Who needs ARP?

Let’s route legacy IP (IPv4) via IPv6 next hops!

```
[cooper@home1 ~]$ arp -a
_gateway (10.6.9.1) at ac:1f:6b:6f:0d:97 [ether] on eth0
home2.cooperlees.com (10.6.9.3) at 00:e0:67:20:1f:c9 [ether] on eth0
? (10.6.9.69) at a0:78:17:93:a8:67 [ether] on eth0
? (10.6.9.8) at 08:00:27:2f:47:9d [ether] on eth0
? (10.6.9.10) at 14:7d:da:d9:ae:f2 [ether] on eth0
home1.cooperlees.com (10.6.9.2) at ac:1f:6b:6f:0d:97 [ether] on eth0
```

Cooper Lees
Production Engineer

James Paussa
Real Network Engineer

Meta

Agenda

Discuss pros + cons of stopping legacy (IPv4) addressing of your networks today!

Intro

Where is IPv6 @ Meta?

Point to Point Addressing - Why?

IPv4 via IPv6 - What? How? Why?

Where can I deploy this?

OSS + Vendor Support

Linux IPv4 via IPv6 Demo/Lab

a. https://github.com/cooperlees/v4v6demo

Who am I?

Cooper Lees

- Production Engineer -
  New Operating System Team
- From Wollongong
- Based Remote in South Lake Tahoe, CA
- Former
  - ANSTO + ICT Networks

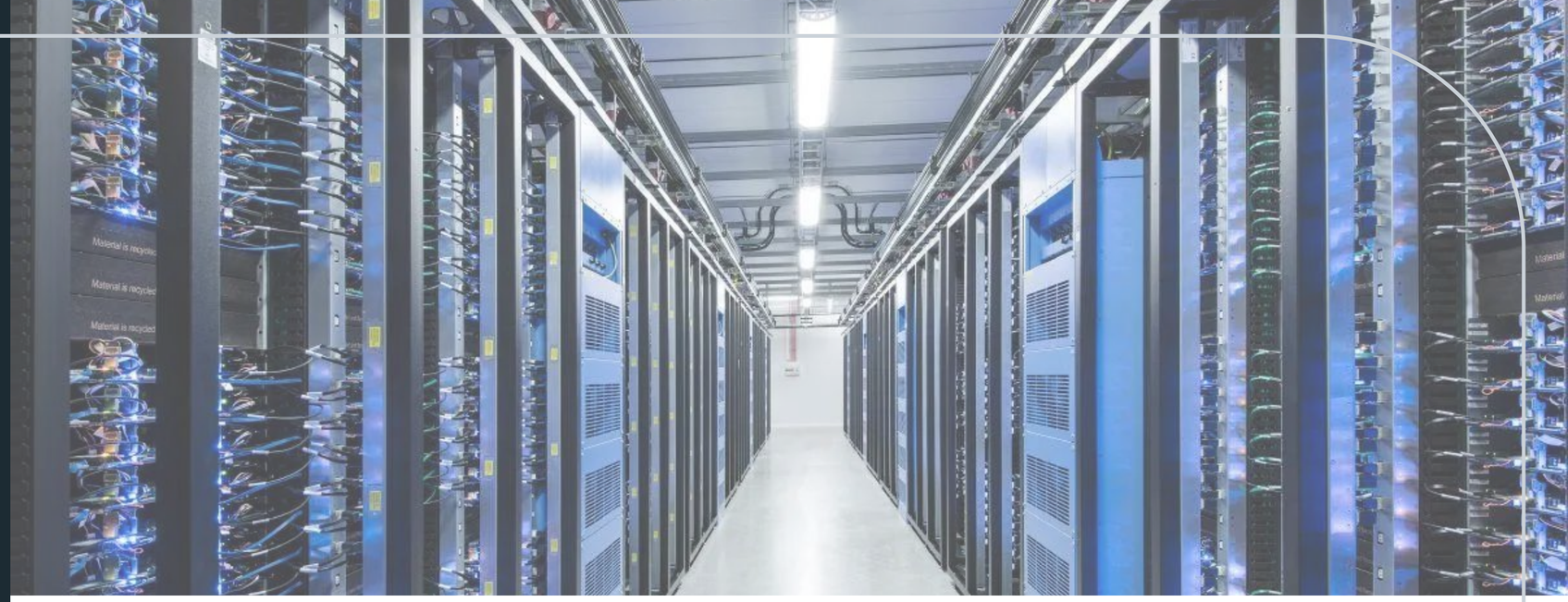
Meta DC Networks Today

- 10s of regions with 2-6 data halls each
- Each data hall is a CLOS Fabric
- Newer DCs are all FBOSS software
  - Rack to Datacenter edge
- 100000s of FBOSS devices

IPv6 in Meta DC Networks Today

- Started going IPv6 only / first in 2013
- Today, all production user traffic is IPv6
  - IPv4 is Layer 7 terminated @ Edge POPs
- All DC network management is via IPv6
  - Legacy management is dual stacked

IPv6 in Meta DC Networks Today



- Servers DHCPv6 + IPv6 PXE etc. to image

- The VLAN between
  - server <> switch
    has **no ARP/IPv4 today globally**

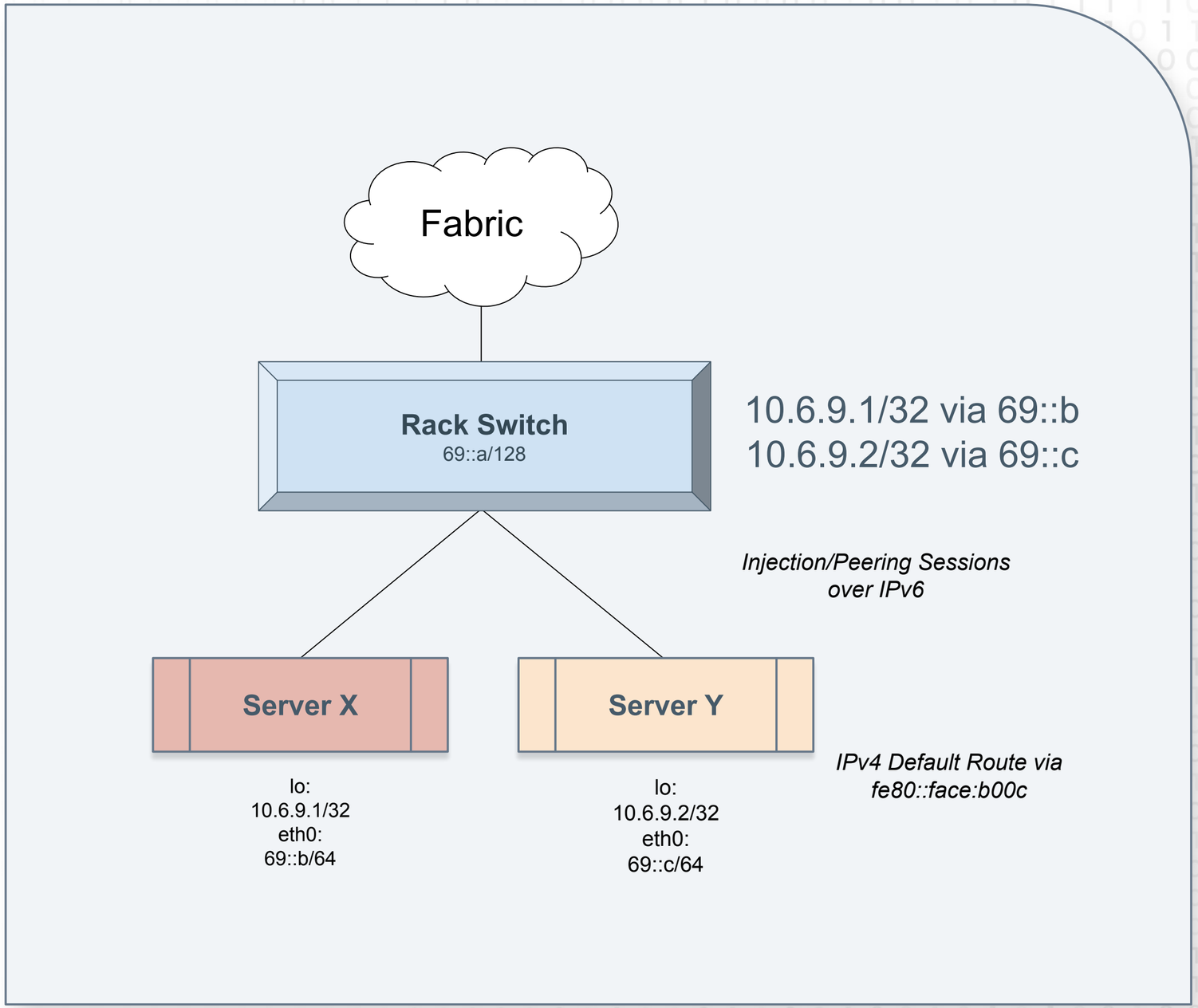
- Side note: In our main Production Fabric
  - We do have a dual Stacked Facilities Network @ each region

IPv4 in Meta DC Networks Today

```
[cooper:rsw008.p104.f01.prn1]$ fboss vips injectors -N

VIP status on rsw008.p104.f01.prn1 :
    devbig1035.prn1: 2401:db00:1c:6707:face:0:13e:0+747082

1 injector(s) are connected

[cooper:rsw008.p104.f01.prn1]$ fboss bgp table | grep -A 1 10.127.30.4
> 10.127.30.4/32, Selected 1/1 paths
*@ from 2401:db00:1c:6707:face:0:13e:0 via
2401:db00:1c:6707:face:0:13e:0 LBW None, IGP, LP:
VIP_HIPRIO/110, ASP: 65000, 69h6m ago
```

- All production IPv4 is an injected VIP
  - Either via BGP or Thrift to Rack Switch
- It’s last hop is always via IPv6

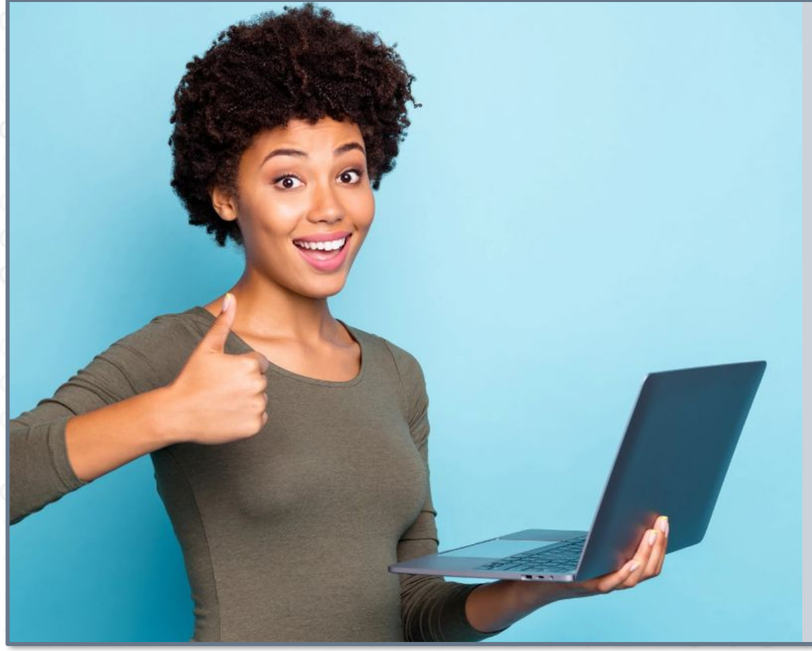
IPv4 in Meta DC Networks Today



Point to Point Addressing

Why have p2p addresses?

Point to Point Addressing



- To have next hops to route to
- Obtain ICMP responses from ingress interface
  - Get per link resolution
- Traditionally prefix has always been the same Address Family as the next-hop.

Why have p2p addresses?

```
Host                                          Loss%   Snt   Last   Avg  Best  Wrst StDev
 1. ns1rtr1-ten-lan.binarylane.cloud           3.4%    29    0.3   1.0   0.2   9.0   2.0
 2. as7575.nsw.ix.asn.au                       0.0%    29    1.6   2.7   0.9  20.4   4.2
 3. et-4-3-0.pe1-brwy-nsw.aarnet.net.au        0.0%    29    1.2   2.5   1.0  14.3   2.7
 4. et-1-1-0.pe1.mcqp.nsw.aarnet.net.au        0.0%    29    5.3   2.5   1.4  11.1   2.4
 5. et-0-3-0.pe1.eskp.nsw.aarnet.net.au        0.0%    29   13.4  14.5  13.4  19.4   1.9
 6. et-5-3-0.pe1.wmlb.vic.aarnet.net.au        0.0%    29   34.8  15.5  13.3  34.8   4.3
 7. 2001:388:cf0c:e::2                         0.0%    29   13.6  14.1  13.5  19.2   1.5
 8. (waiting for reply)
 9. (waiting for reply)
10. mirror.aarnet.edu.au                       0.0%    28   13.4  14.1  13.4  18.4   1.3
```

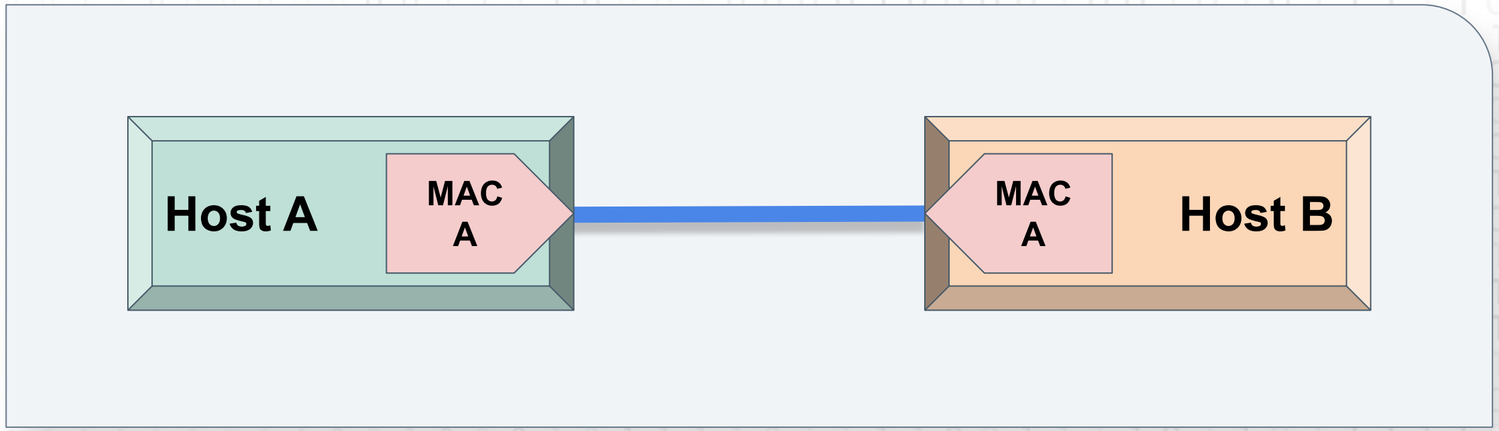
- Do we need that resolution everywhere?
- Would responses from loopbacks be fine in most cases?
- Is the hop before enough information?

IPv4 via IPv6

Disclaimer:

We are only addressing
directly connected routers
+ **IP routing** today …

So how does IPv4 route via IPv6??



- **tl;dr** - It doesn't …
- When a routers sees an IPv6 next hop for an IPv4 Prefix:
  - It source the **Layer 2** address from a different source table
- For ethernet, we use MAC addresses for Layer 2
  - IPv4 == ARP Table
  - IPv6 == Neighbor Table

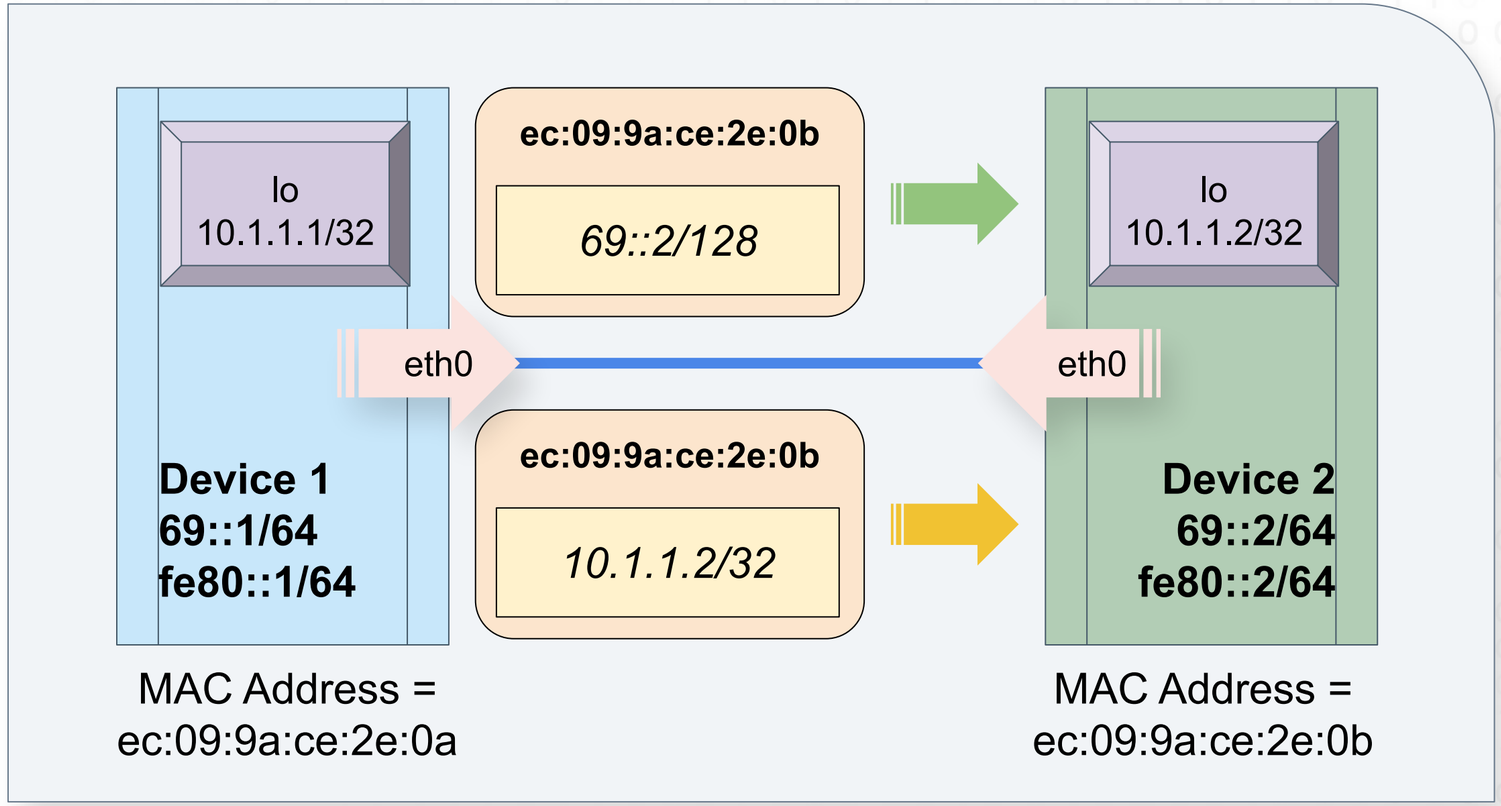
Logically point to point

IPv4 via IPv6

To the receiver … it looks the same … same destination MAC address …

From Device 1 to talk to IPv4 on Device 2

```
ip -4 route add 10.1.1.2/32 via inet6 69::2
ip -4 route add 10.1.1.2/32 via inet6 fe80::2 dev eth0
```



IPv4 needs an IPv4 nexthop …

```
# IPv4 via IPv6 in Linux Kernel since 5.2

[cooper:~]$ ip -4 route

default via inet6 fe80::face:b00c dev eth0 src 10.6.9.9
```

- Nope - Not anymore.
- IPv6 Link Local must have interface defined on routes …

What is a L2 table?

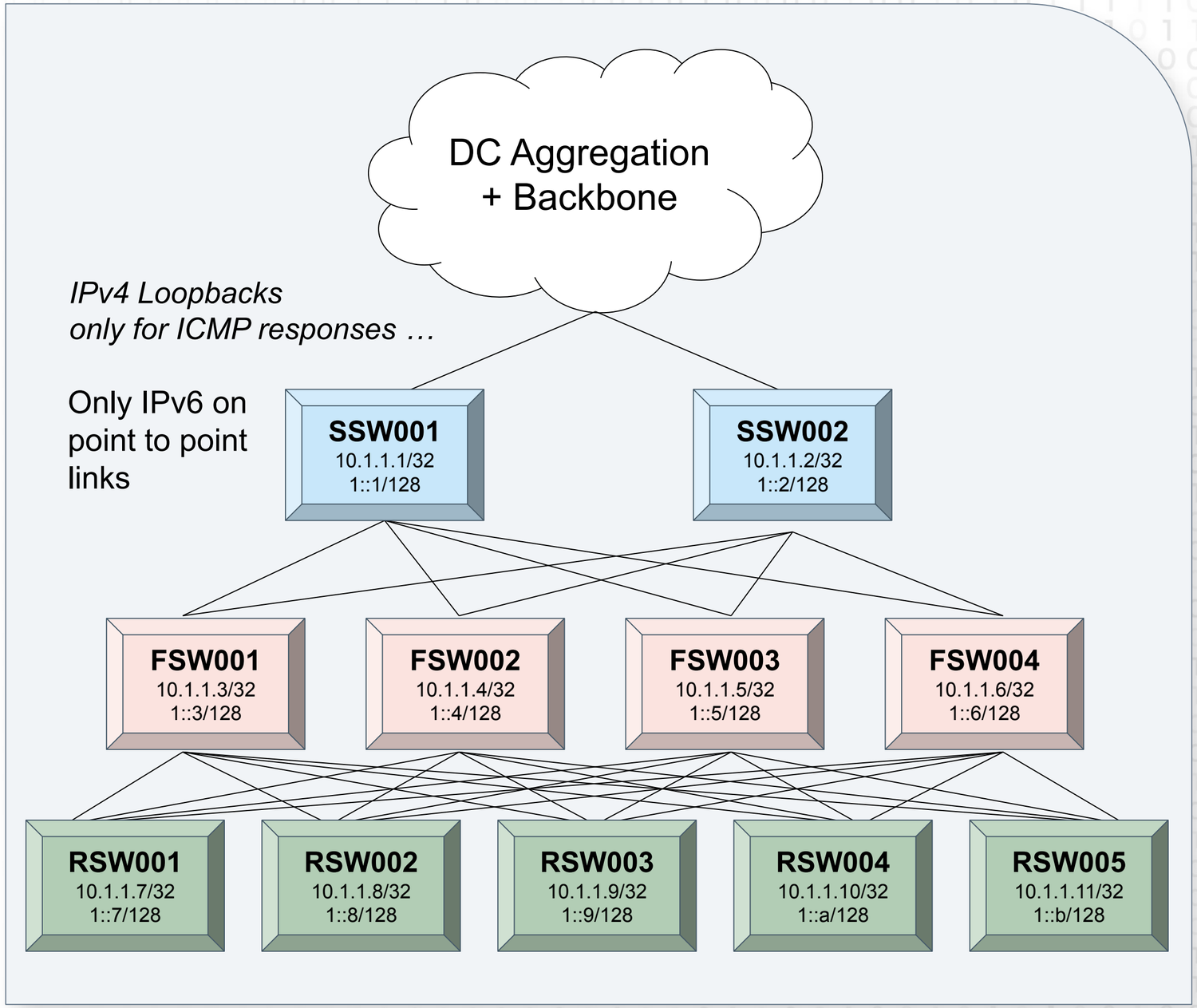
```
# IPv6 on Linux
[cooper:~]$ ip neighbor show
2401:db00:69:6969:face:0:69:0 dev eth0 lladdr ec:0d:9a:ce:b2:42 STALE
fe80::face:b00c dev eth0 lladdr 02:90:fb:61:5d:47 router REACHABLE
# IPv4 on Linux
[cooper:~]$ arp -an
? (169.254.0.10) at 02:90:fb:61:5d:47 [ether] on eth0
```

Yes, I know `arp` is deprecated. So is IPv4 …

`ip -4 neighbor show`

Logically a DC

IPv4 via IPv6

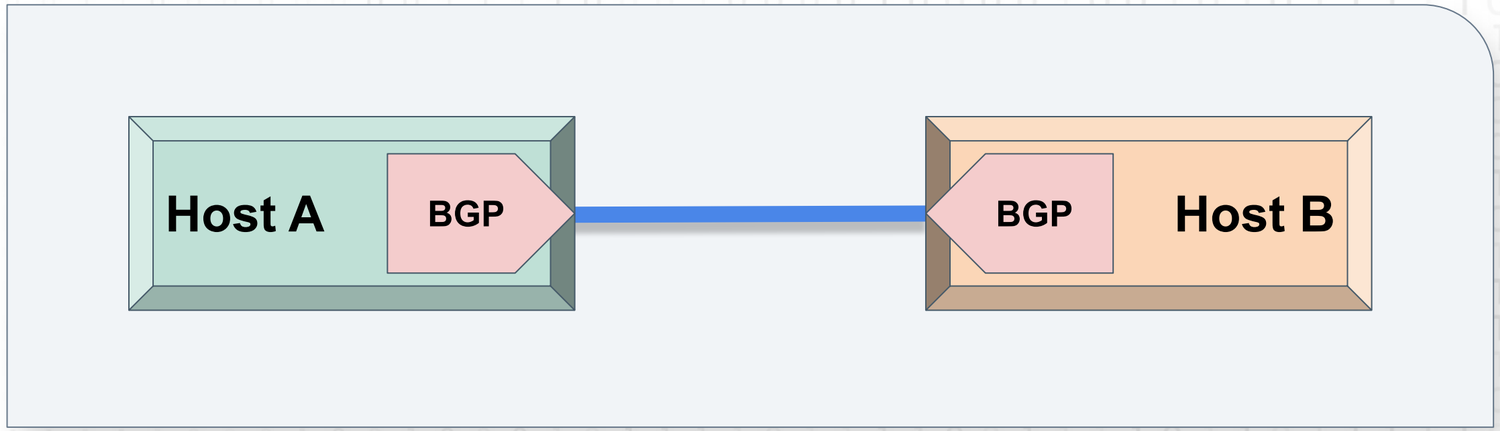


Meta DC

IPv4 via IPv6

```
[cooper:~]$ mtr -4 rsw048-inband.p086.f01.pci1
Host                                    Loss%   Snt   Last   Avg  Best  Wrst StDev
1. po1152.eb07.atn3.tfbnw.net            0.0%     1   36.0  36.0  36.0  36.0   0.0
2. eth2-5-1.ma07-06.atn3.tfbnw.net       0.0%     1   36.4  36.4  36.4  36.4   0.0
3. 10.242.148.1                          0.0%     1   36.4  36.4  36.4  36.4   0.0
4. 10.216.58.43                          0.0%     1   36.4  36.4  36.4  36.4   0.0
5. 10.216.21.15                          0.0%     1   36.3  36.3  36.3  36.3   0.0
16. rsw048-inband.p086.f01.pci1.tfbnw    0.0%     1   36.7  36.7  36.7  36.7   0.0
```

Dynamic Routing Protocols + v4 via v6



- **Routing Protocols**
  - RIB support for 128bit next hops
- BGP
  - Has RFCs for MP-BGP IPv4 via IPv6 capabilities

IPv4 via IPv6

BGP
RFCs

- RFC5549 (200905 - obsoleted)
  - MP BGP allow IPv6 NLRI (Network Layer Reachability Information) for IPv4 prefixes
  - Use cases
    - IPv4 Islands over IPv6 only cores
    - IPv4 VPNs over IPv6 Cores
  - Must advertise capability to peers

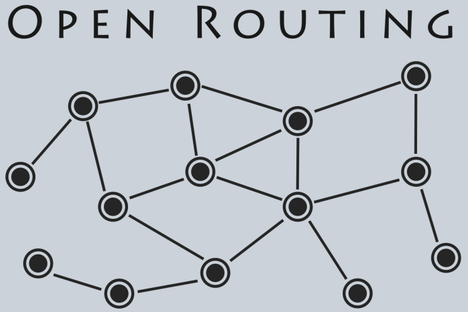
BGP RFCs

- RFC8950 - 202011
  - Changes the next-hop address encoding to a VPN-IPV6 address (from 5549)
    - From 16/32 bytes to 24/48 bytes - “Extended Next Hop Encoding Capability”
  - Adds ability for IPv4 VPN Multicast over IPv6 core
  - Interoperable with 5549 peers

IPv4 via IPv6 Dynamic Routing @ Meta

IPv4 via IPv6

BGP++



- In house C++ BGP daemon “BGP++”
  - Has non VPN RFC5549 capability
- Open/R our Link State IGP
  - Config option to enable IPv4 prefixes with IPv6 next hops
- Arista + Cisco Interop @ Edge POPs
  - interoperating with FBOSS (BGP++) and servers injection via BGP
    - E.g. www.facebook.com VIPs

How can I deploy?

- **2 addressing options**
  - Address Point to Point Links with
    - Site Scope IPv6
    - Global Scope IPv6
  - Use IPv6 Link Local
    - fe80::/10
      - Common link local addressing for Default Gateway
      - e.g. fe80::1 (uplink) fe80::2 (downlink)

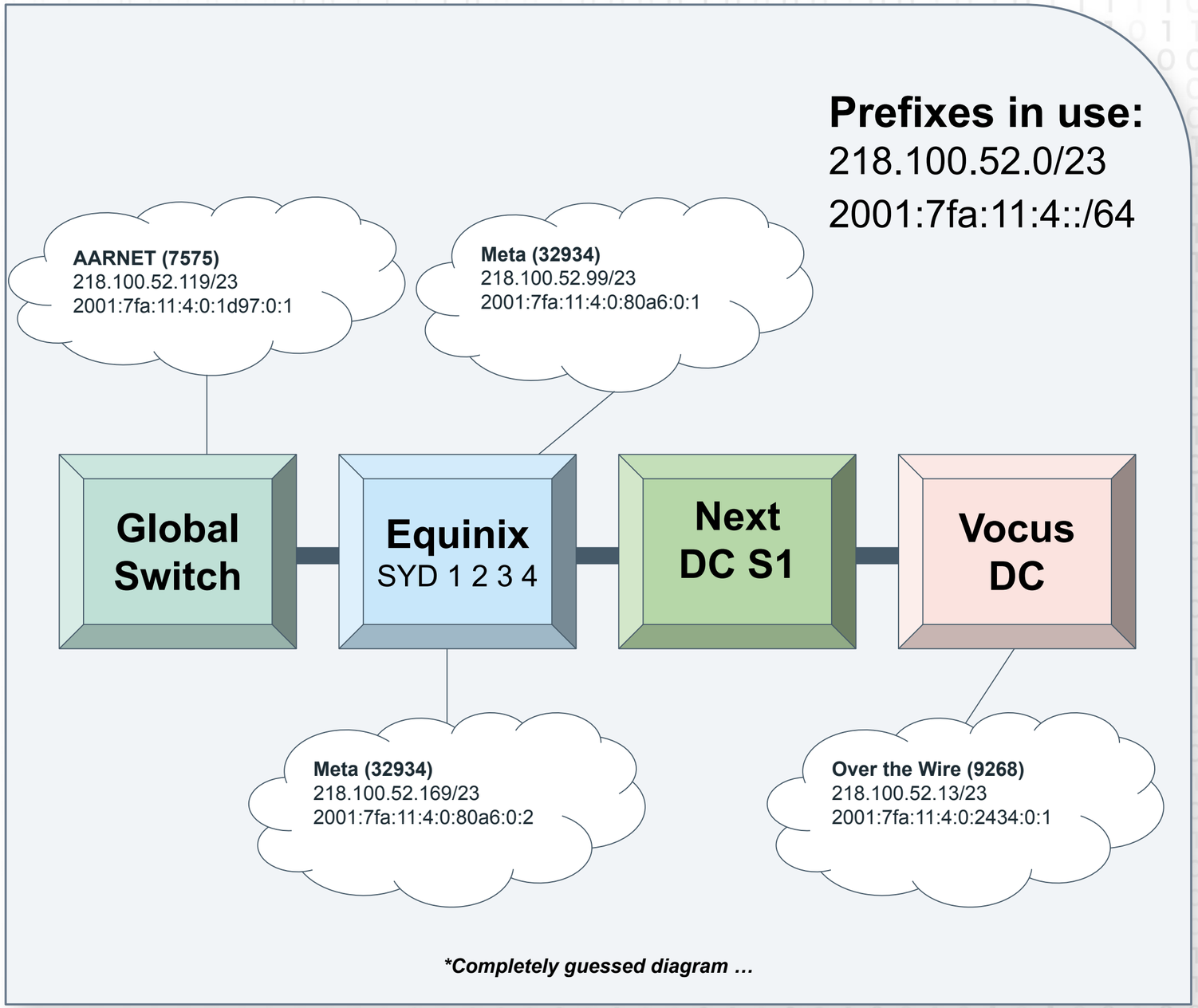
Potential Use Cases

- ISP point to point links
  - Especially if a customer is only receiving a /32
- Peering Exchanges
  - Just route the v4 over the v6 /64
- Server Access network/vlan
  - IPv6 Static address
  - Autoconf / DHCPv6

    And route to those addresses …

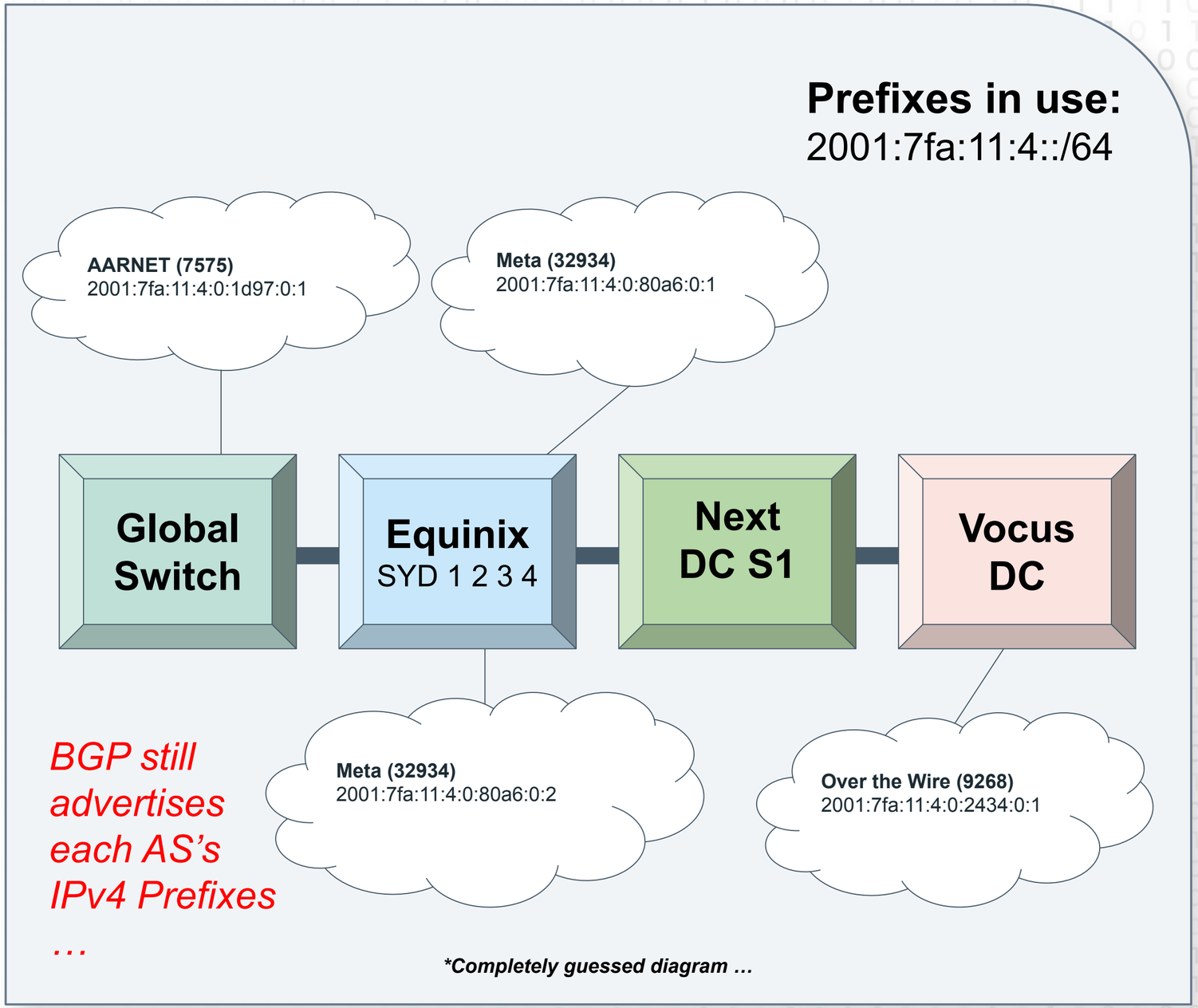
Case Study - NSW IX: **Potential Use Case**

IPv4 via IPv6



Case Study - NSW IX **Remove IPv4 …**

IPv4 via IPv6



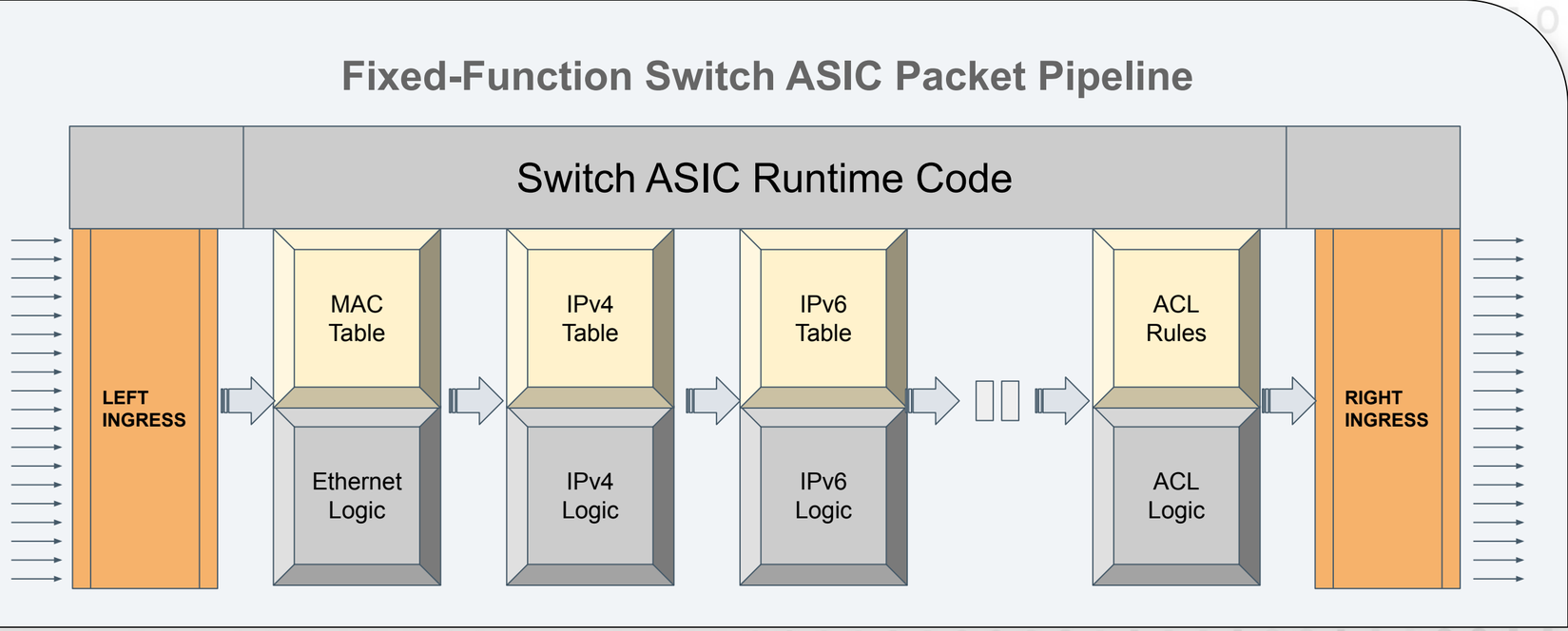
Case Study - NSW IX

- We get a */**23** back …
  - This is just one IX
    - Maybe need some loopbacks …
- No *tunneling* - so full MTU …
- Downsides
  - Loss of interface resolution for traceroute
  - CPE needs RFC5549/8950 support

Why remove IPv4?
Why care?

Reason to remove IPv4

Why remove IPv4?
Why care?

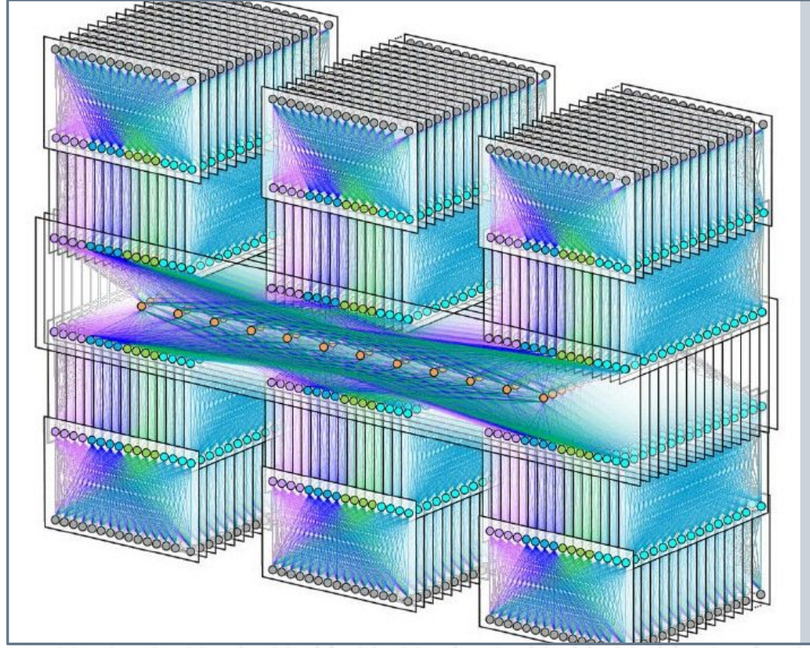


- #1 - Save Public Space
- Limit of RFC1918 address
  - e.g. **Meta has almost used all of 10.0.0.0/8** ***(and that's with no servers getting v4 for many years)***
- Turn off allocating resource for IPv4 in ASICs
  - No v4 store for ARP needed
  - Create more IPv6 FIB capacity
    - FIB = Forwarding Information Base

Start with less tech debt + Save Addressing

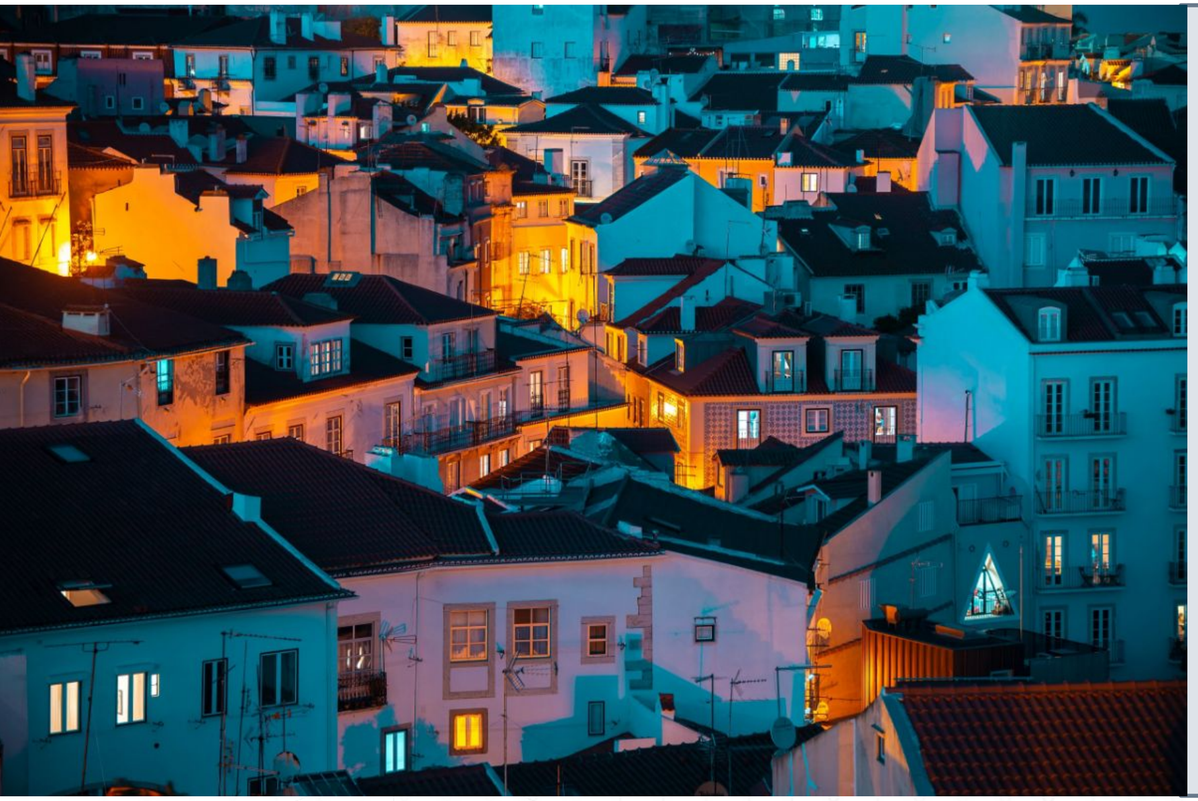
- Use IPv6 on point to point links only
  - Could even use IPv6 Link Local
- Less addressing to allocate
  - We can dream - One day we may not even need IPv4 - No addressing to remove!

Our New Region Designs



- Meta’s Fabrics are only planned to only get larger
- That’s more IPv4 for
  - Loopbacks
  - Point to point links
- Plan: use the same IPv4 Loopback prefix for each new Fabric being turned up (for ICMP replies)

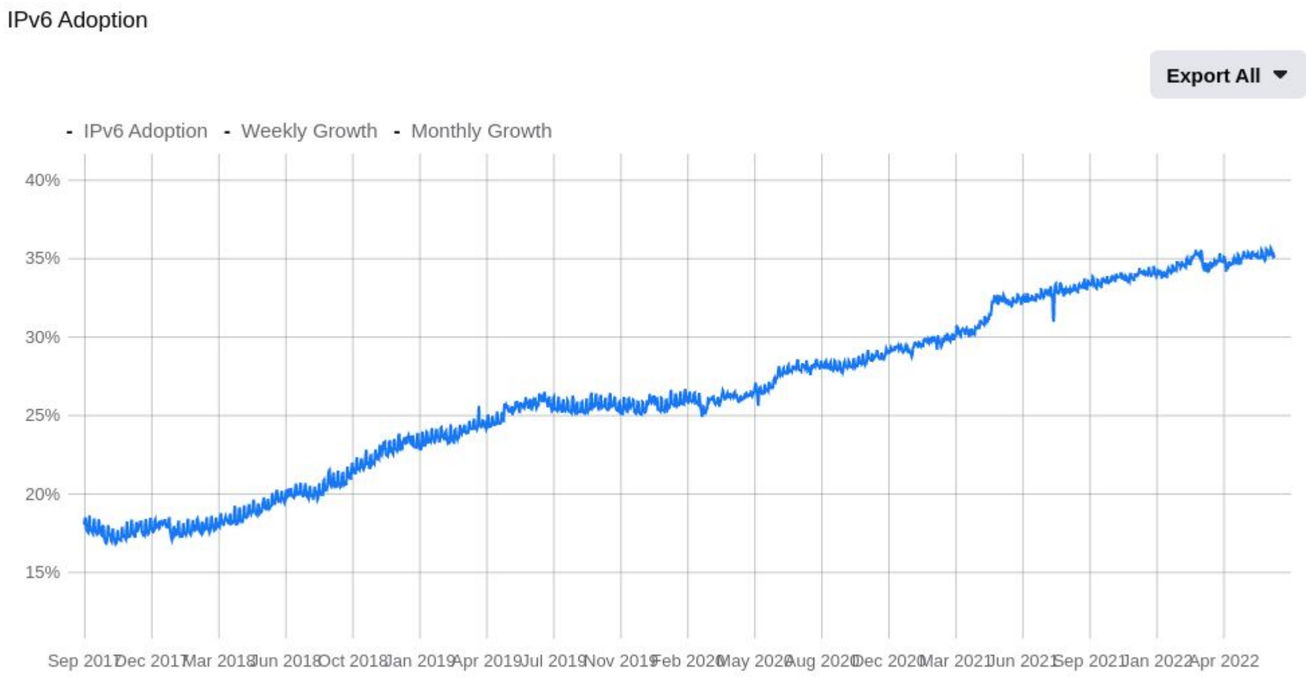
Why even IPv4?



- **You all …**
- IPv4 only ISPs
- IPv4 only Internet Services
  - You will be surprised at some big players that are still legacy only …

IPv6 Globally

Why even IPv4?

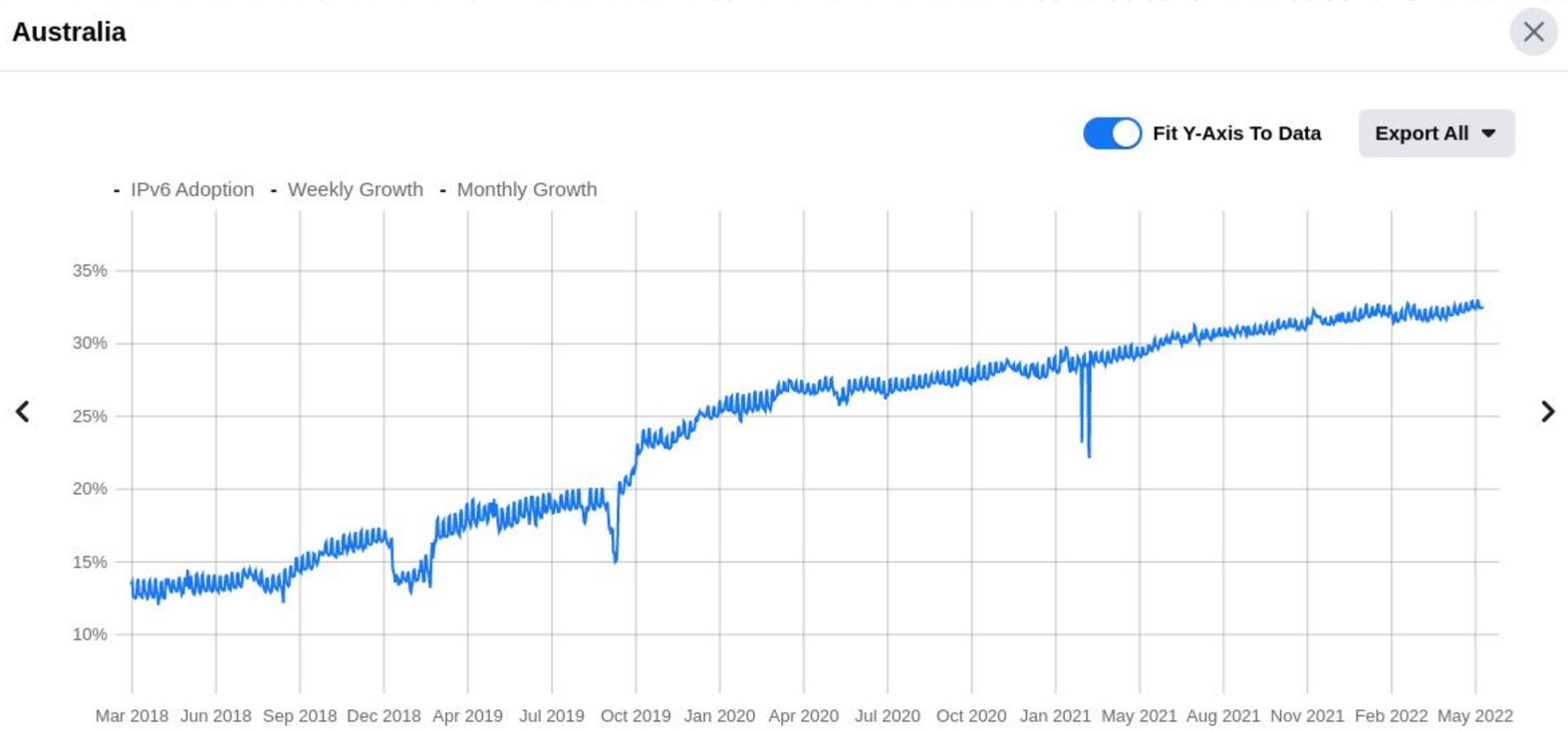


https://www.facebook.com/ipv6/?tab=ipv6_total_adoption

- 35% of Meta’s global users are IPv6 users
  - Includes Instagram, WhatsApp etc.

IPv6 in Straya 🇦🇺

Why even IPv4?

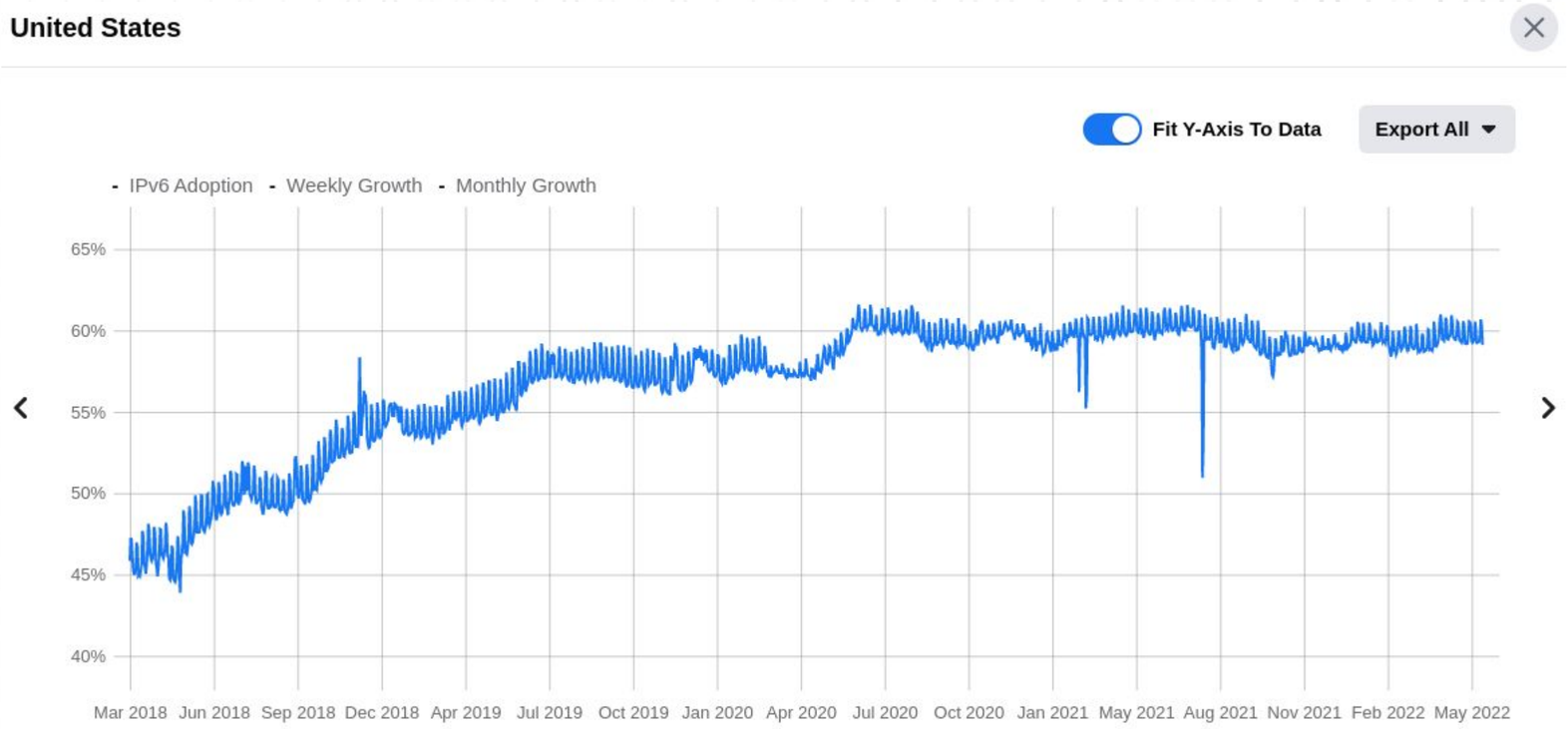


https://www.facebook.com/ipv6/?tab=ipv6_country

- 32% of Meta's AU users are IPv6 users
  - Catch up people!

IPv6 in Murica 🦅

Why even IPv4?



https://www.facebook.com/ipv6/?tab=ipv6_country

- 60% of Meta’s US users are IPv6 users
  - Can’t let the yanks better us!
  - Mainly mobile networks pushing this up …

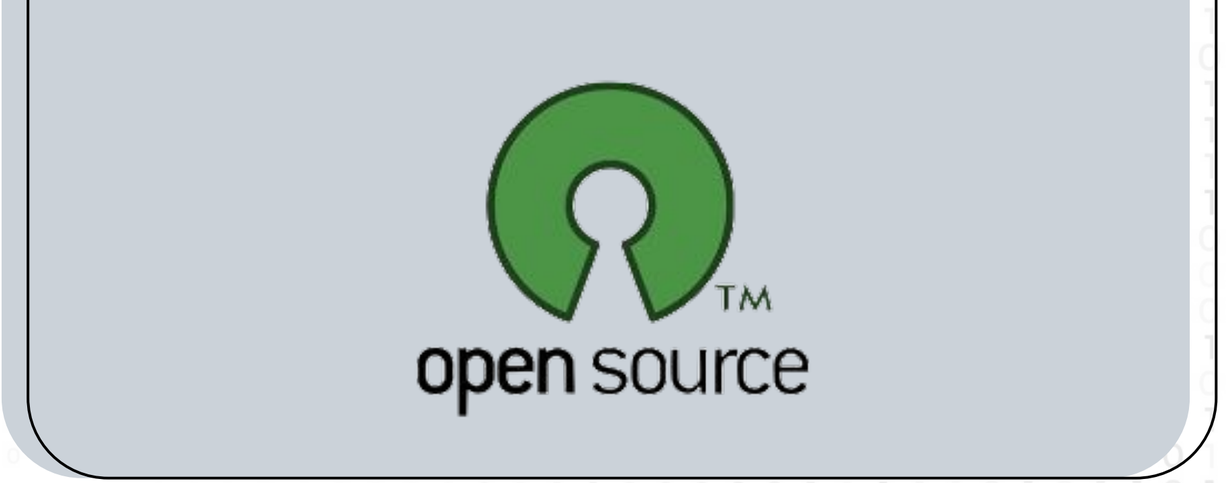
Things that won’t go away for a long time …

- Edge IPv4 VIPs
- IPv4 (dual stacked) Proxy Servers
- Some Data Center controlling hardware
  - E.g. temperature sensors
- We’re pushing all our V4 to edge POPs only

OSS Support

OSS Software

OSS + Vendor Support



- Linux Kernel >=5.2
  - Netlink (static route) support + forwarding
- Routing Daemons
  - ExaBGP - https://github.com/Exa-Networks/exabgp
    - But can not natively program netlink
    - Has simple plugin API
  - FRRouting
  - Open/R - https://github.com/facebook/openr

OSS Support: ExaBGP

```
template {
    neighbor nt {
        family unicast {
            ipv4 unicast;
        }
        nexthop {
            ipv4 unicast ipv6;
        }
    }
}


cooper@home1:~$ exabgp-cli show bgp summary
Peer              AS         up/down state       |     #sent     #recvd
fd00::1           65069     2:19:04 established            3         69
```

- ExaBGP
  - >= 4.1.0
  - `pip install exabgp`

OSS Support: FRR

```
home1.cooperlees.com# show run
…
router bgp 65069
 neighbor ALL_PEERS capability extended-nexthop


home1.cooperlees.com# show bgp ipv4
*  10.250.254.0/24  fd00:1::3           0 65070 65001 65002 ?
*>                  fd00::2             0 65001 65002 ?

cooper@home1:~$ ip route get 10.250.254.69
10.250.254.69 via inet6 fd00::2 dev wg0 src 10.255.0.3 uid 6969
    cache
```

- FRR
  - >= 7.0 - https://frrouting.org/
  - `apt/dnf install frr`

OSS Support: Open/R

```
{
  …,
  "v4_enabled": false,  # Default
  "v4_over_v6_nexthop": true,
  …
}

[netops@rsw069.p069.f69.crl6 ~]$ fboss openr unicast-routes
> 10.163.56.0/26
  via fe80::b4a9:fcff:fe0a:7bb3%fboss4007  weight 1
  via fe80::d8c4:97ff:feeb:6dcb%fboss4003  weight 1
  via fe80::b4a9:fcff:fe1b:622f%fboss4004  weight 1
  via fe80::b4a9:fcff:fe0a:7b2f%fboss4008  weight 1
  via fe80::d8c4:97ff:fed0:5325%fboss4005  weight 1
  via fe80::d8c4:97ff:feeb:6963%fboss4006  weight 1
  via fe80::d8c4:97ff:fed0:5607%fboss4001  weight 1
  via fe80::d8c4:97ff:fed0:5754%fboss4002  weight 1
```

- Open/R
  - >= many commits ago - https://github.com/facebook/openr
  - cd openr && ./build/build_openr.sh

Vendor Support

Vendor Support: Arista

```
Arista1(config-router-bgp)#show active
router bgp 1
 router-id 0.0.1.1
 bgp default ipv4-unicast transport ipv6
 neighbor 2000:0:0:40::2 remote-as 2
 neighbor 2000:0:0:40::2 maximum-routes 12000
 network 10.0.0.0/8
 address-family ipv4
   neighbor 2000:0:0:40::2 next-hop address-family ipv6
originate

Arista2#show ip bgp
BGP routing table information for VRF default Router identifier
0.0.1.1, local AS number 2
…
Network Next Hop Metric LocPref Weight Path
* > 10.0.0.0/8 2000:0:0:40::1 0 100 0 1 ?
```

- Arista
  - > = EOS-4.22.1.F

    Full RFC5549 non tunnel support
  - All Platforms

Vendor Support: Cisco

```
router bgp 65101
  address-family ipv6 unicast
    table-policy set-global-ipv6-nexthop  # Don’t prefer LL


show ip route
…
B    31.13.75.9/32 [20/0] via 2401:db00:f053:18:face:0:31:0
(nexthop in vrf default), 4d02h
                   [20/0] via 2401:db00:f053:18:face:0:7b:0
(nexthop in vrf default), 4d02h
```

- Cisco
  - NX-OS support (not verified) …
  - IOS-XR supports
    (seems natively if capability asked for)
    - >= 7.3.3

Vendor Support: Juniper

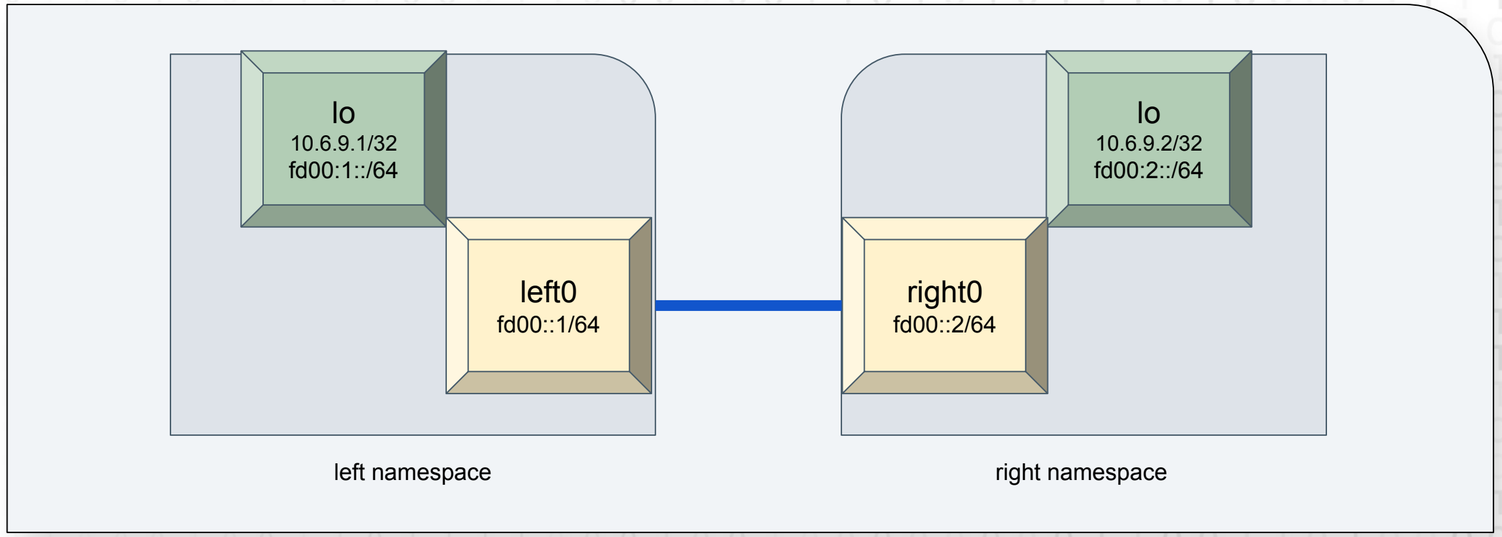
```
set protocols bgp group ebgp-v6 type external
set protocols bgp group ebgp-v6 export p1
set protocols bgp group ebgp-v6 peer-as 64496
set protocols bgp group ebgp-v6 neighbor 69::1
set protocols bgp group ebgp-v6 family inet unicast
extended-nethop
set protocols bgp group ebgp-v6 family inet6 unicast

set policy-options policy-statement p1 from protocol static
set policy-options policy-statement p1 then accept
```

- Juniper
  - >= 17.3R1
  - PTX >= Junos OS Evolved Release 21.2R1
  - Seems tunnels only or IPv4 still existing on the neighbor router is required from JunOS docs
  - More Info

Demo / Lab

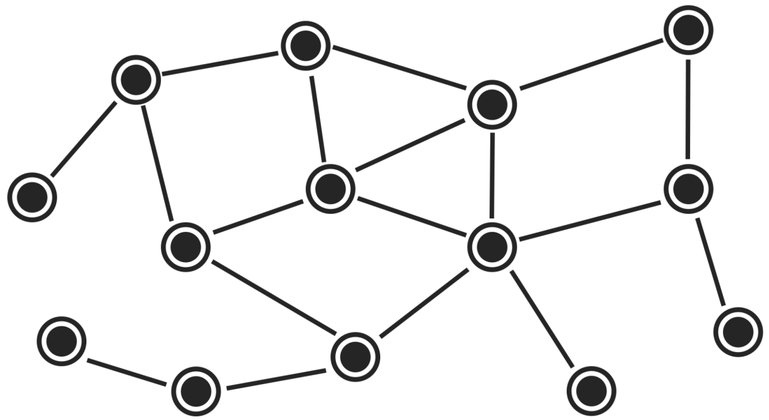
Risky Live demo time …

Labs are fun!



- Get a Linux box and run over to
  - https://github.com/cooperlees/v4v6demo
- We create two Network Namespaces
  - The get a v6 point to point link
  - We add routes to make the IPv4 address on lo reachable!

**Potentially guaranteed you'll become a Network god!*



Open/R v4 via v6

- Facebook's Open/R also has a similar lab
  - But you'd need to build Open/R from source
- https://github.com/facebook/openr/tree/main/openr/orie/labs/001_point_to_point_loopback

Summary

We don’t always need to have IPv4 addressing on point to point links now …

a. Let's save the tech debt where it makes sense

Think IPv6 first with all new systems

If you’re not Dual Stacked, please, prioritize it

Try IPv4 via IPv6 in your labs today!

Questions?

THANK YOU FOR YOUR TIME