The Who, How & Why of IPv6 Transit and Peering

• Relationships
  o Peers, Transit providers, agreements

• Technology
  o BGP peering
  o BGP filtering
IPv6 Transit and Peering Relationships

• Who should I Peer with?
  o Everyone you Peer with on IPv4

• Who should I buy Transit from?
  o Whoever you buy IPv4 Transit from

• What terms should I negotiate?
  o The same terms as are in your IPv4 agreements

• Revisit this once IPv6 traffic levels come up
Carpe Diem?

• This may be an opportunity to expand your peering
• Acquire new IPv6-only peers
  ○ In addition to IPv4-only and dual-stack peers
• The faster you build out IPv6, the more likely this is
IPv6 Peering and Transit

TECHNOLOGY
IPv6 BGP Peering

• Assuming existing IPv4 BGP sessions
  ○ Not a greenfield
  ○ eBGP and iBGP already established
Single Session

- Exchange IPv6 routes over existing IPv4 peering?
  - Single BGP session between peers
  - Leverages Multiprotocol BGP (MP-BGP / RFC 4760)
    - Adding IPv6 NLRI bounces the session
  - IPv4 and IPv6 have a shared fate
Dual Session

• Establish new, IPv6-only peering?
  o Two BGP sessions between peers
  o IPv6 topology independent of IPv4, and vice verse
    • Outages: IPv6 session goes down if IPv6 reachability is lost
    • Maintenance: IPv4 and IPv6 sessions don’t affect each other
    • Operational Clarity
Junos Configuration Example

```
routing-options {
    router-id 10.10.10.5
    autonomous-system 65000;
}
protocols bgp group v6_PEERS {
    type external;
    neighbor 2001:db8:8000:4200::1 {
        peer-as 64109;
    }
}
```
IOS Configuration Example

router bgp 65000
   bgp router-id 10.10.10.5
   no bgp default ipv4-unicast

! 

address-family ipv6
   neighbor 2001:DB8:8000:4200::1 activate
   no synchronization
exit-address-family

NOTE: When you add “address-family” the BGP config splits into three parts: ipv4, ipv6, and the “main” BGP config.
## IPv6 BGP Session Verification Commands

<table>
<thead>
<tr>
<th>Juniper Command</th>
<th>Cisco IOS Command</th>
<th>Co-Ordinating Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>show ipv6 neighbors</td>
<td>show ipv6 neighbors</td>
<td>Show IPv6 neighbor cache information</td>
</tr>
<tr>
<td>show interface</td>
<td>show ipv6 interface</td>
<td>Show IPv6 interface information</td>
</tr>
<tr>
<td>show route protocol bgp table inet6.0</td>
<td>show bgp all</td>
<td>Show IPv6 unicast routes [Network, Next Hop, Metric, LocPrf, Weight, AS Path, etc.]</td>
</tr>
<tr>
<td>show bgp summary</td>
<td>show bgp ipv6 unicast summary</td>
<td>Show IPv6 unicast peers [AS, Up/Down, # of learned prefixes, etc.]</td>
</tr>
<tr>
<td>show bgp neighbor</td>
<td>show bgp ipv6 unicast neighbors</td>
<td>Show detailed information about each peer</td>
</tr>
<tr>
<td>show route advertising-protocol bgp &lt;ipv6 addr&gt;</td>
<td>show bgp ipv6 unicast neighbors [&lt;ipv6 addr&gt; advertised-routes]]</td>
<td>Show IPv6 prefixes advertised to a peer</td>
</tr>
<tr>
<td>show route receive-protocol bgp &lt;ipv6 addr&gt; table inet6.0</td>
<td>show bgp ipv6 unicast neighbors [&lt;ipv6 addr&gt; received-routes]]</td>
<td>Show IPv6 prefixes received from a peer – requires “neighbor soft-reconfiguration”</td>
</tr>
<tr>
<td>show route &lt;prefix&gt;/&lt;length&gt;</td>
<td>show bgp ipv6 &lt;prefix&gt;/&lt;length&gt;</td>
<td>Show information about a specific IPv6 prefix</td>
</tr>
<tr>
<td>show policy &lt;policy-name&gt;</td>
<td>show ipv6 prefix-list [summary</td>
<td>detail]</td>
</tr>
<tr>
<td></td>
<td>show ipv6 prefix-list &lt;name&gt;</td>
<td></td>
</tr>
<tr>
<td>Use traceoptions with flags</td>
<td>debug bgp ipv6 unicast</td>
<td>Debug BGP IPv6 packets</td>
</tr>
<tr>
<td>clear bgp neighbor &lt;neighbor&gt;</td>
<td>clear bgp ipv6 unicast *</td>
<td>Clear the BGP Session to all peers</td>
</tr>
<tr>
<td>ping [ipv6]</td>
<td>ping [ipv6]</td>
<td>Send echo messages</td>
</tr>
<tr>
<td>traceroute [ipv6]</td>
<td>traceroute [ipv6]</td>
<td>Trace route to destination</td>
</tr>
</tbody>
</table>
IPv6 BGP Filtering

- Methodology not special, generally mirrors IPv4
  - Explicit, bogons/martians, maximum-prefix, prefix size, etc.
  - The devil’s in the details
- Three high-level “themes:”
  - Filtering customers
  - Filtering Peers and Transit providers
  - Filtering your own routes
Filtering Routes Coming From Customers

• Explicit filter:
  o Allow customer network(s)
  o Deny all else
  o Customer is responsible for updating you
    • Should be infrequent if at all in IPv6

• Prefix size?
  o In IPv4 it’s common to allow down to /32 (for blackholing, etc.)
  o Orders of magnitude more addresses in v6
  o If allowed in IPv6 (/128); **ensure proper maximum-prefix limits**
  o Filter out more specifics before announcing to Peer / Transit
    • Could filter to /48 if customer needs to leverage more specifics
Ingress Customer Prefix Filter Example

• Assume your customer’s prefix is 2001:db8::/32
• IOS:
  
  ```
  ipv6 prefix-list ipv6-from-customer permit 2001:db8::/32
  ipv6 prefix-list ipv6-from-customer deny 0::/0 le 128
  ```

• Junos:
  
  ```
  policy-statement ipv6-from-customer {
    from {
      family inet6;
      route-filter 2001:db8::/32 exact next policy;
    }
    then reject;
  }
  ```
Filtering Routes Coming From Peers & Upstreams

1. Generate prefix filters from IRR:
   - Everyone should register in an Internet Routing Registry (IRR)
   - Allows tool-based filter generation
   - As good as the data in the IRR
   - We have a chance to do this right from the beginning

2. Bogon and max-prefix
   - Some networks don’t use an IRR (yell at them)
   - Bogon filter rejects the crazy stuff (reserved, small prefix, etc.)
   - A maximum-prefix limit protects against route overload
   - Better than nothing

NOTE: The number of IPv6 prefixes being announced from Peers and especially from Upstreams is growing rapidly. Be prepared to monitor and update your limits frequently.
# IPv6 Bogon / Martian Filtering Basics

<table>
<thead>
<tr>
<th>Prefix</th>
<th>Description</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>::/0</td>
<td>Default Route</td>
<td>Deny, Exact</td>
</tr>
<tr>
<td>::/8</td>
<td>Contains Loop back Address (::1/128), Unspecified Address (::/128), IETF reserved Address (formerly IPv4-compatible IPv6 Address) (::/96), and IPv4-mapped IPv6 Address (::ffff:0:0/96)</td>
<td>Deny, Or Longer</td>
</tr>
<tr>
<td>2000::/3</td>
<td>Global Unicast</td>
<td>Allow, /48 Or Shorter</td>
</tr>
<tr>
<td>2001::/32</td>
<td>Toredo</td>
<td>Allow, Exact</td>
</tr>
<tr>
<td>2001:db8::/32</td>
<td>Documentation Address</td>
<td>Deny, Or Longer</td>
</tr>
<tr>
<td>2002::/16</td>
<td>6to4</td>
<td>Allow, Exact</td>
</tr>
<tr>
<td>3ffe::/16</td>
<td>Former 6bone</td>
<td>Deny, Or Longer</td>
</tr>
<tr>
<td>fe80::/10</td>
<td>Link-local Unicast</td>
<td>Deny, Or Longer</td>
</tr>
<tr>
<td>fec0::/10</td>
<td>IETF reserved Address (formerly Site-local Address)</td>
<td>Deny, Or Longer</td>
</tr>
<tr>
<td>fc00::/7</td>
<td>Unique-local Address</td>
<td>Deny, Or Longer</td>
</tr>
<tr>
<td>ff00::/8</td>
<td>Multicast Address</td>
<td>Deny, Or Longer</td>
</tr>
</tbody>
</table>
Example “Loose” Bogon Filter - IOS

```plaintext
ipv6 prefix-list ipv6-bogon-loose deny 0::/0
ipv6 prefix-list ipv6-bogon-loose deny 0::/8 le 128
ipv6 prefix-list ipv6-bogon-loose permit 2000::/3 le 48
ipv6 prefix-list ipv6-bogon-loose deny <your prefix> le 128
ipv6 prefix-list ipv6-bogon-loose permit 2001::/32
ipv6 prefix-list ipv6-bogon-loose deny 2001::/32 le 128
ipv6 prefix-list ipv6-bogon-loose deny 2001:db8::/32 le 128
ipv6 prefix-list ipv6-bogon-loose permit 2002::/16
ipv6 prefix-list ipv6-bogon-loose deny 2002::/16 le 128
ipv6 prefix-list ipv6-bogon-loose deny 3ffe::/16 le 128
ipv6 prefix-list ipv6-bogon-loose deny fe80::/10 le 128
ipv6 prefix-list ipv6-bogon-loose deny fec0::/10 le 128
ipv6 prefix-list ipv6-bogon-loose deny fc00::/7 le 128
ipv6 prefix-list ipv6-bogon-loose deny ff00::/8 le 128
ipv6 prefix-list ipv6-bogon-loose deny 0::/0 le 128
```
Example “Loose” Bogon Filter - Junos

```c
policy-statement ipv6-bogon-loose {
    from {
        family inet6;
        route-filter 0::/0 exact;
        route-filter 0::/8 orlonger;
        route-filter 2000::/3 prefix-length-range /3-/48 next policy;
        route-filter <your prefix> orlonger next policy;
        route-filter 2001::/32 exact next policy;
        route-filter 2001::/32 longer;
        route-filter 2001:db8::/32 orlonger;
        route-filter 2002::/16 exact next policy;
        route-filter 2002::/16 longer;
        route-filter 3ffe::/16 orlonger;
        route-filter fe80::/10 orlonger;
        route-filter fec0::/10 orlonger;
        route-filter fc00::/7 orlonger;
        route-filter ff00::/8 orlonger;
        route-filter 0::/0 orlonger;
    }
    then {
        reject;
    }
}
```
Strict Bogon Filters

• Not all of 2000::/3 has been allocated to RIRs
  - http://www.iana.org/assignments/ipv6-unicast-address-assignments/ipv6-unicast-address-assignments.xml

• Other prefixes may be considered “bogon”
  - http://www.team-cymru.org/Services/Bogons/fullbogons-ipv6.txt

• RIRs hand out different space in different sized blocks
  - Provider Assigned (PA) vs Provider Independent (PI) space
  - Micro-allocations, etc.

• Strict filters must be actively maintained
  - Constantly changing landscape by design
Filtering Your Own Routes

- Only allow you and your customer’s space
- Don’t deaggregate unnecessarily
  - Only /48 and shorter (larger) prefixes
- Use communities to identify and filter routes
  - Level of hierarchy – Filter more specifics
  - Type of customer – BGP customers more likely to multihome
  - Use – No need to advertise specific infrastructure prefixes
  - Locational – Allows regionalization for TE, etc.
Some Parting Thoughts

• Take the opportunity to get it right!
• Use an IRR
  o Register in an existing database
  o Register your own database
  o Synchronize your own filters with your IRR
• Consider using a comprehensive IPAM solution
  o Massive address space
  o DNS/DHCP complexities
• Consider registering in the PeeringDB
A Few References

• BGP Configuration:

• Route Filtering:
  - http://www.team-cymru.org/ReadingRoom/Templates/IPv6Routers/
  - http://www.space.net/~gert/RIPE/ipv6-filters.html

• IRR:
  - http://www.irr.net/
  - https://www.isc.org/software/irrtoolset
Questions?

@ChrisGrundemann
@c.grundemann@cablelabs.com
http://gogonet.gogo6.com/profile/cgrundemann