Deploying a Production IPv6 Network in 30 Minutes or Less (or its free)

Richard A Steenbergen <ras@nlayer.net> nLayer Communications, Inc.

Identifying Our Target Audience

- There are many networks full of experienced IPv6 experts who have been running it successfully for years.
  - Sorry, this talk isn't for you.
- This talk is for the people who haven't done IPv6 because:
  - They're concerned about the potential impact to their IPv4 network.
  - They don't want to spend money on new hardware to do it.
  - They haven't adapted their management software to handle IPv6.
  - They're letting others walk on the bleeding edge first.
  - They don't see a commercial driver to make the effort worth it.
  - They just haven't gotten around to it.
  - Or they just don't like IPv6 (there *are* valid reasons).
- In other words: This talk is for you Cogent. ☺

The Only Known Alternative to Deploying IPv6

Goals of This Talk

- There are many possible ways to deploy IPv6
- But for the purposes of this talk, our goals are:
  - Learn how to deploy a dual-stack, native IPv6 network
    - No tunnels on dedicated low-end routers, or other cheating.
  - To do it without unnecessary hardware upgrades.
  - To do it without requiring major mgmt software changes.
  - To do it without requiring major management changes.
  - To make it scalable and easy to maintain.
  - To make it generally painless.
  - And most importantly, to not break IPv4 in the process.

A Quick Overview of the IPv6 Basics

Insert some quote from Sun Tzu here, to make this talk seem wiser than it actually is.

Zen and the Art of the IPv6 Address

What's wrong with you people? Decimals don't belong in AS numbers and colons don't belong in IP addresses.
–Matt Levine

- It's really not that hard, honest
  - IPv4 addresses had 4 groups of 8 bits separated by a .
  - IPv6 addresses now have 8 groups of 16 bits separated by a :
  - Oh and those decimal numbers (0-255) are now in hex (0-FF)
    - Just so we could implement 0xDEADBEEFCAFEBABE jokes.
    - You can also put decimal IPv4 addresses inside IPv6 addresses.
      - For example: 2001:db8::192.168.50.10 == 2001:db8::c0a8:320a
  - To ease the number headache, long runs of 0 can be replaced with a double colon (::), but only once per address.
    - 2001:db8::/32 is 2 blocks of 16 bit groups (hence the /32), follows by 0's
    - 2001:0db8:0000:0000:0000:0000:0000:0000/32 is the long way of writing it.
  - If in doubt about your hex math, use a calculator.

Woohoo more bits! Now what do we do?

- IPv6 addressing is best summed up as follows:
  - It's just like IPv4, but with 96 more bits.
  - An IPv4 host (/32) is now an IPv6 /128, etc.
- Except for the one minor glitch in The Matrix:
  - IPv6's "*suggested*" addressing scheme is to use a /64 as a host address, and reserve the last 64 bits for MAC-address based auto IP addressing.
  - But nothing about the protocol forces you to do this, it's just a suggestion.
  - And maybe not even a good one, since DHCP solved this problem already.
  - The concept (called an EUI-64 address) is a particularly bad idea on routing links, where you want to be able to identify the address of your neighbor. Therefore, you should *really* use static IPs on router circuits.

The IPv6 Dual-Stack Addressing Trick

The IPv6 Dual Stack Addressing Trick

- The real problems with IPv6 occur at Layer 8.
  - Most of us use some type of software to run our networks.
    - Customer Databases
    - DNS Management Tools
    - IP Allocation Database / Provisioning Tools
  - And a lot of it doesn't support IPv6 without major rewrites.
- I wanted to ease the transition from IPv4 to IPv6
  - So I thought about how to reduce the administrative overhead of an existing IPv4 network now deploying IPv6.
  - The best way to do that was to make the IPv6 addressing and DNS process map naturally from existing IPv4.
  - And the IPv6 Dual Stack Addressing Trick was born.
    - I'll take suggestions on a cooler name afterwards.

The IPv6 Dual Stack Addressing Trick

- Ok so how does it work?
  - Lets say you have the IPv6 block 2001:db8::/32
  - We peel off the first /96 from it to represent IPv4 space
    - You can use any block, but if you start with a lot of 0's it makes the address shorter and thus much easier on the eyes.
  - And convert the existing IPv4 address into IPv6
    - Convert from IPv4 to IPv6 prefix lengths by adding 96.
    - Convert to hex (a lot of software does this automatically too).
    - A circuit with the existing IPv4 address of 192.168.172.1/30
      - Becomes 2001:db8::192.168.172.1/126
      - Or written in the normal IPv6 hex form: 2001:db8::c0a8:ac01/126
    - Now every IPv4 enabled circuit in the network instantly becomes IPv6 enabled, with no modifications to our existing tools.

The IPv6 Dual Stack Addressing Trick

- Advantages
  - Instant IPv6 addressing with minimal work for every router, circuit, peer, and customer interface within your network.
  - Very easy to copy DNS data from IPv4 in-addr zone.
  - No special software/tools modifications required.
- Disadvantages
  - Requires an existing IPv4 address. Fine for dual stack but eventually you'll need a new scheme to go IPv6 only.
  - Doesn't accommodate customer allocations (/48s), you'll still have to allocate those, but it really helps on circuits.
  - May require manual translation to hex in older software.

The IPv6 Dual Stack Addressing Trick

- Juniper
  - All known JUNOS versions which support IPv6 support mixed hex and decimal addressing.
  - Address stays in the configuration as it was entered.
  - “Ping” will quickly convert from mixed to hex format.
- Cisco
  - Automatically converted to hex in CLI on recent IOS
    - 12.2(33)SR, SXH/SXI+ trains, etc
  - Not automatically converted on older code
    - 12.2(18)SXF, 12.0, etc
  - Not supported in XR either, go figure.

Routing Protocol Support

Routing Protocols - IGP

- Two main choices: OSPF and ISIS
  - If you're even thinking the word RIPng, **leave the room now**.
  - Seriously, enough is enough, can we let RIP die already?
- OSPF
  - Requires a new version of the protocol for IPv6, called OSPFv3, which is configured and operates completely independently from the IPv4 OSPF protocol.
- ISIS
  - ISIS supports IPv4 and IPv6 topologies simultaneously.
    - Which makes configuration and operation a lot simpler.
  - But you must choose: Single or Multi Topology Config.

Routing Protocols - BGP

- Modern BGP implementations support Multiprotocol
  - Multiple address families (AFIs) carried over a single bgp session regardless of the address of the BGP neighbor.
- Multiprotocol
  - Easier to configure and maintain, only a single session.
  - Easy to add IPv6 to your existing IPv4 BGP sessions.
- Native
  - Lets BGP act as a liveness test for your link/address.
- Recommendations?
  - I prefer multiprotocol for IBGP, and native for EBGP.

Show Me Some Configs Already

Tap Tap Tap - Is This Thing On?

- Cisco
  - Turn on IPv6: router(config)#ipv6 unicast-routing
  - Some platforms may require: router(config)#ipv6 cef
  - Certain hardware platforms also require recarving the TCAM to support IPv6, usually followed by a reboot.
    - 3560 type “top of rack” switches
      - router(config)#sdm prefer dual-ipv4-and-ipv6 default
      - **Note that this reduces your TCAM for IPv4 routes significantly.**
    - 6500/7600 type switches
      - router(config)#mls cef maximum-routes ipv6 64
- Juniper
  - Nothing special to get started.

Interface Configuration - Cisco

Existing Configuration

```
interface TenGigabitEthernet8/1
 description Something Here
 ip address 123.234.56.1 255.255.255.252

 ip router isis
 isis metric 10
```
-or-
```
 <network statement under router ospf>
 ip ospf cost 10
```

New Configuration

```
interface TenGigabitEthernet8/1
 ipv6 address 2001:db8::123.234.56.1/126
 (ipv6 address 2001:db8::7bea:3801/126)

 ipv6 router isis
 isis ipv6 metric 10
```
-or-
```
 ipv6 ospf 10 area 0
 ipv6 ospf cost 10
```

- ISIS multi-topology requires you to explicitly configure both the “isis metric” and the “isis ipv6 metric” values.
- ISIS single topology mode uses the “isis metric” value for all topologies.

Interface Configuration - Juniper

Existing Configuration

```
xe-1/2/3 {
    description Something;
    unit 0 {
        family inet {
            address 123.234.56.1/30;
        }
    }
}
```

New Configuration

```
xe-1/2/3 {
    description Something;
    unit 0 {
        family inet6 {
            address 2001:db8::123.234.56.1/126;
        }
    }
}
```

IGP Configuration – Cisco OSPF

Existing Configuration

```
router ospf 10
  router-id ###.###.###.###
  ispf
  log-adjacency-changes
  process-min-time percent 5
  nsf
  traffic-share min across-interfaces
  redistribute connected metric-type 1 subnets
  redistribute static subnets metric-type 1 route-map
      REDIST-STATIC
  passive-interface Loopback0
  maximum-paths 8
```

New Configuration

```
ipv6 router ospf 10
  log-adjacency-changes
  maximum-paths 8
  redistribute connected metric-type 1
  redistribute static metric-type 1 route-map
     REDIST-STATIC
```

IGP Configuration – Juniper OSPF

Existing Configuration

```
protocols {
    ospf {
        export REDIST-OSPF;
        area 0.0.0.0 {
            interface xe-1/2/3.0 {
                metric 10;
            }
            interface lo0.0 {
                passive;
            }
        }
    }
}
```

New Configuration

```
protocols {
 ospf3 {
        export REDIST-OSPF;
        area 0.0.0.0 {
            interface xe-1/2/3.0 {
                metric 10;
            }
            interface lo0.0 {
                passive;
            }
        }
    }
}
```

IGP Configuration – Cisco ISIS

Existing Configuration

```
router isis
 net 49.0001.####.####.####.00
 is-type level-2-only
 authentication mode md5 level-2
 ispf level-2
 metric-style wide
 no hello padding
 log-adjacency-changes
 nsf ietf
 redistribute connected
 redistribute static ip route-map REDIST-STATIC
 passive-interface Loopback0
```

New Configuration

```
address-family ipv6
  multi-topology
  maximum-paths 8
  redistribute connected metric 0
  redistribute static route-map REDIST-STATIC
 exit-address-family
```

IGP Configuration – Juniper ISIS

Existing Configuration

```
protocols {
    isis {
        export REDIST-ISIS;
        level 1 disable;
        level 2 {
            authentication-key "$password";
            authentication-type md5;
            wide-metrics-only;
        }
        interface lo0.0 {
            passive;
        }
    }
}
```

New Configuration

```
protocols {
    isis {
        topologies ipv6-unicast;
    }
}
```

Does It Work? - ISIS

```
user@juniperrouter> show isis adjacency detail
another.router
  Interface: xe-1/2/3.0, Level: 2, State: Up, Expires in 9 secs
  Priority: 64, Up/Down transitions: 1, Last transition: 14w5d 15:54:19 ago
  Circuit type: 2, Speaks: IP, IPv6, MAC address: 0:1d:71:98:56:80
  Topologies: Unicast, IPV6-Unicast
  Restart capable: Yes, Adjacency advertisement: Advertise
  LAN id: another.router.02, IP addresses: 10.20.30.2
  IPv6 addresses: fe80::21d:71ff:fe98:5680

ciscorouter#sh isis nei det
Tag null:
System Id      Type Interface   IP Address      State Holdtime Circuit Id
anotherrouter  L2   Te8/1       10.20.30.1      UP    11       anotherrouter.02
  Area Address(es): 49.0001
  SNPA: 0005.857b.afc0
  IPv6 Address(es): FE80::205:8500:467B:AFC0
  State Changed: 14w5d
```

IBGP Configuration – Cisco

Existing Configuration

```
router bgp 1234
 neighbor 1.2.3.4 remote-as 1234
 neighbor 1.2.3.4 update-source Loopback0
 !
 address-family ipv4
  neighbor 1.2.3.4 activate
  neighbor 1.2.3.4 next-hop-self
  neighbor 1.2.3.4 soft-reconfiguration inbound
  neighbor 1.2.3.4 send-community both
 exit-address-family
```

New Configuration

```
router bgp 1234
address-family ipv6
  neighbor 1.2.3.4 activate
  neighbor 1.2.3.4 next-hop-self
  neighbor 1.2.3.4 soft-reconfiguration inbound
  neighbor 1.2.3.4 send-community both
 exit-address-family
```

IBGP Configuration – Juniper

Existing Configuration

```
group IBGP {
    type internal;
    description "IBGP Sessions";
    import IBGP-IN;
    export IBGP-OUT;
    family inet {
        unicast;
    }
    local-address 20.30.40.50;
    peer-as ####;
    neighbor 1.2.3.4 {
        description "Another Router";
    }
}
```

New Configuration

```
group IBGP {
    family inet6 {
        unicast;
    }
}
```

- Note – Adding this will flap your BGP session.

Does It Work? - BGP

```
user@juniperrouter> show bgp summary
Groups: 74 Peers: 103 Down peers: 11
Table          Tot Paths  Act Paths Suppressed    History Damp State    Pending
inet.0           2379302     297170          0          0          0          0
inet6.0            11057       2031          0          0          0          0
Peer                     AS      InPkt     OutPkt    OutQ   Flaps Last Up/Dwn State
1.2.3.4                1234    6943416    6207121       0       0    28w4d1h Establ
  inet.0: 15612/65796/65796/0
  inet6.0: 127/881/881/0
```

```
ciscorouter#sh bgp ipv6 unicast summary
BGP router identifier 4.3.2.1, local AS number 1234
BGP table version is 5248900, main routing table version 5248900
1646 network entries using 238670 bytes of memory
3292 path entries using 250192 bytes of memory

Neighbor        V    AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/PfxRcd
1.2.3.4         4  1234 1403131 1253210  5248900    0    0 15w4d        1646
```

Static Routes

Cisco Configuration

```
ip route 10.20.30.40 255.255.255.0 null0
ipv6 route 2001:db8::/32 null0
```

Juniper Configuration

```
routing-options {
    static {
        route 10.20.30.0/24 {
            discard;
        }
    }
    rib inet6.0 {
        static {
            route 2001:db8::/32 {
                discard;
            }
        }
    }
}
```

Prefix-Lists

Cisco Configuration

```
ip prefix-list EXAMPLE permit 10.20.30.0/24
ipv6 prefix-list EXAMPLE permit 2001:db8::/32
```

- Prefix-lists can share the same name for IP and IPv6, which list is used is determined based on how the prefix-list is referenced.

Juniper Configuration

```
policy-statement EXAMPLE {
    term ipv4 {
        from {
            route-filter 10.20.30.0/24 exact;
        }
        then accept;
    }
    term ipv6 {
        from {
            route-filter 2001:db8::/32 exact;
        }
        then accept;
    }
    then reject;
}
```

Wait This Seems Too Easy…
Where are all the “Gotchas”?

And Next-Hop Self Means What Exactly?

- A tricky difference between Juniper and Cisco
  - Both use “next-hop self” to set export next-hop to a specific address.
  - Cisco uses “update-source lo0”, so NHS means…
    - IPv4 routes will use the IPv4 address on lo0 for a next-hop
    - IPv6 routes will use the IPv6 address on lo0 for a next-hop
    - Cisco does the “right thing”.
  - Juniper uses “local-address 1.2.3.4”, so NHS means…
    - BGP protocol next-hop will be set to the local-address 1.2.3.4
    - Even if the route is not the same protocol as the session local-address.
    - For example, an IPv6 route sent next-hop self over an IPv4 signaled BGP session from local-address 1.2.3.4 will have its next-hop set to ::1.2.3.4, not an actual IPv6 local-address from the router.
    - This is probably not what you want (IMHO this is a design flaw).

Juniper Next-Hop Self Workaround

- If you can’t make it send a correct next-hop, you can make the next-hop it does send valid. Configure a secondary address ::1.2.3.4/128 on your Juniper loopback family inet6 interface…
- This will help for next-hop self, next-hop peer is still a problem.

```
interface {
    lo0 {
        unit 0 {
            family inet {
                address 1.2.3.4/32;
            }
            family inet6 {
                address 2001:db8::1.2.3.4/128 {
                    primary;
                    preferred;
                }
                address ::1.2.3.4/128;
        }
    }
```

IPv6 and MPLS Shortcuts

- It turns out that support for MPLS traffic engineering for IPv6 is rather limited on most current platforms.
  - “To traffic engineer, first you need traffic.” –Vijay Gill
  - IOS does not appear to support IPv6->MPLS next-hop mapping, 6PE is the only option for MPLS.
  - JUNOS recently added support for IPv6 MPLS shortcuts as of 9.3, prior code has no support either.

IPv6 Packet Filtering

- IPv4 uses ARP to resolve IP<->MAC on Ethernet
- IPv6 uses ICMP Neighbor Discovery instead
  - So people who blindly filter all ICMP are going to be in for a rude shock when their LAN stops working.
  - Or people who incorrectly rate-limit ICMP may be in for a more subtle but equally bad shock in the future.
  - Cisco attempts to help work around this with an implicit accept for Neighbor Discovery before the implicit deny on an ACL. Guess what happens to people who later add an explicit deny to their ACL without understanding this behavior, or the need for an explicit ND rule.

Other IPv6 Hardware Limitations

- Older platforms may not be able to do ECMP or link-aggregate hashing on IPv6 packets.
  - This can result in extremely poor load-balancing.
  - Fortunately there isn’t much IPv6 traffic out there, yet.
- The size of the IPv6 address can consume a lot of routing lookup resources in hardware.
  - For example, Cisco SUP720 packets can’t do a full hardware lookup on 128 bits of IPv6 address + layer 4 ACLs.

Summary

- IPv6
  - It's not the perfect protocol.
  - It's still finding its place in the world of production routing.
  - A lot of the currently deployed hardware will probably have significant problems scaling to large amounts of IPv6 traffic in the future.
  - Many routing protocols are still dependant on IPv4 in order for IPv6 to operate, IPv4 isn't going away soon.
- But it's still relatively simple to deploy a working IPv6 network today.

Send questions, complaints, to:

Richard A Steenbergen <ras@nlayer.net>