Zero-Configuration Multicast Address Assignment

Nate Karstens
Garmin
National Marine Electronics Association (OneNet Committee)

First Problem

• Coexistence of devices with varying link speeds
The Solution

• Each data stream assigned a different multicast address
• Devices only request streams they are interested in or can handle
• Multicast snooping forwards packets only to interested ports
Link-Local IPv6 Unicast Address

- RFC 4291
### Multicast Addresses

- **RFC 1884 (RFC 4291): Multicast Address**

```
<table>
<thead>
<tr>
<th>8</th>
<th>4</th>
<th>4</th>
<th>112 bits</th>
</tr>
</thead>
<tbody>
<tr>
<td>11111111</td>
<td>flgs</td>
<td>scop</td>
<td>group ID</td>
</tr>
</tbody>
</table>
```

- **RFC 3306: Unicast-Prefix-based IPv6 Multicast Address**

```
<table>
<thead>
<tr>
<th>8</th>
<th>4</th>
<th>4</th>
<th>8</th>
<th>8</th>
<th>64</th>
<th>32</th>
</tr>
</thead>
<tbody>
<tr>
<td>11111111</td>
<td>flgs</td>
<td>scop</td>
<td>reserved</td>
<td>plen</td>
<td>network prefix</td>
<td>group ID</td>
</tr>
</tbody>
</table>
```

- **RFC 4489: Link-Scoped IPv6 Multicast Address**

```
<table>
<thead>
<tr>
<th>8</th>
<th>4</th>
<th>4</th>
<th>8</th>
<th>8</th>
<th>64</th>
<th>32</th>
</tr>
</thead>
<tbody>
<tr>
<td>11111111</td>
<td>flgs</td>
<td>scop</td>
<td>reserved</td>
<td>plen</td>
<td>IID</td>
<td>group ID</td>
</tr>
</tbody>
</table>
```

*flgs are 0RPT where:*
- **0** = reserved
- **R = 0** -> does not embed Rendezvous Point address
- **R = 1** -> embeds Rendezvous Point address
- **P = 0** -> address not assigned based on network prefix
- **P = 1** -> address is assigned based on the network prefix
- **T = 0** -> permanently assigned multicast address
- **T = 1** -> dynamically assigned

*flgs are 0011 (R = 0, P = 1, T = 1)*

*reserved = 0*

*plen = 0xff*

*IID = IPv6LL IID*

*Allocated as per RFC 3307*
Group ID is:

• 4.1 Permanent IPv6 Multicast Addresses
  Allocated by IANA ([IPv6 registry])
  \( T = 0, \ P = 0 \)
  0x00000001 to 0x3FFFFFFF

• 4.2 Permanent IPv6 Multicast Group Identifiers
  Allocated by IANA
  0x40000000 to 0x7FFFFFFF

• 4.3 Dynamic IPv6 Multicast Addresses
  \( T = 1 \)
  0x80000000 to 0xffffffff
The Problem: Transmitting on Ethernet

- RFC 2464 (Transmission of IPv6 Packets over Ethernet Networks)
  Section 7 specifies destination multicast MAC address mapping:
  - First two octets are 33:33
  - Last four octets are last four octets of IPv6 multicast address

- Link-Scoped IPv6 Multicast Address

```
|   8   |  4 |  4 |   8   |    8   |       64       |    32    |
+--------+----+----+--------+--------+----------------+----------+
|11111111|flgs|scop|reserved|  plen  |       IID      | group ID |
+--------+----+----+--------+--------+----------------+----------+
Different nodes can generate different IPv6LL addresses with the same MAC address!
Solution Requirements

• Zero-configuration
  • Most customers do not have expertise on network configuration

• No Internet connection
  • Networks are typically single-subnet

• Unique Ethernet destination address
  • RFC 4541 indicates most switch vendors do not support looking at IPv6 destination address
  • Switch parts at the desired price point do not support Source-Specific Multicast

• Decentralized (avoid single-point of failure)
  • Existing solution for dynamic assignment, MADCAP (RFC 2730), relies on a server

• Multiple streams from same host
Overview of Proposed Solution

- Update RFC 3307 section 4.3 to designate a range for zero-configuration allocations
- Application generates random group ID in zero-configuration range
- Application uses mDNS (RFC 6762) to ensure group ID is unique
  - mDNS is a zero-configuration technology
  - Both probing before initial use and continuous monitoring
- Application uses group ID to generate Link-Scoped IPv6 Multicast Address
Update RFC 3307 Section 4.3

- RFC 3307 specifies range of 0x80000000-0xFFFFFFFF for both:
  - Server-based allocation (section 4.3.1)
  - Host-based allocation (section 4.3.2)

- Proposed change:

<table>
<thead>
<tr>
<th>0x80000000-0xBFFFFFFF</th>
<th>MADCAP [RFC2730]</th>
</tr>
</thead>
<tbody>
<tr>
<td>0xC0000000-0xCFFFFFFF</td>
<td>mDNS-based zero-configuration algorithm described above</td>
</tr>
<tr>
<td>0xD0000000-0xFEFFFFFF</td>
<td>Reserved for future zero-configuration algorithms</td>
</tr>
<tr>
<td>0xFF000000-0xFFFFFFFF</td>
<td>Solicited-node multicast addresses [RFC4291], Section 2.7.1</td>
</tr>
</tbody>
</table>
Uses mDNS to Ensure Group ID Is Unique (1/2)

• DNS uses PTR records to perform reverse lookups. Examples from RFC 8501 (Reverse DNS in IPv6 for Internet Service Providers):
  192.0.2.1 => 1.2.0.192.in-addr.arpa.
  2001:0db8:0f00:0000:0012:34ff:fe56:789a =>
  a.9.8.7.6.5.e.f.f.f.4.3.2.1.0.0.0.0.0.0.0.0.f.0.8.b.d.0.1.0.0.2.ip6.arpa.

• Proposal uses a PTR record for layer 2 address:
  33:33:CF:ED:24:68 =>
  8.6.4.2.d.e.f.c.3.3.3.3.eth-addr.arpa

• The application uses the mDNS probing algorithm described in RFC6762 section 8.1 to continuously query for a PTR record with the generated string for the name.
Uses mDNS to Ensure Group ID Is Unique (2/2)

• If the probing algorithm completes without any conflict, then the application begins advertising its own PTR record using that name.

• The PTRDNAME field consists of a unique application identifier, in the form of a DNS label, followed by the device's host.
  • Example PTRDNAME: “application.example.local.”
  • Integrating a unique identifier in this manner allows for multiple applications to be on the same host.

• The application retains the group ID value in long-term storage and use it the next time the multicast stream is transmitted.
  • Conflicts should not be repeated each power cycle.
Thank You!