Automotive Applicability of P802.1CQ Address Assignment

v00: 2022-08-30

Roger Marks
(EthAirNet Associates)
roger@ethair.net
+1 802 capable

30 August 2022
Note

• Contributor is not an expert in P802.1DG or Automotive Ethernet requirements.

• Contribution is seeking to:
  ▫ Review P802.1CQ and possible automotive applications
  ▫ Get a better understanding of P802.1DG requirements and usage of MAC addresses
  ▫ Engage P802.1DG experts in reviewing P802.1CQ (“Multicast and Local Address Assignment”)
  ▫ Encourage P802.1DG experts to evolve P802.1CQ to serve P802.1DG requirements
Background

• **P802.1CQ (“Multicast and Local Address Assignment”):**
  - specifies protocols, procedures, and management objects for locally-unique assignment of 48-bit and 64-bit addresses in IEEE 802 networks. *Peer-to-peer address claiming and address server capabilities are specified.*

• **P802.1CQ/D0.8 is in TSN Task Group Ballot, through Sept. 5**
  - deadline could be extended

• specifically uses the **Block Address Registration and Claiming (BARC)** protocol to assign blocks of unicast and multicast addresses
  - Some assignment protocol options may be applicable to Automotive In-Vehicle Ethernet.
  - Network-cognizant block address assignments may be applicable to Automotive In-Vehicle Ethernet
IEEE 802 Addresses

- $m$ bit splits the IEEE 802 address space into unicast and multicast addresses
- $x$ bit splits the IEEE 802 address space in universal and local addresses
  - Universal addresses are factory-assigned and static
    - Should be unique among all devices over an intended span of 100 years
    - IEEE Registration Authority manages the remaining address space to ensure that it lasts
- The Structured Local Address Plan (SLAP) of IEEE Std 802 (per 802c-2017) designates 25% of local addresses for Standard-Assigned Identifiers (SAI), to be specified by IEEE Std 802.1CQ
- P802.1CQ draft describes the Block Address Registration and Claiming (BARC) protocol to assign blocks of unicast and multicast addresses
  - Within the SAI quadrant
BARC assigns MAC Addresses in Blocks

- An Address Block (AB) is a set of local BARC addresses.
- An AB includes equal-sized unicast and multicast contiguous sub-blocks.
- No BARC address falls within more than one AB.
- Registrable Address Block Identifier (RABI)
  - identifies a Registrable Address Block (RAB)
    - holding Registrable Addresses (RAs)
  - RABIs are held in inventory of a Registrar
  - may be assigned to Claimants
- Claimable AB Address (CABA)
  - identifies Claimable Address Blocks (CABs)
    - holding Claimable Addresses (CAs)
  - claimable by a Claimant without using a Registrar
  - CABA is a multicast MAC address, not in any AB, and used as a DA
- A large set of Temporary Unicast Addresses (TUAs) is specified
  - useful for initial discovery by Claimant lacking a unicast address
BARC Address Structure

<table>
<thead>
<tr>
<th>N0</th>
<th>r</th>
<th>i</th>
<th>j</th>
<th>k</th>
</tr>
</thead>
<tbody>
<tr>
<td>N1</td>
<td>z</td>
<td>y</td>
<td>x</td>
<td>m</td>
</tr>
<tr>
<td>N2</td>
<td>0000 for CA or TUA</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>N3</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>N4</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>N5</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>N6</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>N7</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>N8</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>N9</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>N10</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>N11</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- address block includes subblocks of
  - $16^k$ claimable addresses, or
  - $16^2^k$ registrable addresses
- for claimable addresses, $i$ distinguishes
  - Claimable Addresses (CAs) from
  - CABAs
    - CABA is a multicast address that identifies a CAB

- $r=1$ for registrable addresses; $r=0$ for claimable addresses and TUAs
- $m$ is the multicast (I/G) bit; $x$ is the local (U/L) bit; $zyx=111$ for “SAI**”

*per IEEE Std 802 [2], “Specification of the use of the SAI quadrant for SLAP address assignments is reserved for the standard forthcoming from IEEE P802.1CQ”

12 nibbles per 48-bit address

<table>
<thead>
<tr>
<th>r</th>
<th>i</th>
<th>jk</th>
<th>zy</th>
<th>x=U/L</th>
<th>m=I/G</th>
</tr>
</thead>
<tbody>
<tr>
<td>CA</td>
<td>0</td>
<td>1</td>
<td>block size</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CABA</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TUA</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>RA</td>
<td>1</td>
<td>RABI Option</td>
<td>block size</td>
<td></td>
<td></td>
</tr>
<tr>
<td>RABI</td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>I/G</td>
<td>1</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>I/G</td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Claiming (sketch)
(might not be optimal for P802.1DG)

Claimant of CABA_X AB listens to CABA_X multicast address

1. CABA_1: DISCOVER state
2. CABA_1: CLAIMED state
3. CABA_6: DISCOVER state
4. CABA_6: CLAIMED state

LAN

- CABA_4
- CABA_5
- CABA_6
- CABA_3
- CABA_2
- CABA_1
Operation with Registrars (sketch)

(1) CABA₁: DISCOVER state

(2) (unicast) RABI₁: OFFERED state

(3) (unicast) RABI₁: REQUESTED state

(4) (unicast) RABI₁: REGISTERED state
Advisor/Registrar (sketch)  
(seems applicable to P802.1DG)

- Claimant may send a Proposal to an Advisor  
  ▫ many variations supported  
- Advisor may suggest a different Proposal and may refer Claimant to a separate Registrar  
  ▫ (not illustrated)  
  ▫ Advisor’s Proposal may be based on, e.g., network structure  
- Registrar at Advisor may offer a RABI, based on Proposal  
  ▫ illustrated below  
  ▫ Offer may be based on, e.g., network structure
IEEE 1722 MAAP Support

- IEEE Std 1722 specifies MAAP
  - *The media access control (MAC) Address Acquisition Protocol (MAAP) is designed to provide a way to allocate dynamically the multicast MAC addresses needed by AVTP.*

- MAAP is a peer-to-peer protocol for claiming a range of addresses from the set specified in IEEE Std 802

- P802.1CQ is backward compatible with MAAP
  - Uses the same EtherType
  - A device may issue a MAAP Claim (v2)
    - Legacy MAAP devices can respond to defend against that MAAP Claim
    - A BARC registrar can respond to that MAAP Claim with an offer
Claimable Address Blocks

CABA and CA, CAB Size 0-3

<table>
<thead>
<tr>
<th>N0</th>
<th>r</th>
<th>i</th>
<th>j</th>
<th>k</th>
</tr>
</thead>
<tbody>
<tr>
<td>N1</td>
<td>z</td>
<td>y</td>
<td>x</td>
<td>m</td>
</tr>
</tbody>
</table>

CABA and CA, CAB Size 0-3

C=CSI=0

<table>
<thead>
<tr>
<th>CABA</th>
<th>CAB</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 0 0 0</td>
<td>0 1 0 0</td>
</tr>
<tr>
<td>1 1 1 1</td>
<td>1 1 1 *</td>
</tr>
<tr>
<td>0 0 0 0</td>
<td>0 0 0 0</td>
</tr>
<tr>
<td>X3</td>
<td>X3</td>
</tr>
<tr>
<td>X4</td>
<td>X4</td>
</tr>
<tr>
<td>X5</td>
<td>X5</td>
</tr>
<tr>
<td>X6</td>
<td>X6</td>
</tr>
<tr>
<td>X7</td>
<td>X7</td>
</tr>
<tr>
<td>X8</td>
<td>X8</td>
</tr>
<tr>
<td>X9</td>
<td>X9</td>
</tr>
<tr>
<td>X10</td>
<td>X10</td>
</tr>
<tr>
<td>0</td>
<td>*</td>
</tr>
</tbody>
</table>

C=CSI=1

<table>
<thead>
<tr>
<th>CABA</th>
<th>CAB</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 0 0 1</td>
<td>0 1 0 1</td>
</tr>
<tr>
<td>1 1 1 1</td>
<td>1 1 1 *</td>
</tr>
<tr>
<td>0 0 0 0</td>
<td>0 0 0 0</td>
</tr>
<tr>
<td>X3</td>
<td>X3</td>
</tr>
<tr>
<td>X4</td>
<td>X4</td>
</tr>
<tr>
<td>X5</td>
<td>X5</td>
</tr>
<tr>
<td>X6</td>
<td>X6</td>
</tr>
<tr>
<td>X7</td>
<td>X7</td>
</tr>
<tr>
<td>X8</td>
<td>X8</td>
</tr>
<tr>
<td>X9</td>
<td>X9</td>
</tr>
<tr>
<td>X10</td>
<td>X10</td>
</tr>
<tr>
<td>0</td>
<td>*</td>
</tr>
</tbody>
</table>

C=CSI=2

<table>
<thead>
<tr>
<th>CABA</th>
<th>CAB</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 0 1 0</td>
<td>0 1 1 0</td>
</tr>
<tr>
<td>1 1 1 1</td>
<td>1 1 1 *</td>
</tr>
<tr>
<td>0 0 0 0</td>
<td>0 0 0 0</td>
</tr>
<tr>
<td>X3</td>
<td>X3</td>
</tr>
<tr>
<td>X4</td>
<td>X4</td>
</tr>
<tr>
<td>X5</td>
<td>X5</td>
</tr>
<tr>
<td>X6</td>
<td>X6</td>
</tr>
<tr>
<td>X7</td>
<td>X7</td>
</tr>
<tr>
<td>X8</td>
<td>X8</td>
</tr>
<tr>
<td>X9</td>
<td>X9</td>
</tr>
<tr>
<td>X10</td>
<td>X10</td>
</tr>
<tr>
<td>0</td>
<td>*</td>
</tr>
</tbody>
</table>

C=CSI=3

<table>
<thead>
<tr>
<th>CABA</th>
<th>CAB</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 0 1 1</td>
<td>0 1 1 1</td>
</tr>
<tr>
<td>1 1 1 1</td>
<td>1 1 1 *</td>
</tr>
<tr>
<td>0 0 0 0</td>
<td>0 0 0 0</td>
</tr>
<tr>
<td>X3</td>
<td>X3</td>
</tr>
<tr>
<td>X4</td>
<td>X4</td>
</tr>
<tr>
<td>X5</td>
<td>X5</td>
</tr>
<tr>
<td>X6</td>
<td>X6</td>
</tr>
<tr>
<td>X7</td>
<td>X7</td>
</tr>
<tr>
<td>X8</td>
<td>X8</td>
</tr>
<tr>
<td>X9</td>
<td>X9</td>
</tr>
<tr>
<td>X10</td>
<td>X10</td>
</tr>
<tr>
<td>0</td>
<td>*</td>
</tr>
</tbody>
</table>

2 contiguous subblocks per CABA (one unicast, one multicast)

- \( \approx 6.9 \times 10^9 \) Size 0 CABA
- \( \approx 4.3 \times 10^9 \) Size 1 CABA
- \( \approx 2.7 \times 10^8 \) Size 2 CABA
- \( \approx 1.7 \times 10^7 \) Size 3 CABA

* indicates wildcard (any value)
Registrable Address Blocks

RABI and RA, RABI Size 0-6

<table>
<thead>
<tr>
<th>N0</th>
<th>N1</th>
<th>N2</th>
<th>N3</th>
<th>N4</th>
<th>N5</th>
<th>N6</th>
<th>N7</th>
<th>N8</th>
<th>N9</th>
<th>N10</th>
<th>N11</th>
</tr>
</thead>
<tbody>
<tr>
<td>r</td>
<td>i</td>
<td>j</td>
<td>k</td>
<td>z</td>
<td>y</td>
<td>x</td>
<td>m</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

2 contiguous subblocks per RABI (one unicast, one multicast)

- 1.1E12 Size 0 RABIs
- 1 RA/subblock
- 4.3E9 Size 2 RABIs
- 256 RAs/subblock
- 1.7E7 Size 4 RABIs
- 65536 RAs/subblock
- 1.7E7 RAs/subblock

* indicates wildcard (any value)
BARC PDU

- Format is compatible with MAAP
- All fixed-size fields
- One 8-bit state field
- Three 48-bit fields
  - Content depends on the State field
address assignments to a node:
1 or more unicast addresses (SA in egress and DA in ingress)
0 or more multicast addresses (DA in egress)
Automotive Addresses - Unicast

- **P802.1DG/D1.4 6.3** (“In-vehicle network addressing constraints”)
  - Automotive in-vehicle networks (IVNs) are unlike many other Ethernet networks in that every device on the network (see Clause 8.2) is supposed to be pre-configured, administratively authorized and installed by the OEM or its assignees. As such, the OEM can exercise control over the layer 2 and layer 3 addresses assigned to every node in that portion of the network. The OEM can dictate what network addresses and VLAN IDs are used for all TSN and non-TSN devices... can make the selection of TSN features much simpler since every device is known when the network is designed. Those MAC addresses that are not exposed outside of the IVN are not required to be globally unique. As such, it is recommended that IVN designers use locally administered addresses wherever possible.

- **P802.1DG/D1.4** (“13.8 Redundant Paths and Loop ”)
  - The ARL (Address Resolution Lookup) table learning process in switches depends on there being a unique path to a unique end station identified by its MAC address.

- **P802.1DG/D1.4** (“Items remaining to be implemented ”)
  - List requirements for a management based structure (TBD in a future draft) that defines addressing used throughout the IVN. Recommend (should) that globally unique addresses should not be used inside the vehicle.
Unicast Address Assignment scenarios

• **Fully Static**
  ▫ **Globally unique, based on EUI**
    • Strains the remaining IEEE inventory of EUIs
    • Complicates inventory management and network implementations
  ▫ **Local, burned in at component manufacture, per part number**
    • Does this complicate the use of a part in, e.g., multiple vehicle models?
  ▫ **Local, burned in at assembly**
    • Is this practical?

• **Fully or Quasi-Dynamic**
  ▫ **Assigned within operating network**
    • Too slow to repeat assignment and dissemination at each vehicle startup?
  ▫ **One-time assignment, possibly with in-service reassignment**
    • With non-volatile memory storage independent of vehicle shutdown.
    • Is this feasible?
  ▫ **Should the assigned address be meaningful?**
    • For example,
      • Address reflects the nature of the component.
      • Address reflects a component and a stream.
    • Address assignment could be based on the part identifier.
      • For example, network recognizes a right rear brake actuator and assigns a set of addresses indicative of a right rear brake actuator as used in this network.

• **Are some of these realistic? Are there others?**
Bridge Learning of Address Block

One (short) FDB entry covers (e.g.) 65536 addresses.
Zonal Address Block Assignment

Zone A
- N0
- N1
- N2
- N3
- N4
- N5
- N6
- N7
- N8
- N9
- N10
- N11

Zone B
- 1
- 2
- 3
- 4

LAN
- Unicast: RA
- Type?
- Flow?

Bridge
- Stores route to node, not individual addresses

Zone A & Zone B
- LAN stores route to Zone A & Zone B
- Not individual addresses

RABI
- Root
- Zone ID
- Port
- Type?

RA
- Root
- Zone B
- Type?
- Flow?
Automotive Addresses - Multicast

- P802.1DG/D1.4 ("F.2.2 Stream Reservation Protocol (SRP)")
  - Multicast addresses are a carefully managed resource in the design of a vehicle network and should not be assigned dynamically. The control instance (likely using SOME/IP) can more securely handle this task.
    - Content and title of Subclause 14.8 [“SOME/IP (OSI-Layer 5-7)" ] suggests that SOME/IP might not be appropriate for Layer 2 address assignment
    - But 14.9 says “SOME/IP-SD can communicate IPv4, IPv6 and MAC address information.”
- P802.1DG/D1.4 (“F.3 Others items to be considered") (Editor’s Note)
  - Address assignment (CA, IEEE 1722 MAAP)
Address Assignment - Multicast

- Per P802.1DG/D1.4 F.2, TSN SRP functions to
  - Communicate the multicast MAC address for the AVB traffic to allow the sink to open its RX filter.
  - Communicate the multicast MAC address for the AVB traffic and make sure it is forwarded to the correct ports (set up the multicast MAC address’ forwarding path) and make sure it is not flooded to other ports.
- Such multicast MAC addresses are assigned to the Talker for its use.
  - As is done in MAAP of IEEE Std 1722.
- Since Talker needs at least one unicast address assignment and a set of multicast address assignments, is it practical to assign both in a single assignment?
  - Single-step process, for simplicity
  - Unicast and multicast addresses are related
    - The entire address block assignment is visible from any one of its addresses
Stream IDs

<table>
<thead>
<tr>
<th>Stream ID</th>
<th>conventional stream ID</th>
<th>alt. stream ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>RABI</td>
<td>RAB root</td>
<td>RAB root</td>
</tr>
<tr>
<td>N0</td>
<td>1 0 1 0</td>
<td>1 0 1 0</td>
</tr>
<tr>
<td>N1</td>
<td>0 0 0 0</td>
<td>1 1 1 0</td>
</tr>
<tr>
<td>N2</td>
<td>RAB root</td>
<td>RAB root</td>
</tr>
<tr>
<td>N3</td>
<td>1 0 1 0</td>
<td>1 0 1 0</td>
</tr>
<tr>
<td>N4</td>
<td>1 1 1 0</td>
<td>1 1 1 1</td>
</tr>
<tr>
<td>N5</td>
<td>1 0 1 0</td>
<td>1 0 1 0</td>
</tr>
<tr>
<td>N6</td>
<td>1 1 1 0</td>
<td>1 1 1 1</td>
</tr>
<tr>
<td>N7</td>
<td>1 0 1 0</td>
<td>1 0 1 0</td>
</tr>
<tr>
<td>N8</td>
<td>1 1 1 0</td>
<td>1 1 1 1</td>
</tr>
<tr>
<td>N9</td>
<td>1 0 1 0</td>
<td>1 0 1 0</td>
</tr>
<tr>
<td>N10</td>
<td>1 1 1 0</td>
<td>1 1 1 1</td>
</tr>
<tr>
<td>N11</td>
<td>1 0 1 0</td>
<td>1 0 1 0</td>
</tr>
</tbody>
</table>

The MAC address component of the StreamID can, but does not necessarily, have the same value as the source_address parameter of any frame in the actual data stream.
Example: coded extension field

Elementary example: This coding indicates that the stream has listeners at Zones 2, 9, and 15.

Bridges know the route to each Zone, so no lookup is required for forwarding.

Unique ID differentiates among streams from a common Talker and with a common set of destination zones.
Summary

- Address assignment protocol reduces global address consumption and may simplify inventory management.
- BARC assigns address blocks including unicast and multicast addresses.
- It appears that block assignment could be conveniently exploited in network operation.
  - Minimizes forwarding tables and lookup requirements.
  - Address bits can be used for various other purposes.
- BARC offers assignment based on claiming and on Registrar assignment, with an Advisor option.
  - The Advisor-driven option may be best suited for automotive case.
- Address assignment takes some time.
  - But it can survive vehicle reboot and need not be repeated frequently.
Call to Action

- Review P802.1CQ and see if it meets P802.1DG needs.
  - P802.1CQ/D0.8 is in Task Group Ballot through 5 September
  - Perhaps it can be extended, so please request if it would help.
  - Many details have not been detailed here.
  - Parameters (such as timers) remain undetermined.
- Consider whether BARC address blocks are useful.
- Consider whether BARC assignment protocols are useful.
  - Or whether alternatives would be better.
- Consider whether any variations or enhancements would improve the applicability.
More information

- P802.1CQ/Do.8, “(Draft) Standard for Local and Metropolitan Area Networks: Multicast and Local Address Assignment”