IPv6 Neighbor Discovery Prefix Registration

draft-thubert-6lo-prefix-registration

Pascal Thubert
IETF 115
London
6LoWPAN ND (IPv6 Stateful Address Autoconfiguration)

**RFC 6775** (original 6LoWPAN ND)
Defines ARO for registration and DAD operations for stateful AAC

**RFC 8505** (Issued 11/2018)
The protocol agnostic registration for ULA/GUA for proxy ND and routing services
Analogous to a Wi-Fi association but at Layer 3: a deterministic and query-able state for all addresses

**RFC 8929** (Issued 11/2020)
Federates 6lo meshes over a high-speed backbone
ND proxy analogous to Wi-Fi bridging but at Layer 3

**RFC 8928** (Issued 11/2020)
Protects addresses against theft (Crypto ID in registration)

**draft-ietf-6lo-multicast-registration**
Extends RFC 8505 for multicast and anycast

**draft-thubert-6lo-unicast-lookup**
Provides a 6LBR on the backbone to speed up DAD and lookup
Coexistence with classical ND

**draft-thubert-6lo-prefix-registration**
Extends RFC 8505 for prefixes

NEW DRAFT
Redistributing RFC 8505 in routing?

• Already done for host routes with the “R” flag
  • e.g., RFC 9010 into RPL, or even RFC 8929 into IPv6 ND
  • Also draft-thubert-bess-secure-evpn-mac-signaling using BGP, or RIFT
  • Provides a host / router interface that is agnostic to the IGP beyond the router
Multi-link Subnet Routing (non-storing mode)

Parent is default GW, propagates root PIO (L-bit off)
Parent Address in the PIO (with R bit)
RPL Router autoconfigures Address from parent PIO
RPL Router advertises Address via Parent to Root
Root recursively builds a Routing Header back

A::A

A::B

A::C

A::D

C:
::/0 via A::B
A::B connected
A::C self
A:: ~onlink

Target A::C via Transit A::B

B:
::/0 via A::A
A::A connected
A::B self
A:: ~onlink

A: (root)
A::A self
A::B connected
A::C via A::B
A::D via A::B
A:: ~onlink

D:
::/0 via A::B
A::B connected
A::D self
A:: ~onlink

A::C via A::B connected

A::A

A::B

A::C

A::D
6LR advertises A:: in RAs
6LN autoconfigures A::L
6LN registers A::L with « R » flag set
6LR injects the address as external host route in RPL

C:
::/0 via A::B
A::B connected
A::C self
A:: ~onlink

Target A::L via Transit A::C (Ext)

A: (root)
A::A self
A::B connected
A::C via A::B
A::L via A::C
A::D via A::B
A:: ~onlink

A::L via
A::C via A::B connected
Let it be for prefixes!

- Hosts may own prefixes -> and routers may connect to prefixes
  - Network in Node / recursive networking
  - Kubernetes / Private IPv4 realms
  - Directly connected (no routing)
Owned prefix routing (non-storing mode)

Parent is default GW, advertizes owned PIO (L bit on)
RPL Router autoconfigures Address from parent PIO
RPL Router advertises Prefix via Address to Root
Root recursively builds a Routing Header back

A::A

B::B

C::/0 via B::B
B:: connected
C:: connected

D::/0 via B::B
B:: connected
D:: connected

B::/0 via A::A
A:: connected

C:: via B::C

D::3 via B::D via A::B connected

Target C::/ via Transit B::C
Owned prefix routing (non-storing mode)

C::L is reachable but L:: is not
Missing equivalent of RFC 8505/9010 for prefixes

Target C::/ via Transit B::C

A: (root)
A:: connected
B:: via A::B
C:: via B::C
D:: via B::D

L:: unreachable
C::L via B::C via A::B connected
Non LLN (SNAC) Use case 1: Shared Link

1) Register P2
2) Route via 6LR2
3) Redirect P2::g

Internet
Non LLN (SNAC) Use case 2: Hub and Stubs

1) Register P2

2) Inject P2

3) Route via 6LR2
What becomes of DAD?

• Need to consider prefix aggregation and nesting
  • Provisioned Mobile Networks should be unique
  • Auto-allocation?
How would that work?

- RS (SRO) « R » set
- RS (SRO) « R » not set
- Stub registration option EARO with P=3
- Injecting Route
- RS To replaces NS?
Extending the P field

- P is a 2-bits field in EARO, DAR, and RTO
- Defined the Multicast Address Registration draft

<table>
<thead>
<tr>
<th>Value</th>
<th>Meaning</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>00</td>
<td>Registration for a Unicast Address</td>
<td>mcast RFC</td>
</tr>
<tr>
<td>01</td>
<td>Registration for a Multicast Address</td>
<td>mcast RFC</td>
</tr>
<tr>
<td>10</td>
<td>Registration for an Anycast Address</td>
<td>mcast RFC</td>
</tr>
<tr>
<td>11</td>
<td>Unassigned</td>
<td>mcast RFC</td>
</tr>
<tr>
<td>11</td>
<td>Registration for a prefix</td>
<td>This RFC</td>
</tr>
</tbody>
</table>
NS (target = IPv6 address, EARO (ROVR=Crypto-ID PoO))

NA (EARO(status=Validation Requested), Nonce)

NS (EARO, CIPO*, Nonce and NDPSO**)

EARO becomes A stub registration

NA (EARO(status=0))

* Crypto-ID Parameters Option
** NDP Signature Option

RS or NS?

Access link

Extend or replace DAR / DAC ?

EDAR

EDAC
Non LLN (cloud) Use case 3: LISP/EVPN

Provision IPv4 tenant-global subnet 10.1.0.0/28

Provision IPv6 prefix 2001:db8:1::/96 for subnet 10.1.0.0/28 in tenant VRF

RS (StubRegOption, R=1 lifetime, sequence, PoO)

RA(SRO, status = 0)

Advertise (IGP, eVPN)

EDPR

EDPC

Container

Server

6LR

6LBR

MS/MR L-B DNS

IPAM

vSwitch

Ethernet

Fabric

ToR (leaf)
Could do’s

• Adding stub prefix advertisement vs. host today
  • Indicate prefix type e.g., a /96 to embed an IPv4 address
  • Proof of ownership (PoO) per RFC 8928

• Adding policy / ACLs
  • Signal partial micro-segmentation (offload), who can talk to me

• Adding preference to influence load balancing
  • worker capacity (clusters / containers)
  • Access bandwidth /
  • multihoming / preferred interface / anycast

• Tenant ID / VRF ID / RPL instanceID
  • Route tags, RH
Ask

• NS vs RS?
• Name EARO with P=3 an SRO?
• Support of IPv4 with a /96 to embed an IPv4 subnet?
• Proof of ownership (PoO) per RFC 8928