Opportunities for New IEEE 802 Switching Protocols in Local Space

Document Number:

Date Submitted:
2015-01-12

Source:
Roger B. Marks
EthAirNet Associates
404 Montview Blvd
Denver, CO 80207 USA
Voice: +1 802 227 2253
E-mail: roger@ethair.net

Re: 802.1 Local Address Study Group

Venue:
IEEE 802.1 Working Group

Purpose:
To proposal that the P802c PAR be drafted to support a project that will specify a swath of local address space be reserved for development IEEE 802 switching protocols.

Notice:
This document represents the views of the author and is offered as a basis for discussion.
Opportunities for New IEEE 802 Switching Protocols in Local Space

Roger B. Marks
EthAirNet Associates
Key points

- We can’t “partition the local space,” because there are many local spaces, each of which is local.

- Local address structuring can enable interesting switching protocols, such as zone-based switching.

- NAT-like address translation means that local space structure in the network core can be independent of that at the edge.

- Interesting protocols should be standardized in IEEE 802, not simply outsourced via Company ID.

- RAC coordination is important.
What is a Company ID, and why does it matter?

- The last proposed draft P802c PAR would “allocate a portion of the address space for protocols using an IEEE Registration Authority assigned Company ID.”

- a bit confusing; RA documentation does not fully explain
  - [https://standards.ieee.org/develop/regauth/tut/eui.pdf](https://standards.ieee.org/develop/regauth/tut/eui.pdf)

- RA specifies that a Company ID (CID) is 24 bits, with the last two bits of the first byte set to 1 and 0
  - if one were to build an 802 MAC address using the CID as the first three bytes, that MAC address would have the local bit on and the multicast bit off

- But people are talking about a “quadrant” of the CID space. What’s that?
Observations:
• First two LSB in first byte are 802-compatible (local bit set, multicast not).
• Next two LSB in first byte are all “10”; RA seems to staying in this “10” quadrant.
• Structured use of any other bit leads to collision since RA has allocated every option.
  • RA has even allocated every possible PAIR of bits in the first byte.
• Possible inference: The RA has staked out every corner of this “10” quadrant and does not want any trespassing! Bits will not be designated. Only solo addresses will be assigned. Don't expect a clean block assignment in this quadrant.
What is a CID again?

• It’s important to differentiate the CID space (22 bits) from the currently-allocated “10” CID quadrant (20 bits).

• The CID is not limited to the “10” quadrant.

• It’s possible to “allocate a portion of the address space for protocols using an IEEE Registration Authority assigned Company ID” without touching the “10” quadrant.

• Utilizing the other portions of the CID space will require early RAC coordination.
What I think draft P802c PAR originally proposed

CID
protocol identifier
3 bytes

protocol-specific use
3 bytes
What I prefer from P802c

non-802 Protocols

CID
- protocol identifier
- 3 bytes

private CID
- protocol-specific use
- 3 bytes

IEEE 802 Protocols

Header byte

IEEE 802-specified use
- 6 bytes
Local address details
possible use of first three bytes

### non-802 Protocols

<table>
<thead>
<tr>
<th>more assigned CID</th>
<th>RA CID Quadrant</th>
<th>Local bit</th>
<th>Multicast bit</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td></td>
<td>0</td>
</tr>
</tbody>
</table>

### IEEE 802 Protocols

<table>
<thead>
<tr>
<th>IEEE 802 Protocol Identifiers and Parameters</th>
<th>RA CID Quadrant</th>
<th>Local bit</th>
<th>Multicast bit</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td></td>
<td>0</td>
</tr>
</tbody>
</table>
Zonal Address structures

see Zonal Address Partitioning in the Local Space, of 2014-11-01

<table>
<thead>
<tr>
<th>Header byte</th>
<th>1 Zone identifier byte up to 256 zones</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Local (intra-zone) byte</td>
</tr>
<tr>
<td></td>
<td>Local (intra-zone) byte</td>
</tr>
<tr>
<td></td>
<td>Local (intra-zone) byte</td>
</tr>
<tr>
<td></td>
<td>Local (intra-zone) byte</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>CID Space</th>
<th>2 Zone identifier bytes up to 65536 zones</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Local (intra-zone) byte</td>
</tr>
<tr>
<td></td>
<td>Local (intra-zone) byte</td>
</tr>
<tr>
<td></td>
<td>Local (intra-zone) byte</td>
</tr>
<tr>
<td></td>
<td>Local (intra-zone) byte</td>
</tr>
</tbody>
</table>
## example address header

<table>
<thead>
<tr>
<th>Bytes of Zone ID (if zonal)</th>
<th>structure identifier</th>
<th>quadrant identifier</th>
<th>Local bit</th>
<th>Multicast bit</th>
</tr>
</thead>
<tbody>
<tr>
<td>00 = 1 byte</td>
<td>00 = zonal</td>
<td>01 or 11 if RAC agrees to avoid CID allocations</td>
<td></td>
<td></td>
</tr>
<tr>
<td>01 = 2 bytes</td>
<td>01 = reserved</td>
<td>(10 is already off-limits, and 00 is slightly problematic)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>10 = 3 bytes</td>
<td>10 reserved</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11 = 4 bytes</td>
<td>11 reserved</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

1
0
Switching based on mixed conventional and zonal addresses

1. Frame
   - Read source address (SA) and destination address (DA)
   - Examine SA and DA bit patterns to determine if zonal topology is encoded.

2. Zonal DA?
   - yes: Read zone ID
     - Look up egress port in zone table
     - Forward to port
   - no: Use normal bridging

3. Zonal SA?
   - yes: Read zone ID
     - Store zone ID and ingress port in zone table
   - no: Use normal learning
NAT

Core Switch

Address is partitioned to incorporate a zone identifier. Switch selects a port based on zone identifier in the zonal DA. Switch learns zones, not addresses, based on zonal SA.

Access Point (AP)

Local mapping table maps each internal (user-side) address to a corresponding external (core-side) alias address.

Access point replaces access address with core alias address in transmissions to core, and vice versa.

User

terminal selects among many (46?) random bits

or could use a global address
NAT Updating

Address is partitioned to incorporate a zone identifier. Switch selects a port based on zone identifier in the zonal DA. Switch learns zones, not addresses, based on zonal SA.

Core Switch

Local mapping table maps each internal (user-side) address to a corresponding external (core-side) alias address.

Access Point (AP)

Access point replaces access address with core alias address in transmissions to core, and vice versa.

User

terminal selects among many (46?) random bits
or could use a global address

later, AP assigns unique address (could be identical to Core address)

Core address

Access address

Core address

Access address

alias unique within the core
local (could be terminal-selected) or global

alias unique within the core
Unique, once assigned by AP; could be identical to Core address
What might a future 802c standard recommend?

Example 1: If a device is used in a domain in which multiple local address assignment or usage protocols are active, then the first byte of the device address should end in 10 only if the first three bytes of the address are a CID allocated by the IEEE RA and the address is assigned in accordance with the protocol specified by the owner of that CID.
What I’d like an 802c standard to include

Example 2: If the first byte of an address ends in [for example] \{1110 or \textbf{1111}10 or \textbf{11111}10\}, then the address should be interpreted per protocols specified by IEEE 802.
Summary

• Structuring of local space offers opportunity for new switching methods that might work better in some circumstances.
• NAT can allow various local space structures to interoperate.
• Structure needs to be standardized so switches can understand addresses.
• Interesting protocols should be standardized in IEEE 802, not simply outsourced via Company ID.
• RAC coordination is important.