802.1ah: CBP / PIP Interface Stacks

Stephen Haddock

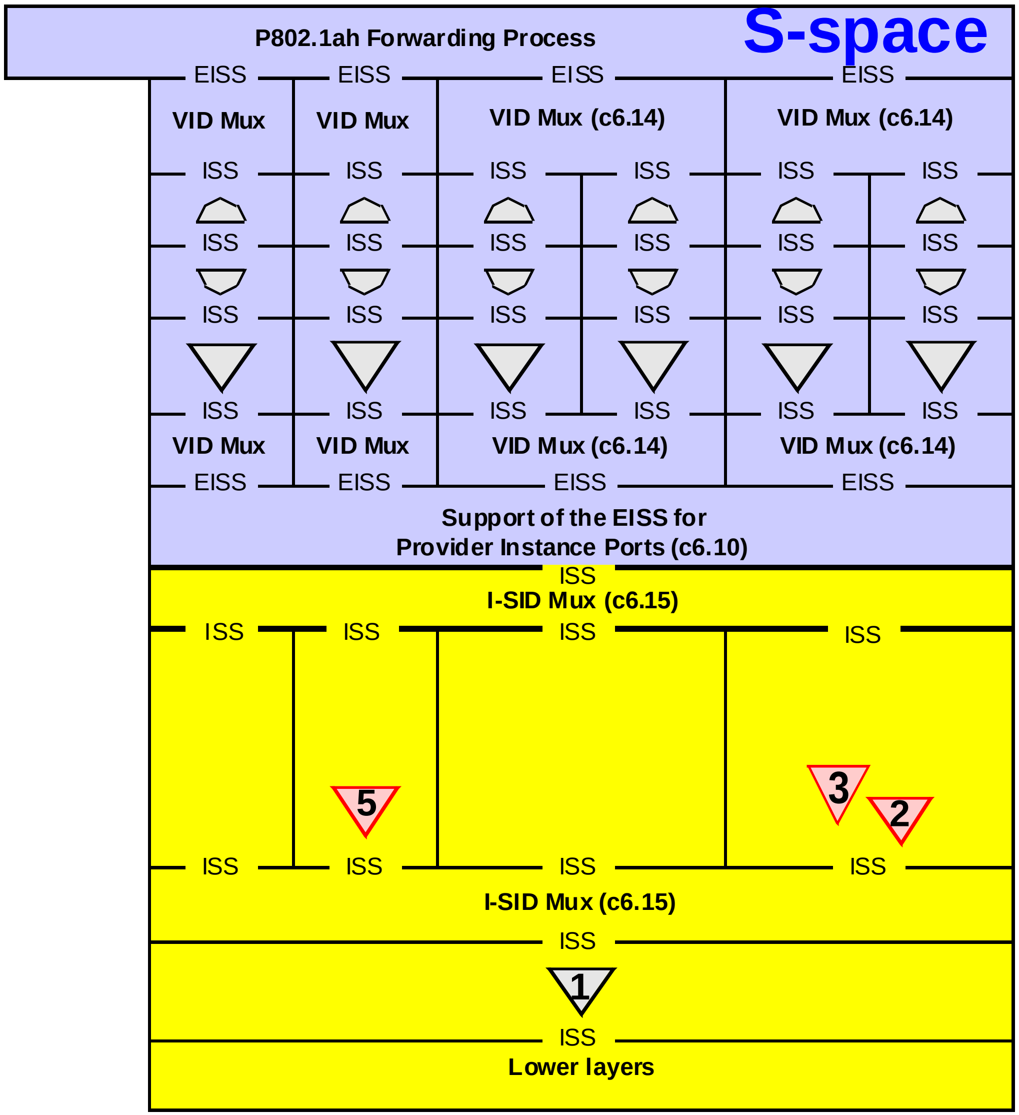
May 29, 2007

802.1 Interim, Geneva

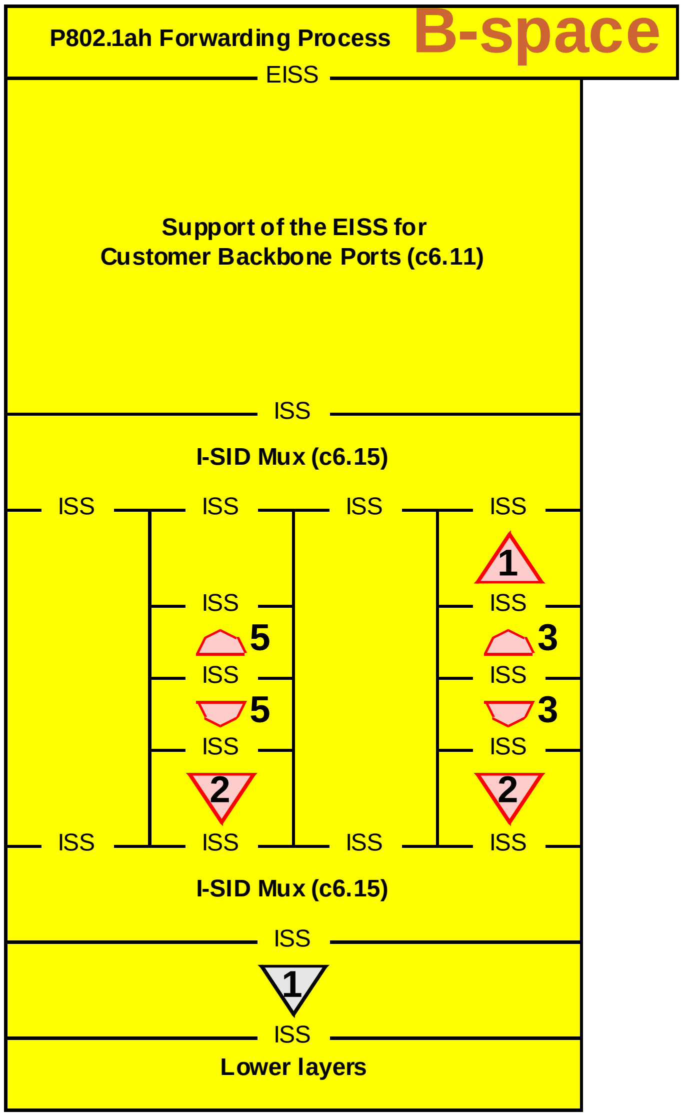
Changes in d3.5

- Incorporated “B-space” model as described in “ah-haddock-CFM-for-PBB-v2” as agreed in the March meeting.
  - At the March meeting I asked for editorial license in deciding how to partition the PIP functionality between clauses 6.9, 6.10, 6.15. The decisions to be made were:
    1. Should the ISID Multiplex Entities always be back-to-back in both PIP and CBP, or should the PIP have just the “lower” ISID Multiplex Entity (refering to Figure 6-6). (Decided yes)
    2. Should the split 6.9 functionality (S-tag handling) be split out of 6.10? (Decided no)
  - Figure 26-2 was not consistent with these decisions.
    - David Martin generated a figure consistent with these:

Corrected 802.1ah-d3.5 Figure 26-2



I-component PIP



B-component CBP

Reasons for decision 1

- Currently 6.10 multiplexes all VIPs to a single ISS interface.
  - Using back-to-back ISID Multiplex Entities below 6.10 splits stack out to per-ISID vertical columns again. Could remove the VIP multiplexing from 6.10 and create an ISS for each VIP, then use just the lower ISID Mux.
- Arguments for using back-to-back ISID Multiplex Entities:
  1. It makes 6.10 a complete self-contained functional block. If not using per-ISID CFM shims then 6.10 performs all necessary functionality and there is no need for an ISID Multiplexer Entity block.
  2. Makes the back-to-back multiplexer with CFM shims between a complete self-contained structure that is identical in both the CBP and PIP (and optional in both CBP and PIP). It is exactly analogous to the back-to-back EISS Multiplex Entity structure used to support per-VID CFM shims.
  3. Currently the use of the "Encapsulated Addresses" type value is completely contained in the ISID Multiplexer Entity subclause. Not using back-to-back ISID Muxes would require 6.10 to pass and mac_sdu with this type value.
- Arguments against:
  1. Removes some multiplexing functionality from 6.10 that is duplicated in the ISID Multiplex Entity.

Reasons for decision 2

- Currently 6.10 operates on both S-tags and I-tags.
  - The S-tag related functionality is largely redundant with 6.9. It is possible to remove the duplicated functionality and specify that each VIP has a 6.9 functional block in the stack above the 6.10 functional block.
- Arguments for splitting 6.9 functionality out of 6.10:
  1. Nice not to deal with two tags in one functional block
  2. May help resolve Panos comment carried from d3.3.
- Arguments for not splitting 6.9 functionality out of 6.10:
  1. Need to add drop_eligible to ISS (need for ISID Mux anyway).
  2. Need to modify 6.9 to pass connection_id between EISS ←→ISS.
  3. Would get per VIP PCP encode/decode tables, priority regeneration tables, S-VID translation tables. Lots of excess management objects. Can possibly make these either optional or shared between all VIPs of PIP.
  4. Exposes an ISS that would have been necessary for “S-space” model CFM shims, but allowing CFM shims here is unnecessary, redundant, and possibly confusing having decided to use the “B-space” model.

Suggested Resolutions

- Several comments were submitted in the d3.5 ballot related to the discrepancy between clause 6 and Figure 26-2.
- Strongly recommend staying with the back-to-back ISID Multiplex Entity structure in both CBP and PIP (decision 1).
- Unfortunately I think we probably need to change my choice on decision 2 in order to resolve Panos' comment #355 (carried from d3.3):
  - Breaking 6.9 functionality out of 6.10 creates an ISS per VIP between the S-tagging operations and the I-tagging operations.
  - The issue Panos points out is that these interfaces are a necessary SAP for higher layer control protocols (RSTP, MSTP, MVRP) that get "tunneled" through the PBBN.
- Make Figure 26-2 consistent with these decisions.