
facebook
INFRASTRUCTURE

IPv6 @FB: From the NIC to the Edge

IPv6 Council - Dec 2017

Mikel Jimenez
Network Engineer, Facebook

Agenda

- Who am I
- Some IPv6 numbers
- Walk-through how Facebook implements IPv6
  - Servers -> Racks -> DC -> Backbone -> Edge
- Other IPv6 applications
- Questions ?

Who am I ?

- Mikel Jimenez
- Network Engineer @FB - Dublin
  - Network Infrastructure Engineering
  - DataCenter Network Engineering
  - BackBone Network Engineering
- I have a lots of IPv6 Tshirts

Agenda

- Who am I ?
- **FB in numbers**
- Walk-through how Facebook implements IPv6
  - Servers -> Racks -> DC -> Backbone -> Edge
- Other IPv6 applications
- Questions ?


2.07 Billion Users
1.37+ Billion Daily Users
85.8% of daily active users
outside US/Canada

Let’s talk about IPv6 :-)

As of
today…

16% user traffic is over IPv6

+50% US mobile traffic is IPv6

+99.99% internal traffic IPv6

So, how do we build this ?

Agenda

- Who am I ?
- FB in numbers
- **Walk-through how Facebook implements IPv6**
  - Servers —> Racks —> DC —> Backbone —> Edge
- Other IPv6 applications
- Questions ?

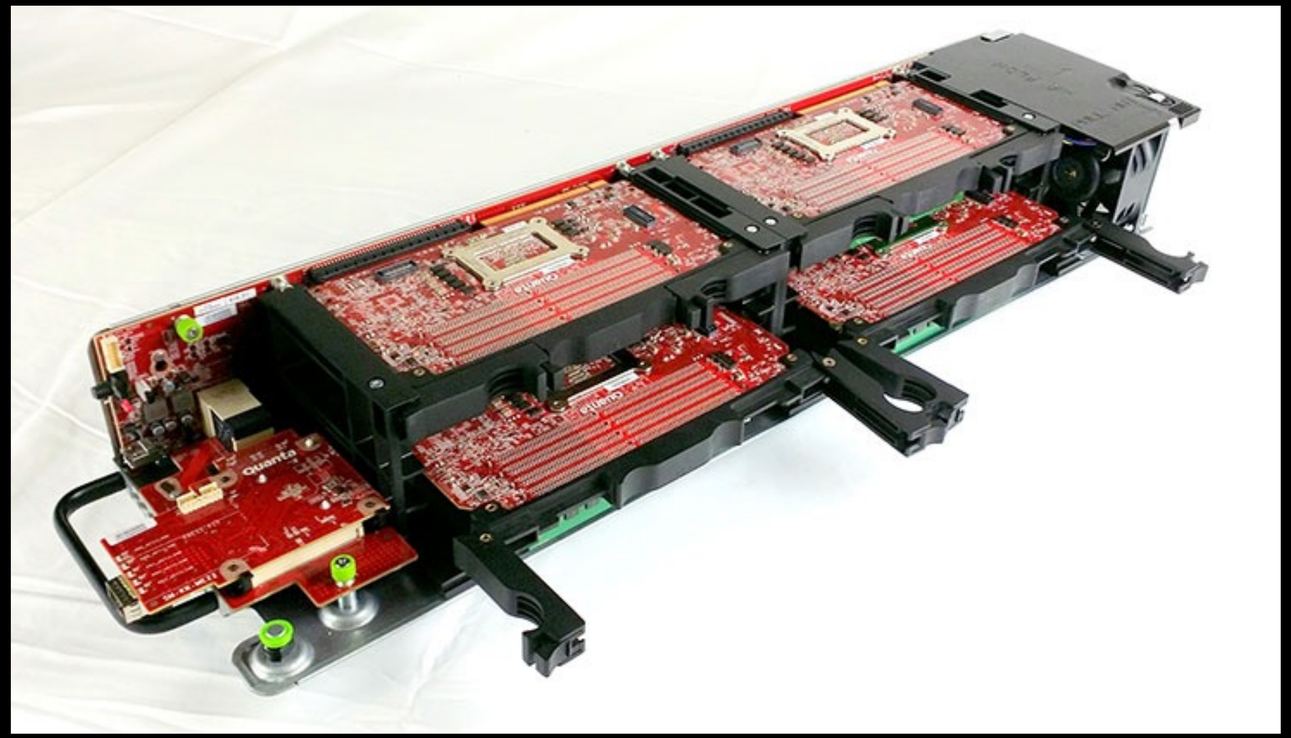
First… servers….

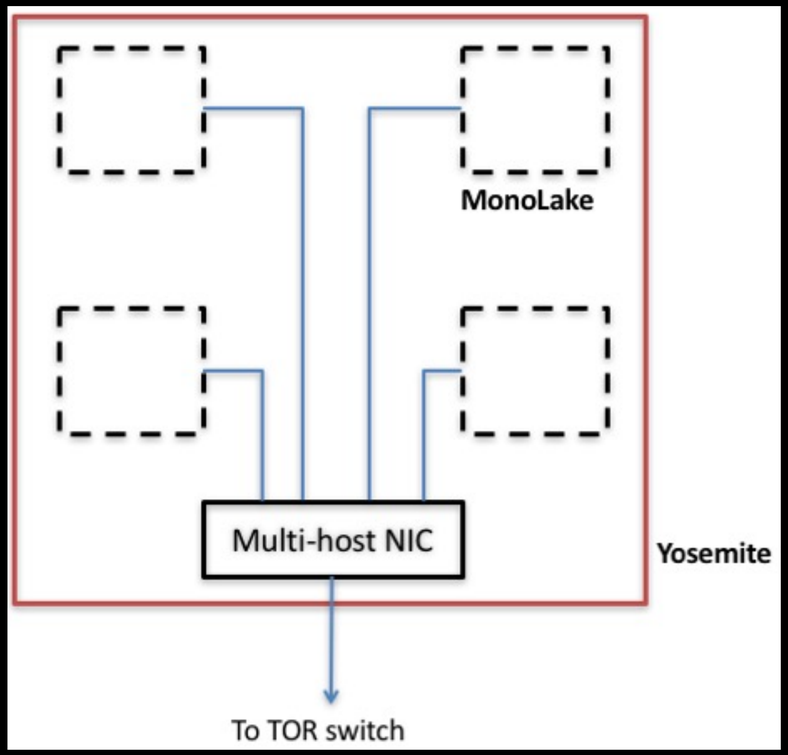
Servers

One NIC per host

Servers

Multi-host NICs





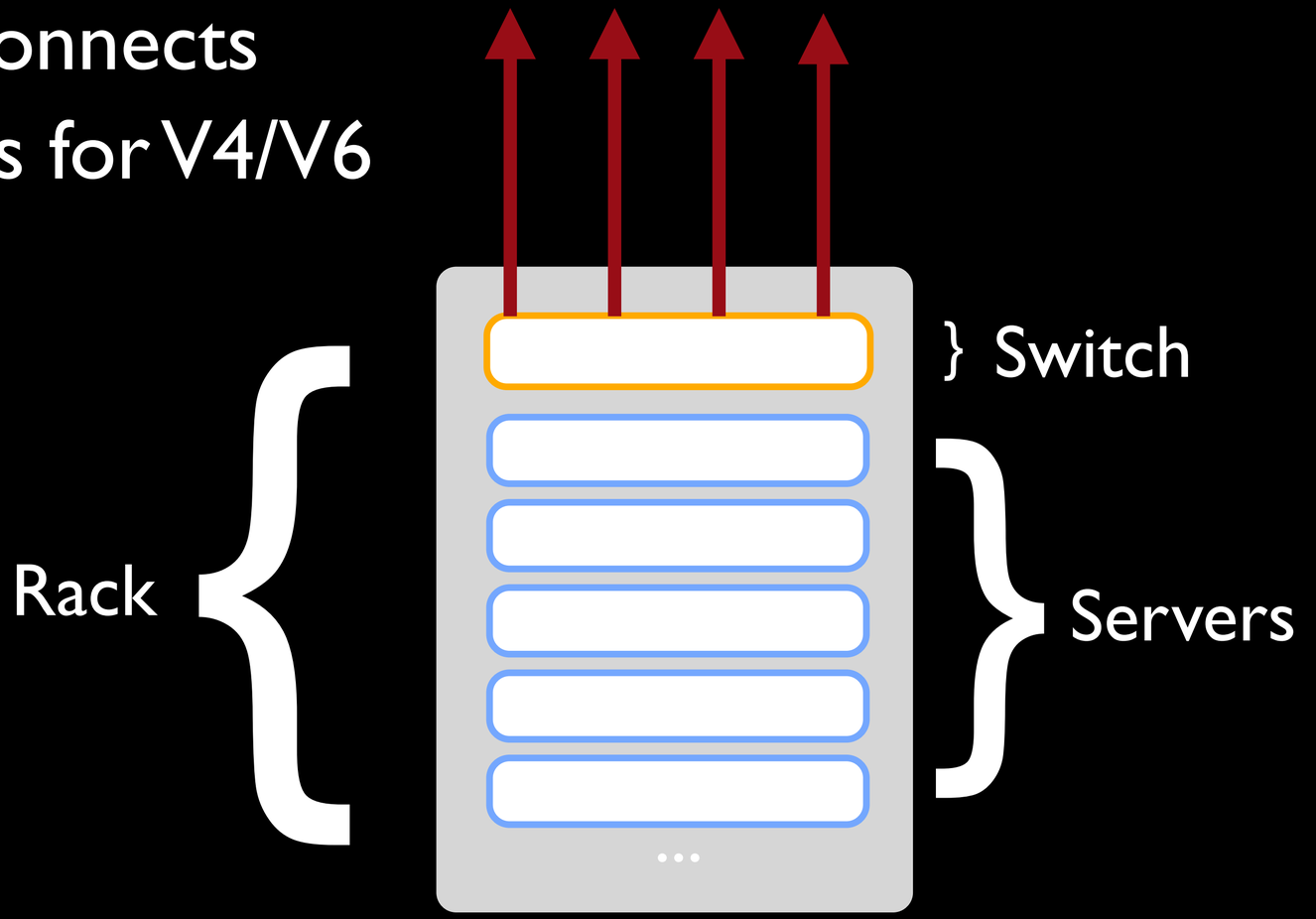
Server configuration

- Static configuration, managed by Chef
- Prefixlen /64
- Same default route across the fleet
  - “default via fe80::face:b00c dev eth0"
- Servers use BGP to announce /64 VIPs
  - TCAM scale friendly
- DHCPv6 used for provisioning purposes
  - RA interval from TOR 4s, important for provisioning

A group of servers

Rack

- /64 per rack
- 4x BGP uplinks, /127 interconnects
- Shared vs Dual BGP sessions for V4/V6
  - Vendor bugs
  - Operational pains



Rack

- Static IPv6 LL address for server facing local VLAN
  - ipv6 link-local fe80::face:b00c
  - Same across all racks, simple
  - Handy to implement default route specific configs like MTU/MSS

```
[root@host ~]# ip link | grep eth0
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 9000 qdisc mq state UP
mode DEFAULT group default qlen 1000

[root@host ~]# ip -6 route | grep mtu
default via fe80::face:b00c dev eth0 metric 10  mtu 1500 pref medium
2001:abcd::/52 via fe80::face:b00c dev eth0 metric 10  mtu 9000 pref medium
```

We have lots of racks

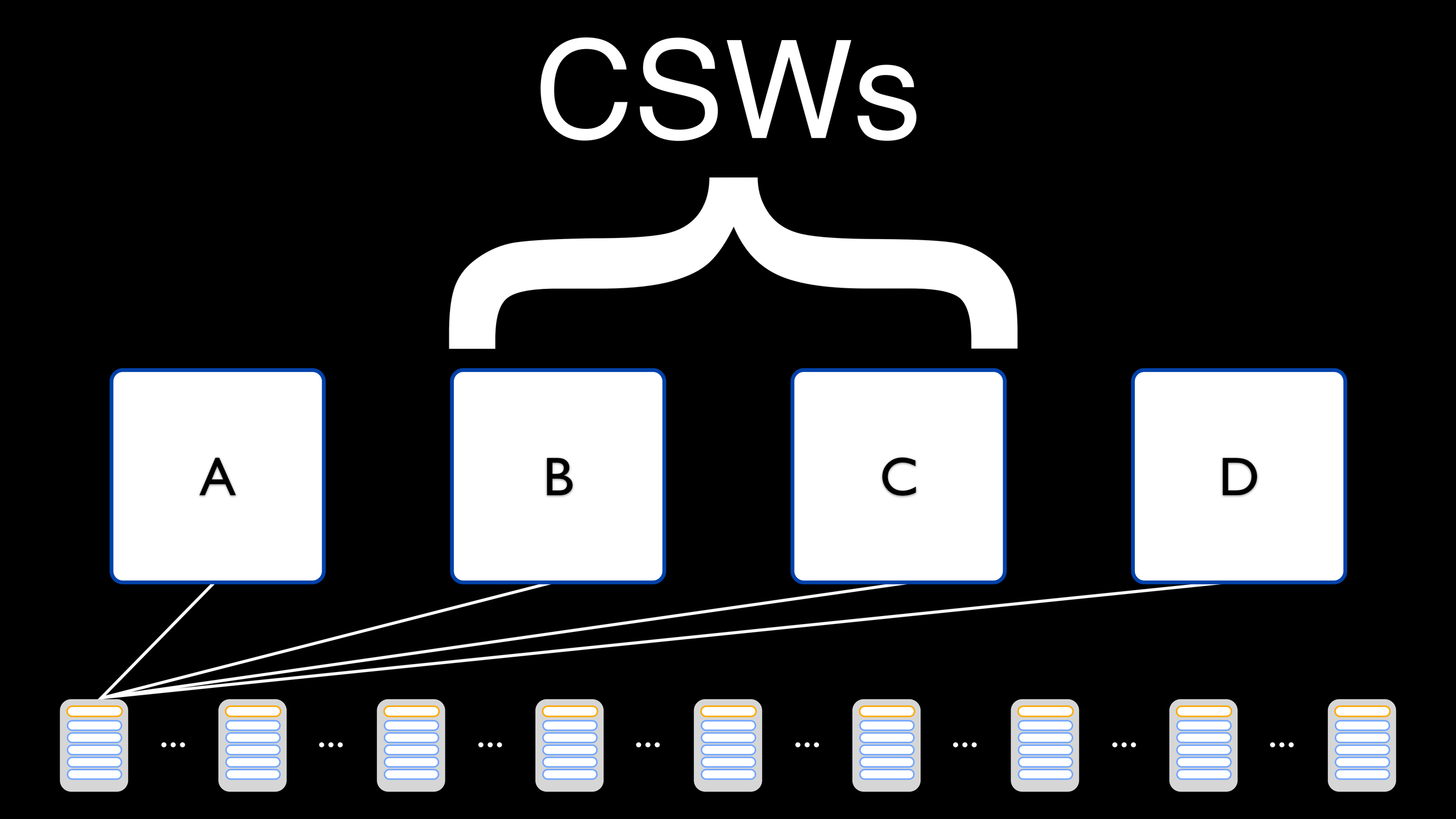
Racks talk to each other

2 Data center architectures

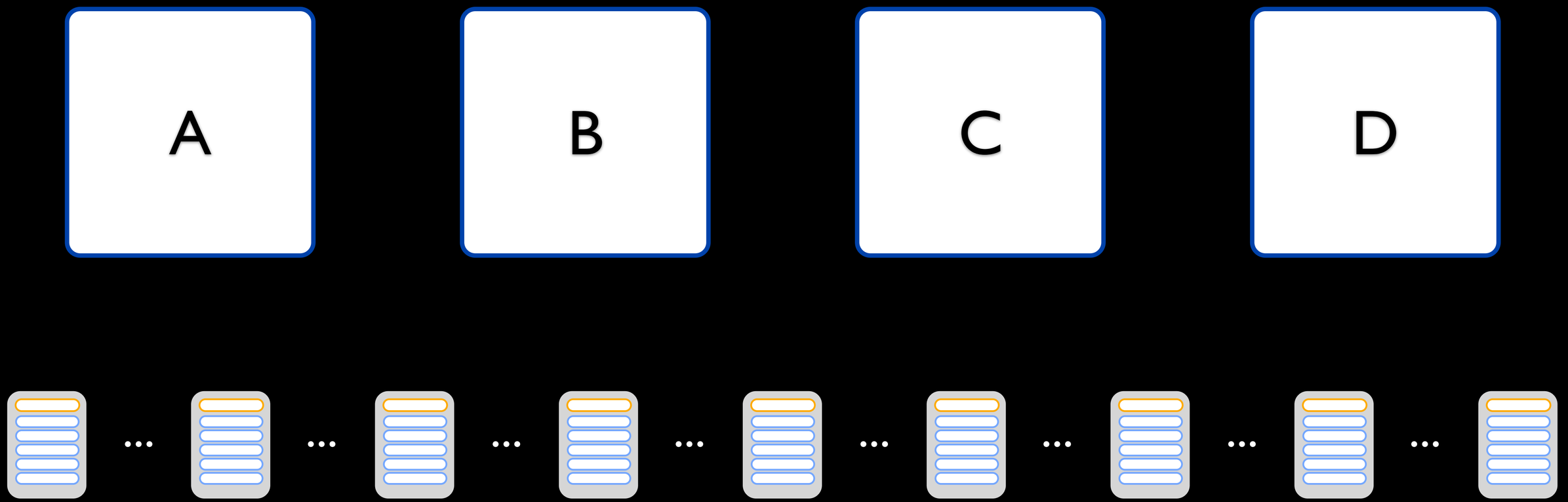
"4 post clusters"

4 post Clusters

- Legacy topology
- Built on big radix 4x cluster switches
- [ie]BGP the only routing protocol
- ECMP is your friend
- A very big unit of deployment


CSWs
A
B
C
D

Cluster



4 post Clusters

- Aggregating hundreds of racks in a big unit of compute
- Dual stack
  - /64 per rack aggregated in a /52 per cluster
  - /24 per rack on IPv4
- Too many BGP sessions!
  - Scaling pains
  - Had to move from dual v4 and v6 sessions to MP-BGP over v4

4 post Clusters - The "final" version

- IPv6 only services mind
- RFC 5549 support not there
  - Advertising IPv4 Network Layer Reachability Information with an IPv6 Next Hop
- Keep MP-BGP over IPv4 sessions the cope with BGP scale
- Non-routed reusable IPv4 address space for interconnects
- Non-routed reusable IPv4 address space for server VLAN
- The only routed/global IP space is IPv6

Data center Fabric

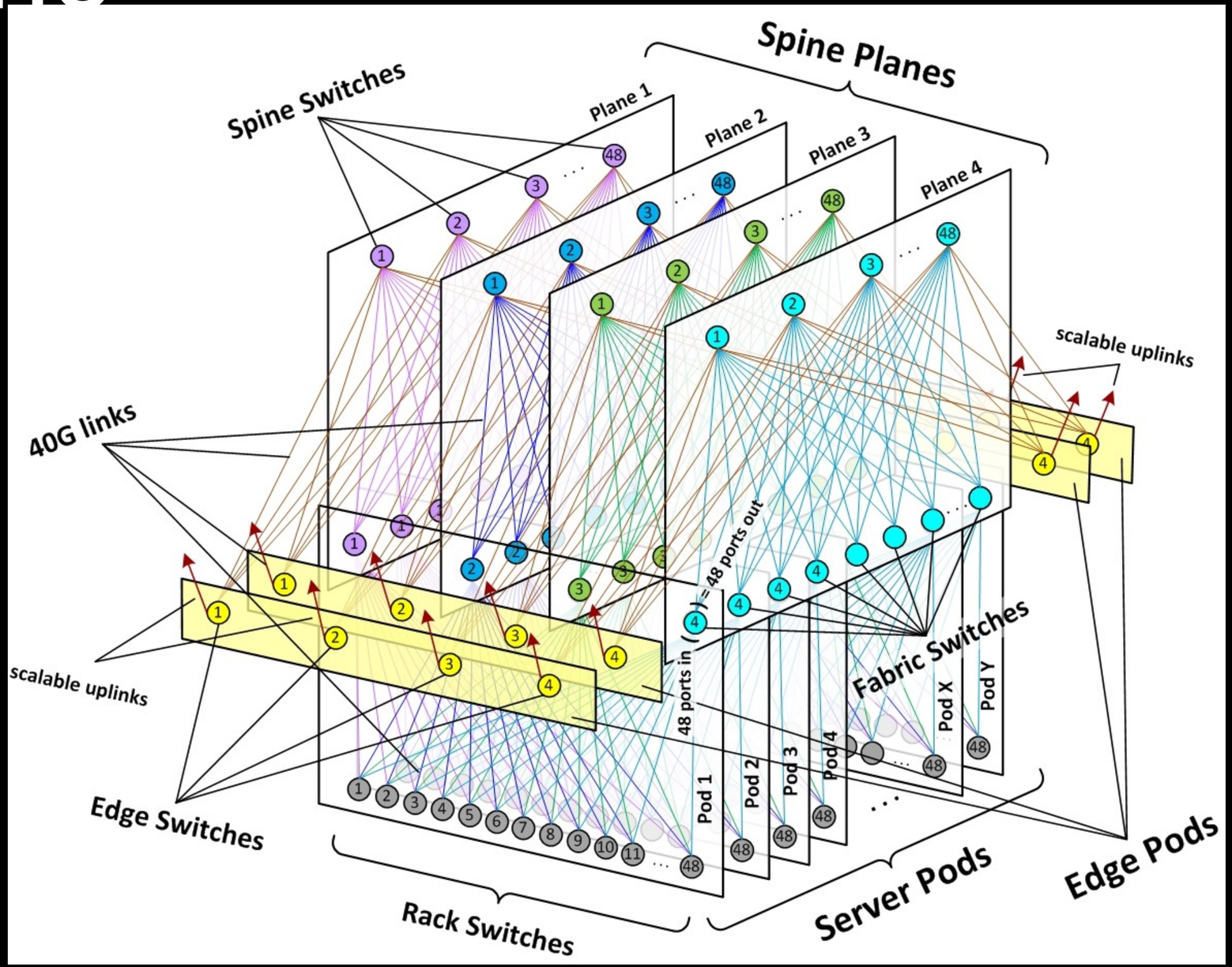
Fabric

- Massive scale, building wide Data center Fabric
- Built with smaller/simpler boxes
- 40G, 100G and beyond

Fabric

- Dual stacked
- Separate BGPv4 and BGPv6 sessions (Yes!!)
- Server POD as building block: 48 racks
- Similar aggregation concepts as previous design
  - /64 per Rack
  - /59 per Pod
  - /52 per cluster (group of PODs)

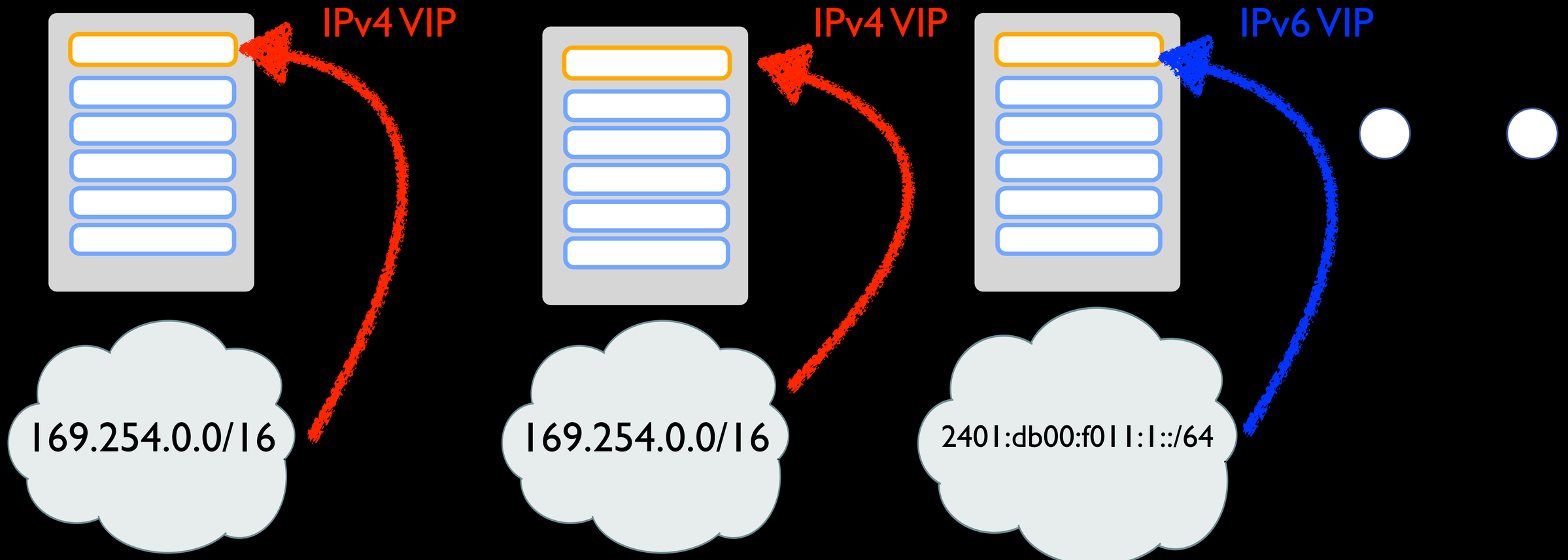
Fabric


Spine Planes
Spine Switches
Plane 1
Plane 2
Plane 3
Plane 4
scalable uplinks
40G links
scalable uplinks
Edge Switches
48 ports in ( ) = 48 ports out
Fabric Switches
Pod 1
Pod 2
Pod 3
Pod 4
Pod X
Pod Y
Rack Switches
Server Pods
Edge Pods

IPV4 services in IPv6 only clusters ?

Rack

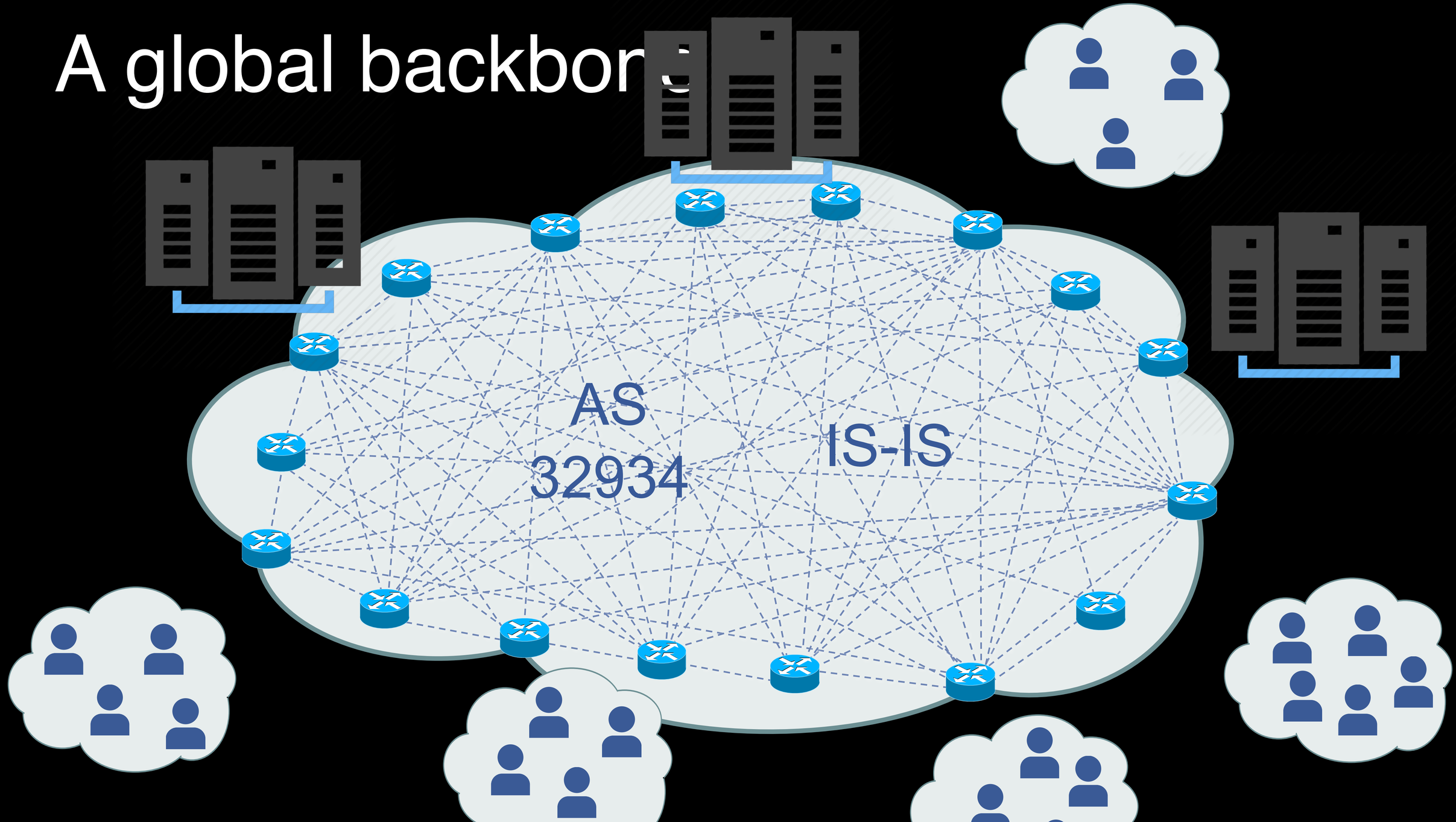
- All racks with same 169.254.0.0/16 address space server facing VLAN for IPv4 VIP injection
- Every rack with different /64, regular BGP VIP injections



We have lots of DCs...

and we need to connect them :)

A global backbone



Backbone

- Global presence
- Used for DC-DC and POP-DC connectivity
- IS-IS as IGP protocol
- Based on MPLS/RSVP-TE
- BGP free core

Backbone: IGP Routing IPv6

- In the early days, we IGP routed IPv6 traffic because there wasn't much
- As traffic started ramping up we ran into problems
- We had RSVP-TE and no one had a RSVP v6 implementation
- Remember: BGP free core
- Again, no one had a working RFC 5549 implementation

Decisions...

Options	Pros	Cons
IPv6 Tunneling	Less BGP state, Simplest Configuration	Bounce BGP Sessions
BGP Labeled Unicast (6PE)	Less BGP State, No LSR Dual Stacking, End to End LSPs	Bounce BGP Sessions, New BGP AFI/SAFI
IGP shortcuts	No BGP changes, flexible for Dual Stack Environments	More BGP state, LSP metrics Need to change

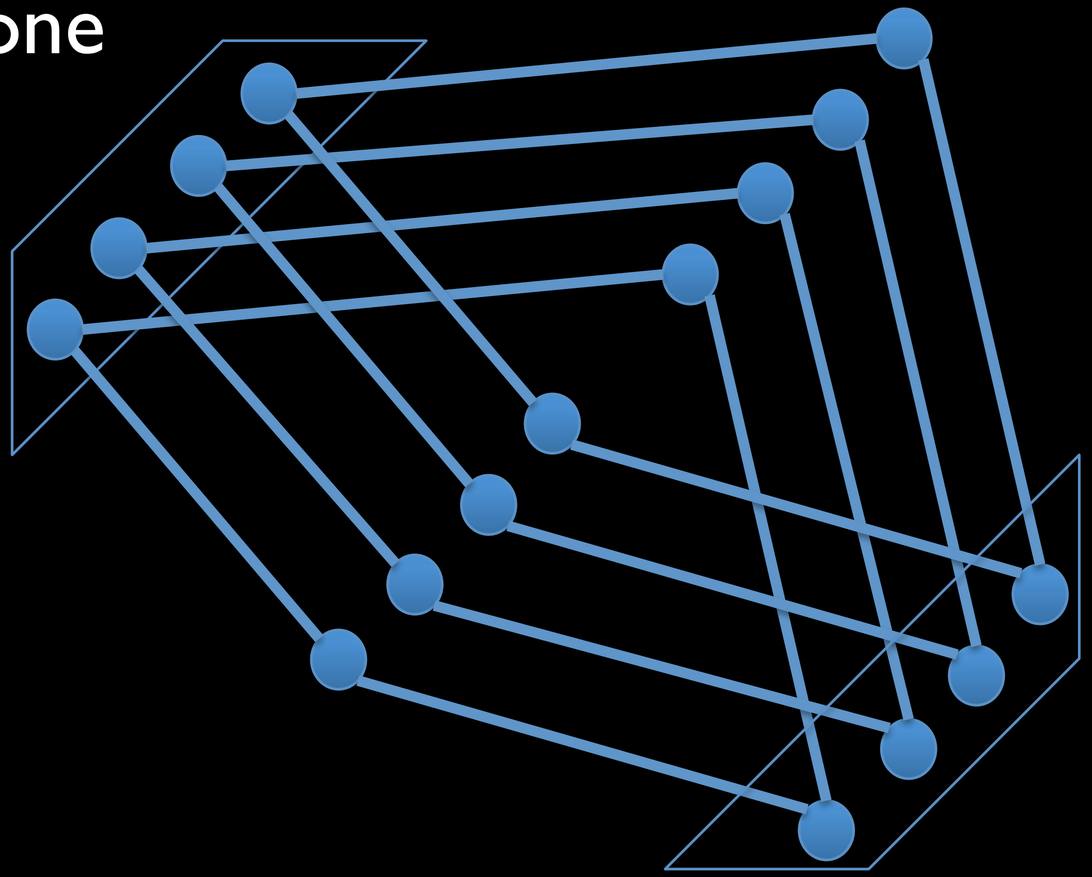
Decisions…

Options	Pros	Cons
IPv6 Tunneling	Less BGP state, Simplest Configuration	Bounce Sessions, Dual Stacked LSRs
BGP Labeled Unicast (6PE)	Less BGP State, No LSR Dual Stacking, End to End LSPs	Bounce Sessions, New BGP AFI/SAFI
IGP shortcuts	No BGP changes, flexible for Dual Stack Environments	More BGP state, LSP metrics Need to change

Not only one BackBone…

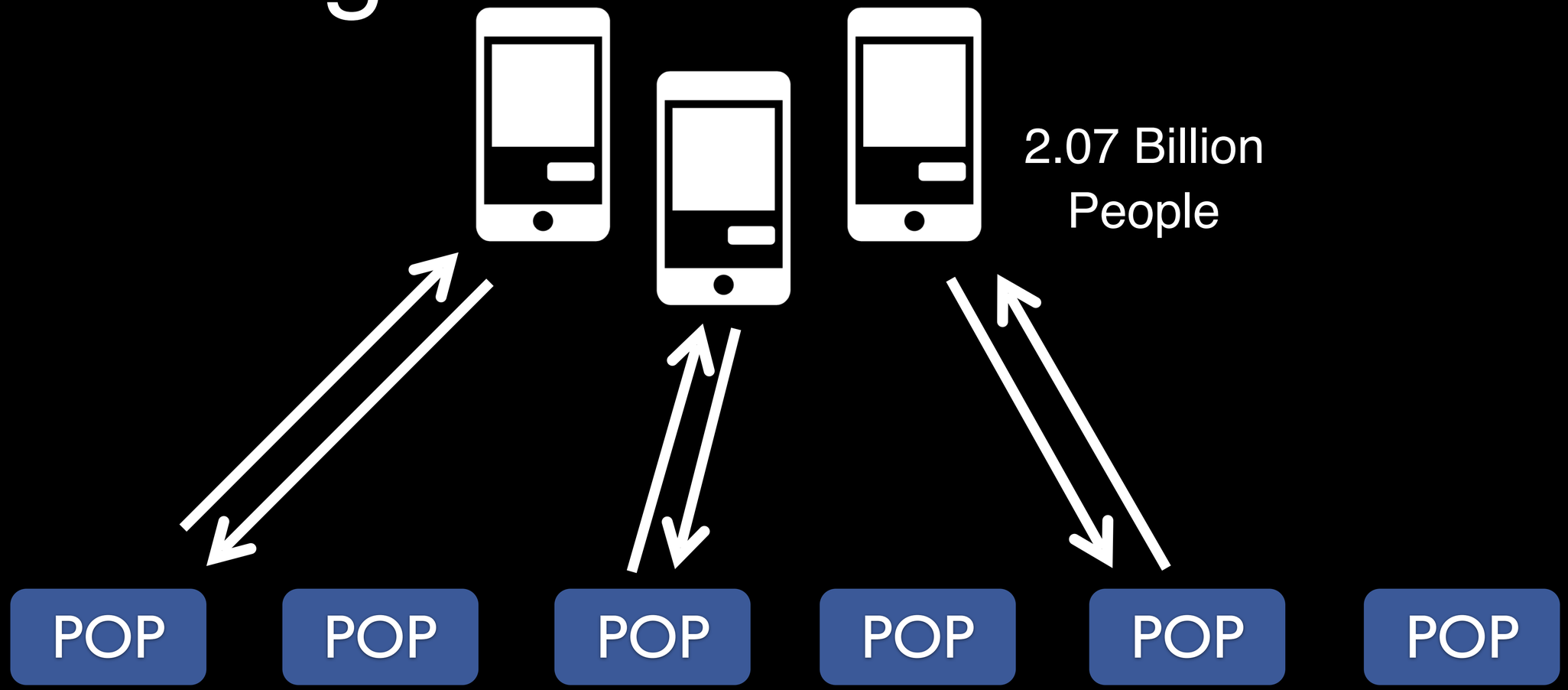
ExpressBackbone

- Dedicate DC-DC SDN BackBone
- 4 parallel planes
- IPv6 the only routed protocol
- OpenR as IGP

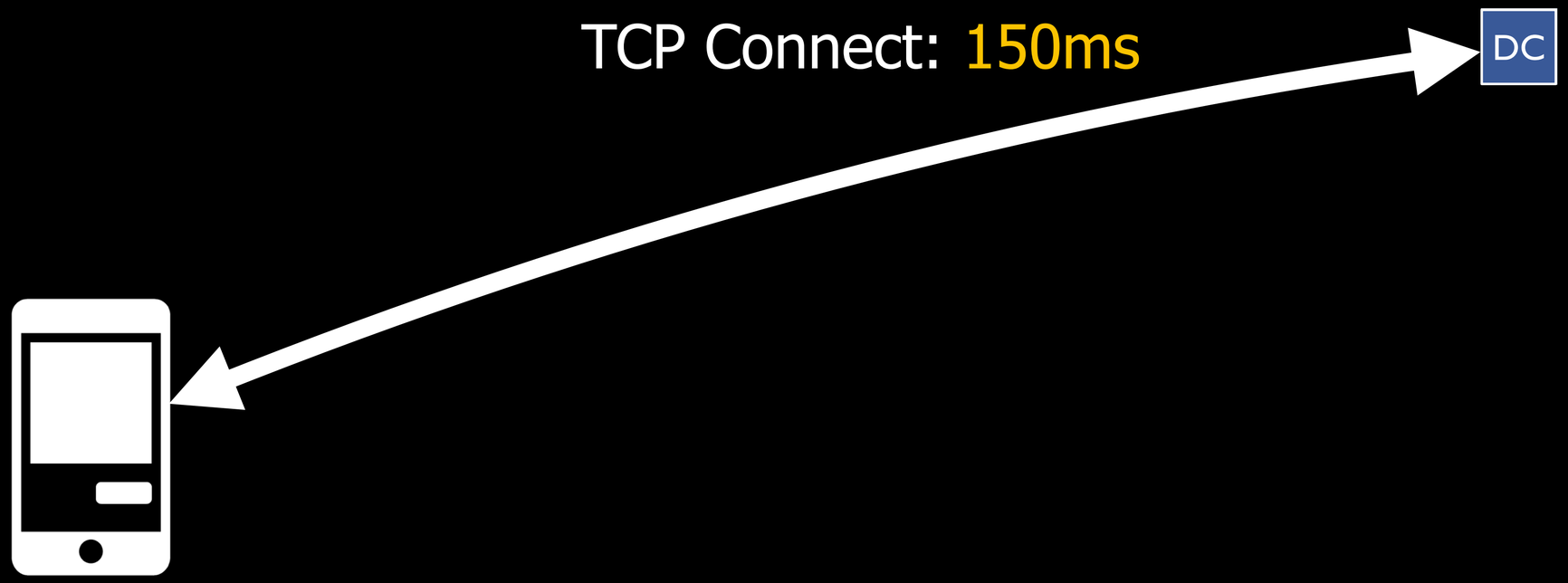


How do users reach Facebook ?

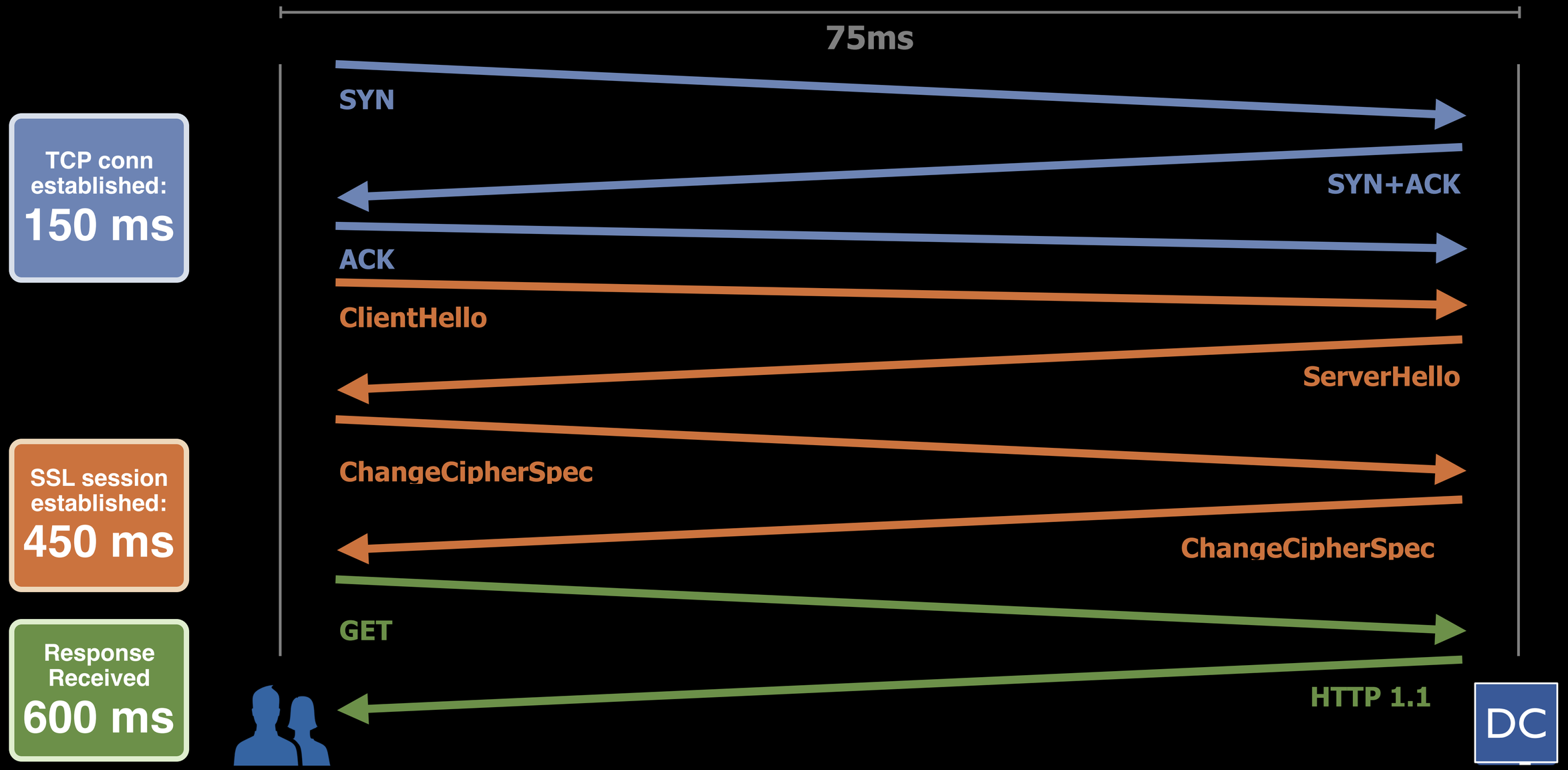
Our edge connects to the world



LocationX -> Oregon



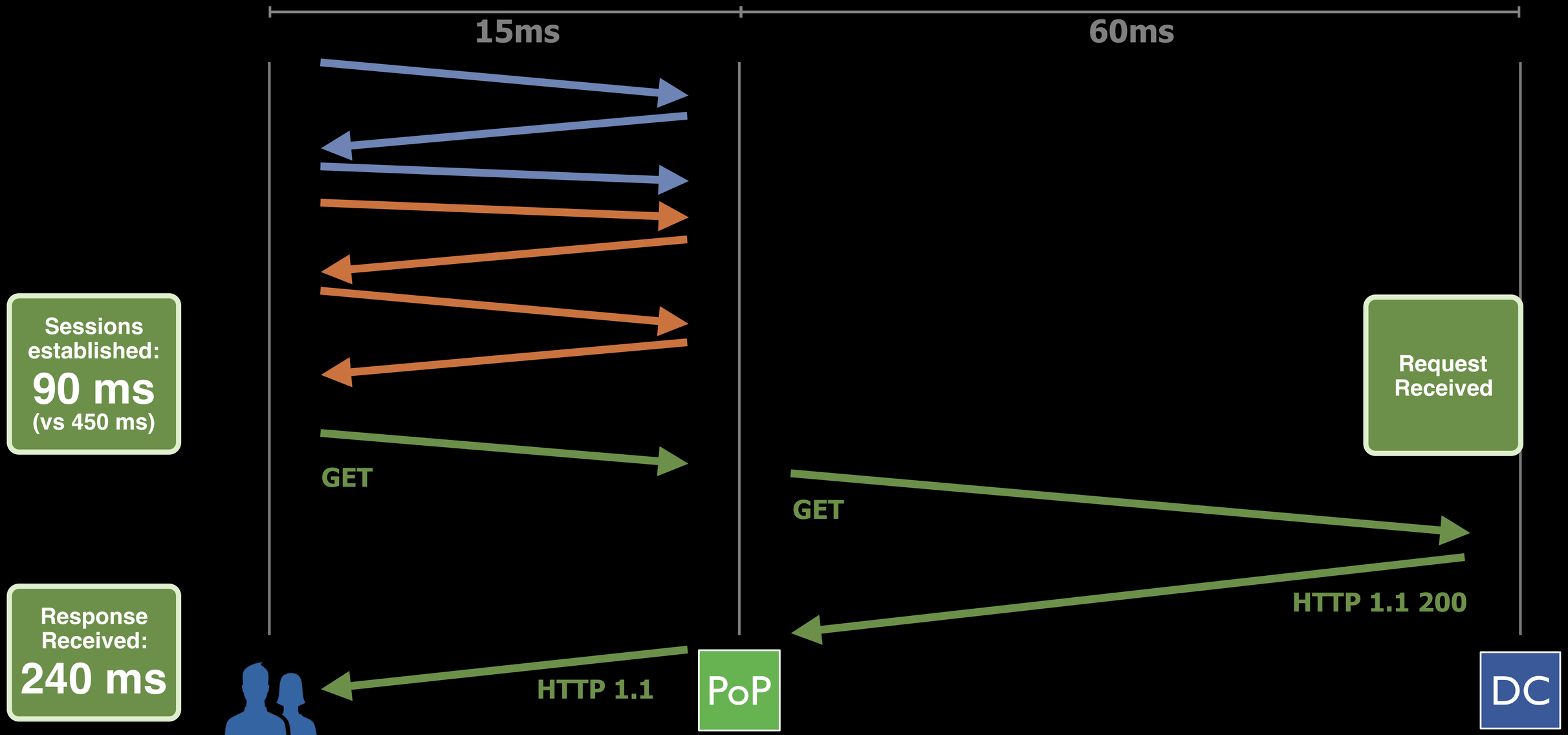
HTTPS LocationX -> Oregon



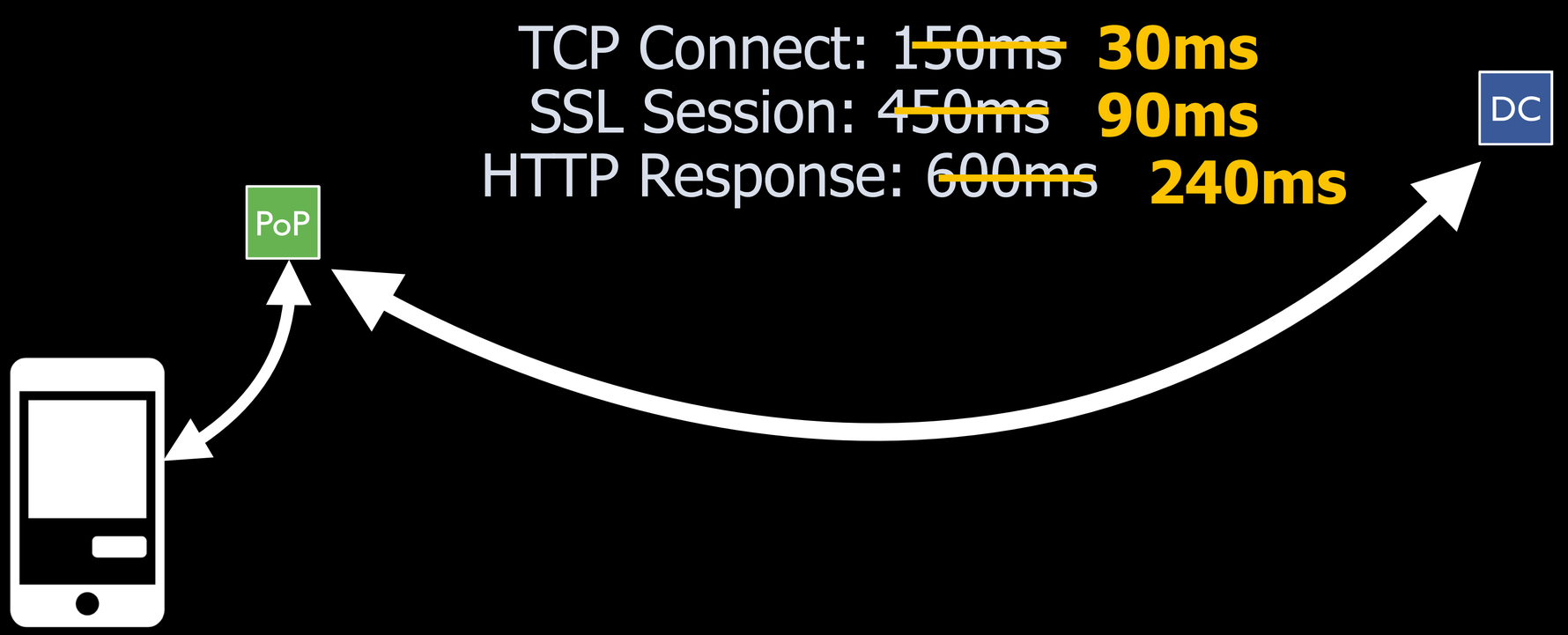
LocationX -> Oregon

TCP Connect: 30ms
SSL Session: ??
HTTP Response: ??

HTTPS LocationX -> POP -> Oregon

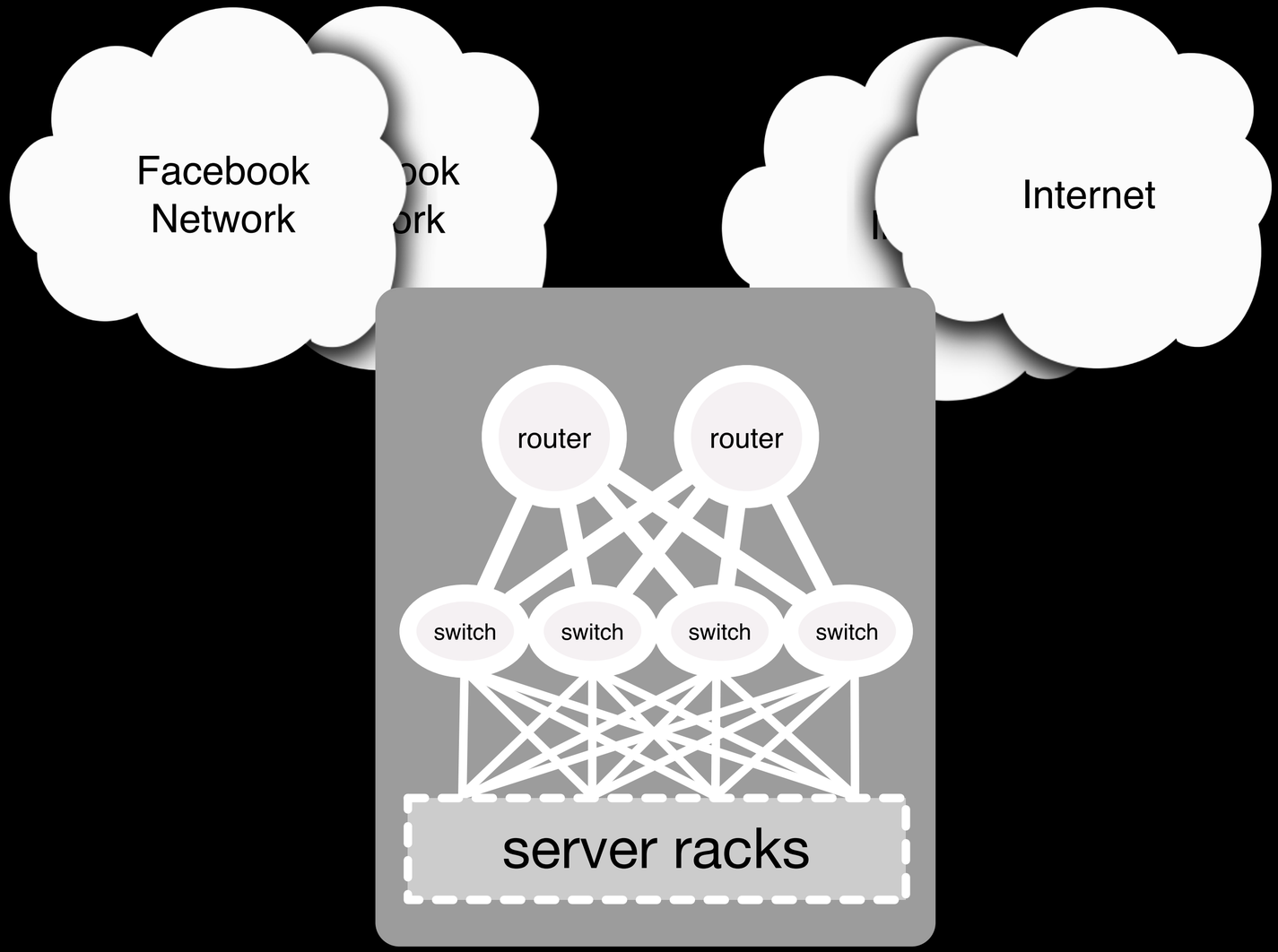


LocationX -> Oregon

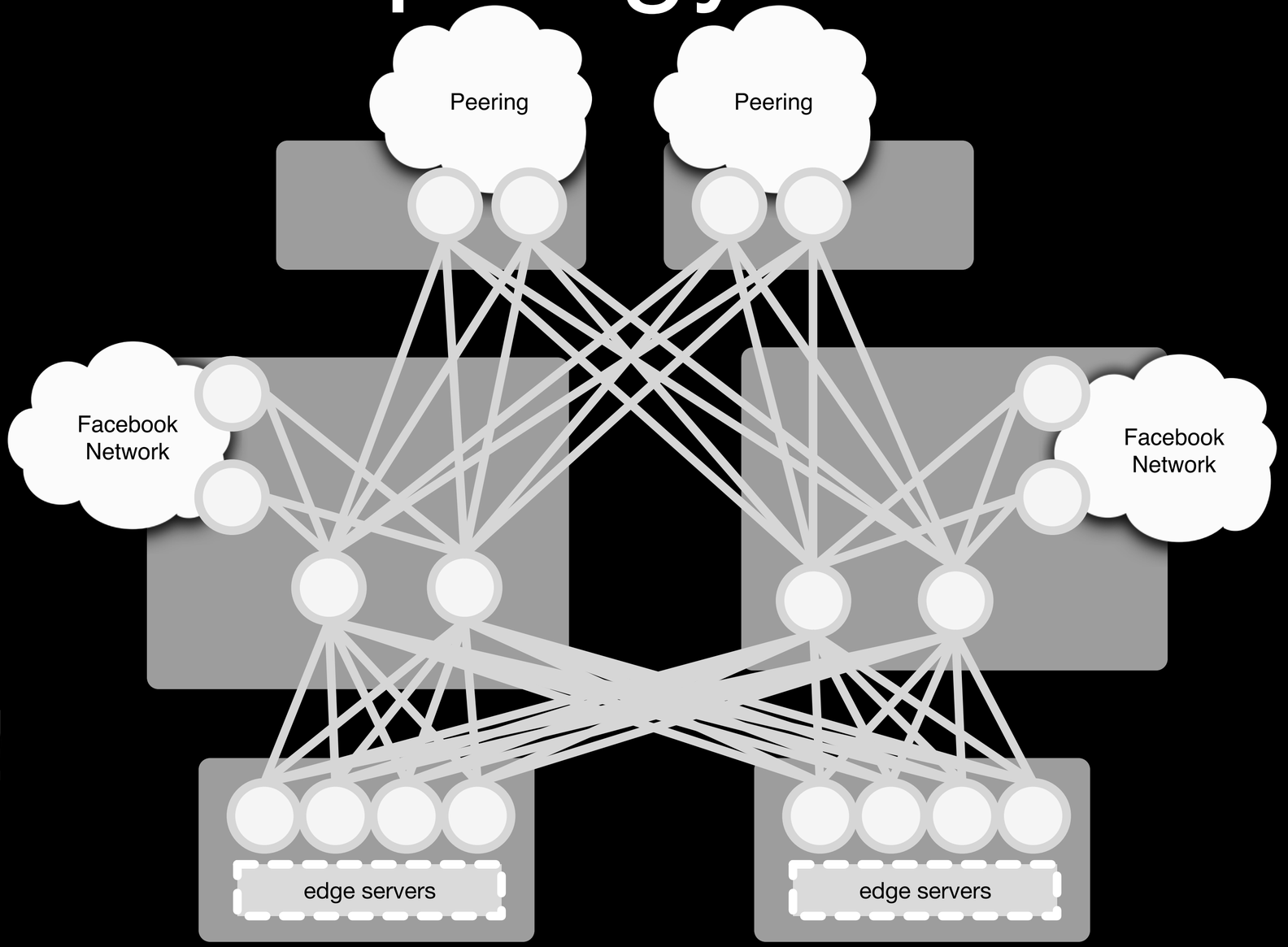


These locations are not representative of actual PoP locations

edge routers -> edge clusters



-> edge metro topology



100G
Everywhere!

Edge

- Inherited a lot of concepts from the DC
- BGP the king
- /64 per rack, /52 per cluster, /48 per metro
  - Multiple clusters in the metro, /48 external announcement
- All Edge->Origin traffic is IPv6
  - Users connecting to us via IPv4 are proxied back using IPv6
- All east-west traffic inside the POP is 100% IPv6.

No NATs :-)

Agenda

- Who am I ?
- FB in numbers
- Walk-through how Facebook implements IPv6
  - Servers -> Racks -> DC -> Backbone -> Edge
- **Other IPv6 applications**
- Questions ?

IPv6: other applications

IP per task

FB runs on containers

Current challenges

- Right now: IP per host
- We need a port per container/task
  - Every restart - new port
  - Service directory churn
- Port collision, complex allocation logic
- Painful traffic accounting

IPv6 to the rescue!

/64 per host

- Every server at Facebook gets a dedicated /64
- Adapted container address allocation
- IP Address per task
  - Each task get's it's own IPv6 /128
  - Each task get's it's own port number space
  - Simplifies task scheduling and accounting
  - Port collisions gone (W00000TTT!!!)

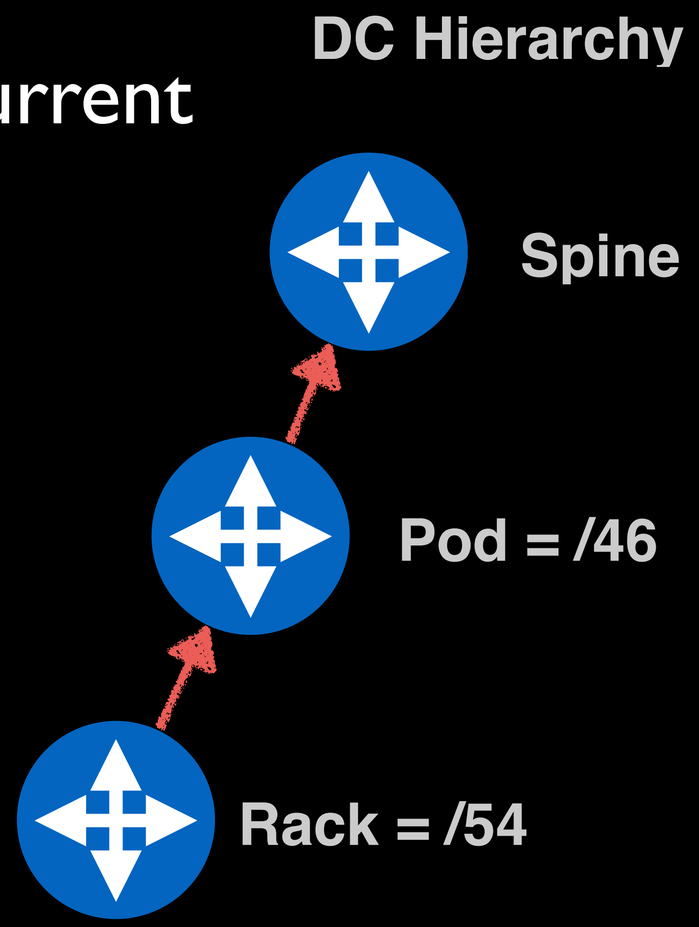
/64 per host

- Uses the new /64 as an address pool
- The ::1 address in /64 reserved for physical host <> IP
- Controlled rollout, preferred lifetime = 0

```
$ ip -6 a ls
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qlen 1000
    inet6 2803:6082:18e0:e825::1/64 scope global deprecated
       valid_lft forever preferred_lft forever
    inet6 2401:db00:11:d03a:face:0:25:0/64 scope global
       valid_lft forever preferred_lft forever
    inet6 fe80::f652:14ff:febe:fe54/64 scope link
       valid_lft forever preferred_lft forever
```

/64 per host

- Overlay addressing schema on top of current
- Hierarchical allocation
  - /54 per rack
  - /44 per cluster (/48 in Edge)
  - /37 per DC Fabric



ILA: Identifier Locator Addressing

ILA

- Location independence addressing, mobility
- Splits 128 bits of IPv6 in 2
  - Locator: First /64 bits, routable
  - Identifier: Task ID
  - draft-herbert-nvo3-ila, draft-lapukhov-ila-deployment

```
$ ip -6 a ls
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qlen 1000
    inet6 2803:6082:18e0:e825::1/64 scope global deprecated
       valid_lft forever preferred_lft forever
    inet6 2401:db00:11:d03a:face:0:25:0/64 scope global
       valid_lft forever preferred_lft forever
    inet6 fe80::f652:14ff:febe:fe54/64 scope link
       valid_lft forever preferred_lft forever
```

Facebook's IPv6 Deployment Timeline

2008 — First IPv6 discussions based on RIR IPv4 depletion warnings

2009 — Discussions and testing around IGP selection to support IPv6 moving forward. IS-IS is selected as Facebook's new IGP.

2010 — IGP migration from OSPF to IS-IS completed

2011 — **World IPv6 Day**, Dual stacking load balancer VIPs and the start of dual stacking the backbone

2012 — **World IPv6 Launch**, backbone dual stacked, IGP shortcuts deployed

2013 — Dual stacking work in the Data Center and Edge POPs

2014 — First native IPv6 clusters deployed. We start actively migrating services to IPv6 from IPv4

2015 — All clusters with one exception were turned up native IPv6.

2017 — **+99% of internal traffic and 16% of external traffic is now IPv6.**
IP per task rolled out
ILA being rolled out

???? — IPv6 everywhere…

To make IPv6 a reality…

- Alignment between Network and application teams
- Make it part of the roadmap and success criteria
- It helps if management is invested on the mission
- Is not gonna be an easy ride
- Iterate, test, and iterate again
- Less documents, more deployments
- Start yesterday

Questions?


facebook
INFRASTRUCTURE