All of the registries, for the most part, assign initial blocks for
- Service provider /32
- Enterprise /48
What makes up a good addressing plan?

- Depends on the type of network, the size of the network, and problem to be solved
- Points to consider
  - Documentation
  - Ease of troubleshooting
  - Aggregation
  - Standards compliance
  - Growth
  - SLAAC
  - Existing IPv4 addressing plan
  - Human factors
Encode every IPv4 address in your network in an IPv6 address

At first it seems relatively simple:

10.10.10.10 (A0A0A0A)

2001:DB8:A0A0:A0A0::

Easy, right?
6to4?

• Don’t 6to4 and 6rd already do this?
  ▪ No, 6to4 defines how to make IPv6-only resources available to devices that have only an IPv4 path
• 6to4 style allocation requires a /16
• Yes, it does define how to associate IPv4 addresses with IPv6 ones…
• Requires a /32 assignment if a minimum subnet size of /64 is to be preserved
  ▪ Do you have or can you get a /32?
  ▪ Provides no information about the subnet mask
  ▪ Results in very large subnets
  ▪ Light documentation requirements as your existing IPv4 documentation is your IPv6 documentation
Algorithmic Approaches

- Subnetting issue
  
  10.10.10.0/24 (A0A0A0)

  2001:DB8:A0A:A00::/56

  Do we count the significant digits for the subnet?

  2001:DB8:A0A:A00::/56
Algorithmic Approaches

- What to do about hosts??
  - 10.10.10.17/24 (A0A0A10)
  - 2001:DB8:A0A:A10::/56?

- At least for non-static addresses, SLAAC is functional with no collisions, but a /56? That can be wasteful even by IPv6 standards.
What if we “round down” to /64?
10.10.10.17/24 (A0A0A10)

2001:DB8:AAA0::10/64?

Better, but let’s look at a point to point link.
Algorithmic Approaches

Point to Point Link Example

- Point to Point Link:
  10.10.10.1/30 (A0A0A10) for the remote site
  10.10.10.2/30 (A0A0A10) for the local site

If we follow the previous rule to the letter we get:
  2001:DB8:AAA0::1/64
  2001:DB8:AAA0::2/64

But using /64s on router-to-router links can be dangerous, causing loops on some platforms
Algorithmic Approaches
Point to Point Link Example

Better to use a /127:

2001:DB8:AAA0::1/127
2001:DB8:AAA0::2/127

Um, wait a minute. **What’s wrong here?**
2001:DB8:AAA0::1/127
2001:DB8:AAA0::2/127

• Those are **NOT** in the same subnet!! A /127 could be ::0 and ::1, or ::2 and ::3, but **NEVER** ::1 and ::2!!

• As a matter of fact, **NO** IPv4 /30 can ever cleanly map into a /127!!
Link Numbering Issues

- OSPFv3 masks this problem, unlike in IPv4
- Separation of addressing from the link state database means that OSPFv3 neighbor relationships will establish, even on links with mismatched addressing and/or masks
- Link-local based forwarding prevents address mismatches from being easily detected because traffic flows normally and traceroutes don’t appear too strange
To detect link numbering errors, look for “Uturn” routing:

```
$ traceroute6 2620:144:B0C::
traceroute to 2620:144:B0C:: (2620:144:b0c::), 30 hops max, 80 byte packets
  1 2620:144:8fc:: (2620:144:8fc::) 26.747 ms 26.730 ms 26.716 ms
  2 2620:144:b0c::2 (2620:144:b0c::2) 29.137 ms 29.222 ms 29.264 ms
  3 2620:144:8fc:: (2620:144:8fc::) 29.355 ms 29.335 ms 29.350 ms
  4 2620:144:8fc:: (2620:144:8fc::) 29.438 ms !H 29.433 ms !H 29.413 ms !H
```

Note hop 2 is the misnumbered address. This traceroute should have looked like this:

```
$ traceroute6 2620:144:B0C::
traceroute to 2620:144:B0C:: (2620:144:b0c::), 30 hops max, 80 byte packets
  1 2620:144:8fc:: (2620:144:8fc::) 32.473 ms 32.447 ms 32.427 ms
```
Link Numbering Issues

- 2620:144:b0c::2/127
  (should really be 2620:144:b0c::)
- 2620:144:b0c::1/127
- 2620:144:8fc::
Link Numbering Issues

• Should you number your links at all or just use link-local?
• Loopback interfaces usually show up so you know which routers traffic is following, so why waste address space on links?
Using equal cost multipath?

```
$ traceroute6 2001:DB8::5:2
traceroute to 2001:DB8::5:2 (2001:DB8::5:2),
30 hops max, 80 byte packets
  1  2001:DB8::6:1 (2001:DB8::6:1)  22.723 ms
    26.730 ms  26.716 ms
  2  2001:DB8::1:1 (2001:DB8::1:1)  80.233 ms
    * ms  72.173 ms
  3  2001:DB8::5:2 (2001:DB8::5:2) * ms
    99.223 ms  29.350 ms
```

Which link did it take?
Does your management system use link numbering for monitoring or circuit identification?

Are you really saving any significant addressing by not assigning addresses?
$ traceroute6 2001:DB8::5:2
traceroute to 2001:DB8::5:2
(2001:DB8::5:2), 30 hops max, 80 byte packets
  2  2001:DB8::4 (2001:DB8::4) * ms  88.322 ms * ms
  3  2001:DB8::5:2 (2001:DB8::5:2) * ms  90.123 ms  100.110 ms

Better, now we know which link is having issues.
• Hybrid Approach for multiple netblocks (Interop example)

For example:

199.45.0.0/21
199.45.8.0/22
45.0.0.0/15
Interop Example

199.45.0-11
199.45 is not significant, it’s the same across the space

45.0.0.0/15
45 is not significant, everything else is
At first thought, just drop the insignificant digits:

199.45.1.0 becomes 1.0 becomes 2001:DB8:100::/48 (or smaller subnet if desired)

Cool. 45.1.0.0 is 10.0.0 which is 2001:DB8:....

uh oh!
• Because of the re-occurrence of the 199.45.0.0 and 45.0.0.0 we couldn’t even just drop the 199 to disambiguate
• Why not just use the whole address?
• We only have a /32 to work with... how do we convert both 199.45.1.32/30 and 45.1.32.0/23
• Let’s take just the significant octets from the 199.45 space:
  199.45.8.10
• We’ll just encode those digits right after our prefix:
  2620:144:810::/64
What if we make a rule that the minimum IPv4 allocation size in 45/15 is /24?

Now we only have a single significant digit: 45.10.1.2

But since it overlaps with the 199.45 space, we’ll add 128: 10+128=138

2620:144:BA00::/48
Networks smaller than /64 can be desirable, especially using /127s for point to point links.

To avoid future breakage, allocate a /64 in your documentation but use the smaller block.

Similarly, reserve /48s for EVERYTHING you can, there’s no reason to allocate densely, there’s plenty of space.

If you have a complex network, allocate in a sparse way to enable easy aggregation.
You can indeed add convenience and save on documentation by using an algorithmic approach.

But **ONLY** if you have reasonably few IPv4 blocks, if you have 100s, you’ll probably need a different approach unless you can get a large enough v6 allocation.
Questions?

- Brandon Ross
- Network Architect
- Torrey Point Group, LLC
- bross@torreypoint.com

- See us at booth #8