Questions about the IEEE 802 Architecture

Notice to WG Chair: This contribution is “previously published” per the IEEE SA Copyright Policy, as it incudes material previously published in IEEE Std 802, IEEE Std 802.1Q, IEEE Std 802.1AX, and IEEE Std 802.1CB.

Note: Personal views of contributor expressed herein.

Roger Marks (EthAirNet Associates; Huawei)

roger@ethair.net
+1 802 capable

27 January 2022

What is the IEEE 802 Family?

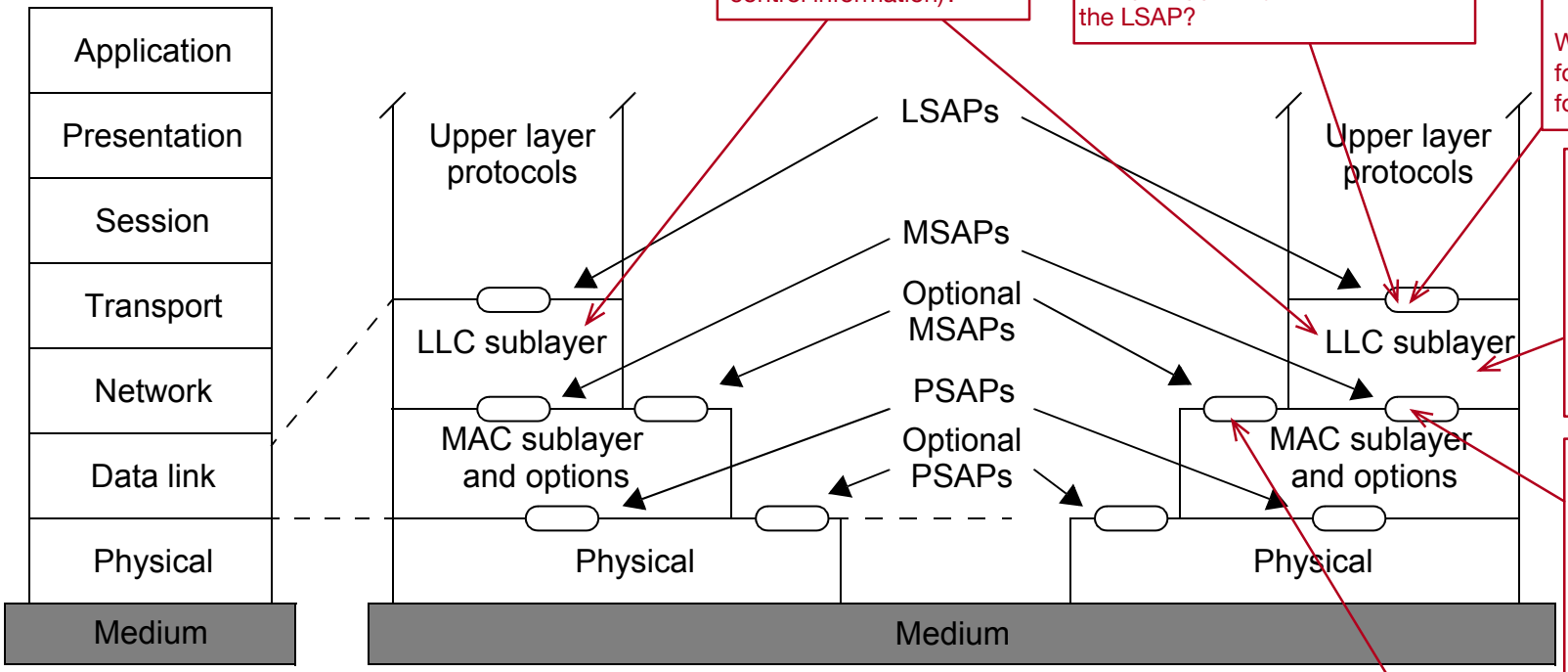
- IEEE Std 802 says:
  - *This standard serves as the foundation for the family of IEEE 802 standards published by IEEE for local area networks (LANs), metropolitan area networks (MANs), personal area networks (PANs), and regional area networks (RANs).*
  - *several types of medium access technologies are currently specified in the family of IEEE 802 standards*
- What is a "Family of Standards"?
- In what way are the standards "related":
  - The nature of the relationship is not explained or explored.
- What do IEEE 802 standards have in common?
  - They largely share addresses.
- So, is this a family, or simply a group of "roommates" with some shared addresses?

Architecture in IEEE Std 802

- IEEE Std 802 covers “Overview and Architecture”
- The Scope includes
  - “the IEEE 802 architecture is defined”
- The word “architecture” appears sparsely.
- No content purports to specify the architecture.
- The closest to a specification of the architecture is in Clause 5 (“Reference models (RMs)”), which says
  - “Figure 3 shows the architectural view of IEEE 802 RM for end stations and its relation to the OSI/RM. A variation of the model applies within bridges, as described in 5.3.2.”
- Where is the IEEE 802 architecture specified?

IEEE 802 Reference Model

MSAP MAC service access point
LSAP link service access point



Can two LLC entities communicate (e.g. share control information)?

How does the application express QoS requests?

How does LLC arbitrate when multiple upper layers have data for the LSAP?

What are the characteristics of the Link Layer service provided to the Link Layer client at the LSAP, analogous to the LLC specification in IEEE Std 802.2?

What parameters are passed?

What are the allowed frame formats? Which formats are forbidden or blocked?

What are the functions of the LLC?

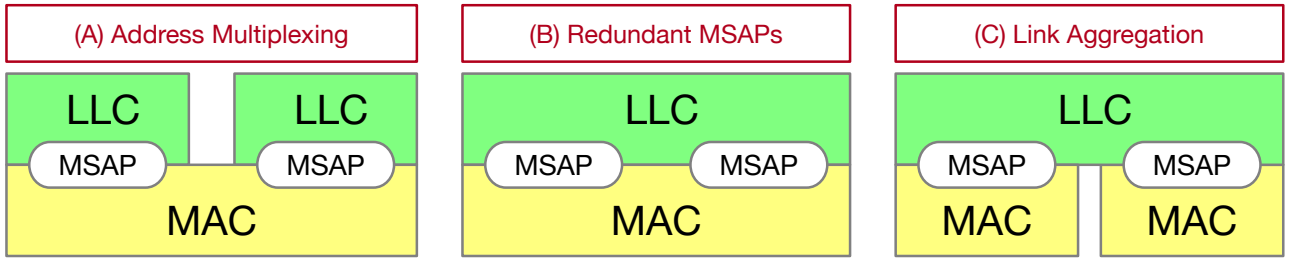
How are the parsing rules used to identify the service parameters (from LSAP and from MSAP)?

What are the allowed frame formats?

How is the MSAP characterized by MAC address; e.g., can an MSAP have multiple MAC addresses; e.g. multicast addresses).

Does the MAC handle this? Are there arbitration issues when both LLCs send data to the MAC?

Figure 3—IEEE 802 RM for end stations



Do the MSAP and Optional MSAP allow configurations (A), (B), and (C)? The figure seems to suggest (A). IEEE 802 says “MAC sublayer provides one or more MAC service access points (MSAPs) as interfaces to the LLC sublayer in an end station”; this could be (A) or (B). Is (B) allowed? Do we need (B)? Is Link Aggregation (C) within the architecture?

IEEE 802: LLC and HLPDE

IEEE Std 802 says "The higher layer protocol discrimination entity (HLPDE) is used by the LLC sublayer to determine the higher layer protocol to which to deliver an LLC sublayer protocol data unit (PDU). Two methods may be used in the HLPDE. The two methods are… (EPD) and (LPD)."
- Does the MAC care which one is used?
- How does the recipient LLC know which one was used?
- Does the LL Client need to be able to support both, know the HLDPE method of the LLC, and format frames accordingly?
- Can each LL Client select its preferred identifier type (e.g. DSAP/SSAP, EtherType, OUI-based, etc.)?
- What does the IEEE Std 802 Introduction (which "is not part of IEEE Std 802-2014") mean by "While the protocol identification mechanism specified by ISO/IEC 8802-2 (IEEE Std 802.2™, withdrawn) is still used, its use for new standards has been deprecated." In which standard is it deprecated?
- Does the HLPDE function in the down direction as well as up?

The LLC sublayer contains a variety of entities, as illustrated in Figure 6.

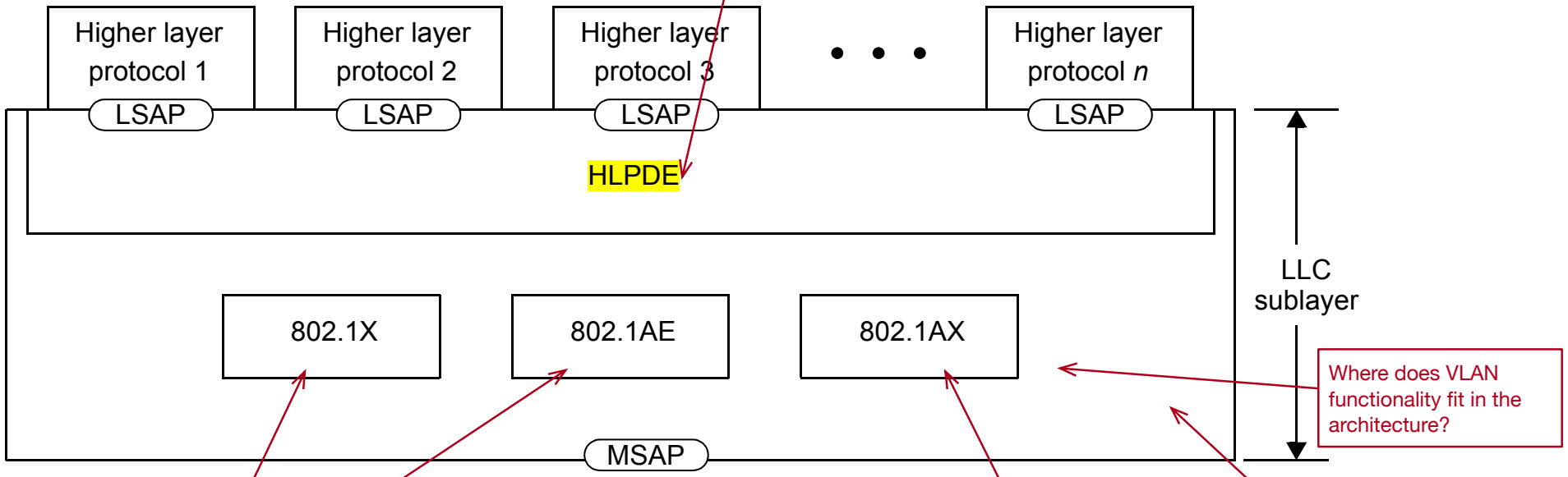


Figure 6—LLC sublayer in 802 RM

Where does VLAN functionality fit in the architecture?

What's else is missing? Other methods using end-station tags; e.g.:
- 802.1CB Redundancy-tag for Frame Replication and Elimination for Reliability (FRER)
- CN-tag for congestion notification

How do 802.1X Port-Based Network Access Control and 802.1AE MAC Security fit into the sublayered architecture?

How does 802.1AX Link Aggregation fit into the architecture? Does this imply multiple MSAPs (to multiple MAC entities) below an LSAP (C), even though only one is shown?

(C) 802.1AX Link Aggregation in the LLC

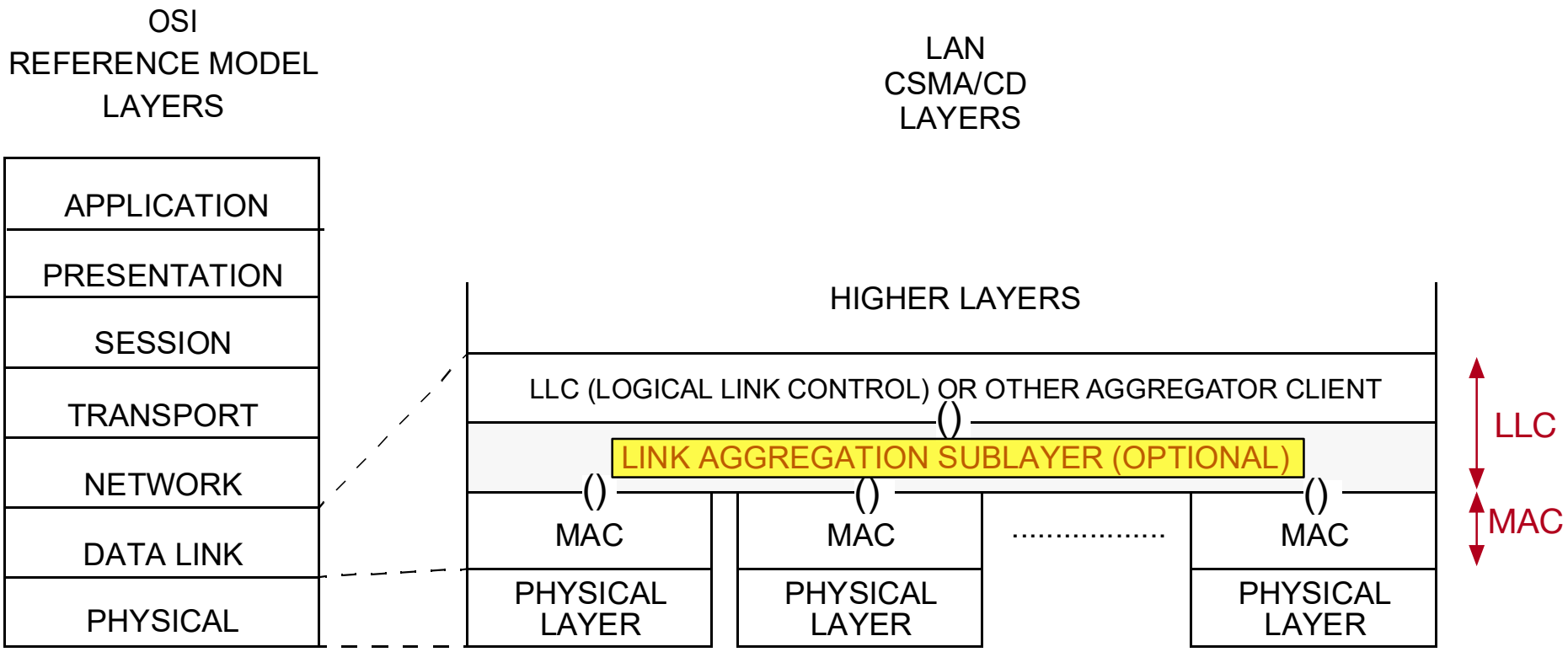


Figure 6-1—Architectural positioning of Link Aggregation sublayer

Tags and Shims

What is the role of tags and shims in the 802 architecture? They are not mentioned in IEEE Std 802.

Per 802.1Q:

•*Tag header: A header that allows priority information, and optionally, Virtual Local Area Network (VLAN) identification information, to be associated with a frame.*

•*Tagged frame: A frame that contains a tag header immediately following the Source MAC Address field of the frame.* [Note: Not every frame with a tag is a "tagged frame".]

•*shim: A protocol entity that uses the same service as it provides.*
NOTE—Within this standard, shims make use of the Internal Sublayer Service (ISS) or the Enhanced Internal Sublayer Service (EISS).

A VLAN-aware end station can use the EISS Multiplex Entity (6.17) to provide multiple SAPs, one per VID of interest, to separate MAC Clients.
-IEEE Std 802.1Q-2018

The EISS Multiplex Entity enables shims defined for the ISS to use the EISS. Figure 6-6 illustrates two EISS Multiplex Entities placed back-to-back.

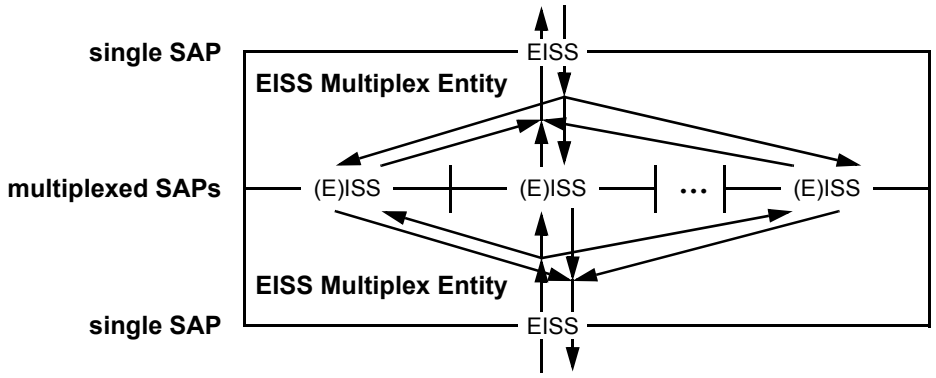


Figure 6-6—Two back-to-back EISS Multiplex Entities

Tags and Shims in End Stations

Without detailing bridge architecture, IEEE Std 802 still should consider the architecture of tags in end stations.

Many tags cannot appear in end stations.

802 does not provide a unified list of tags. They are in various standards.

Rules for tag ordering can be difficult to ascertain.

tag	Ethertype	standard	end station
C-tag	81-00	802.1Q	yes
CN-tag	22-E9	802.1Q	yes
R-tag	F1-C1	802.1CB	yes
Sec-tag	88-E5	802.1X/802.1AE	yes
S-tag	88-A8	802.1Q	no
B-tag	88-A8	802.1Q	no
I-tag	88-E7	802.1Q	no
F-tag	89-4B	802.1Q	no
E-tag	89-3F	802.1BR	no

Protocol stack per 802.1CB (FRER)
(Informative Annex C.1)

In the 802 architecture, everything between MAC and upper layers is LLC, so this is all within LLC.

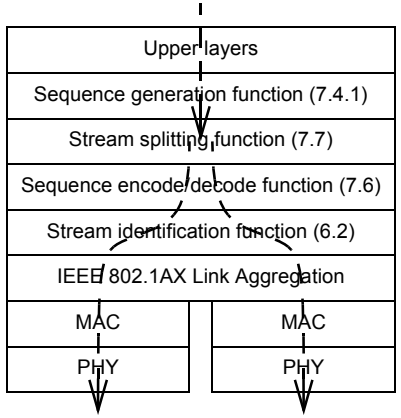


Figure C-2—Protocol stack for End System B in Figure C-1

In the 802 architecture, everything between MAC and upper layers is LLC, so this is all within LLC.

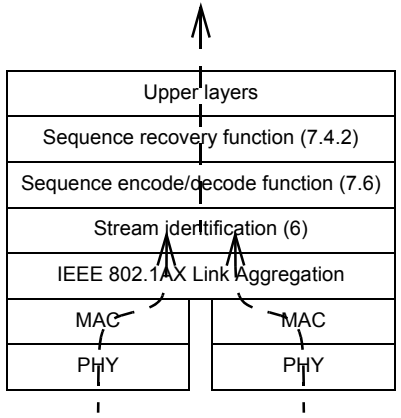


Figure C-3—Protocol stack for End System G in Figure C-1 and Figure C-4

Field	Offset	Length
Destination MAC address	**0**	**6**
Source MAC address	**6**	**6**
C-TAG EtherType	**12**	**2**
Priority, DE, VLAN ID	**14**	**2**
R-TAG EtherType	**16**	**2**
Reserved	**18**	**2**
Sequence number	**20**	**2**
Payload Length/EtherType	**22**	**2**
Data	**24**	***n***
Frame Check Sequence	**24+*n***	**4**

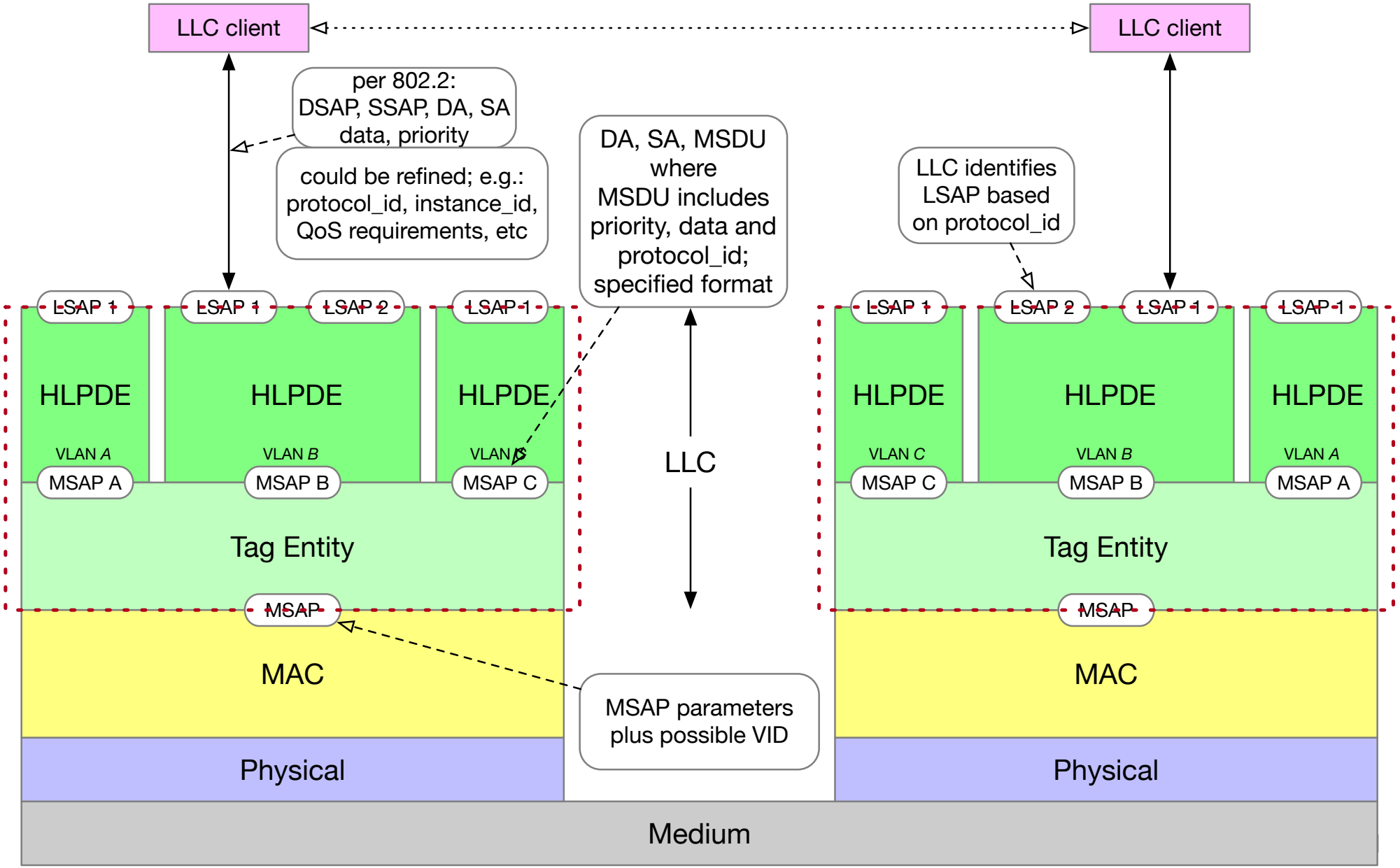
Figure 8-3—Example Ethernet frame format

Tags and Shims in the 802 architecture

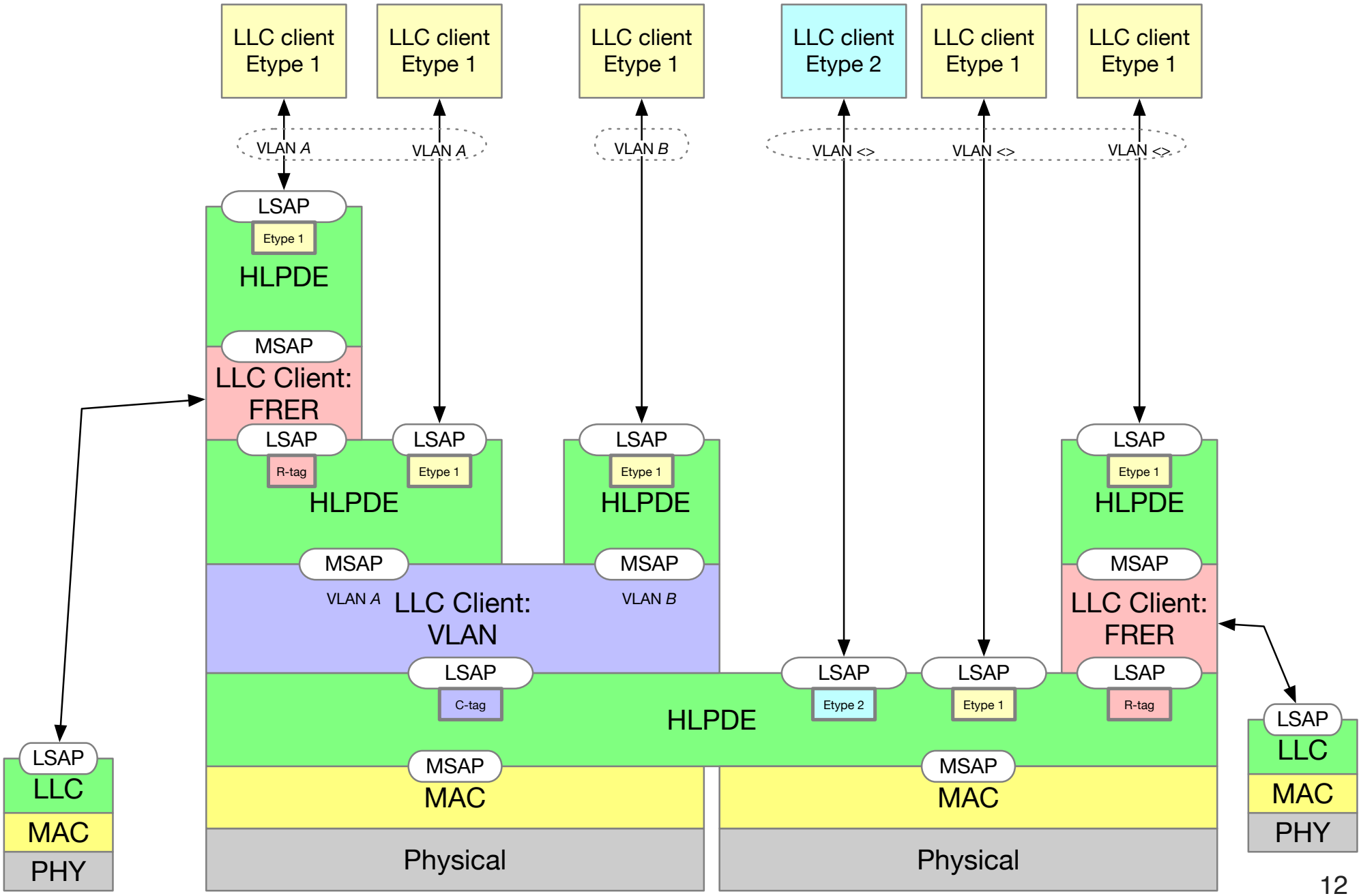
The following figures illustrate various models of Tags and Shims in the 802 architecture.

The perspective evolves as the slides proceed, and some views are replaced by subsequent ones.

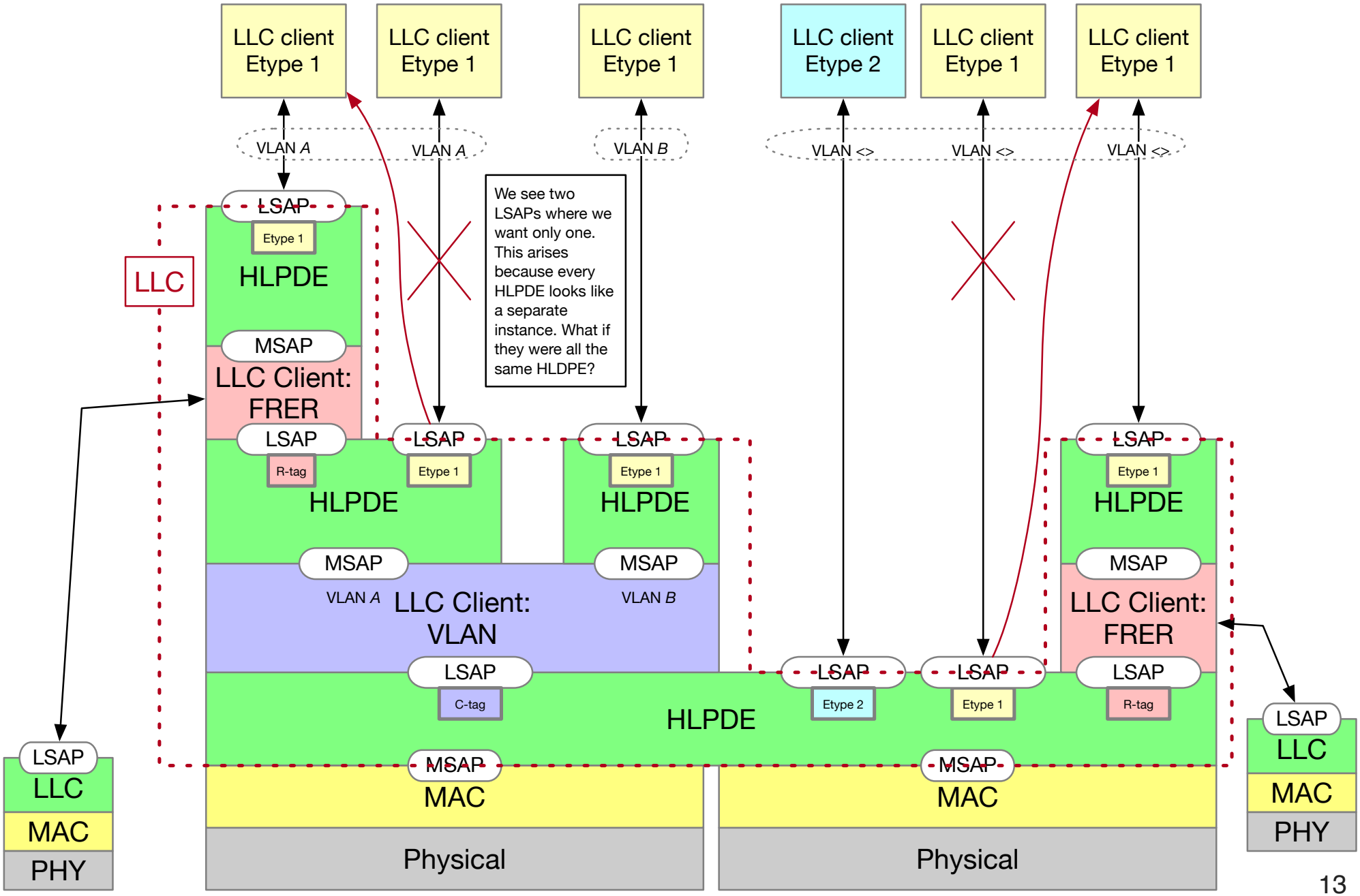
Tags in the architecture



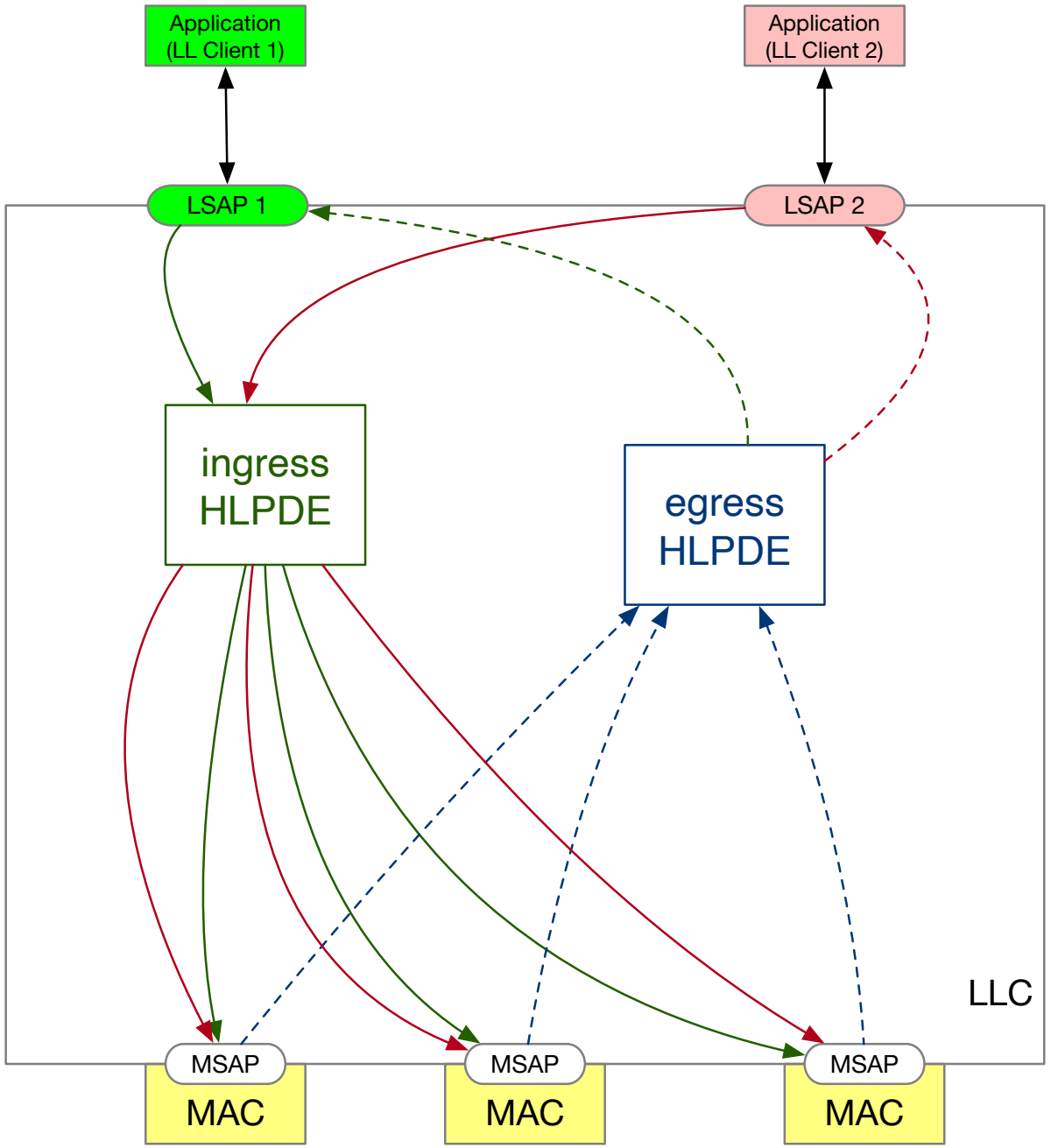
Model with multiple tags



Model with multiple tags



LLC Model

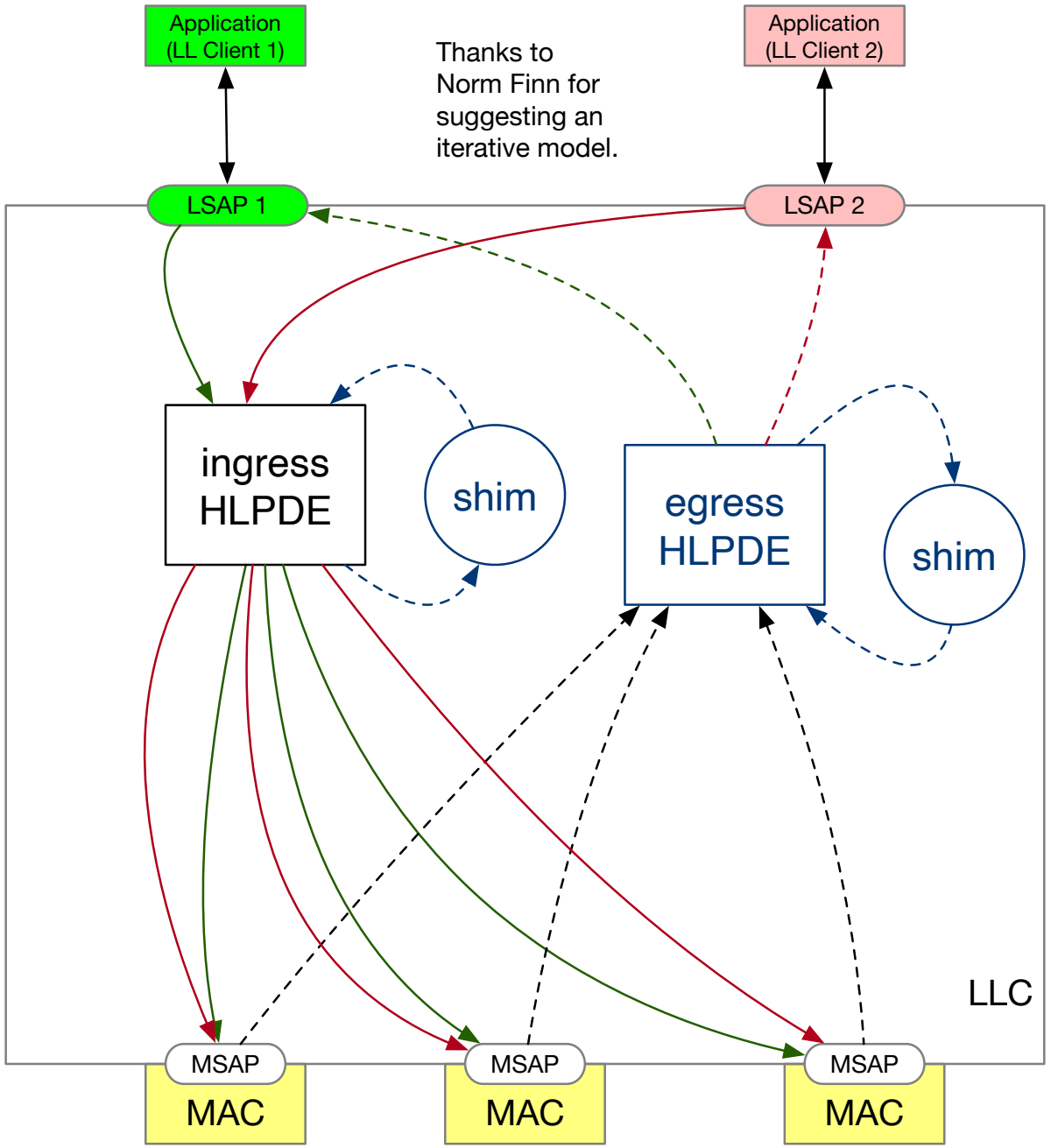


In 802.1CB FRER case, we can represent the Annex C.1 functionality within LLC.

ingress HLPDE:
- Sequence generation function
- Stream splitting function
- Sequence encode/decode function
- Stream identification
- 802.1AX Link Aggregation
- protocol ID insertion
- C-tagging
- MAC address insertion

egress HLPDE:
- MAC address filtering
- C-detagging and demultiplexing
- protocol ID demultiplexing
- 802.1AX Link Aggregation
- Stream identification
- Sequence encode/decode function
- Sequence recovery function

LLC Model showing shim iterations



In 802.1CB FRER case, we can represent the Annex C.1 functionality within the LLC.

ingress LLC function, per 802.1CB Annex C.1:
- Sequence generation function
- Stream splitting function
- Sequence encode/decode function
- Stream identification
- 802.1AX Link Aggregation
- protocol ID insertion
- C-tagging
- MAC address insertion

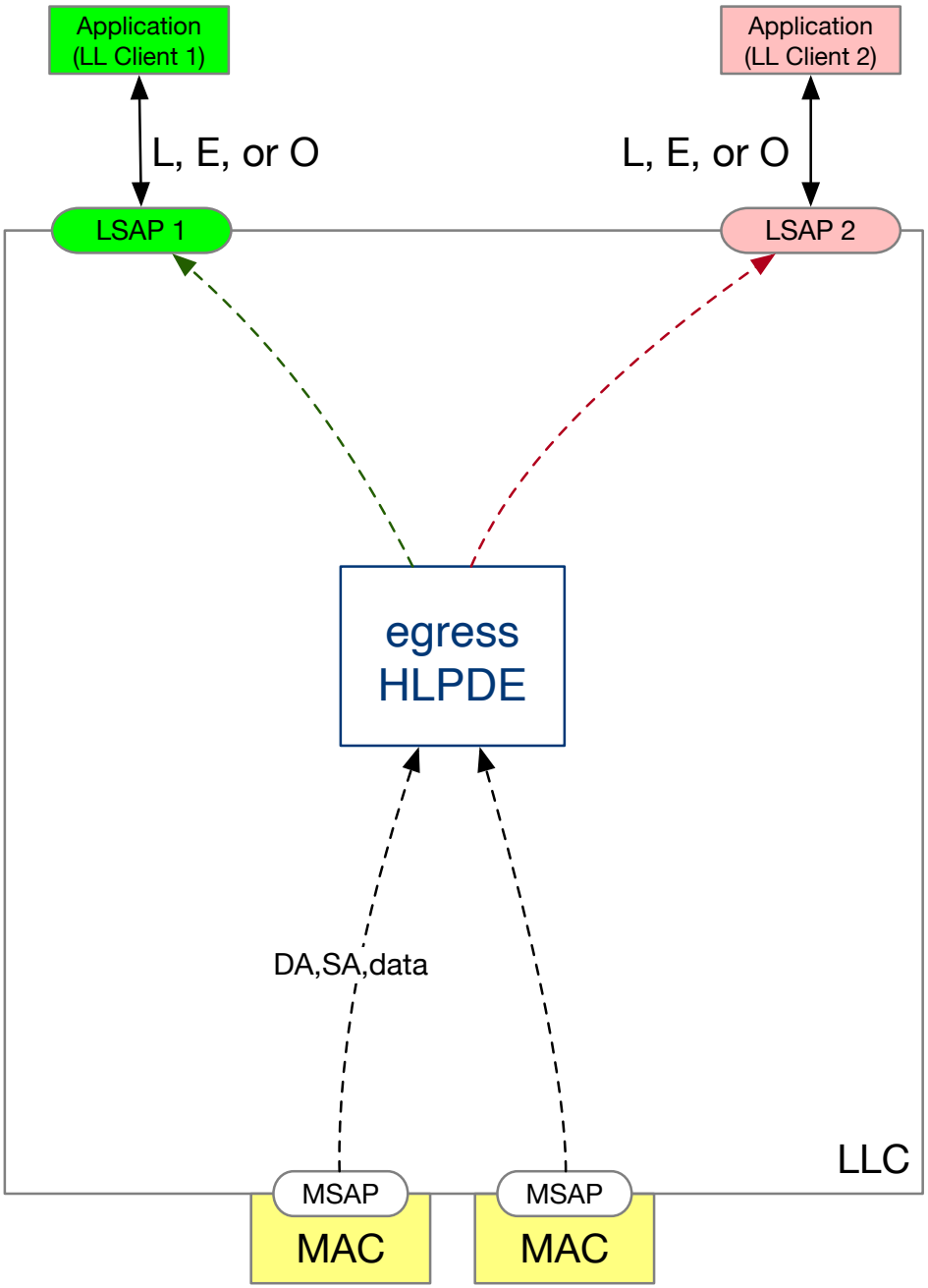
egress LLC function, per 802.1CB Annex C.1:
- MAC address filtering
- C-detagging and demultiplexing
- protocol ID demultiplexing
- 802.1AX Link Aggregation
- Stream identification
- Sequence encode/decode function
- Sequence recovery function

egress LLC function, first shim:
- C-tag detection
- C-detagging and demultiplexing

egress LLC function, second shim:
- R-tag detection
- R-detagging
- 802.1AX Link Aggregation
- Stream identification
  *may peek into next tag (protocol ID)
- Sequence encode/decode function
- Sequence recovery function

egress LLC function, third shim:
- protocol ID demultiplexing

HLDPE Egress (Ethernet): Base Process

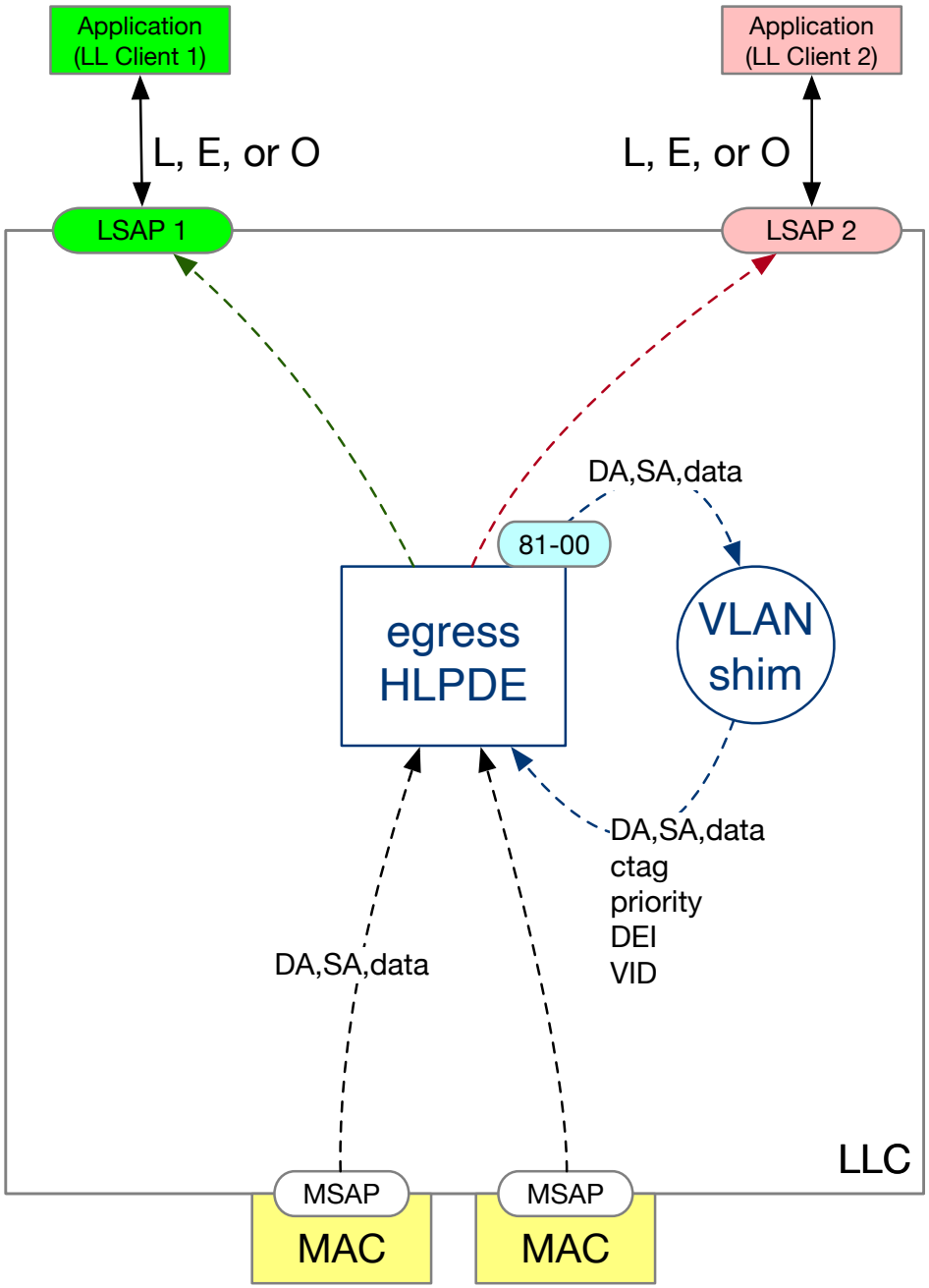


HLDPE Base Process: protocol discrimination
receive parameters including DA, SA, data, MSAP, etc.

X == first 3 bytes of data field
- 0x05DC<X<0x0600: discard
- X<0x05DD:
  - Y == first 3 bytes of data field
  - if Y = 0xAAAA03
    - Z == next 3 bytes of data field
      - if Z = 0x000000
        - E == next 2 bytes of data field
          - if E is a known 802 tag type
            - discard
          - else
            - if E is 0x80F3
              - forward parameters to Ethertype E'
            - else
              - forward parameters to Ethertype E
      - else
        - O == next 5 bytes of data field
        - forward parameters to O-Identifier type O
  - else
    - L == next 2 bytes of data field
      - forward parameters to LSAP Identifier L
- X>0x05FF:
  - Y == first 2 bytes of data field
  - if Y = 0xBB87
    - O == next 5 bytes of data field
    - forward parameters to O-Identifier type O
  - else
    - E == next 2 bytes of data field
    - forward parameters to Ethertype E

EtherType			
0x88B7	O Identifier		
Length	DSAP/ SSAP	0x03	
Length	0xAAAA03	0x000000	EtherType
Length	0xAAAA03	O Identifier	

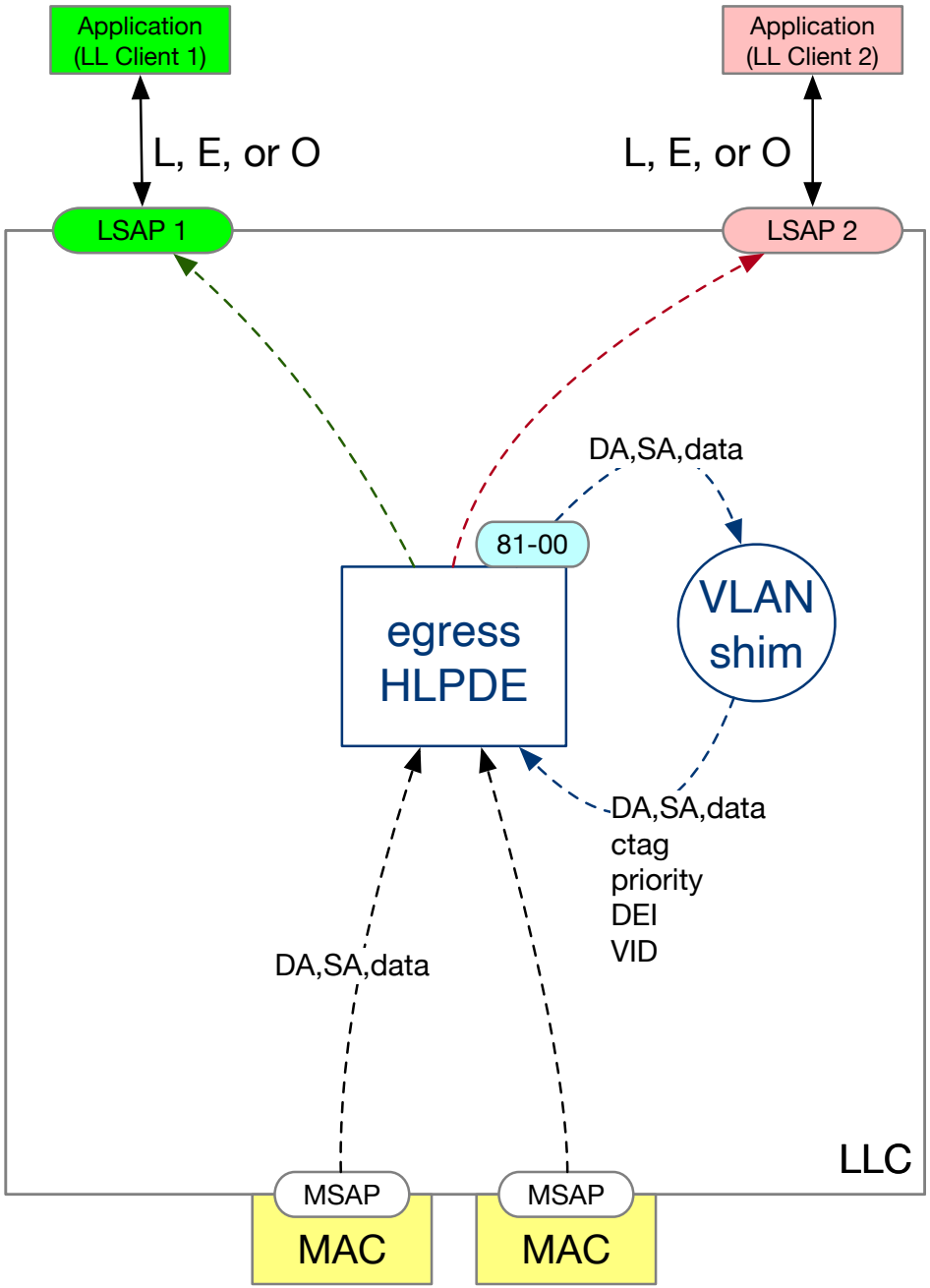
VLAN Shim



- if ctag, cntag, rtag, sectag… (tag flags that shall not precede VLAN tag) then discard
- strip first 2 bytes of data
- X == first 2 bytes of data
- strip first 2 bytes of data
- priority == first 3 bits of X
- DEI == next bit of X
- VID== next 12 bits of X
- return data, ctag=true, priority, DEI, VID

Now we see how to update the HLPDE Egress Base Process.

HLDPE Egress with VLAN Shim

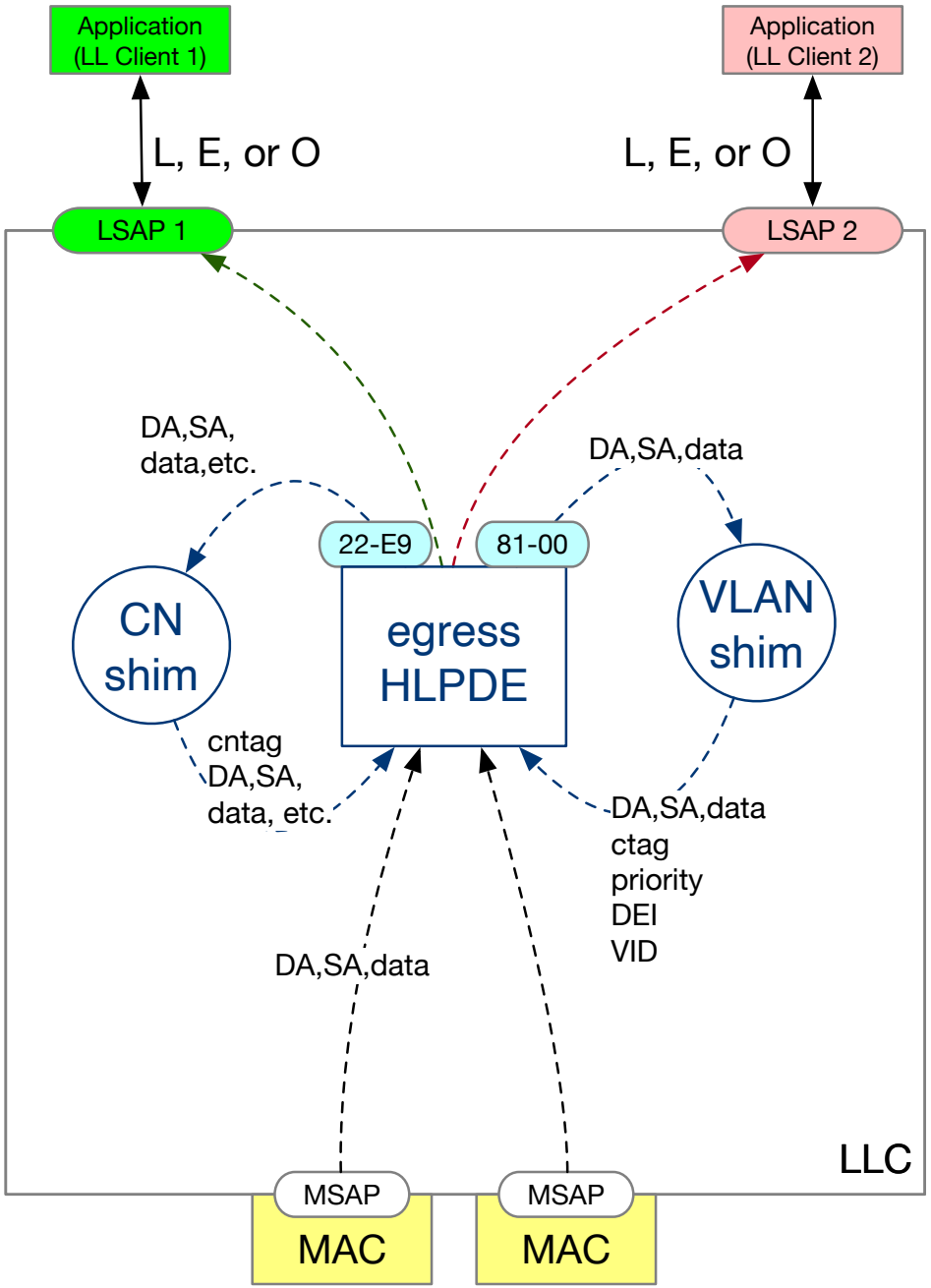


receive parameters including DA, SA, data, MSAP, ctag, priority, DEI, VID, etc.

- 0x05DC<X<0x0600: discard
- X<0x05DD:
  - Y == first 3 bytes of data field
  - if Y = 0xAAAA03
    - Z == next 3 bytes of data field
    - if Z = 0x000000
      - E == next 2 bytes of data field
      - if E is a known 802 tag type
        - discard
      - else
        - if E is 0x80F3
          - forward to Ethertype E' at VID
        - else
          - forward to Ethertype E at VID
    - else
      - O == next 5 bytes of data field
      - forward to O-Identifier type O at VID
  - else
    - L == next 2 bytes of data field
    - forward to L at VID
- X>0x05FF:
  - Y == first 2 bytes of data field
  - if Y = 0xBB87
    - O == next 5 bytes of data field
    - forward to O-Identifier type O at VID
  - else
    - E == next 2 bytes of data field
    - else
      - forward parameters to Ethertype E at VID

Note: Instead of stripping tags, we could preserve the frame but instead return a full frame with a pointer to the next tag. That may be better? It would not require returning a list of parameters.

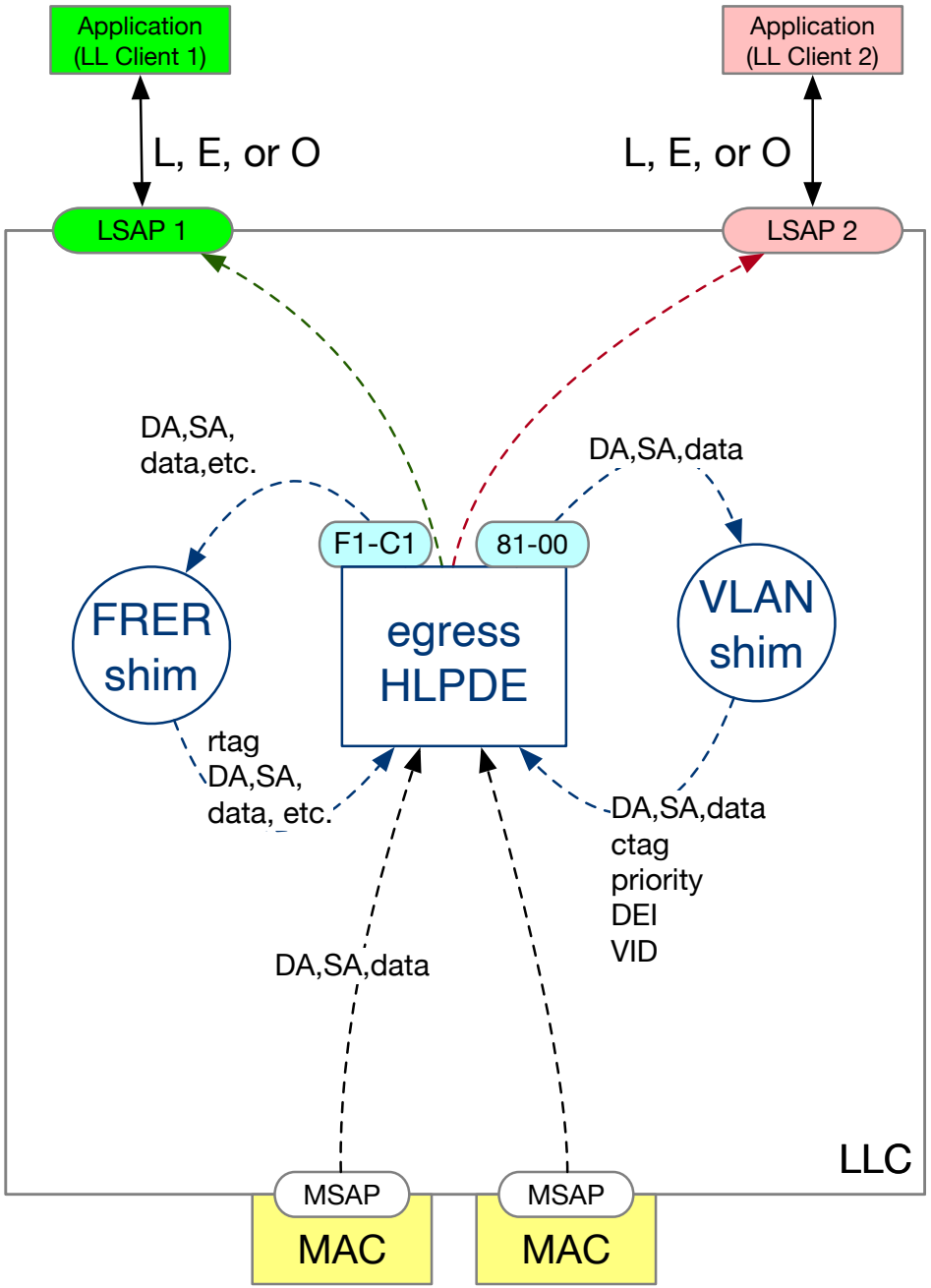
Congestion Notification (CN) Shim



- if cntag (etc; tag flags that shall not precede CN-tag) then
  discard
- strip first 2 bytes of data
- X == first 2 bytes of data
- strip first 2 bytes of data
- pass X to Congestion Notification Message creation function
- return cntag=true and modified parameters

- No updates to HLDPE parameters required to support CN Shim?

FRER Shim



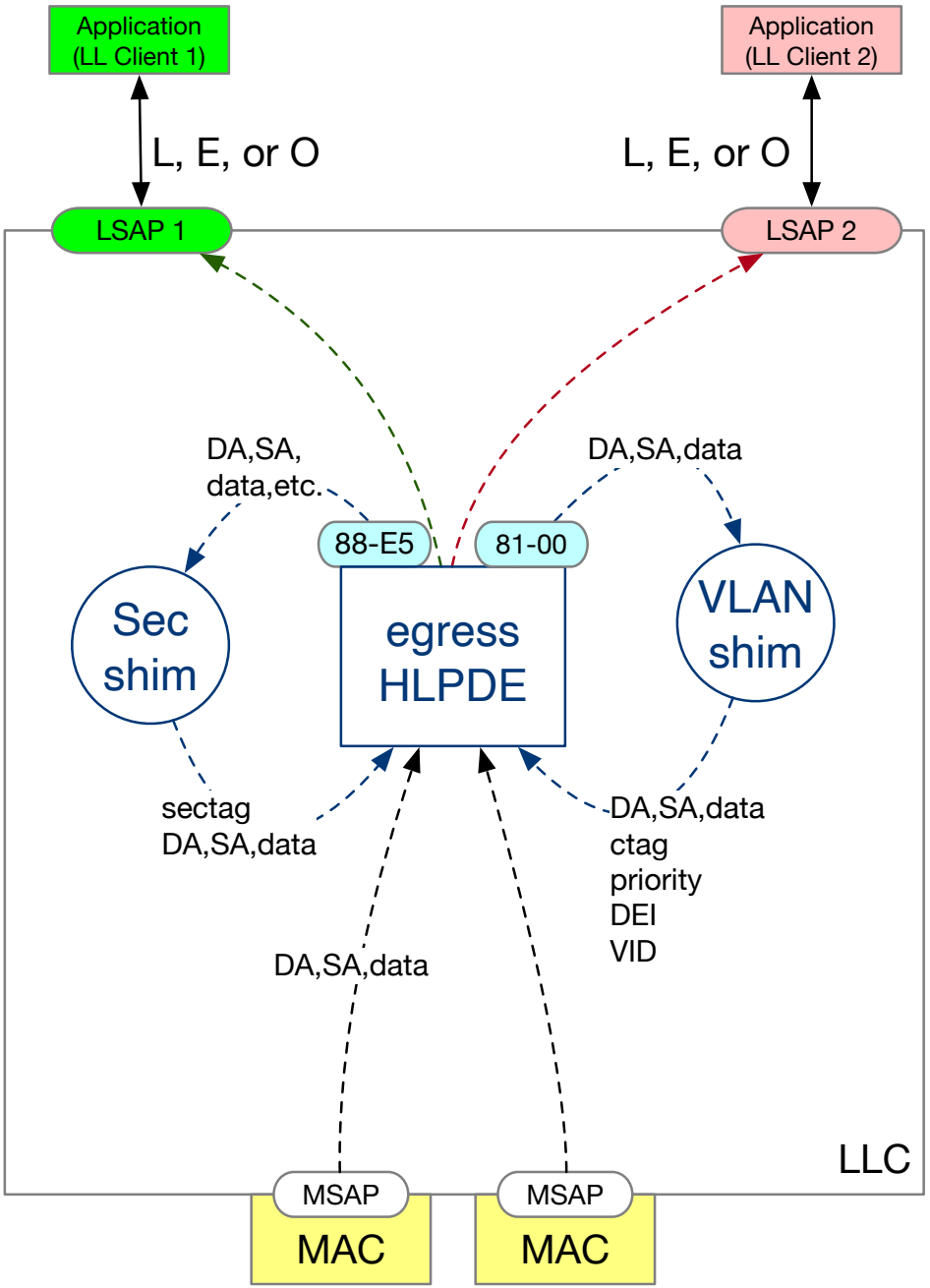
- if (tag flags that shall not precede R-tag) then
  discard
- strip first 2 bytes of data
- pass input parameters to FRER function

Note: Input parameters include DA, SA, data, and MSAP, along with parameters passed to HLPDE from prior shims; e.g., ctag, priority, DEI, VID. FRER uses these parameters in, e.g., stream identification.

- receive parameters from FRER function (not 1:1 with ingress frames)
- return rtag=true and parameters

- No updates to HLDPE parameters required to support FRER Shim?

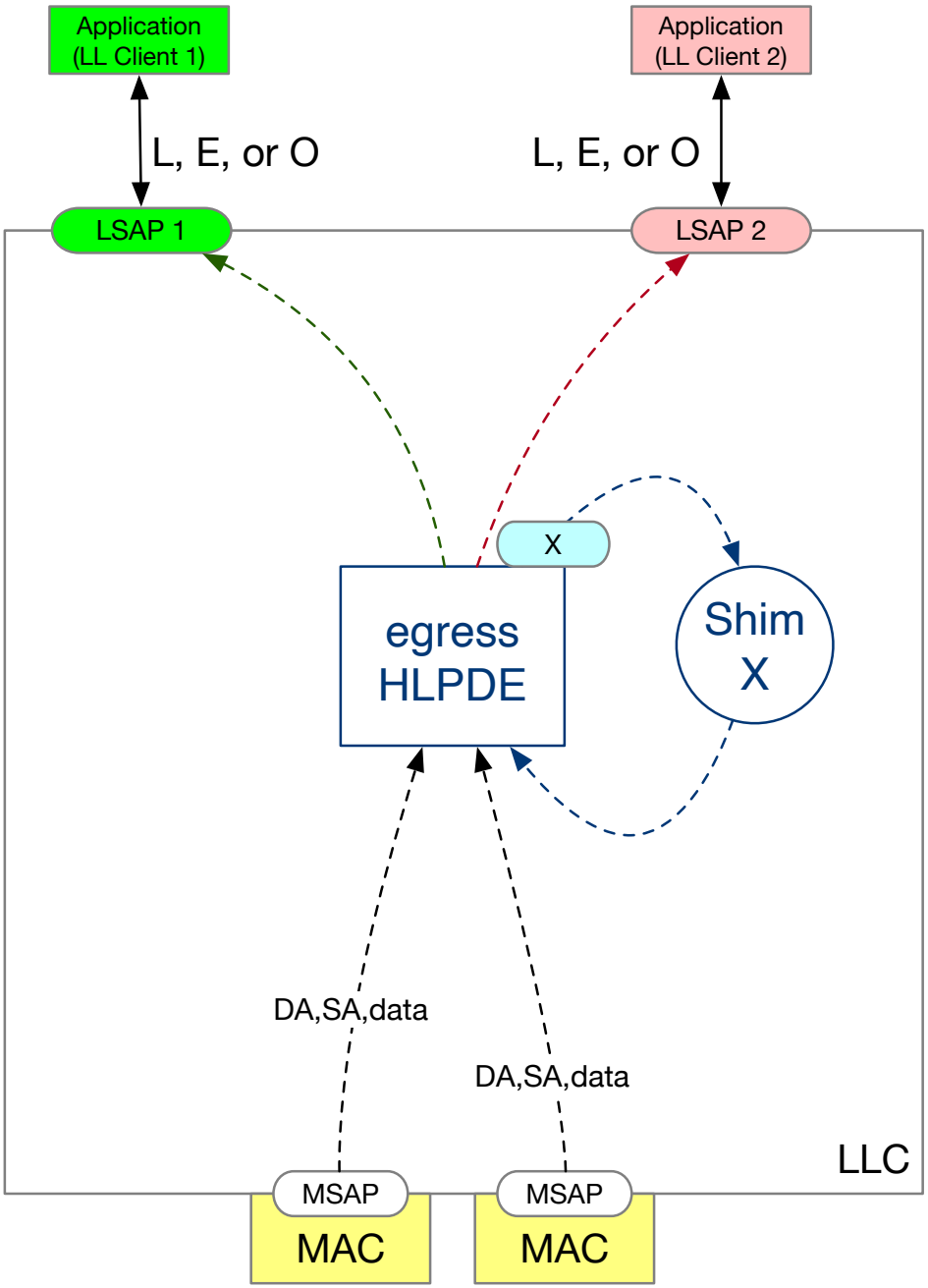
MacSec Shim



- if (any tag flag except ctag) then
  discard
- process Sec

- No updates to HLDPE parameters required to support Sec Shim?

Future Shims



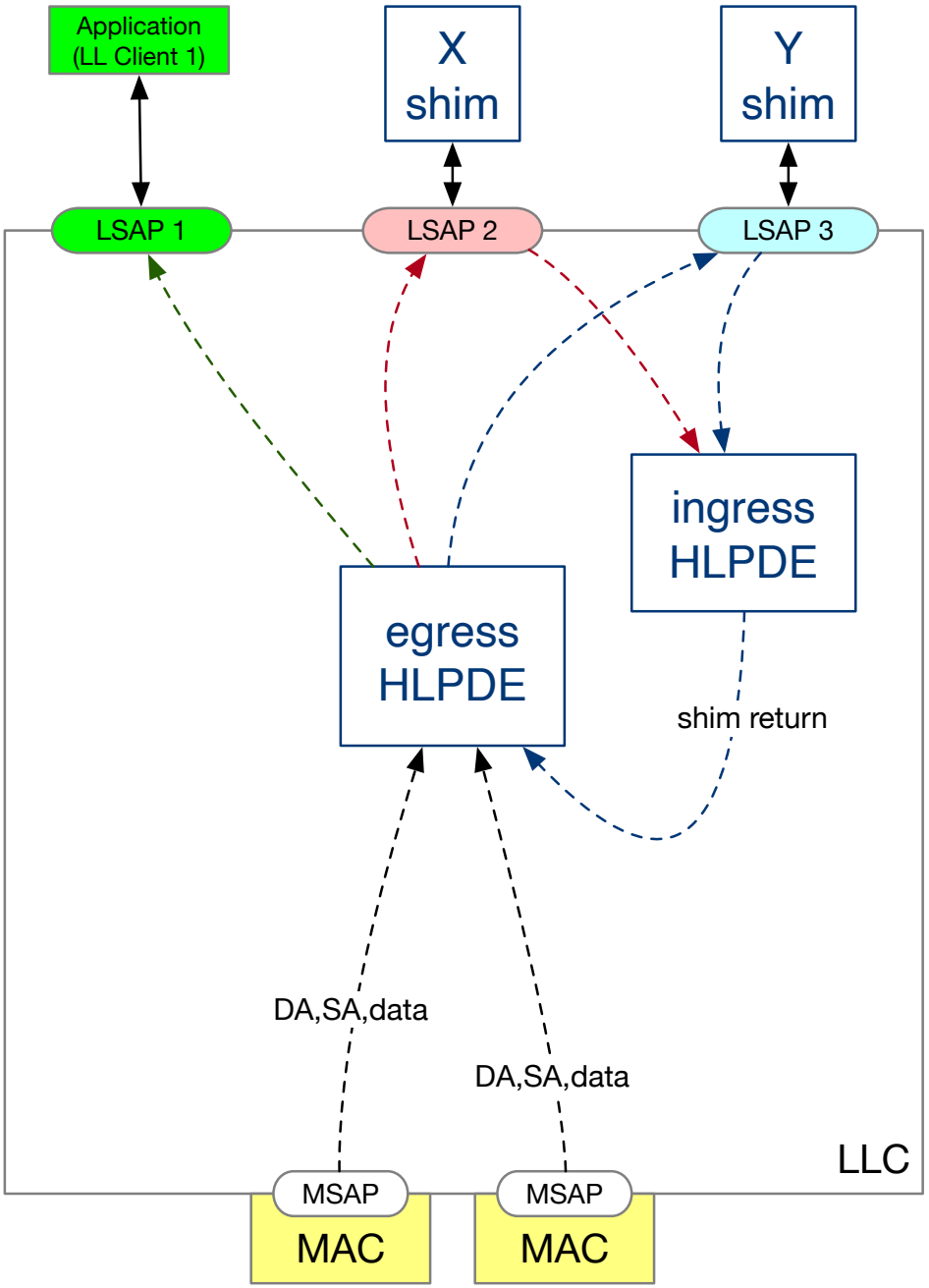
Does a new standard shim require an update to the HLPDE spec?

A generic 802 Shim Ethertype could be introduced for future shims, with a subtype field to distinguish them. This is probably not needed?

Should be possible to add a generic shim allowing new Ethertype.

Could be complicated to add enough flexibility to HLDPE to allow adding a shim requiring HLPDE to handle specific parameters.

Generic Non-standard End Station Shim



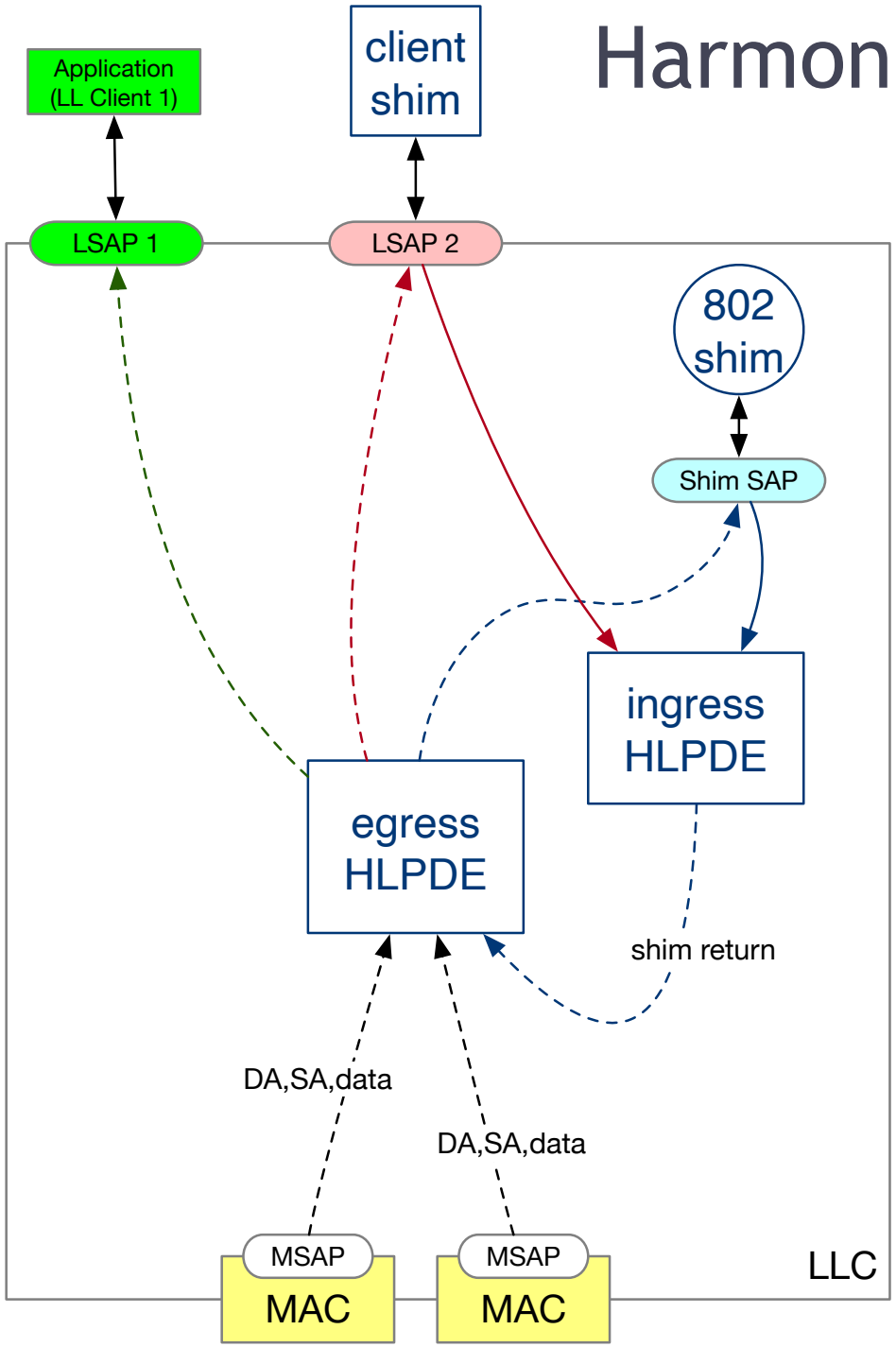
Possible architecture to model a generic non-standard end station shim.

This introduces the ingress HLPDE as a means to return frame to the egress HLPDE, with a “shim return” pathe.

Need to understand how to trigger the ingress HLPDE to use this path. For example, ingress HLPDE recognizes a local DA.

Strict rules for shim return would be needed.

Harmonized Shim Model

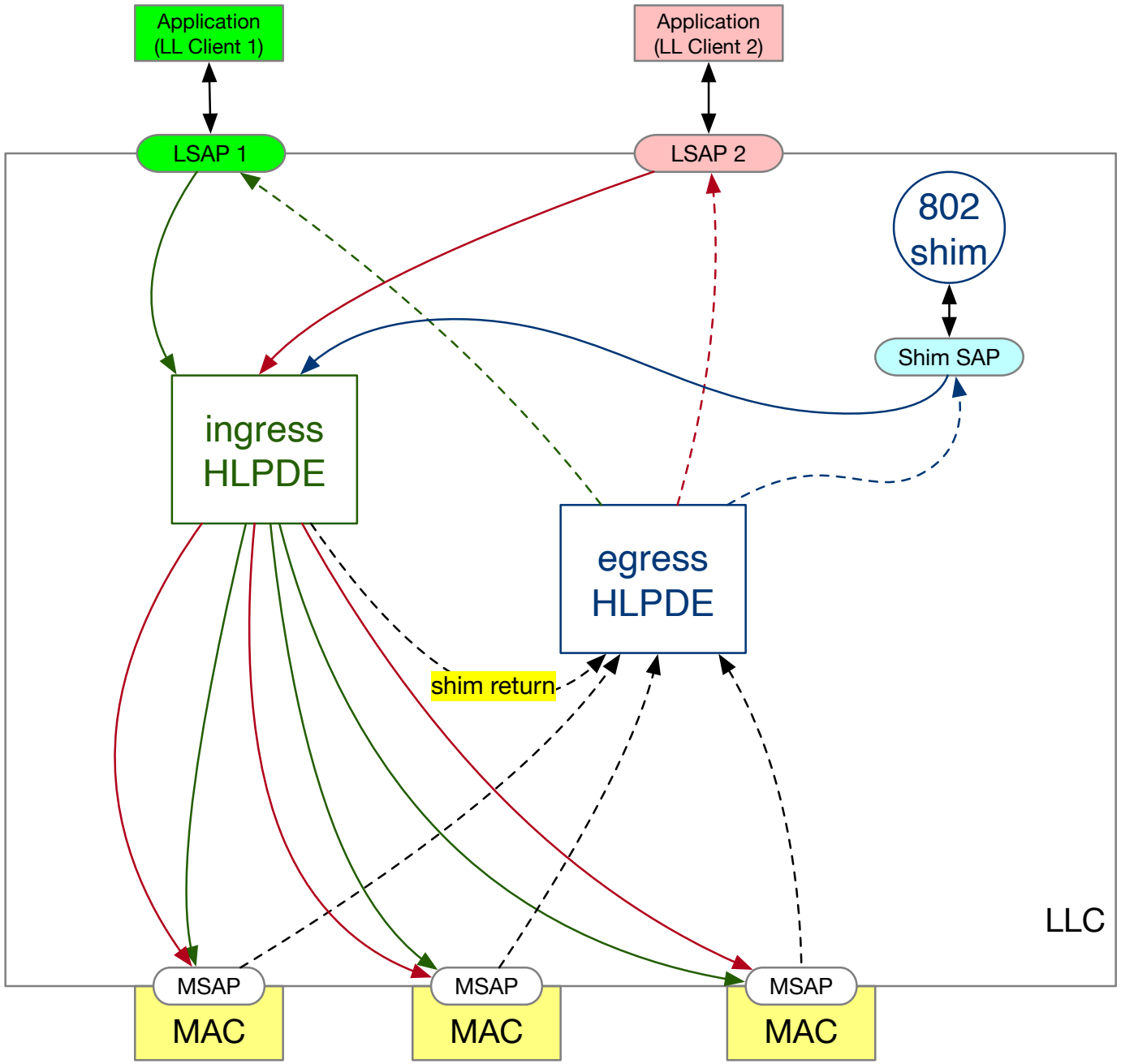


Possible architecture to model both standard and non-standard shims.

Shim SAP works like an LSAP and uses the shim return path, but Shim is within the 802 Link Layer.

All shims return parameters the same way, using a limited set of parameters.

LLC Model with Shims



Need to think further about how shim return is triggered and how it functions.

Still need to model for the entire HLPDE ingress function.

Link Layer Service

- What is the Link Layer Service that IEEE 802 provides to the application at the upper layers?
- Open Systems Interconnection (OSI) specifies 7 layers.
- ITU-T X.212 (OSI "Data Link Service [DLS] Definition") describes the service provided by the DLS to the Network Layer
  - includes
    - Characterization of the service provided
    - Primitive actions and events
  - Abstract specification, but many details (body is 26 pages)
  - specifies that the DLS "transfers DLS user data transparently, without restricting or interpreting the content or format of the data."
  - Specifies that DLS user can request QoS characteristics
  - etc.
- IEEE Std 802.2 describes the LLC service
- What is IEEE 802 Link Layer Service at the LSAP?

Control and Management

- IEEE Std 802 says "IEEE 802 network management provides protocols for exchange of management information between stations. The media-independent control function (MICF) is a parallel control plane that provides control functions for different MAC and PHY sublayers."
  - MICF is specified in IEEE Std 802.21.
- Is this sufficient to describe all control in the IEEE 802 network?

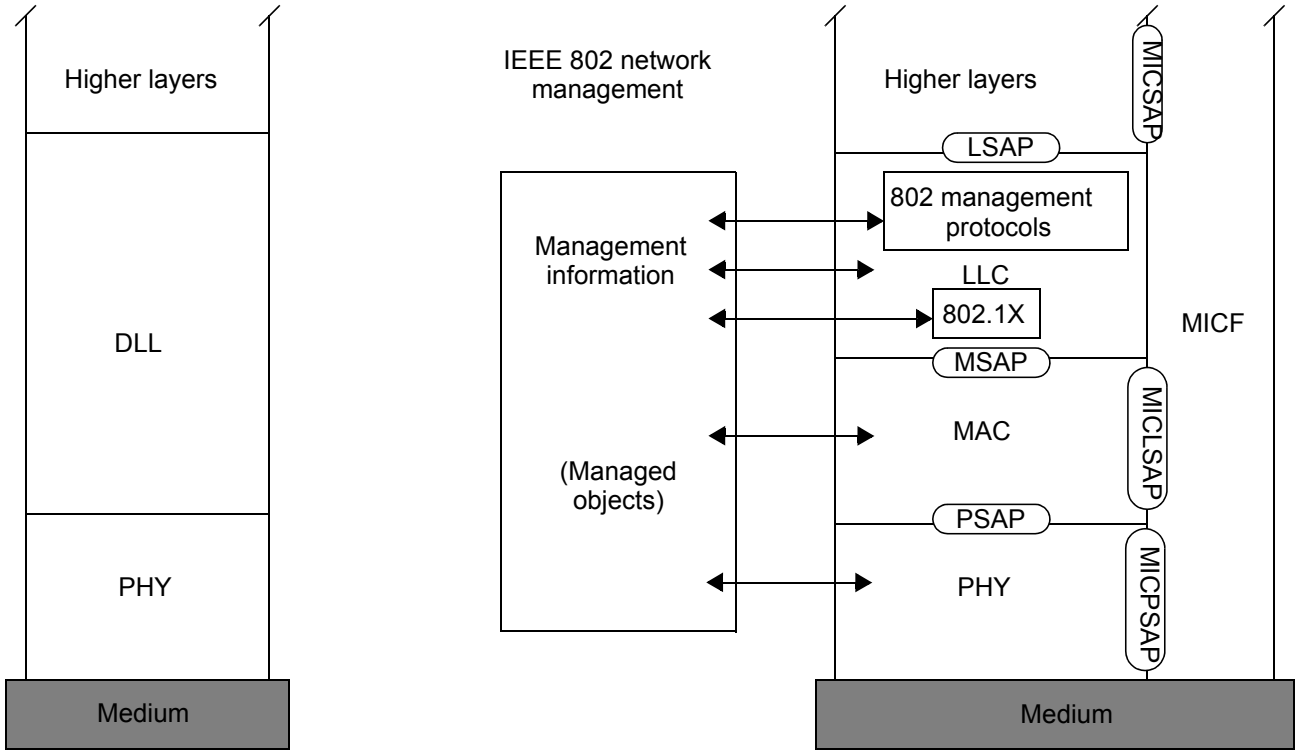


Figure 5—IEEE 802 RM with end-station management, security, and MIH

Followup Questions

- It is possible to answer the questions herein in a way that is consistent with existing IEEE 802 standards?
- Would a clarified IEEE 802 architecture and IEEE 802 Link Layer service provide an easier way for applications to make use of IEEE 802 networks, independent of the specific MAC in use?
- Would a clarified IEEE 802 architecture and IEEE 802 Link Layer service provide an easier route to improving IEEE 802 functionality and making it available to network clients?

Further information

- 802.1-21-0045-02-ICne (2021-09-23)
  - ELLA: What is the IEEE 802 Link Layer Service?
- 802.1-21-0060-04-ICne (2021-11-03)
  - ELLA: Proposed Aspects of IEEE Std 802 Revision
- 802.1-21-0073-00-ICne (2021-11-18)
  - ELLA: What's missing from the IEEE 802 Architecture?