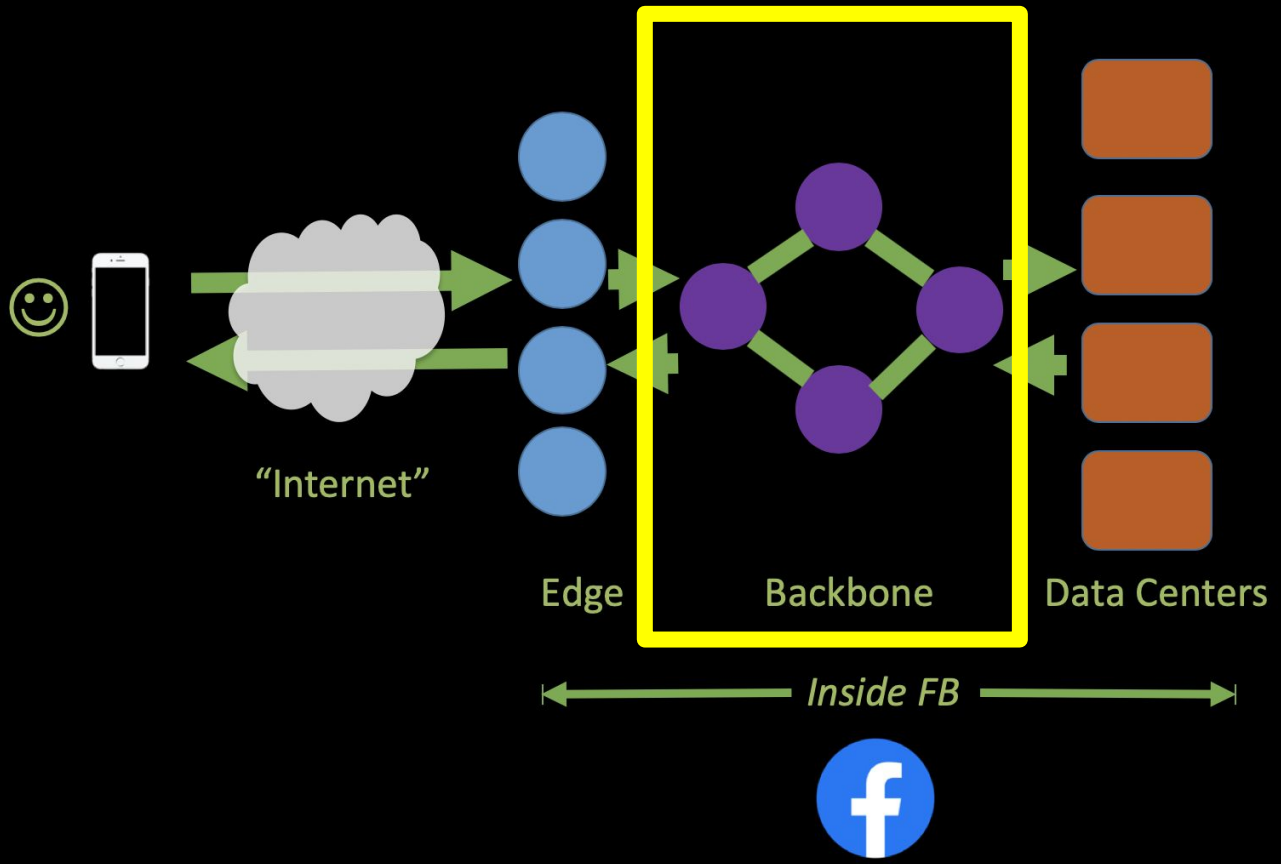
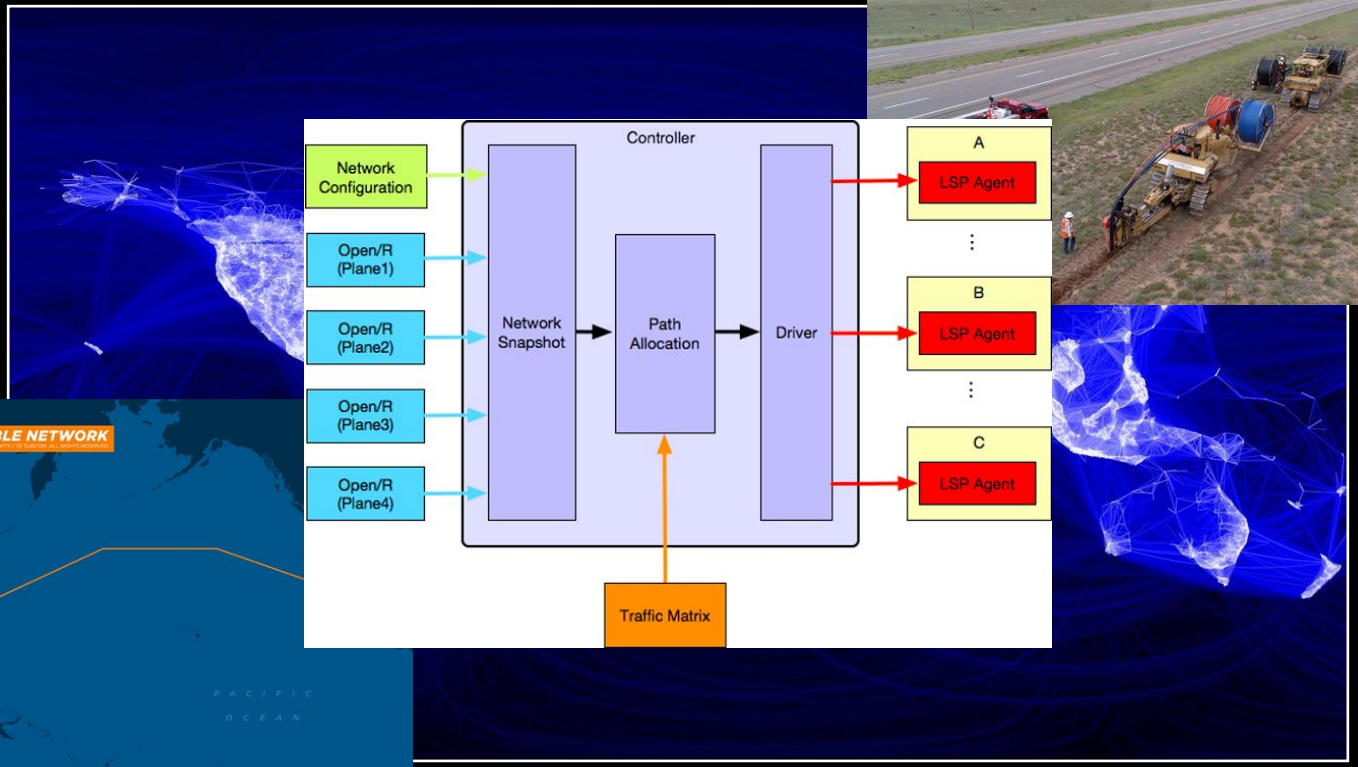


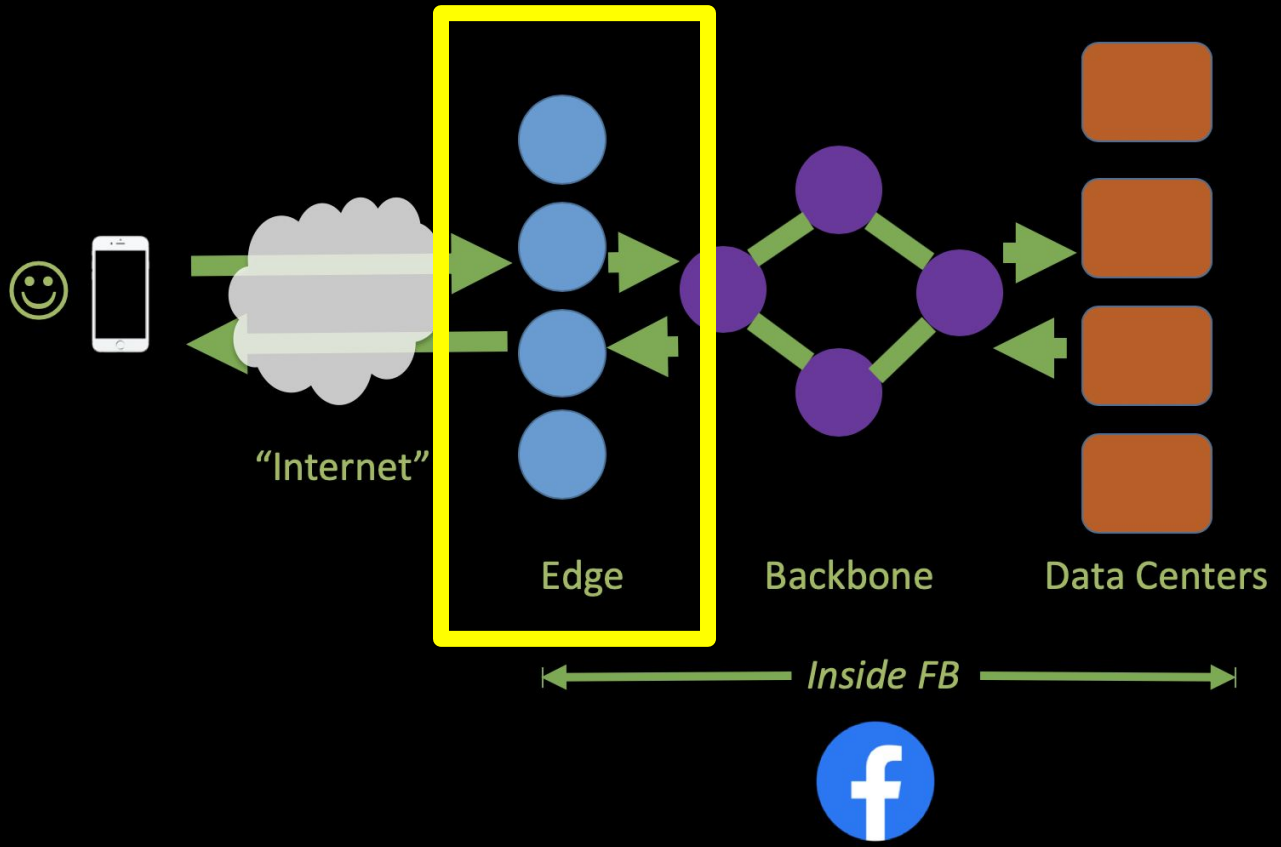

Figure 8-2: Switch Main Board Architecture




NETWORK INFRA




NETWORK INFRA


"Internet"
Edge
Backbone
Data Centers
Inside FB

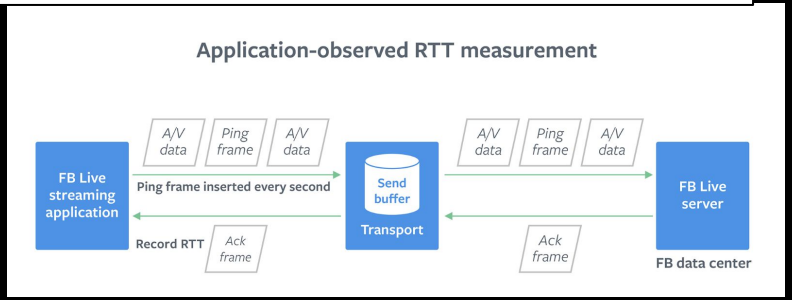
facebook research

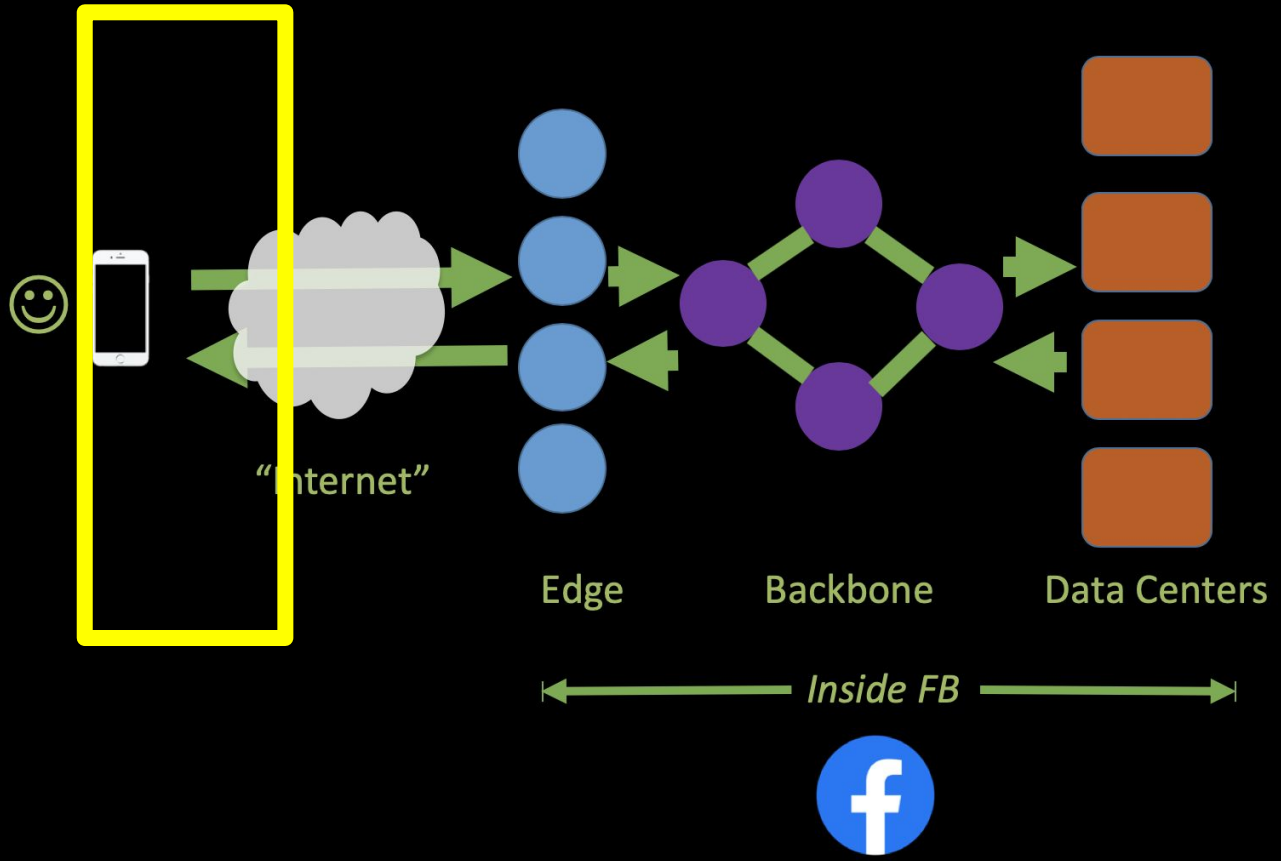
## Internet Performance from Facebook's Edge*

Brandon Schlinker[†♯] Italo Cunha[‡♮] Yi-Ching Chiu[†] Srikanth Sundaresan[♯] Ethan Katz-Bassett[♮]
[†] University of Southern California [♯] Facebook [‡] Universidade Federal de Minas Gerais [♮] Columbia University

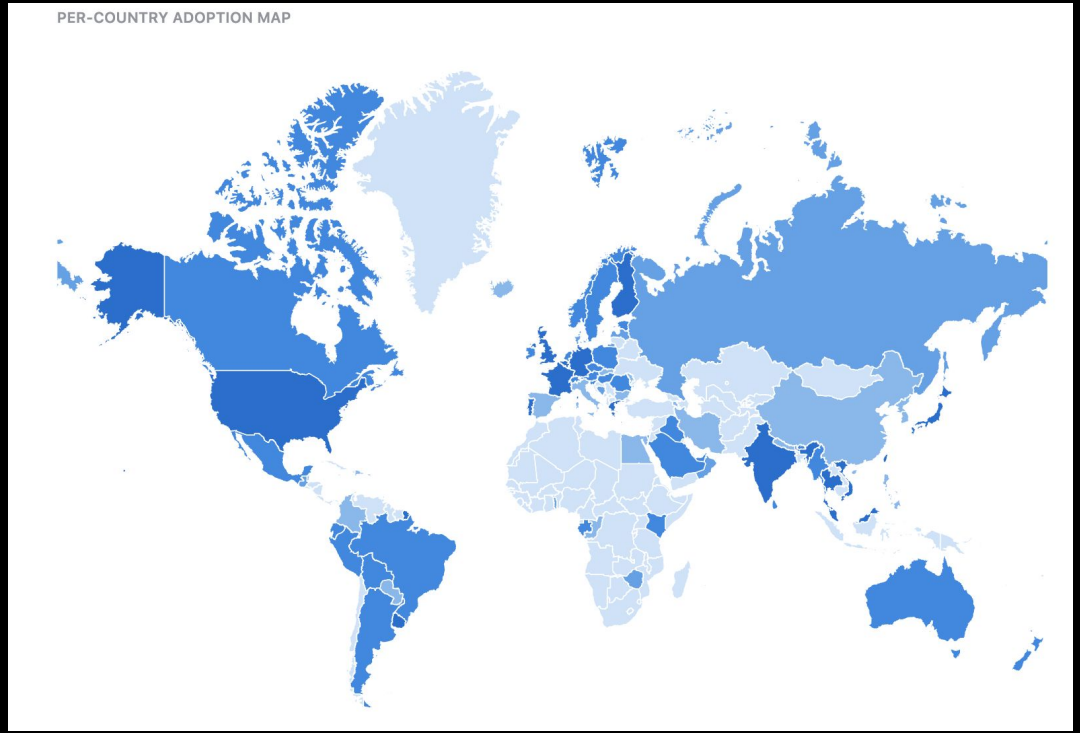
POSTED ON NOV 17, 2019 TO NETWORKING & TRAFFIC, VIDEO ENGINEERING

## Evaluating COPA congestion control for improved video performance



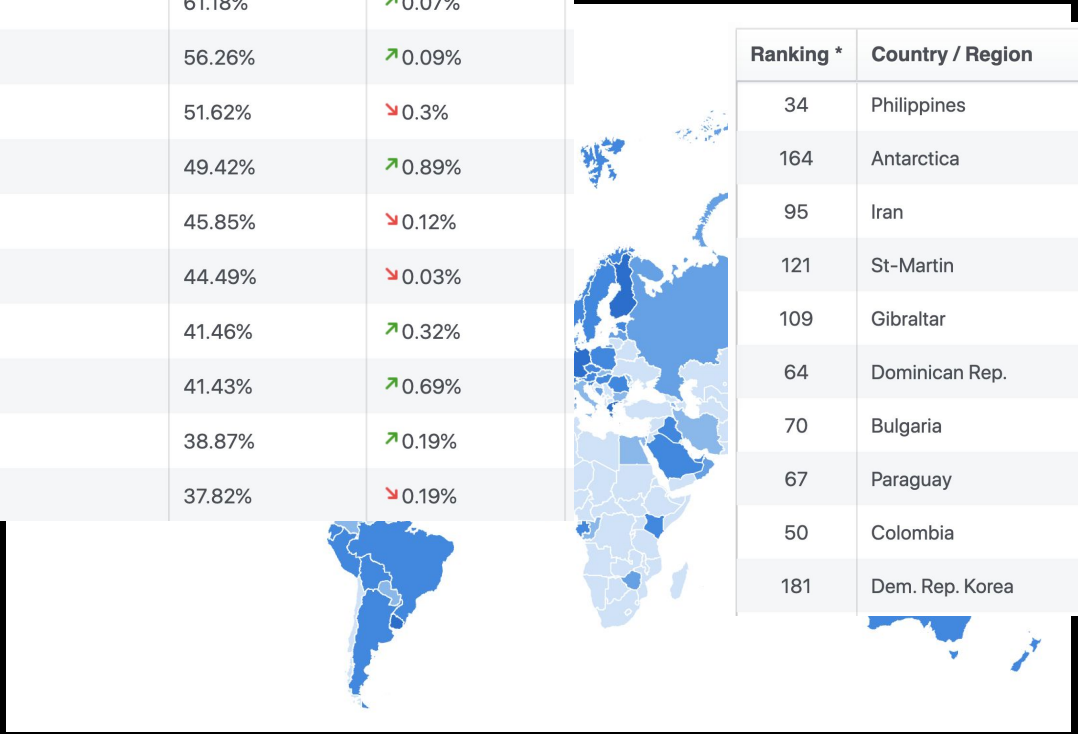

"Internet"
Edge
Backbone
Data Centers
Inside FB

# facebook.com/ipv6



# facebook.com/ipv6

| Ranking * | Country / Region | IPv6 Adoption | Weekly Growth |
|---|---|---|---|
| 2 | India | 61.18% | ↗0.07% |
| 1 | United States | 56.26% | ↗0.09% |
| 18 | Belgium | 51.62% | ↘0.3% |
| 7 | Germany | 49.42% | ↗0.89% |
| 21 | Greece | 45.85% | ↘0.12% |
| 11 | Taiwan | 44.49% | ↘0.03% |
| 4 | Vietnam | 41.46% | ↗0.32% |
| 8 | Malaysia | 41.43% | ↗0.69% |
| 38 | Finland | 38.87% | ↗0.19% |
| 10 | France | 37.82% | ↘0.19% |



| Ranking * | Country / Region | IPv6 Adoption |
|---|---|---|
| 34 | Philippines | 2.12% |
| 164 | Antarctica | 1.94% |
| 95 | Iran | 1.91% |
| 121 | St-Martin | 1.89% |
| 109 | Gibraltar | 1.73% |
| 64 | Dominican Rep. | 1.38% |
| 70 | Bulgaria | 1.31% |
| 67 | Paraguay | 1.31% |
| 50 | Colombia | 1.18% |
| 181 | Dem. Rep. Korea | 1.17% |

# connectivity.fb.com/

Hungary

In June 2018, Magyar Telekom, subsidiary of Deutsche Telekom, deployed their first Terragraph network in Mikebuda, Hungary.

Terragraph improved local network speeds from 5mbps to 650mbps.

Source: Magyar Telekom

reliable, high-speed internet in Uganda. Through this build we've improved network coverage in Northwest Uganda by 40%.

Source: Facebook and Industry Analysis

# More info

- engineering.fb.com/category/networking-traffic/
- research.fb.com/category/systems-and-networking/
- connectivity.fb.com/

# Connect with us!

**Krizia Torres, University Recruiter**
*kriziatorres@fb.com*

**Visit our Careers Page!**
*facebook.com/careers/university*

**Connect with us**
*@FacebookCareers on Facebook*
*@FacebookLife on Instagram*
*@Universities on Facebook*

PhD student? Contact Nate Lee (natelee@fb)

# Thanks!