RFC8950 for a V4LESS(-AS) World:

Diving into the magic of IPv4-with-IPv6 Next-Hops

Tobias Fiebig
Technische Universität Wien

Daniel Wagner
DE-CIX

Stop Doing IPv4 Driven Addressing Plans

- IPv4 around for +40 years and still no solution for making clean addressing plans!
- Why is there CIDR if it makes IP?
- Want to work on Layer 3? Find the Layer 2 address first!
- ‘Yes, please give me 50 different prefixes in a SINGLE metro!’

- There is a solution: Embrace our lord an savior RFC8950!

ARP & NDP

ARP:

- Figure out to which MAC address to send packets for an IPv4 address
- Construct ethernet frame with fitting destination MAC

NDP:

- Figure out to which MAC address to send packets for an IPv4 address
- Construct ethernet frame with fitting destination MAC

RFC5549, RFC8950, and a draft

RFC5549 & RFC8950:

- RFC5549 in 2009: What if we just put an IPv6 address into the nexthop field of an IPv4 prefix in BGP?

draft-ietf-intarea-v4-via-v6-08:

- How to actually handle an IPv4 route with an IPv6 nexthop
- Essentially:
  - Ask for MAC you'd have to send packets for the IPv6 nexthop to
  - Send the IPv4 packet there
- Passed IETF Last Call!
- Telechat date: 2026-05-21

RFC8950 & Draft Vendor Support

BGP AFI v4 with v6 Nexthop:

- JunOS
- Arista
- Cisco
- ExaBGP (no FIB)
- FRR
- Bird

OS-level support:

- JunOS
- Arista
- Cisco
- Linux (netlink)
- FreeBSD
- **NOT** OpenBSD :-(

RFC8950 & Draft Vendor Support

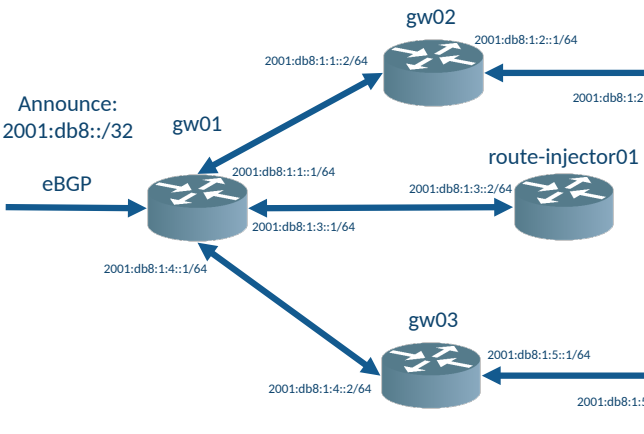
BGP AFI v4 with v6 Nexthop:

- *JunOS*
- Arista
- Cisco
- ExaBGP (no FIB)
- *FRR*
- *Bird*

OS-level support:

- *JunOS*
- Arista
- Cisco
- *Linux (netlink)*
- FreeBSD
- **NOT** OpenBSD :-(

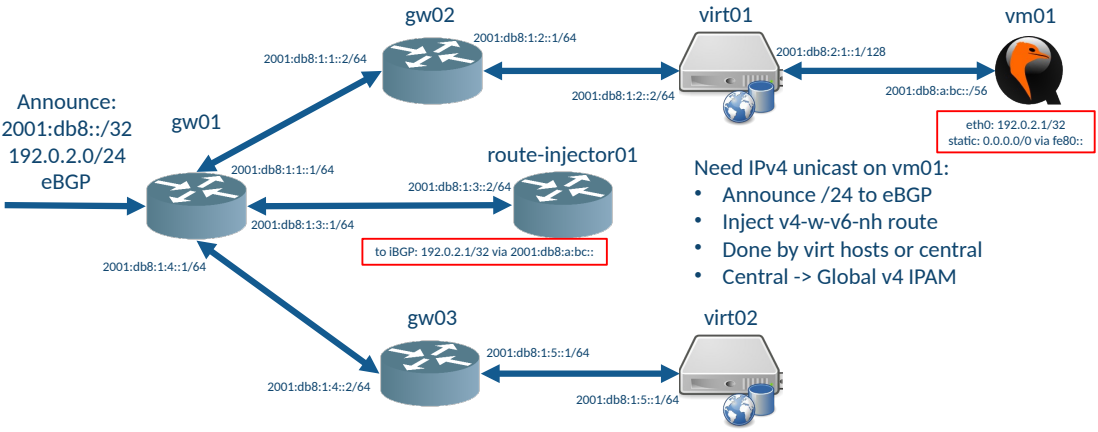
Addressing With Port... er v4-w-v6-nh



Setup:

- Numbered iBGP between GW's
- Virt. hosts speak BGP, announce VM assigned GUA to iBGP or other IGP
- Route assigned GUA to VM LL Address

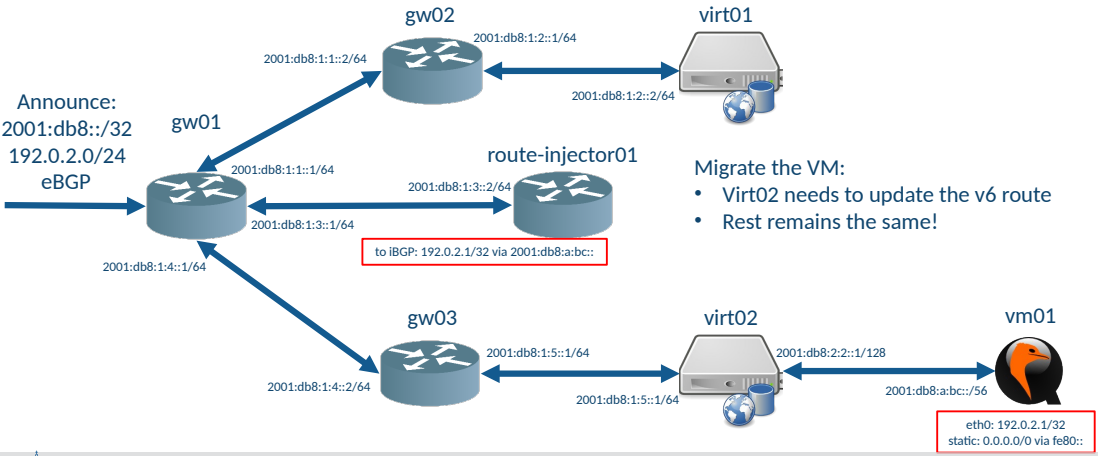
Addressing With Port... er v4-w-v6-nh



Need IPv4 unicast on vm01:

- Announce /24 to eBGP
- Inject v4-w-v6-nh route
- Done by virt hosts or central
- Central -> Global v4 IPAM

Addressing With Port... er v4-w-v6-nh



Migrate the VM:

- Virt02 needs to update the v6 route
- Rest remains the same!

Advantages of v4-w-v6-nh

- Fine grained (/32) routing of IPv4
- IPv4 as a complete add-on
- Clean IPv6 centric addressing plan
- No need for IPv4 transfer/router/network/broadcast addresses
  - This includes eBGP!
  - Really use all you IPv4 addresses IPv4 ‘follows’ an IPv6 address (prefix) around
- Technically there does not even have to be a loopback IPv4 address on routers; Nice for traceroute/PMTUD though
- Legacy services can be connected behind v4-w-v6-nh transport item This can be done partially (only for end-hosts, only for transport...)

Making it run: Linux

Config/Commands:

- `ip r a 192.0.2.0/24 via \ inet6 fe80::  dev eth0`
- `ip r a 192.0.2.0/24 via \ inet6 2001:db8::1`

Caveats:

- Easy to do in ifupdown :-)
- No support in any ‘framework’ (netplan, systemd-network...)
- if.up actions can help, but are not reliable.

/etc/networkd-dispatcher/routable.d/50-ifup-hooks:

```
#!/bin/bash
if [ "$IFACE" == "ens18" ];
then
    if [ "$STATE" == "routable" ];
    then
        if !  ip -4 r s | grep fe80::  > /dev/null;
        then
            ip -4 r a default via inet6 fe80::  dev ens18
        fi;
    else
        if ip -4 r s |grep fe80::  > /dev/null;
        then
            ip -4 r d default via inet6 fe80::  dev ens18
        fi;
    fi;
fi;
```

Making it run: Bird

Config/Commands:

- Route Injection

```
protocol static {
    ipv4;
    route 192.0.2.1/32 unreachable { bgp_next_hop = 2001:db8::1; };
}
```

- BGP Neighbor (template):

```
template bgp v4v6 {     ...
    ipv4 {
        ...
        extended next hop yes;
        add paths tx;
    };
}
```

Caveats:

- Works.
- ‘`add paths tx;`‘ is a surprise tool that will help us later.

Making it run: FRR

Config/Commands:

- Bird

```
...
router bgp 215250
  ...
  neighbor 2001:db8::1 capability extended-nexthop
  ...
  bgp allow-martian-nexthop
  !
  address-family ipv4 unicast
    neighbor 2001:db8::1 activate
    ...
```

Caveats:

- No static route injection (missing in the Interface)
- Older versions (pre-10.1ish) start sending RAs when activating extended nexthop
- Re-qualification of next-hops needs

  ‘`bgp allow-martian-nexthop`’:

  192.0.2.0/28 via 192.0.2.254 from 2001:db8::1

  192.0.2.254 via 2001:db8::1 from 2001:db8:1b66:1

Making it run: JunOS

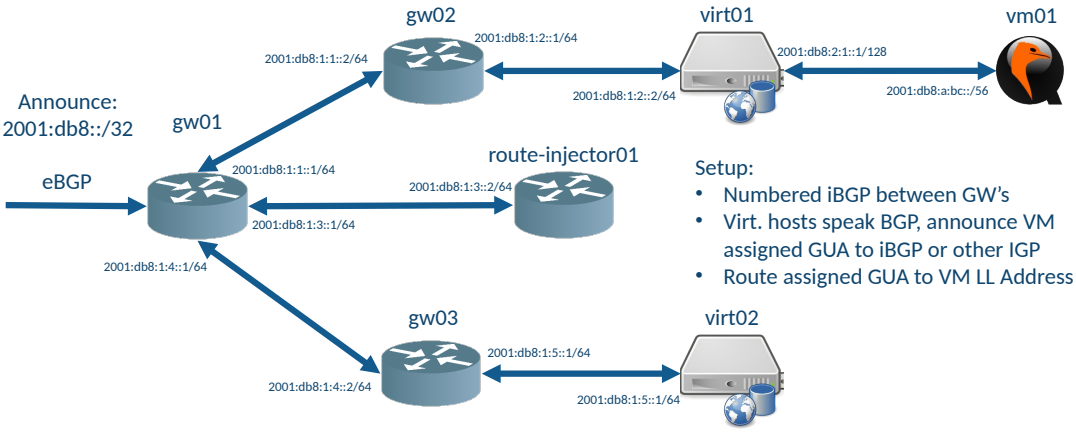
Config/Commands:

```
interfaces {
 xe-0/0/12 {
   family inet;
   family inet6 {
     address 2001:db8::2/64;
   }
 }
}
...
protocols {
  bgp {
    group v4v6 {
    ...
    neighbor 2001:db8::1 {
      ...
      ...
      family inet {
        unicast {
          extended-nexthop;
          add-path {
            receive;
          }
        }
      }
      family inet6 {
        unicast {
        add-path {
          receive;
        }
      }
    }
  }
}
```

Caveats:

- No static routes (AFAIK)
- Interfaces MUST have `family inet;` to pass IPv4 traffic!
- `add-path receive` will help us later!

Recall this!



Setup:

- Numbered iBGP between GW's
- Virt. hosts speak BGP, announce VM assigned GUA to iBGP or other IGP
- Route assigned GUA to VM LL Address

Making it run: Linux (with port... er Puppet)

Objective:

- Unify host config with netplan
- Handle routes from Puppet (this should be... fixed)
- Allow configuration of Anycast (BGP-on-host) and central injection
- Also do everything for IPv6 (because...)

Components:

- Puppet module (self-written and ugly)
- Bird route injectors in iBGP
- Bird on hosts (when anycasting)
- A lot of exported resources. :-)

class as215250v4v6::bird - Bird RR

class as215250v4v6::bird

```
Array $ibgp_peers = [],
String $asn = '64600',
String $local_ip = "${facts['networking']['ip6']}",
String $routerid = '10.0.0.0',
Hash $static = {},
Hash $static6 = {},
Array $blackhole = [],
Array $blackhole6 = [],
Boolean $v4v6collect = true,
```

Description:

- Configure reflector
- Configure static routes
- Configure blackhole routes (to drain route loops ;-))
- Set whether the reflector should collect exported resources

Puppet Example: Bird RR

```
class{'as215250v4v6::bird':
  asn => '59645',
  routerid => '10.6.66.12',
  ibgp_peers => [
    '2a06:d1c0::1',
  ],
  static => {
    '195.191.197.37/32' => '2a06:d1c1:f0::3:3',
  },
  static6 => {
    '2a06:d1c1:ee::/48' => '2a06:d1c0::dead:beef:5002',
  },
  blackhole => [
    '195.191.196.0/25',
  ],
  blackhole6 => [
    "2a06:d1c0::/30",
  ],
}
```

- Fully functional route reflector configured into the iBGP mesh
- Happily injecting routes
- Skipped *some* config lines for brevity. ;-)

class as215250v4v6 - The hosts

class as215250v4v6

```
Array $v4addr = [],
String $v4prefix = '32',
Array $v6addr =
['2001:db8::1/64'],
Array $v6addr_export = [],
String $if = 'eth0',
Array $ns = ['2a06:d1c7::'],
String $search =
'example.com',
String $gw = 'fe80::',
String $gw4 = 'v6',
Boolean $export_ip = true,
Array $bgp_communities = [],
Hash $routes = {},
Boolean $bgp_export = false,
```

- Configure the host via netplan
- Use an `exec` to set the v4-via-v6 default route
- Either export IPs as 'exported resources' (`@@as215250v4v6::v(4|6)route {}`) or per BGP (peer exported)
- Attach BGP communities to BGP exported routes
- Should(tm) see some... cleanup...

Puppet Example: Anycasted Service

```
class{'as215250v4v6':
 if => 'eth0',
 v4addr => [
   '195.191.197.20',
 ],
 v6addr => [
   "2a06:d1c2:a00d:300::1/56",
 ],
 v6addr_export => [
   "2a06:d1c7::/128",
 ],
 gw => 'fe80::200:5eff:fe00:256',
 ns => [
   '2a06:d1c7::64',
 ],
 search => 'ams01.as59645.net',
 bgp_communities => [
   '59645:204:59645',
 ],
 bgp_export => true,
}
```

- 195.191.197.20/32 and 2a06:d1c7::/128 announced in iBGP
- 59645:204:59645 ← AMS01 location community;
  For prepending in other PoPs. No route-selection on RR.
  This is why we use add-path!
- Of course also with IPv6!
- Exported via BGP, neighbor configured on RR via exported resource:

```
@@as215250v4v6::epeer{"${peerip} - Exported Peer from ${fqdn}":
  peer => "${peerip}",
  peer_asn => '65000',
}
```

- RR configured by collecting exported resource from RR:

```
As215250v4v6::Eexport <<||>>
```

RFC8950 on the IX

- Currently a Euro-IX WG looks into RFC8950 at the IX:
  `https://github.com/euro-ix/rfc8950-ixp`
- IXP prefixes should usually not be globally reachable anyway
- One *should* originate ICMP for IX interfaces from lo anyway
  (Otherwise PMTUD and traceroutes suffer from uRPF)
- There are no concerns about too many members anymore

RFC8950 on IXes: Prod-Action

- BCIX does RFC8950 on their RSes (rewrites next-hops)
  - ca. 112k unique IPv4 prefixes
- Bird (Maria Matejka) and DECIX (Daniel Wagner) are working on `draft-marenamat-grow-route-server-nh-translation-01`

Auto-Configuration

- Not really a thing(tm)
- `draft-equinox-intarea-dhcpv4-route4via6-02`
  - Add field to DHCPv4 (which can be used on v6 only networks via RFC7341, DHCPv4-over-DHCPv6)
  - Expired; Not a lot of enthusiasm
- `draft-vanmook-intarea-ipv6-resolved-gateway-00`
  - Ask IANA to designate an IP that can be used to signal ‘just use your IPv6 default gateway for IPv6 as well’
  - Fresh in (30th of April 2026)
  - Hopefully going *somewhere*!

RFC8950 on Transport & eBGP for me

- Removed all transfer IPv4 in AS59645 (except for that one OpeBSD router); Works since over a year on JunOS / VyOS
- Using private ASN eBGP underlay for LL/Loopback distribution in AS59645
- Setup a dedicated test setup which uses the no-IPv4-except- on-leaf approach using FRR (edge/dist) and bird (injection)
- Notable eBGP sessions without IPv4 (outside AS215250):
  - 2x Upstream from AS59645 to AS215250 (default route)
  - 1x Peering AS59645 to AS215250 (harvesting higher LPREF)
  - 1x Peering AS211286 to AS215250 (harvesting higher LPREF)

L3 Fabric: No ARP, No LACP, No Regrets

- 2 - N TORs per rack
- Full mesh among TORs
- Each server gets one NIC per TOR
- Only L3 interfaces; TOR speak BGP with each server
- RFC8950 only + ECMP

- *No configs yet; Still waiting to actually build it.*
  (Always looking for volunteers ;-))

V4LESS-AS: Testing RFC8950 in Practice

- Dedicated test setup:
  AS215250 / 45.91.12.0/24 / 2a06:d1c3::/32
  + 193.31.54.0/24 (standard hijack)
  + 193.31.55.0/24 (forged origin hijack)
- Running different test scenarios:
  - lo with/without IPv4
  - On-path MTU breaks
  - eBGP
  - RPKI/IRR invalid IPv4 prefixes
- RIPE Atlas & NLNOG RING nodes for introspection
- See: `https://measurement.network/services/v4less-as/`

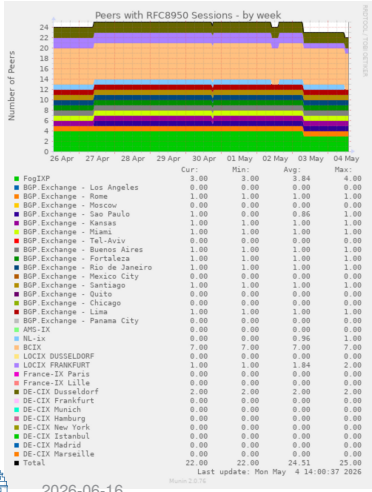
V4LESS-AS: How it works

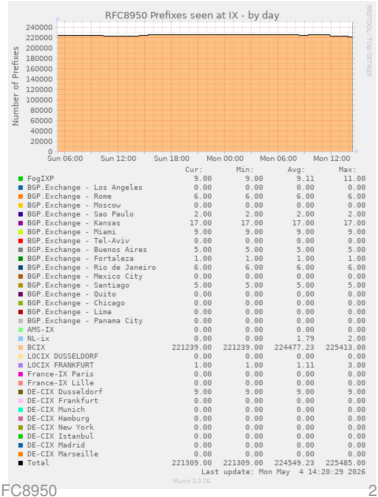
- Available at IXPs with pre-configured passive higher LPREF sessions for all members
- Fragmenting backhaul tunnels for clean 1500 MTU from the border
- You can establish an RFC8950 session and use the RIPE Atlas/NLNOG Ring nodes for testing

Currently active IXPs:

- BCIX
- FogIXP
- France-IX
  (Lille, Paris)
- DE-CIX
  (Düsseldorf, Frankfurt, München, Hamburg, New York, Istanbul, Madrid, Marseille)
- AMS-IX
- NL-IX
- TREX
  (Box is sitting in my office...)

V4LESS-AS: How it's going...





Caveats of v4-w-v6-nh

- Traceroutes are less (only router loopback) or completely useless (not even IPv4 loopback)
- Does not work for some client OSes yet
- Auto-Configuration is... a looooooong way out.
- Needs vendor support on routing infra (Extreme!)
- Works best with a clean IPv6 addressing policy
- You still need a working IPv6 IPAM

What About Vendors w/o RFC8950 Support?

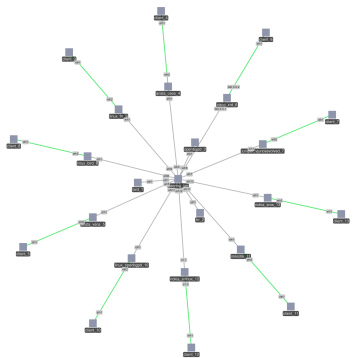
- Euro-IX RFC8950 Working Group tested RFC8950 interoperability between vendors
- Initially started to provide configs for IXP members to activate RFC8950 quickly

`https://github.com/euro-ix/rfc8950-ixp`

What About Vendors w/o RFC8950 Support?

- Euro-IX RFC8950 Working Group tested RFC8950 interoperability between vendors
- Initially started to provide configs for IXP members to activate RFC8950 quickly
- Set up a containerlab and spawned routers from a broad range of vendors
- Quickly identified non-supporting ones...
- Hard to attract new customers at IPv6-only IXPs
- Hard to motivate big IXPs to move to IPv6-only

`https://github.com/euro-ix/rfc8950-ixp`



Yet Another IETF Draft

- Basic idea:
  - Create a legacy-IPv4 network besides the IPv6-only peering LAN
  - Enable the IXP hardware / Route Server to translate between the networks
- Benefits:
  - No additional IXP hardware required
  - Integration of legacy customer hardware until upgrade / software patch
- Drawbacks:
  - Requires collision-free legacy IP address space
  - Additional logic to manage

`https://datatracker.ietf.org/doc/draft-marenamat-grow-route-server-nh-translation/`

State of the IETF Draft

- Waiting for adoption by GROW
- Discussion points
  - Take an collision-free IP block from the experimental IPv4 range?
  - How to handle additional complexity?
- All technical details in the draft
- Your opinion or support on the GROW ML is highly appreciated!

`https://datatracker.ietf.org/doc/draft-marenamat-grow-route-server-nh-translation/`

Key Take-Aways

- RFC8950 / v4-w-v6-nh is the future
  - Allows you to fully leverage all IPv4 you have
  - Build a clean IPv6 centric addressing scheme/architecture
  - Have IPv4 as a flexible add-on with central IPv4 IPAM
- Setup a session to AS215250 at a common IX and test it out
- Reach out to contact@measurement.network to sponsor a presence at an IX near you!
  - Needs: IX port, VM with 8-16GB memory, 2 cores, additional IPv6 only interface with static routing for mgmt / backhaul

`https://measurement.network/services/v4less-as/`

V4LESS-AS URI