Link Layer Addresses Assignment Mechanism for DHCPv6

IETF-101 (London)
DHC WG
Monday, 19 March 2018
17:40 – 18:40 (GMT)
Viscount

Last Edit: 2018-03-19 15:30 GMT (TM)

1

Background (1/2)

- RFC 7241 defines cooperation between IEEE 802 and IETF and there are periodic discussions

- IEEE 802c split "local" MAC address space into 4 quadrants to provide for different allocation schemes

- IEEE 802cq is working on defining allocation mechanisms

- Ralph Droms, Russ Housley, Suresh Krishnan thought that DHCPv6 might be usable as an MAC address allocation (802cq) mechanism

Background (2/2)

- Ralph Droms reached out to Bernie

- Tomek and Bernie discussed and decided to work on it

- Hence, the new I-D: draft-bvtm-dhc-mac-assign-00

- More background about 802c/cq in Pat Thaler's "Emerging IEEE 802 Work on MAC Addressing" slides from IETF-96 (https://datatracker.ietf.org/meeting/96/materials/slides-96-edu-ieee802work-0/)

Why?

Number of tries	Possible combinations	Collision chance	No collision chance
23 people	365 days	49,95%	50,05%
1024 VMs	2^{24} (One OUI)	3,07%	96,93%
4824 VMs	2^{24} (One OUI)	50,01%	49,99%
1M VMs	2^{45} (Local address quadrant)	1,41%	98,59%
1M VMs	2^{46} ("I know better than IEEE")	0,71%	99,29%

- Birthday paradox: https://en.wikipedia.org/wiki/Birthday_problem
- Roughly the same probability for IPv6 uniqueness, and we do DAD
- Calculator: https://instacalc.com/28845

Use Cases

- Hypervisor – to allocate the Virtual Machines
  - Lots of VMs
  - May have short or long life
  - May be possible to reuse addresses for different network segments based on data center
- IoT devices
  - Often short lived/disposable
  - Little need for global MAC address
- Individual clients
- …

Why DHCPv6?

- Existing infrastructure: protocol, network, tools

- Servers already know how to manage and assign resources

- Protocol easily extensible

- We are in DHC and …

IA_LL Option

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|           OPTION_IA_LL          |           option-len          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                         IAID (4 octets)                        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                          T1 (4 octets)                         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                          T2 (4 octets)                         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
.                                                                .
.                         IA_LL-options                          .
.                                                                .
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```
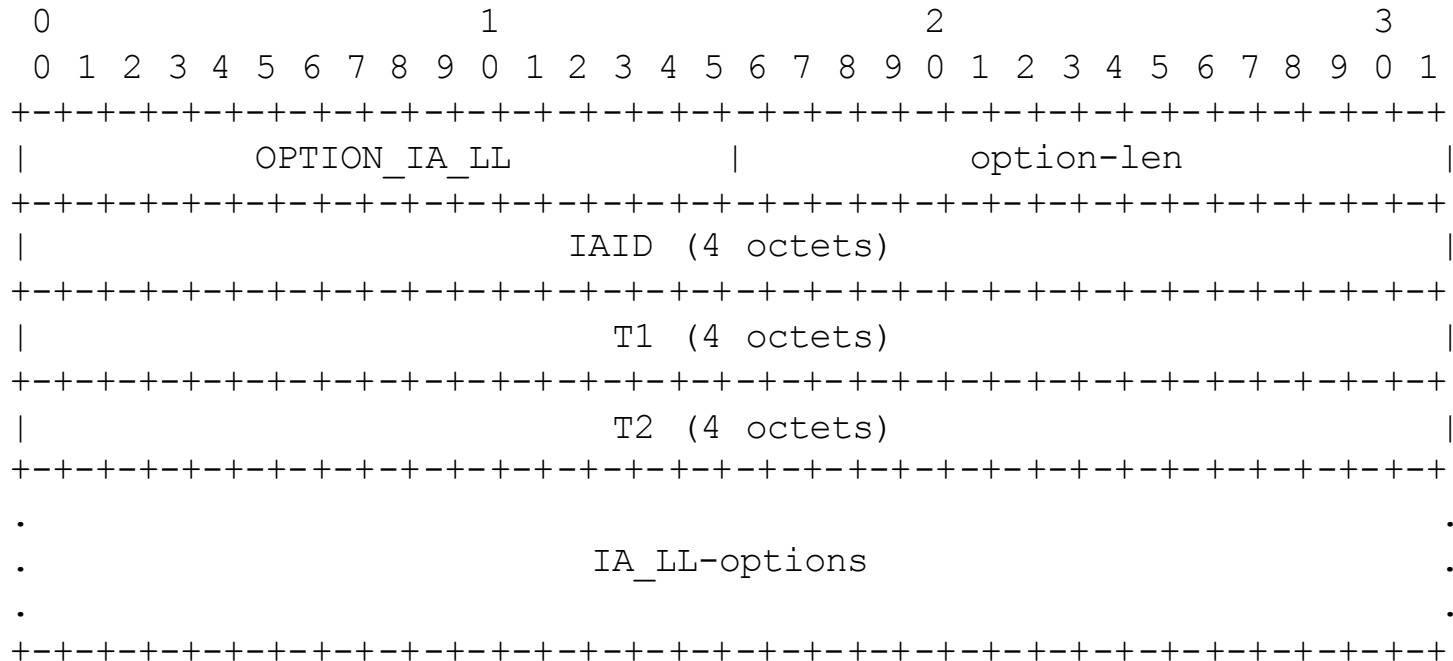
- New Identity Association – IA_LL for link-layer addresses
- Just like IA_NA and IA_PD.

LLADDR Option

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|           OPTION_LLADDR        |           option-len          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|          link-layer-type       |           link-layer-len     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
|                      link-layer-address                       |
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                        extra-addresses                        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                         valid-lifetime                        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
.                                                               .
.                         LLaddr-options                        .
.                                                               .
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```
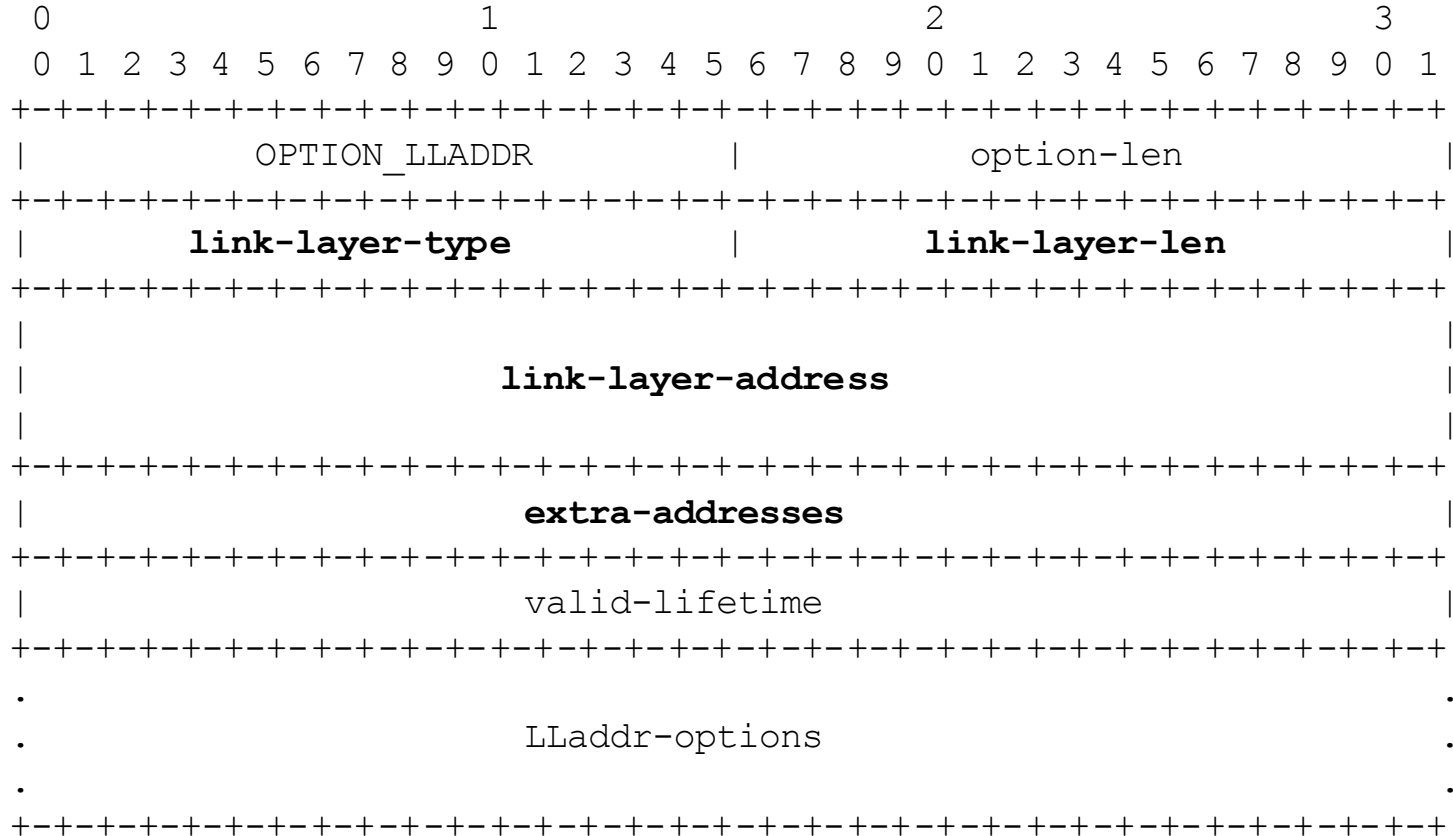
- New container for link-layer address (similar to IAADDR, IAPREFIX)
- Address block = a number of consecutive LL addresses
- Minimal block is 1 address (extra-addresses = 0)
- Link-layer-len is usually 6
- No preferred-lifetime

Client / Server Operation (1)

- Essentials the same as address / prefix delegation, but simpler overall
- Confirm, Decline, Reconfigure, and Information-Request not used
  - Perhaps Decline could be useful if conflict found?

Client / Server Operation (2)

- For hypervisor model
  - Hypervisor is client, but does not use resulting link-layer addresses
  - Instead, address is provided to VMs
  - Those client(s) could do standard DHCPv6
- If "true" client (e.g. IoT) wants a link-layer address
  - Could use Temporary MAC address for anonymity (https://mentor.ieee.org/802.11/dcn/02/11-02-0109-00-000i-temporary-mac-address-for-anonymity.ppt) to get link-layer address
  - Clarify not to do DUID-LL based on temporary MAC
  - Then, use assigned address (for normal DHCPv6, …)

Open Issues

- Use of rapid-commit?
(forbidden, allowed, mandatory)
- Reconfigure?
- Hypervisor – what to do if address expires?

Please discuss on dhc list and report here:

https://github.com/dhcwg/dhcp-mac/issues

Next Steps

- Provide feedback to authors
  - On Draft
  - Address open Issues
  - Indicate interest / support of work (or lack thereof)
- Adopt as Working Group item?
  - Interest? Who will determine the consensus?
  - Defer call?
- If adopted, will need a shepherd (both chairs are authors)

Question or comments …

&

THANKS