IPv6 Workshop

Doug Southworth, dsouthworth@tacc.utexas.edu

Corey Eichelberger, ceichelberger@tacc.utexas.edu

Who are we?

Doug Southworth

- 20+ years in IT (K-12, government, R&E)
- MSCE, Full stack dev, network guy - you name it
- Currently at TACC as a network performance analyst/engineer
- I know just enough to be dangerous about most IT subjects

Who are we?

Corey Eichelberger

- ~15 years working IT in R&E
- Many different hats over the years
  - Mostly Networking and Security
  - Systems/Storage
  - Other duties as assigned*
- HPC
- EPOC - Engagement and Performance Operations Center
- **Disclaimer: I haven't touched Windows since ~2013 and haven't worked in a campus environment since 2016.*

Thank you for having us!

IPv6 - Fundamentals

- This discussion attempts to contain:
  - Fundamentals that underpin IPv6
  - Addressing nuances and subnet allocations

Fundamentals that underpin IPv6 - History

- IPv6 adds to the most recent version of the Internet Protocol we all know and use (the IP in TCP/IP)

- IETF draft in 1998 to help solve the impending IPv4 exhaustion issues. (Ratification in 2017).

- M-21-07.pdf

  page 3: " By 2025 80% Federal networks must be IPv6 **only**! "

Fundamentals that underpin IPv6 - size

- IPv6 addressing is beyond vast at 2^128 (340 undecillion) addresses**
  - 340 undecillion - in the range of atoms in the entire earth.
  - A job that paid 390 trillion dollars/hr: one would have to work 24 x 7 x 365 for about 100 quadrillion years to earn 340 undecillion dollars (same timeframe as comfortable retirement these days).

- IPv4 is sitting at ~4.3 billion** addresses

- [Mandatory xkcd on IPv6]

TL;DR - We will have to work very hard to exhaust IPv6!

** with some slack for special addresses etc…

Fundamentals that underpin IPv6 - implications

- Few quick implications of all this delicious space:
  - Keep a Full internet routing table(s)?
    - IPv6 full tables eventually will be non-trivial.
  - One /64 is a internet of internets!
    - A single /64 subnet has 18,446,744,073,709,551,616 usable host addresses.
    - There are “only” 4,294,967,296 IPv4 addresses**
    - I have to count the commas…
  - Planning and design just as important if not more.

Fundamentals that underpin IPv6 - implications

- Keep it easy on yourself keep the prefixes non-fragmented!
  - BGP advertise as large of blocks as possible vs small ‘chunks’
- DNS
  - Size of IPv6 space
    - zone files can have ‘just a few’ entries… Hand curation for critical systems only.
- Security!
  - One does not simply:
    - `nmap -T4 -F 2001:db8::/32`
    - `Firewalls still let the packet hit the floor.`

Fundamentals that underpin IPv6 - Hexadecimal

Binary	Decimal	Hex
0000	0	0
0001	1	1
0010	2	2
0011	3	3
0100	4	4
0101	5	5
0110	6	6
0111	7	7
1000	8	8
1001	9	9
1010	10	A
1011	11	B
1100	12	C
1101	13	D
1110	14	E
1111	15	F

Fundamentals that underpin IPv6 - Addresses

Addressing fundamentals (take a peek at RFC-5952):

Quick example of the same address ‘tuned’:

2001:0db8:0000:1000:0000:0000:00a0:0000

2001:db8:0:1000::a0:0

- lower case is “highly” preferred - no need to shout your HEX
- leading zeros can be clipped
- :: is your friend but, can only be used *once*
- browsers may need [] as a hint (but we have DNS right?)
  - example: http://[2001:db8:0:1000::a0:0]:9100/metrics

Fundamentals that underpin IPv6 - Addresses

Quick example of the same "good" address:

2001:0db8:0000:1000:0000:0000:00a0:0000

2001:db8:0:1000::a0:0

Quick examples of "bad" addresses:

2001;0db8::1

2001:DB8::1::a0:0

Quick example of "ugly" addresses:

2001:dB8:AbCd::000e

Fundamentals that underpin IPv6 - Scopes

Why have one address when you can have many!

IPv6 Address Scopes:

- Global
  - Global network “out there”

- Link or link-local - fe80::/10
  - Immediate L2 collision domain

- Unique-local (ULAs)
  - fc00::/7 (space per RFC BUT, use a generator)
    - e.g. - https://unique-local-ipv6.com/
  - Can BREAK Dual Stack:
  - ULA use for EDGE CASES only.
    - Use case: Truly back-to-back IPv6 ONLY network DB server
    - Think back-end ONLY DB “quorum” cluster traffic.

Fundamentals that underpin IPv6 - Specials

Special-Purpose address allocations (small sample):

- ::1/128 - loopback
- ::/128 - unspecified address
- fe80::/10 - Link-Local unicast space
- 2001:db8::/32 - documentation space
- 3fff::/20 - augmented documentation space
- fc00::/7 - unique local
- 2002::/16 - 6 to 4 (e.g. - instruments that just cannot IPv6)

Internet Assigned Numbers Authority (IANA) has the:

Special-Purpose Address Registry

Fundamentals that underpin IPv6 - Multicast

ff00::/8 - destination addresses - [IANA link]

interface local

- ff01::1 (all nodes [iface - local])
- ff01::2 (all routers [iface local])

link-local

- ff02::1 (all nodes [link - local])
- ff02::2 (all routers [link local])

site-local

- ff05::2 (all routers [site local])

Do not confuse with Special-Purpose fc00::/7 ULA space

Fundamentals that underpin IPv6 - Allocations

Per - ARIN - FYI

"

2.14. Serving Site (IPv6)

When applied to IPv6 policies, the term serving site means a location where an ISP terminates or aggregates customer connections, including, but, not limited to points of presence (POPs), datacenters, central or local switching offices or regional or local combinations thereof.

2.15. Provider Assignment Unit (IPv6)

When applied to IPv6 policies, the term "provider assignment unit" shall mean the prefix of the **smallest block** a given ISP assigns to end sites **(recommended /48)**.

"

Fundamentals that underpin IPv6 - Allocations

ARIN (and your regional provider) are your best friends on your IPv6 adventures!

“

4.10. Dedicated IPv4 Block to Facilitate IPv6 Deployment

ARIN shall allocate a contiguous /10 from its last /8 IPv4 allocation from IANA. This IPv4 block will be set aside and dedicated to facilitate IPv6 deployment. Allocations and assignments from this block must be justified by immediate IPv6 deployment requirements. Examples of such needs include: IPv4 addresses for key dual stack DNS servers, and NAT-PT or NAT464 translators. ARIN staff will use their discretion when evaluating justifications.

“

Fundamentals that underpin IPv6 - Addressing

Few daily driver “tools” to stop second guessing: python3 and sipcalc

- python3 - Simple binary to and from hexadecimal
  - hex()
    - >>> hex(**0b**0111)
    - ‘**0x**7’
  - bin()
    - >>> bin(0x7)
    - '0b111'
  - int()
    - >>> int(0x7)
    - 7
    - >>> int(0b111)
    - 7

Fundamentals that underpin IPv6 - Addressing

Few daily driver “tools” to stop second guessing:

- sipcalc - Help keep your zen intact

```
:~$ sipcalc 2001:db8::/64
-[ipv6 : 2001:db8::/64] - 0
[IPV6 INFO]
Expanded Address  - 2001:0db8:0000:0000:0000:0000:0000:0000
Compressed address        - 2001:db8::
Subnet prefix (masked)    - 2001:db8:0:0:0:0:0:0/64
Address ID (masked)       - 0:0:0:0:0:0:0:0/64
Prefix address            - ffff:ffff:ffff:ffff:0:0:0:0
Prefix length             - 64
Address type              - Aggregatable Global Unicast Addresses
Network range             - 2001:0db8:0000:0000:0000:0000:0000:0000 -
                            2001:0db8:0000:0000:ffff:ffff:ffff:ffff
```

Fundamentals that underpin IPv6 - Addressing

2001:0db8:8765:4321:aaaa:bbbb:cccc:dddd

2001:0db8::/32 - “Organization”
(/32 has 65,536 - possible /48 s)

2001:0db8:8765::/48 - “ Standard prefix length”
/48 gets split into 65,536 - possible /64s

Recall: a single /64 is a quintillion addresses!
2001:0db8:8765:**4321**::/64

Keep it easy and stay in the /56 and /64. There is tech to ‘solve’ the /64 and beyond
(DHCPv6,SLAAC,RDNSS, etc..)

Fundamentals that underpin IPv6 - Planning

2001:0db8:8765:4321:aaaa:bbbb:cccc:dddd

Again single /64 is a quintillion addresses!
2001:0db8:8765:**4321**::/64

4321 (a /56)- could be each building or Location / Function
`00..`**`21`**`..ff` (a /64)- each vlan and P2P connection.

`2001:0db8:8765:4300:0000:0000:0000:0000 –`
`2001:0db8:8765:43ff:ffff:ffff:ffff:ffff`

Fundamentals that underpin IPv6 - Client

Link Local:

“IPv6 link-local addresses are addresses that can be used to communicate with nodes (hosts and routers) on an attached link. Packets with those addresses are not forwarded by routers.”

fe80::/10 - prefix (we won't run out)

fe80::/64 - your machine’s range

Put the link in the local with EUI-64

Fundamentals that underpin IPv6 - Client (Link Local + EUI-64 generation)

Say we have MAC: **70:07:12:34:56:78**

1. Insert a 0xFFFE into the middle 70:07:12:**ff:fe**:34:56:78
2. 7th bit-flip: bin(0x7),bin(0x0) -> = 0111 00**0**0
3. hex the 7th bit: 0111 00**1**0 -> hex(0x7),hex(0x2) = 72
4. Combine link local (fe80::/64) + (72:07:12:ff:fe:34:56:78)

fe80::7207:12ff:fe34:5678/64

Fundamentals that underpin IPv6 - ICMP

ICMPv6, that other ping

- ICMPv6 many control types.
- ICMPv6 types are integral providing error and information messages not just echo.
- Can easily be blocked by host based firewalls or even appliance based firewalls.
- tcpdump for the win.

Fundamentals that underpin IPv6 - Client Neighbor Discovery Protocol (NDP)

ICMPv6 - (Type 133) Router Solicitation
ICMPv6 - (Type 134) Router Advertisement
ICMPv6 - (Type 135) Neighbor Solicitation
ICMPv6 - (Type 136) Neighbor Advertisement
ICMPv6 - (Type 137) Redirect

A ICMPv6 packet is 40 bytes long, and the first 8 bits of the ICMPv6 header specify its type.

Why the focus on ICMP and Link Local?

- ICMP is a critical service with IPv6!
  - Legacy ICMP blocking to “hide” from ping of death and such… ?
- Link-Local - “red phone” to your router and neighbors
  - Link-Local IS a primary communication method

Neighbor Solicitation and Advertisements

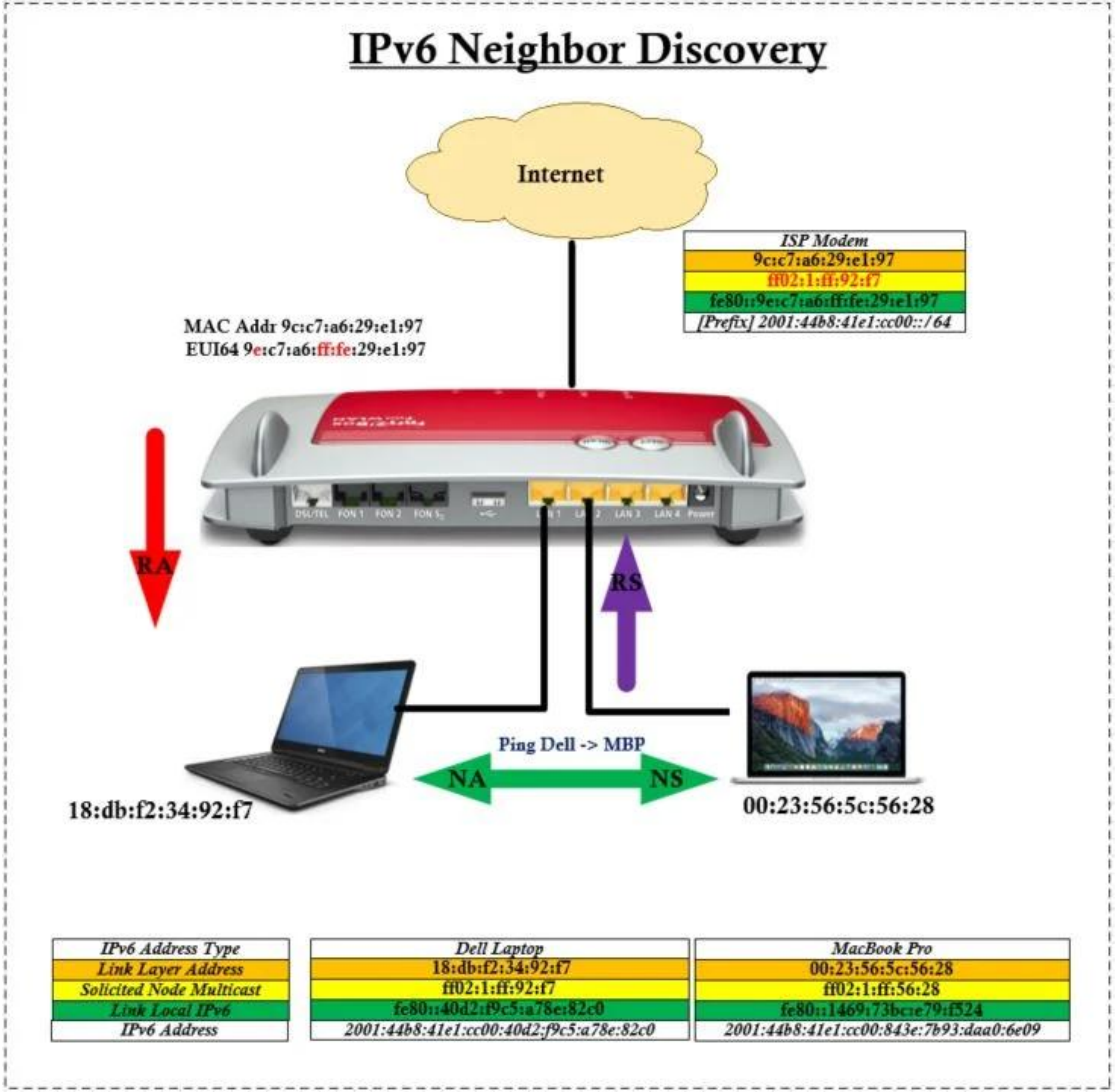
- Who is out there… [east west via north south]

- A Client Solicitation (typical):
  - Neighbor Solicitation sent via ICMPv6 (Type135) from
  - **Source**: Unicast IPv6 Address to
  - **Destination**: Solicited Node Multicast address of all routers (ff02::1)
- Info asked for: What is your link-layer address? Answer provided with data given of target link-layer address

Neighbor Solicitation and Advertisements

- The Answer via Advertisement:

- A Client Advertisement (typical):
  - All nodes reply sending Node Advertisement to the Solicit message ICMPv6 (type 136)
  - **Source**: Unicast IPv6 Address
  - **Destination**: unicast address of requestor -or - all nodes multicast (ff02::1)
- Answer: Here is my IPv6 and my Link-layer address (fe80)



RA - Router Advertisement

RS - Router Solicitation

NA - Neighbor Advertisement

NS - Neighbor Solicitation

Example of Neighbors from client.

```
root@gort:~# ip -6 neigh show
fe80::5054:ff:fee2:6fe dev enp3s0 lladdr
52:54:00:e2:06:fe REACHABLE

2602:43:649:a00::1 dev enp3s0 lladdr 40:8d:5c:f9:68:88
router REACHABLE

2602:43:649:a00::53 dev enp3s0 lladdr
52:54:00:e2:06:fe STALE

fe80::428d:5cff:fef9:6888 dev enp3s0 lladdr
40:8d:5c:f9:68:88 router REACHABLE
```

Fundamentals that underpin IPv6 - Client Router Advertisement (RA)

A Router Advertisement:

- Is every 200 seconds
  - configurable and able to be ‘asked for’ by clients
- Data about how to join the network
  - Lifetime
  - Reachable Time
  - MTU, Retransmission, etc…
- Flags are the “Flavor” of the network (DHCP / SLAAC / Mixed use, etc…)
  - **M**anaged Address Configuration (M)
  - **O**ther Stateful Configuration (O)
  - **A**ddress Autoconfiguration (A)
  - On-**L**ink (L)

Fundamentals that underpin IPv6 - Client Router Advertisement (RA) - FLAGS

- Router Advertisement Flags - How do you want your clients to ask?
  - **M**anaged Address Configuration (M)
    - DHCPv6 (1 = yes)
  - **O**ther Stateful Configuration (O)
    - Use DHCPv6 for more info (1 = yes)
  - **A**ddress Autoconfiguration (A) = SLAAC (0 = no)
  - On-**L**ink (L) = Topology
    - 0 = talk with the router for details
    - 1 = neighbors are on same ‘link’)

SLAAC with RDNSS RA via tcpdump

```
TCPDUMP of a ROUTER SOLICIT ( ICMPv6 type 134 ) message:

grobb3@tars:~$ sudo tcpdump -i br0 "icmp6 && ip6[40] == 134" -vv

10:39:03.025724 IP6 (hlim 255, next-header ICMPv6 (58) payload length: 168) fe80::428d:5cff:fef9:6889 > ip6-
allnodes: [icmp6 sum ok] ICMP6, router advertisement, length 168

          hop limit 64, Flags [managed, other stateful], pref medium, router lifetime 1800s, reachable time
0ms, retrans timer 0ms

            prefix info option (3), length 32 (4): 2602:43:62f:5300::/64, Flags [onlink, auto], valid time
86400s, pref. time 14400s

            route info option (24), length 24 (3):  ::/0, pref=medium, lifetime=1800s

            rdnss option (25), length 56 (7):  lifetime 1800s, addr: pi.hole addr: 2602:43:62f:5300::53 addr:
_gateway

            dnssl option (31), length 24 (3):  lifetime 1800s, domain(s): robbs.casa.

            source link-address option (1), length 8 (1): 40:8d:5c:f9:68:89
```

A IPv6 packet is 40 bytes long, and the first 8 bits of the ICMPv6 header specify its type

Fundamentals that underpin IPv6 - Client Router Solicitation (RS)

Client Ask: Dearest all IPv6 routers (ff02::2) on this segment tell me about the global-unicast prefix via link-local.
NOTE: Only the router will reply all other nodes discard in that they do not answer RS requests.

Router Answer: I heard an ask about the global-unicast on my link-local. Via my link-local dearest all nodes (ff02::1) global-unicast has the following prefix (2001:db8:000a:000e::) with with length of /64.

Fundamentals that underpin IPv6 - Client

How to get a client IPv6 IP Address… (a client point of view)

- Link-Local - cool we are done… fe80 + EUI-64
- Manual Stateful assignment
  - Humans…
  - And, we already have a Link-Local
- SLAAC
  - Stateless Address Auto-configuration
  - (No DNS hand-off ratified to RFC ‘yet’..)
  - there is RDNSS :-D
  - DAD
- DHCPv6
  - Vendor options - <3 (I’m biased http boot etc…) <3
  - DAD
  - ISC - Kea <-> Stork

Fundamentals that underpin IPv6 - Client (Manual Method)

1) Talk with a network administrator
2) Be assigned an address and gateway (DNS too)
3) Apply to your hardware in the usual method.

Error prone, time consuming, was that an echo-niner?
We already have link-local for 'free'.

Fundamentals that underpin IPv6 - Client (SLAAC)

Node configuration via **link-local** address

(local prefix + EUI-64 identification from MAC from process above)

DAD - Duplicate Address Detection

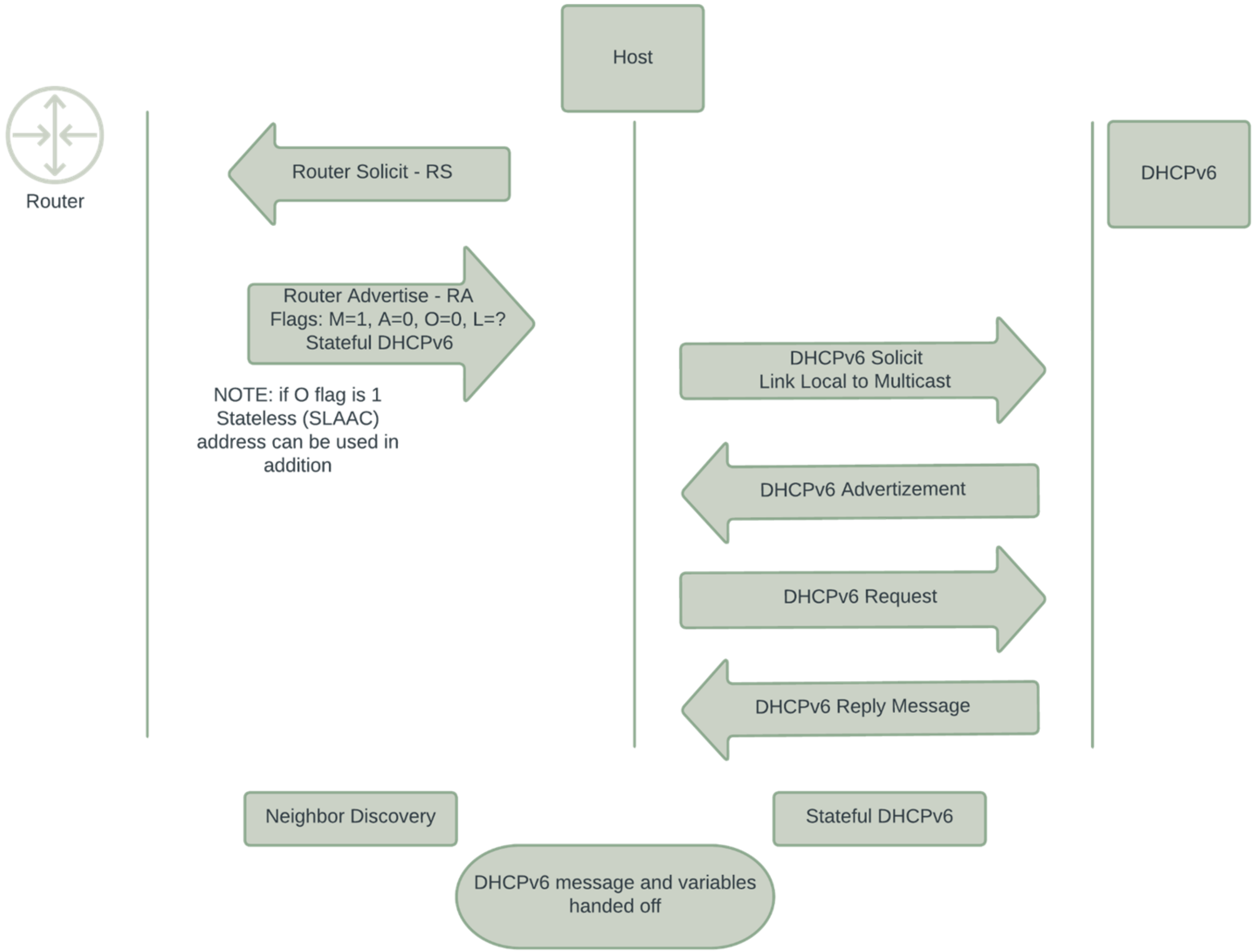
Solicited-node multicast (ff02:1::<last 6 hex>)

Node sends Router Solicitation via link local (ff02::2 - all routers) router hands back to source via (ff02:1 - all nodes)

Node configures Global Address and fires of DAD again in addition to a Neighbor Solicitation (ICMPv6)

Uhh where is my DNS (if there is not a RDNSS reply).
Can be fast and you get to know your neighbors.
TL;DR - We're Up with global link.

Fundamentals that underpin IPv6 - DHCPv6



Fundamentals that underpin IPv6 - DHCPv6

- Listeners are UDP -
  - Clients listen on port **546/UDP**
  - Servers and relays listen on port **547/UDP**

- DHCP unique identifier (DUID)
  - There currently four DUID types:
    - DUID-LLT (Type 1): using Link-Layer address plus Time (32-bits, seconds since 01.01.2000 00:00 UTC modulo 2^32) (MAC based)
    - DUID-EN (Type 2): using vendor assigned Enterprise Number
    - DUID-LL (Type 3): using Link-Layer address (MAC based)
    - DUID-UUID (Type 4): using a Universally Unique Identifier (UUID)

- Prefix and address delegation does not need to be based on DUID!!
  - RFC-6939 host reservation based on MAC (isc-dhcp / kea)

Fundamentals that underpin IPv6 - DHCPv6

```
"reservations": [
            {
                "hw-address": "3c:18:a0:44:5e:4f",
                "hostname": "burp.robbs.casa.",
                "ip-addresses": [ "2001:db8:401:a000::00bb" ],
                "option-data": [
                    {
                        "name": "dns-servers",
                        "data": "2001:db8:401:a000::53"
                    }
                ]
            }
```

Fundamentals that underpin IPv6 - DHCPv6

```
"reservations": [
            {
                "duid": "00:01:00:01:24:a8:d2:b1:00:50:56:8d:19:d1",
                "hostname": "burp.robbs.casa.",
                "ip-addresses": [ "2001:db8:401:a000::00bb" ],
                "option-data": [
                    {
                        "name": "dns-servers",
                        "data": "2001:db8:401:a000::53"
                    }
                ]
            }
```

Fundamentals that underpin IPv6 - DHCPv6

Client reservations - DUID vs MAC

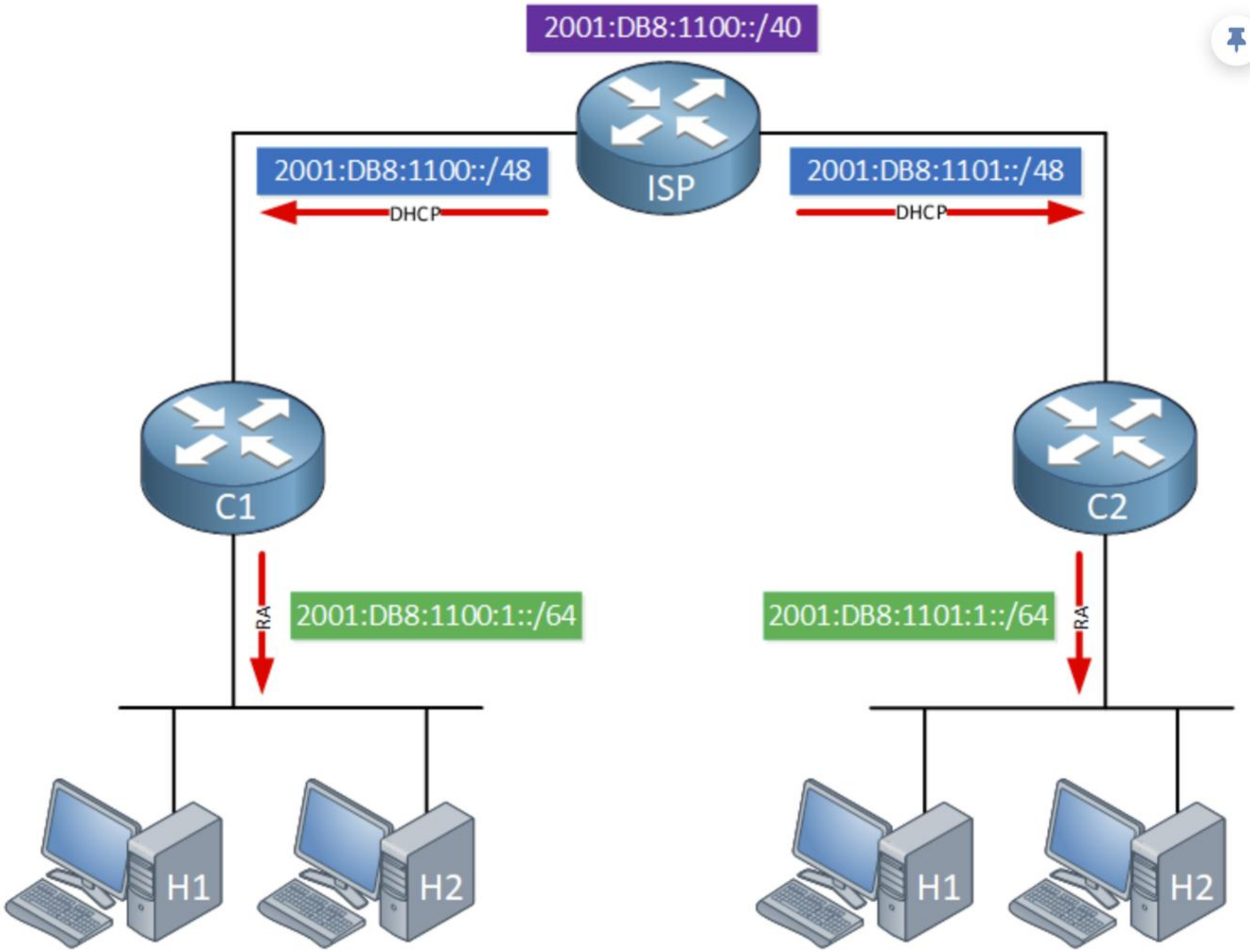
- Up to personal preferences
- DUIDs can be problematic to debug when OSes are cloned or multiple relays are involved.
- MACs can be cloned or changed (VMs and such)
- Local segment DUIDs are elegant 4 types to use
- The choice is up to what works best for you…

Fundamentals that underpin IPv6 - DHCPv6

Prefix Delegations - PD

- Not practical to try to assign prefixes manually when working at scale in IPv6 space.
- ISP hands off a /48 /56 /60 /64 (“depends” - recall ARIN guidance)
- Your local network can now allocate out of this prefix to clients.

This feature can be useful in an ISP environment. Here's an example:



That was a just a bit…

Questions and Discussion?