Zero-Configuration Multicast Address Assignment

Nate Karstens
Garmin
National Marine Electronics Association (OneNet Committee)
OneNet is 40,000x faster than NMEA 2000

NMEA OneNet provides the means for sensing/control messages to work over secure ethernet. Manufacturers will now have the ability to choose what type of product they wish to support eliminating the tradeoffs of one protocol vs another. Even the opportunity exists for manufacturers to choose the best of both worlds for example the reliability of CAN coupled with the high bandwidth data exchange of ethernet.
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

```
+----------+-------------------------+----------------------------+
|1111111010|           0             |       interface ID         |
+----------+-------------------------+----------------------------+
```
Multicast Addresses

• RFC 1884 (RFC 4291): Multicast Address

```
| 8 | 4 | 4 | 112 bits |
+---+----+----+-----------------------------+
|11111111|flgs|scop|                  group ID                   |
+--------+----+----+---------------------------------------------+
```

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

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

```
| 8 | 4 | 4 | 8 | 8 | 64 | 32 |
+---+----+----+---+---+-----+----+
|11111111|flgs|scop|reserved| plen | network prefix | group ID |
+--------+----+----+--------+--------+----------------+----------+
```

• RFC 4489: Link- Scoped IPv6 Multicast Address

```
| 8 | 4 | 4 | 8 | 8 | 64 | 32 |
+---+----+----+---+---+-----+----+
|11111111|flgs|scop|reserved| plen | IID | group ID |
+--------+----+----+--------+--------+-----+----------+
```

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

reserved = 0

plen = 0xff

IID = IPv6LL IID

Allocated as per RFC 3307
RFC 3307: IPv6 Multicast Address Guidelines

Group ID is:

- 4.1 Permanent IPv6 Multicast Addresses
  Allocated by IANA ([IPv6 registry](http://www.iana.org/assignments/ipv6-address-registries))
  \[ T = 0, P = 0 \]
  \[ 0x00000001 \text{ to } 0x3FFFFFFF \]

- 4.2 Permanent IPv6 Multicast Group Identifiers
  Allocated by IANA
  \[ 0x40000000 \text{ to } 0x7FFFFFFF \]

- 4.3 Dynamic IPv6 Multicast Addresses
  \[ T = 1 \]
  \[ 0x80000000 \text{ to } 0xFFFFFFFF \]
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
Solutions

• Source-Specific Multicast
  • Not universally supported by switch hardware

• MADCAP (RFC 2730)
  • Uses a server to allocate addresses
  • Single point of failure
  • Could auto-negotiate MADCAP server and provide failover mechanism
    • Common format for lease database

• ZMAAP
  • Never published, draft expired in 2003
  • Addresses and ports are TBD
Proposal

• Restore ZMAAP and use existing draft as a starting point
• Specify addresses and port for MAAS coordination
• Do not assume single MAAS per host
• Require defending addresses with same destination MAC address
• Allocate Group IDs for zeroconf dynamic addresses
Thank You!