An introduction to IPv6

Terry Froy
<t.froy@qmul.ac.uk>

School of Physics and Astronomy
Queen Mary University of London

WLCG Workshop 2017, Manchester, 19-22 June 2017

GridPP UK Computing for Particle Physics

Queen Mary University of London

A Potted History of IPv4

- The Internet evolved from ARPANET.
- 32-bit address space deemed enough for closed-access ‘experimental network’.
- Overly-generous early allocation policies.
- IPv4 run-out predicted in early 90s.
- The Internet becomes ‘somewhat popular’.
- Lessons learnt from IPv4 are ‘somewhat significant’.

Why we need IPv6...

- IANA has no IPv4 space left.
- Use of NAT with IPv4 reduces performance.
- Multiple layers of NAT are **not** the answer.
- IPv6 restores end-to-end host connectivity to all.
- Larger address space permits continued Internet growth.
- Lack of available IPv4 and slow IPv6 deployment restricts future growth of WLCG.

IPv4 and IPv6 addressing 'differences'

- IPv4 addresses are sized at 32-bits and expressed in decimal octets:

  192.0.2.1 (address)

  192.0.2.1/255.255.255.0 (address and netmask)

  192.0.2.1/24 (address and prefix length)

- IPv6 addresses are sized at 128-bits and expressed in hexadecimal words:

  2001:0db8:0000:0000:0000:0000:0000:0001 (address)

  2001:0db8:0000:0000:0000:0000:0000:0001/64 (address and prefix length)

GridPP UK Computing for Particle Physics

Queen Mary University of London

IPv6 Address Shorthand

- Leading zeroes in an IPv6 address word can be omitted:

  2001:0db8:0000:0000:0000:0000:0000:0001/64

  2001:db8:0:0:0:0:0:1/64

- A **<u>single</u>** contiguous series of zeroes can be replaced with '::':

  2001:db8:0:0:0:0:0:1/64

  2001:db8::1/64

- You can use IPv4 notation to express the last 32-bits of an IPv6 address:

  2001:db8:a::192.0.2.1/64

  2001:db8:a::c000:201/64

  2001:0db8:000a:0000:0000:0000:c000:0201/64

IPv6 Addressing Plan

- There is no single right or wrong way to utilize your available IPv6 address space.
- Use what works for <u>**your**</u> site!
- At UKI-LT2-QMUL, we assign out of:

  2a01:56c0:4033::/48 [GridPP @ QMUL IPv6 Assignment]

  Hosts are assigned IPv6 addresses as per:

  2a01:56c0:4033:<VLAN ID in decimal>::A.B.C.D/64

  [A.B.C.D is the corresponding host IPv4 address]

IPv6 Prefix Lengths

- For networks which require use of dynamic address assignment, you should use a /64 prefix length.
- A single /64 is big enough to hold every single Ethernet interface in the world.
- Different prefix lengths should be used when and where appropriate; statically-addressed point-to-point links are one example.

IPv6 Router Advertisements

- IPv6 routers can be configured to send advertisements, both periodically and on request, with the following information:

  The IPv6 prefix/prefix length in use on this link.

  The IPv6 address of the router.

  Various ‘flags’ (or hints) which tell hosts how to behave.

IPv6 Router Advertisement Flags

- Dynamic addressing in IPv6 is **interesting**...

Flag	Behaviour
A [Autonomous]	Hosts will automatically generate a full 128-bit IPv6 address within the prefix advertised.
L [Link]	Host will install a route for the prefix – this should be set.
M [Managed]	Host should seek out a DHCPv6 server and request an IPv6 address.
O [Other]	Host should seek out a DHCPv6 server and request other configuration information; such as DNS resolvers, static routes, boot server, etc.

IPv6 Dynamic Address Assignment

- Hosts can auto-generate their own addresses using SLAAC (State-Less Auto-Address Configuration) or via DHCPv6 through use of the ‘A’ or ‘M’ flags.
- Until recently, DNS resolver information could only be obtained via DHCPv6 but if your router and hosts support the RDNSS (Recursive DNS Server) attribute, you can use that instead of/in addition to DHCPv6.
- DHCPv6 behaves in a very similar manner to regular DHCP with one important exception...

Device Unique IDentifier [DUID]

- DHCPv6 does not necessarily identify machines by MAC address.
- DUIDs can be one of:

  Link-layer address plus time [DUID-LLT]

  Vendor-assigned UID based on Enterprise Number [DUID-EN]

  Link-layer address [DUID-LL]

  UUID-based DUID [DUID-UUID]
- Not all DHCPv6 clients use the same DUID type by default.

IPv4 DNS

- BIND-style IPv4 example of A/PTR records:

```
$ORIGIN example.wlcg.
host     IN A    192.0.2.1
$ORIGIN 2.0.192.in-addr.arpa.
1        IN PTR  host.example.wlcg.

[tez@tetris] ~]$ host host.example.wlcg
host.example.wlcg has address 192.0.2.1
[tez@tetris] ~]$ host 192.0.2.1
1.2.0.192.in-addr.arpa domain name pointer
host.example.wlcg.
```

IPv6 DNS

- BIND-style IPv6 example of AAAA/PTR records:

```
$ORIGIN example.wlcg.
host    IN AAAA    2001:db8::1
$ORIGIN 0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.8.b.d.0.1.0.0.2.ip6.arpa.
1       IN PTR host.example.wlcg.
```

```
[tez@tetris] ~]$ host host.example.wlcg
host.example.wlcg has address 2001:db8::1
[tez@tetris] ~]$ host 2001:db8::1
1.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.8.b.d.0.1.0.0.2.ip6.arpa domain name pointer host.example.wlcg.
```

- Consider automating the generation of DNS zone data if you are not already doing so...

Packet Fragmentation

- IPv4 supports packet fragmentation.
- IPv6 does **not** support packet fragmentation.
- Packet fragmentation hides broken network paths.
- An intermediate router can break a large IPv4 packet into several smaller ones prior to forwarding.
- This is not permitted in IPv6; so, a router drops the packet and responds to the sender with **ICMPv6 Message Too Big** along with the size of packet that it **will** accept and forward.
- The sender receives the response and transmits a smaller packet.
- Repeat ad-infinitum along the traffic path until packet successfully reaches the destination.

Firewalling ICMPv6

- Be sure to read RFC4890 thoroughly.
  - Make sure your upstream network contacts have read it too… plus your NREN…
  - Excessively filtering ICMPv6 breaks IPv6 connectivity in lots of wonderful subtle ways.
- Does this matter ?
  - Do a Google™ search for ‘pMTU blackhole’
- How do I tell if my IPv6 connectivity is suffering from a path MTU (pMTU) blackhole ?

Testing pMTU Discovery

- Diagnosing a pMTU blackhole is easy (this test assumes your local network MTU is 8252):

  IPv4: `ping -s 8224 -M do 192.0.2.1`

  IPv6: `ping6 -s 8204 2001:db8::1`

- Where did '-s' values come from ?

	ICMP Payload (-s value)	ICMP Header	IP Header	Packet Size	Link MTU
IPv4 Ping	8224	8	20	8252	8252
IPv6 Ping	8204	8	40	8252	8252

Interpreting Results

- If ICMP succeeds and remote/intermediate network uses lower MTU than you, one or more intermediate routers are fragmenting IPv4 packets.
- If ICMP succeeds, your IPv4 traffic path is ‘clean’.
- If ICMP fails with no response, ICMP echo is either blocked and/or pMTU discovery is broken.
- If ping6 succeeds, your IPv6 traffic path is ‘clean’.
- If ping6 results in an ICMPv6 Message Too Big:

  ```
  From ae0.londtn-ban3.ja.net (2001:630:0:10::156) icmp_seq=1 Packet too big: mtu=1500
  ```

  pMTU discovery is working up to the router which sent the message... repeat the test but drop the ICMPv6 payload size as per the value returned with the message.
- If ping6 drops packets, repeat the test with smaller ICMPv6 payloads.
- Share the results with the community (and your site network team!)

Putting IPv6 into Production

- Start with the less-critical services first.
- Enable one service at a time.
- Publish AAAA records in DNS with low TTLs.
- Increase AAAA record TTLs to match your A record TTLs once you are satisfied all is well.

References

- RFC4890 - https://www.rfc-editor.org/rfc/rfc4890.txt

  Recommendations for Filtering ICMPv6 Messages in Firewalls, May 2007 (E. Davies, J. Mohacsi)

Questions ?