P802.1CQ-D0.8, Comment 3
- Supportive Slides -

Johannes Specht (Self)

Jens Bierschenk (Robert Bosch GmbH)

Introduction

- **Comment**
  The main focus in the document is on the dynamic protocol. The commenter believes that there is value to separately describe the forwarding lookup addressing scheme, and allow static configurable associated FDB entries (e.g., YANG) for potential use-cases in static engineered applications. By appropriate definitions of conformance requirements, this could allow conformant implementations that do not implement the dynamic protocol, just the static configuration means.

- **Contained Aspects**
  1. Static configuration (e.g., via YANG)
  2. New FDB Format(s)
  3. Conformance

Motivation for Static Configuration

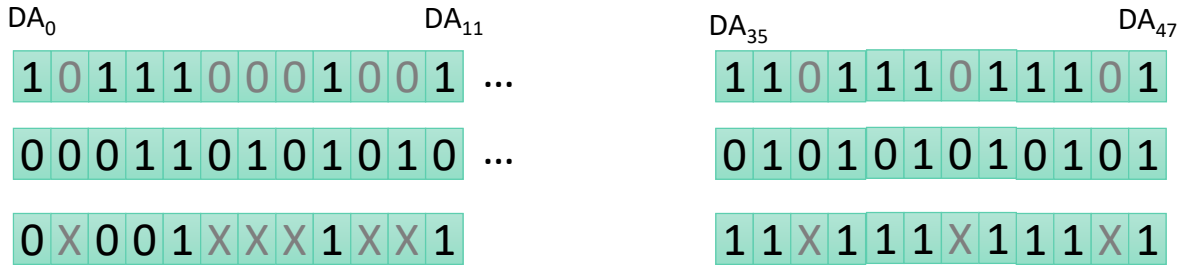
- "Classic" Automotive Applications …
  - ***… do not require*** on-the-fly configuration computation
    - "Classic" Automotive applications rely on configurations entirely engineered prior to car (mass-)production.
    - Engineering can take days, weeks, months, or even years – however, once every configuration parameter in every station is determined, this configuration can be re-produced quickly per car.
  - ***… do not like*** *indeterministic* address configurations
    - Different address ranges assigned per car during mass production:
      In some cars, the FDB may fall back to flooding → QoS guarantees gone → inacceptable
    - Different address ranges assigned per power cycle:
      Sometimes, on car startup (turn key), the FDB may fall back to flooding → QoS guarantees sometimes gone → inacceptable
  - ***… do not like*** *slow* address negotiation processes on startup
    - After turning the key (startup), the network operations state must be reached quickly (e.g., < 100 ms)
      → Typically not enough time for address negotiation by a distributed protocol.
- Examples for achieving this
  - Storing engineered address configurations in NVM *of each ECU* ("classic")
  - Storing engineered address configurations in NVM of a car-central 802.1CQ Registrar, distribute to ECUs on startup, provided that feasibility for Automotive applications (startup time, determinism, etc.) can be proven (further analysis needed)

Motivation: New FDB Formats

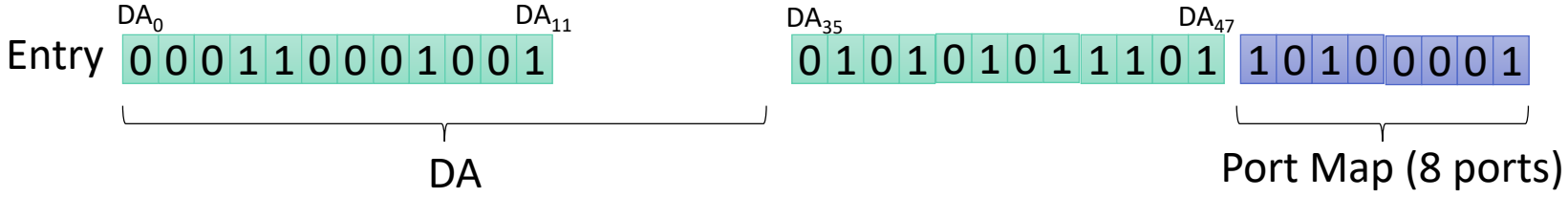
- **Current FDB Formats**
  - The current FDB entry types (802.1Q) are essentially composed of two components (VLANs skipped):
    1. A MAC address, the entry with an **exact DA match** is selected (bit-by-bit comparison)
    2. A port map, one bit per port: 1→transmission port; 0→no transmission port
  - MMRP (802.1Q) supports address blocks. But this is not reflected in the FDB, and not suited for fast startup.
- **The Issue**
  - FDB-size matters in cost sensitive markets - with one FDB entry per Stream, the FDB capacity **requirement** can become … significant.
- **The Idea**
  - **Wildcarding** of **coarse-grained** MAC address bit **blocks** in a **single FDB entry** appears promising …
  - A **block** of streams could be covered by a single FDB entry (**relaxed FDB requirement**)
  - Nothing new in Bridges:
    - Several TCAM-based implementations should be already provide the capability (fine-grained), **but**
      - the capability is not standardized, and
      - would not be friendly to hash-based FDB implementations
    - P802.1CBdb introduced mask-and-match filter
      - **hash-based unfriendly** (fine-grained)
      - **Unavailable for FDB** forwarding port lookup
  - Coarse-grained block-wise wildcarding may mitigate “unfriendliness” for hash-based implementations
    - Requires a **indicator for coarse-grained wildcarding and offset**

Discussion (1)

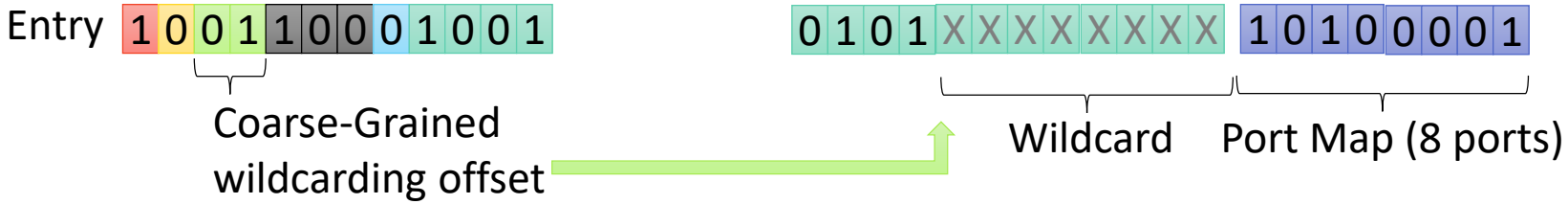
IEEE Std 802.1CBdb Stream Identification (DA only)



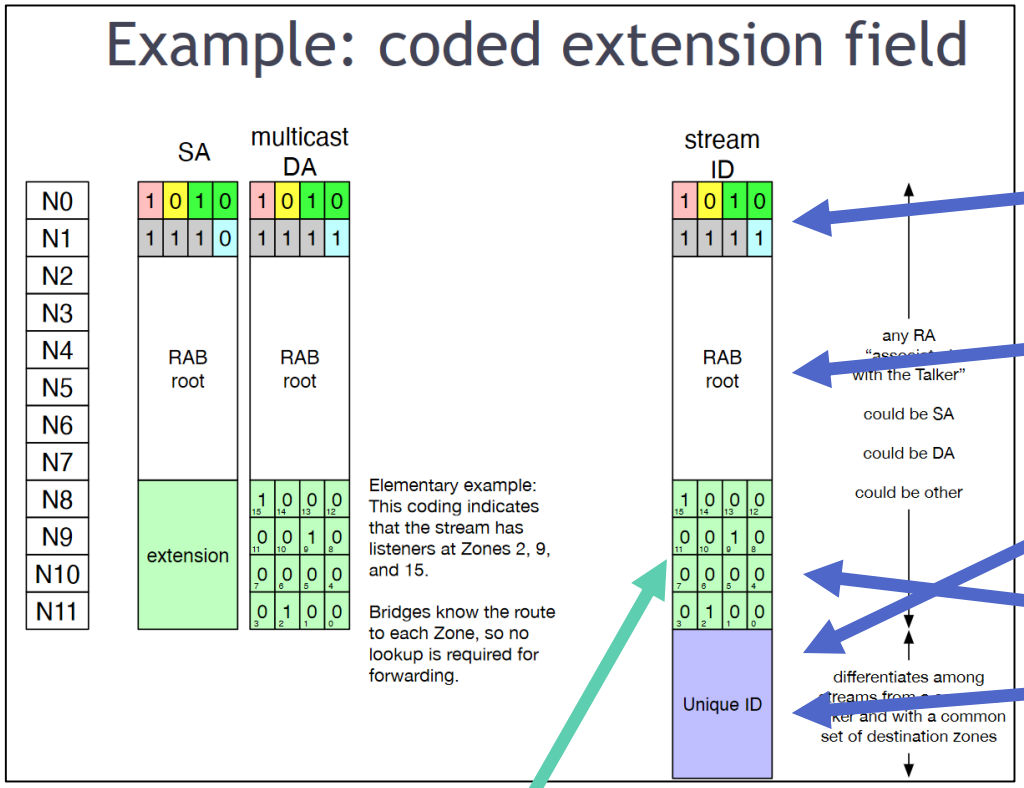
IEEE Std 802.1Q FDB



New FDB Entry



Discussion (2)



Source: https://www.ieee802.org/1/files/public/docs2022/dg-marks-barc-0922-v01.pdf

indicator for coarse-grained wildcarding and offset?

Explicit Match?

Wildcarded area?

Exchange?

New FDB entry type: **Address**?

New FDB entry type: **Port Map**?

802.1 Stds Vehicle?

Thank You for Your Attention!

Questions,
Comments,
Opinions,
Ideas?