ELLA: Proposed Aspects of IEEE Std 802 Revision

Roger Marks (EthAirNet Associates)

roger@ethair.net

+1 802 capable

3 November 2021

IEEE Std 802 - History

- IEEE Std 802-1990
- IEEE Std 802-2001
  - IEEE Std 802a-2003: Playpen EtherTypes
  - IEEE Std 802b-2004: OID hierarchy
- IEEE Std 802-2014
  - IEEE Std 802c-2017: Local MAC Address Usage
  - IEEE Std 802d-2017: URN allocation
  - P802f: YANG Data Model for EtherTypes
    - WG ballot opened, June 2021
- Procedurally, revision should follow before further amendments

IEEE Std 802 - Contents

1. Overview, Scope, Purpose
2. Normative references
3. Definitions
4. Family of IEEE 802 standards
5. Reference models (RMs)
6. General requirements for an IEEE 802 network
7. IEEE 802 network management
8. MAC addresses
9. Protocol identifiers
10. Allocation of OID values
11. Allocation of URN values

- Annex A (informative) Bibliography
- Annex B (informative) Reference models for IEEE 802 standards
- Annex C (informative) Examples of bit ordering for addresses
- Annex D (informative) List of IEEE 802 standards
- Annex E (informative) History
- [Annex F: (informative) YANG representation example (P802f)]

Proposal on how to revise

- 4. *Family of IEEE 802 standards* (pp 7-10)
  - 4.1 High-level view of IEEE 802 networks: revise; explain the commonality
  - 4.2 List of applications: delete
  - 4.3 Internationalization: minimize
  - 4.4 List of standards and projects: delete
- 5. *Reference models (RMs)* (pp 11-18): replace with LL Service & Architecture
  - 5.1 Describes Link Layer (LL) as one LLC over various MACs: replace
    - Describes media-independent handover: delete
  - 5.2 Reference model for end stations (LLC/MAC/PHY): replace
  - 5.3 Interconnection: replace
    - Barely mentions VLANs or priority; no reference to their role in architecture
- 6. *General requirements for an IEEE 802 network* (p 19)
  - Replace with (conformance-related?) description of LL service provided to client
- 7. *IEEE 802 network management* (pp 20-21): replace
- 8. *MAC Addresses* (pp 22-27, plus pp 12-19 of 802c)
  - Revise; merge with LL Service and Architecture
- 9. *Protocol identifiers* (pp 28-34)
  - Lists many 802.3 MAC frame formats, without assigning fields to layers
    - Who's responsible for adding and stripping those fields?
  - Replace; merge with LL Service and Architecture

IEEE Std 802 – Proposed Revision

1. *Overview, Scope, Purpose*
2. *Normative references*
3. *Definitions*
4. *Family of IEEE 802 standards*
5. *Network and application scenarios served by IEEE 802*
6. *IEEE 802 Link Layer Service and Architecture*
7. *General requirements for an IEEE 802 network*
8. *IEEE 802 network management*
9. *Allocation of OID values*
10. *Allocation of URN values*

- *Annex A (informative) Bibliography*
- *Annex B (informative) Reference models for IEEE 802 standards*
- *Annex C (informative) Examples of bit ordering for addresses*
- *Annex D (informative) ~~List of IEEE 802 standards~~*
- *Annex E (informative) History*
- *Annex F: (informative) YANG representation example*

Link Layer Service and Architecture

- Layers, sublayers, and SAPs
  - LLC & LSAP
  - MAC & MSAP
  - PHY
  - VLAN-aware and priority-aware end station
- Link Layer (LL) service specification
  - LL Service User and LL Service Provider
  - LL modes
    - Service Characteristics
      - Transparency
      - Data loss, data insertion, data duplication, misordering, priority, QoS, time-sensitivity, flow control,…
  - LL Service primitives
  - LL Service parameters
  - LL Service sequences of actions and events
- MAC service specification
  - Refer to IEEE Std 802.1AC
- LSAP addresses and protocol discrimination
- MAC address
- Bridging and relay architecture

Service Characteristic statements

- 802.2:
  - *connectionless-mode service… may be useful when higher layers provide any essential recovery and sequencing services so that these do not need replicating in the data link layer.*
- 802.1Q:
  - *The MAC Service (IEEE Std 802.1AC) permits a negligible rate of reordering of frames with a given priority for a given combination of destination address, source address, and flow hash, if present, transmitted on a given VLAN.*
    - *Note: This is an inaccurate characterization of 802.1AC.*
- *IEEE Std 802.1AC*
  - *In general, the MAC Service provider can perform any or all of the following actions: Discard objects, Change the order of the objects*
  - *The MAC Service exhibits a negligible rate of the following: Object duplication, Reordering of objects for a given priority*
  - *Awareness of the characteristics of the MAC Service provided, e.g., the rate at which objects can be discarded, duplicated, or misordered, is part of the MAC Service user's a priori knowledge of the environment.*
  - *Although the MAC Service maintains the integrity of individual MSDUs, it does not necessarily deliver them to the receiving MAC Service user in the order in which they are presented by the transmitting MAC Service user, for example in cases where they have different priorities.*

VLAN/priority-aware end station

- IEEE Std 802:
  - *IEEE Std 802.1Q specifies the method by which the MAC service is supported by virtual bridged LANs, the principles of operation of those networks, and the operation of VLAN-aware bridges,…*
  - Nothing about VLAN-aware end stations.

- 802.1Q (6.3 – Support of the MAC Service) :
  - *On the individual LANs of a Virtual Bridged Network, frames for different VLANs can be distinguished by the addition of a VLAN tag as the initial octets of a frame's MSDU.*
  - *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.*

EISS Multiplex Entity

Per 802.1Q:
On the individual LANs of a Virtual Bridged Network, frames for different VLANs can be distinguished by the addition of a VLAN tag as the initial octets of a frame's MSDU.

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.

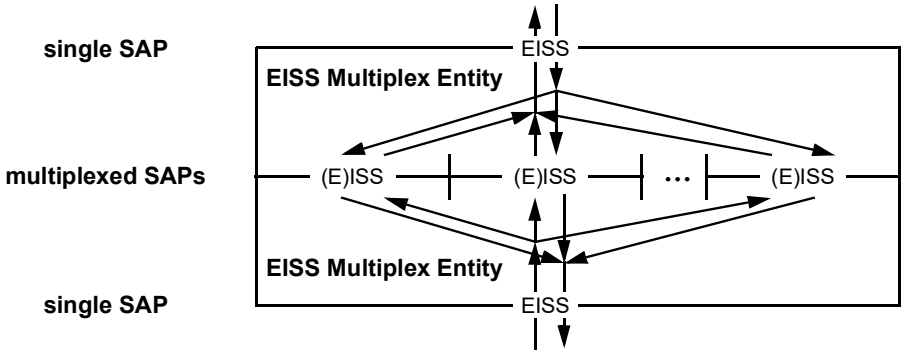
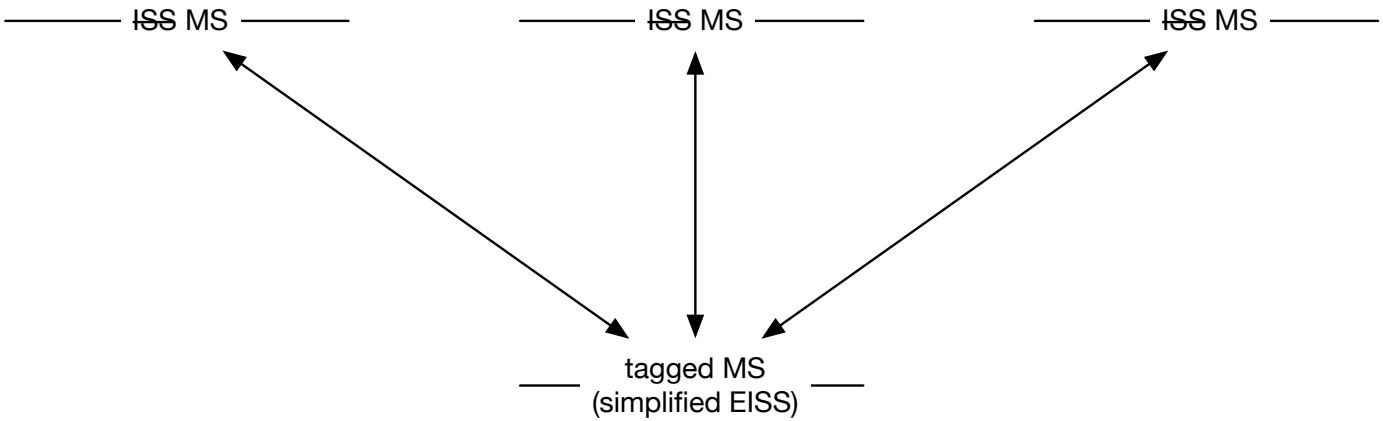
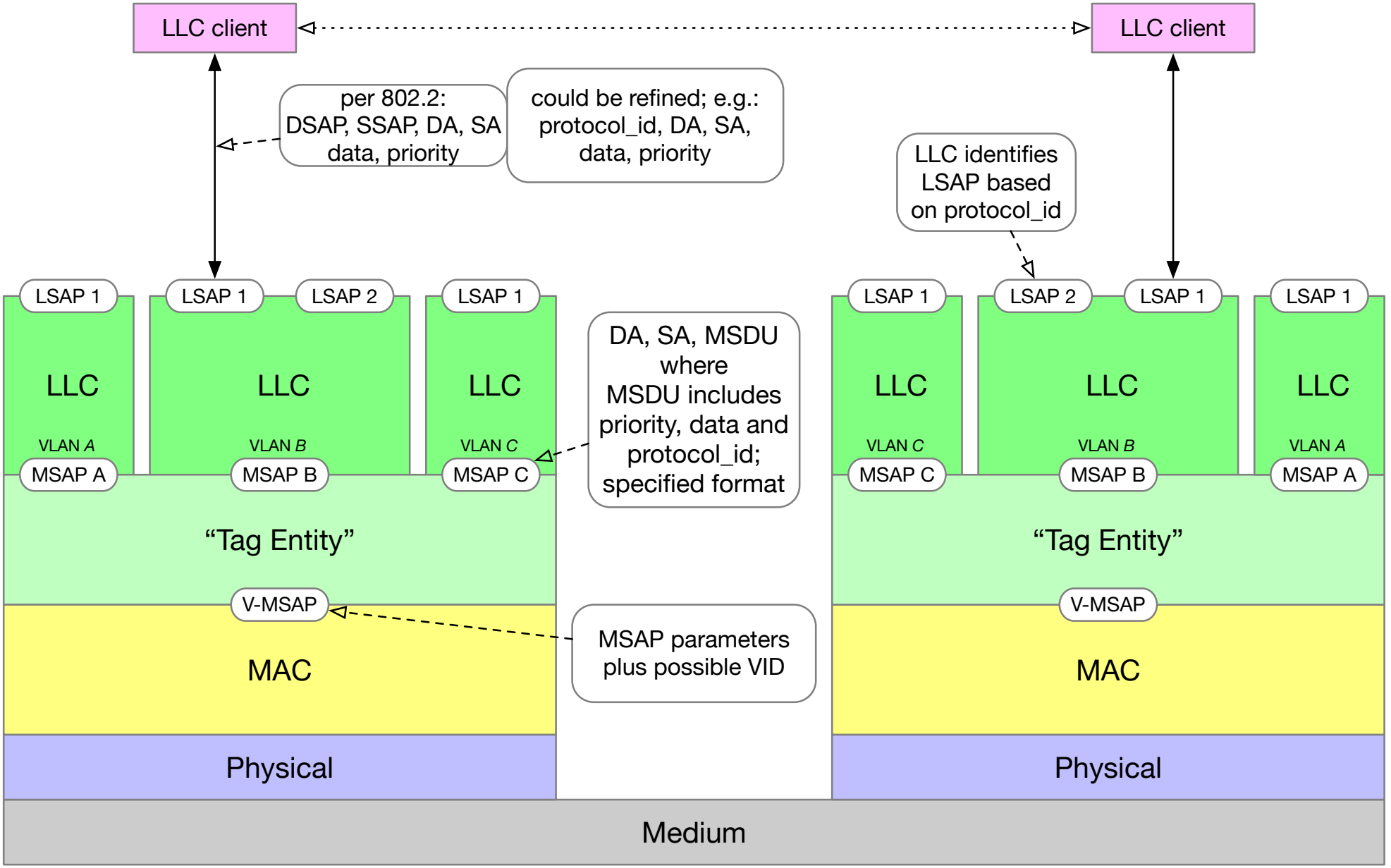


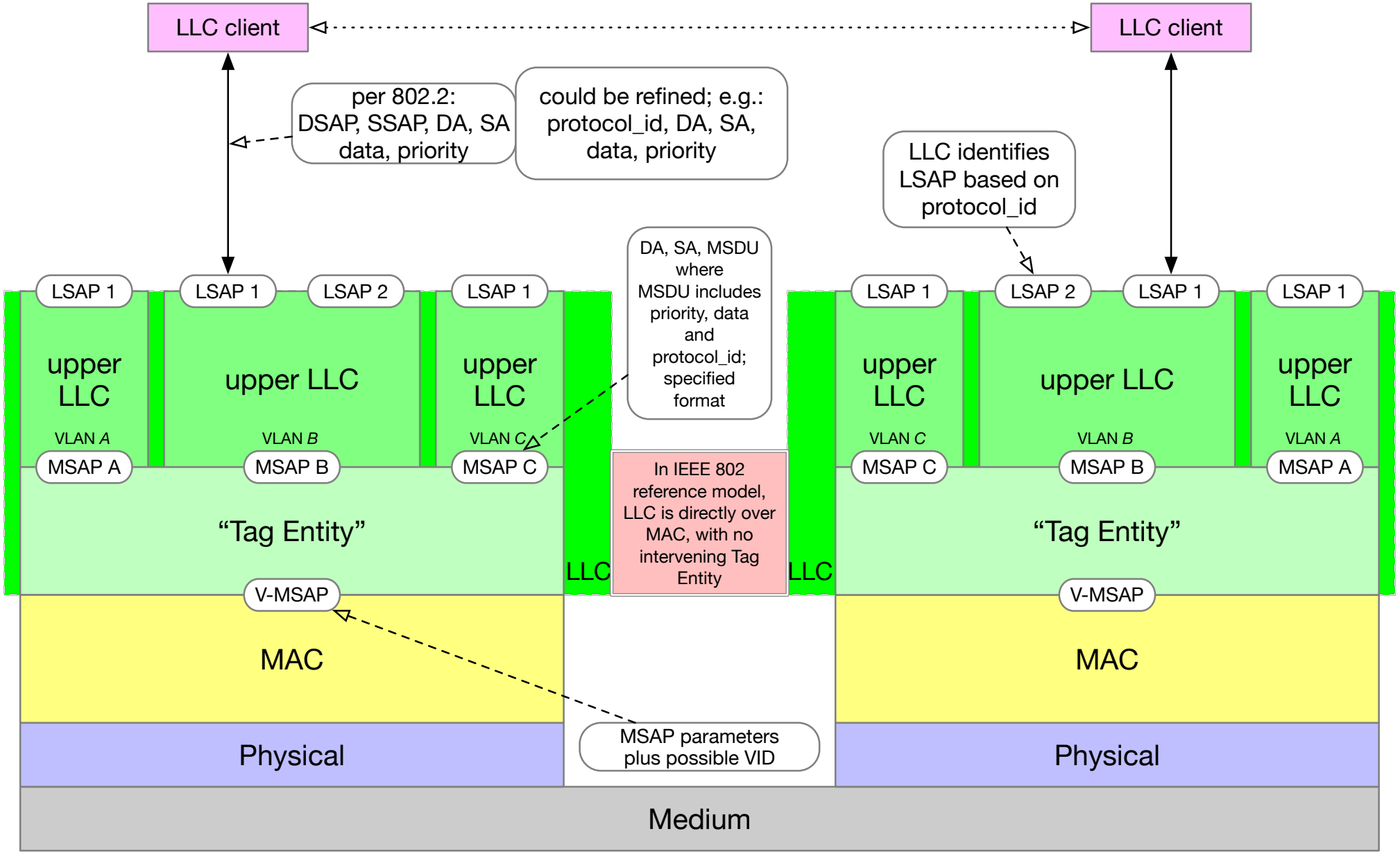
Figure 6-6—Two back-to-back EISS Multiplex Entities



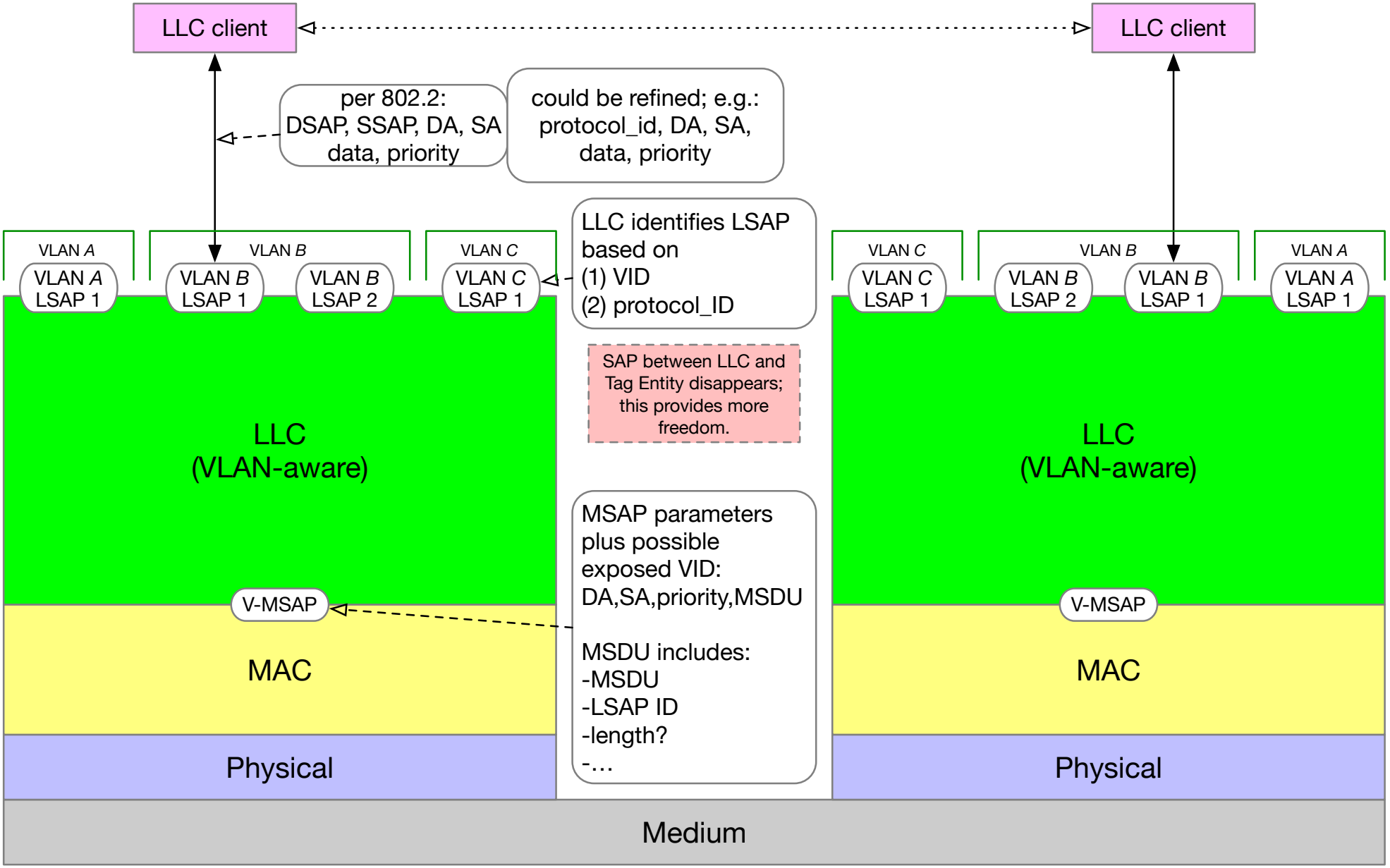
schematic End Station Architecture…



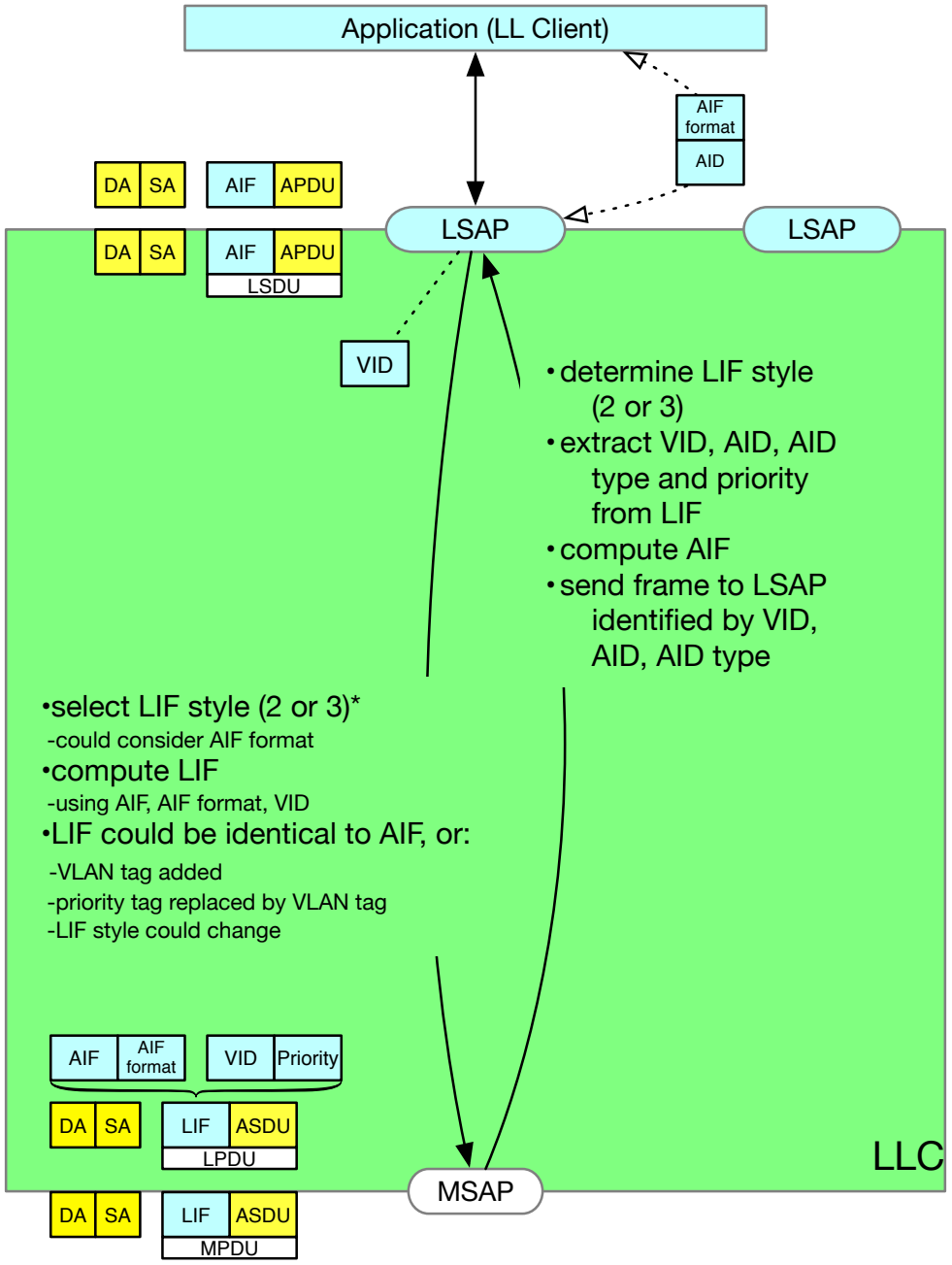
schematic End Station Architecture…



schematic End Station Architecture



LLC operation



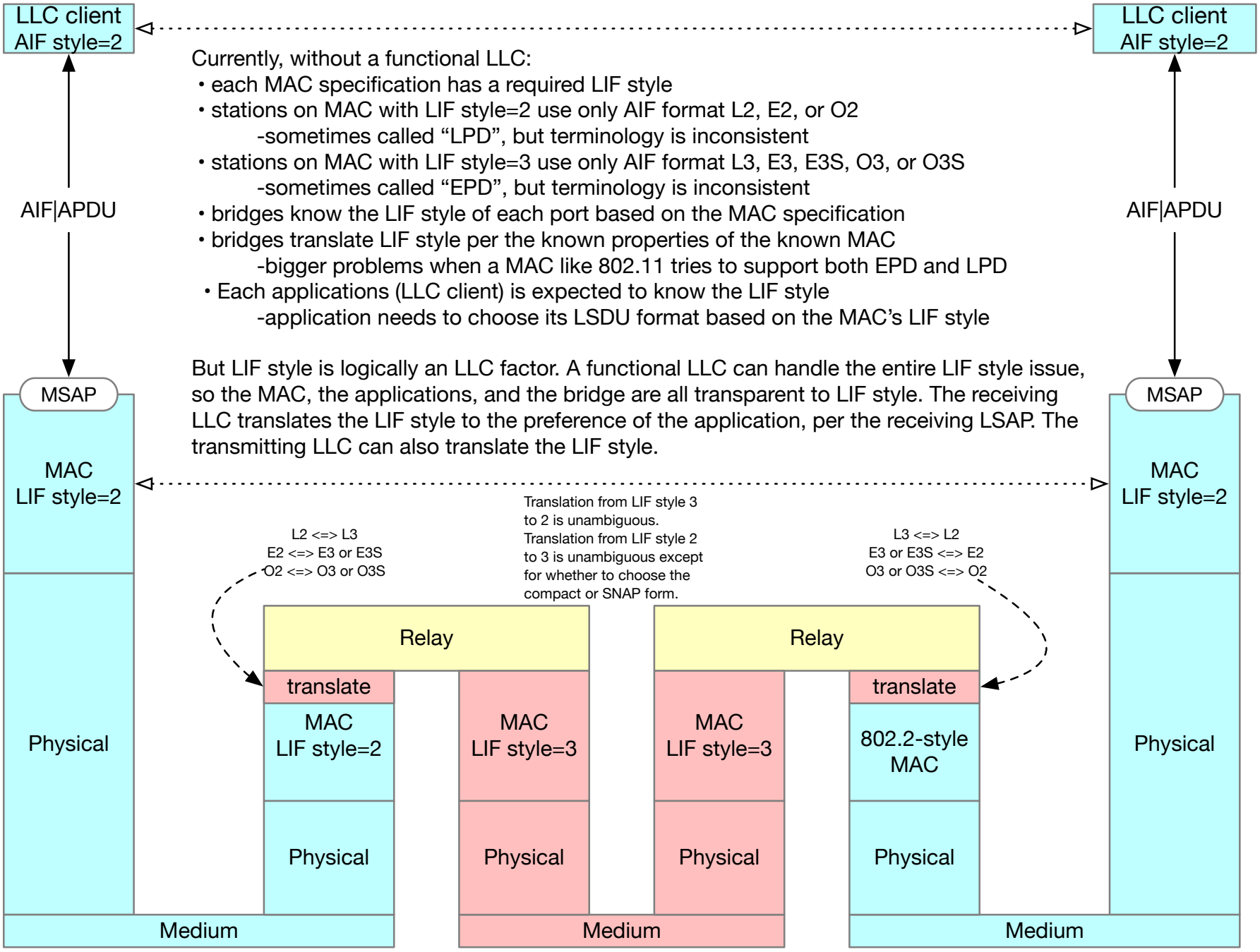
APDU application PDU
AID application identifier
AIF application identifier field
LIF LSAP identifier field
VID VLAN identifier

AID type	AID
L	DSAP SSAP
E	EtherType
O	O Identifier

AID type	LIF style	AIF/LIF format	legacy AIF/LIF alternatives: untagged
L	2	L2	DSAP/SSAP 0x03
E	2	E2	0xAAAA03 0x000000 EtherType
O	2	O2	0xAAAA03 O Identifier
~~O~~	~~2~~	~~*~~	~~0xAAAA03 0x000000 0x88B7 O Identifier~~
L	3	L3	Length DSAP/SSAP 0x03
E	3	E3	EtherType
E	3	E3S	Length 0xAAAA03 0x000000 EtherType
O	3	O3	0x88B7 O Identifier
O	3	O3S	Length 0xAAAA03 O Identifier

*Specified in 802.1AC 12.11, possibly in error. Ignored herein.

LIF style without a functional LLC

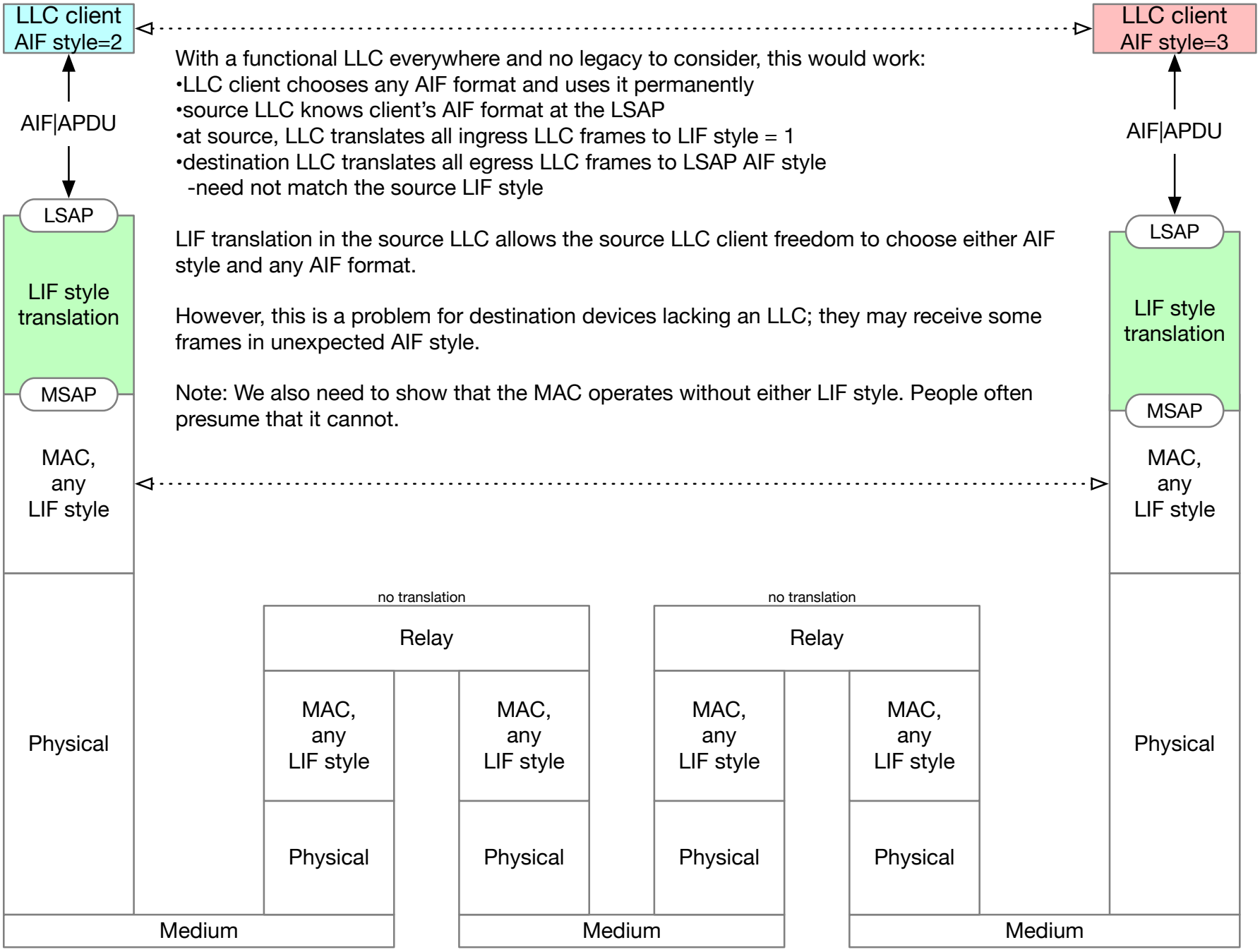


Currently, without a functional LLC:

- each MAC specification has a required LIF style
- stations on MAC with LIF style=2 use only AIF format L2, E2, or O2
  - -sometimes called “LPD”, but terminology is inconsistent
- stations on MAC with LIF style=3 use only AIF format L3, E3, E3S, O3, or O3S
  - -sometimes called “EPD”, but terminology is inconsistent
- bridges know the LIF style of each port based on the MAC specification
- bridges translate LIF style per the known properties of the known MAC
  - -bigger problems when a MAC like 802.11 tries to support both EPD and LPD
- Each applications (LLC client) is expected to know the LIF style
  - -application needs to choose its LSDU format based on the MAC’s LIF style

But LIF style is logically an LLC factor. A functional LLC can handle the entire LIF style issue, so the MAC, the applications, and the bridge are all transparent to LIF style. The receiving LLC translates the LIF style to the preference of the application, per the receiving LSAP. The transmitting LLC can also translate the LIF style.

Functional LLC, Greenfield



With a functional LLC everywhere and no legacy to consider, this would work:

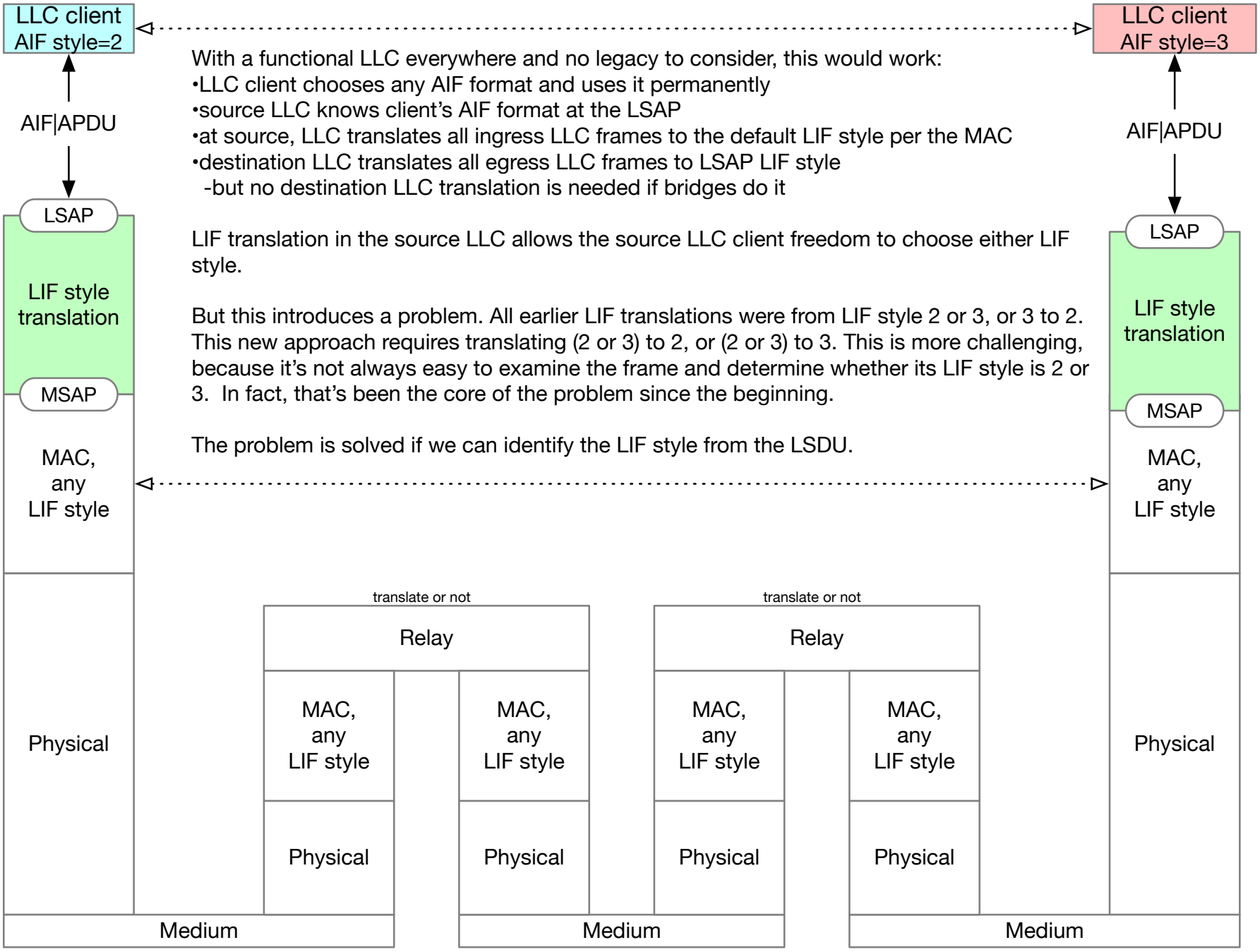
- LLC client chooses any AIF format and uses it permanently
- source LLC knows client's AIF format at the LSAP
- at source, LLC translates all ingress LLC frames to LIF style = 1
- destination LLC translates all egress LLC frames to LSAP AIF style
  - need not match the source LIF style

LIF translation in the source LLC allows the source LLC client freedom to choose either AIF style and any AIF format.

However, this is a problem for destination devices lacking an LLC; they may receive some frames in unexpected AIF style.

Note: We also need to show that the MAC operates without either LIF style. People often presume that it cannot.

Functional LLC, Backward Compatible



With a functional LLC everywhere and no legacy to consider, this would work:

- LLC client chooses any AIF format and uses it permanently
- source LLC knows client's AIF format at the LSAP
- at source, LLC translates all ingress LLC frames to the default LIF style per the MAC
- destination LLC translates all egress LLC frames to LSAP LIF style
  - but no destination LLC translation is needed if bridges do it

LIF translation in the source LLC allows the source LLC client freedom to choose either LIF style.

But this introduces a problem. All earlier LIF translations were from LIF style 2 or 3, or 3 to 2. This new approach requires translating (2 or 3) to 2, or (2 or 3) to 3. This is more challenging, because it's not always easy to examine the frame and determine whether its LIF style is 2 or 3. In fact, that's been the core of the problem since the beginning.

The problem is solved if we can identify the LIF style from the LSDU.

LIF style: untagged and tagged

In order to determine the LIF format, we need two assumptions:

(1) 0xAAAA is never used as an AID. It is not a current EtherType assignment and should be assigned for only SNAP use. This was also proposed (for a different reason) in:
https://www.ieee802.org/1/files/public/docs2009/h-rev-seaman-receive-only-protocol-0509-01.pdf
(2) AIF format L2 is never used if DSAP/SSAP is a length [<0x05DD (1501)] or could be a possible EtherType

With these assumptions:
if frame begins with 0xAAAA03 then LIF style 2 else
if first 2 bytes of frame are >=0x0600 (1536) then LIF style = 3 else
if first 2 bytes of frame are <=0x05DC (1500) then LIF style = 2

How to ensure Assumption (2)?
-solution 1: tag all DSAP/LSAP frames. The LLC converts any L2 format to L2T.
-solution 2: solution 1, minus some exceptions (e.g., 0x4242, 0xFEFE are presumed L2)
-those exceptions could still be used as EtherType in any LIF format except E3
-it's not clear we need an exception for 0x4242; it is currently used only in L3 format

Instead of EPD or LPD, we have universal protocol discrimination (UPD). Applications can choose their favorite AID type and AIF format, regardless of the MAC. Frames will be delivered in the requested AIF format; the same app can choose a different format per LSAP. Bridges need not (but may) translate AIF format.

See also R. Marks, "LLC Theory and Protocol Discrimination <https://mentor.ieee.org/802.11/dcn/21/11-21-0092-01-0arc.pdf>

AIF/LIF: untagged

LIF style	AIF/LIF format							
2	L2	DSAP/ SSAP	0x03					
2	E2	0xAAAA03	0x000000	EtherType				
2	O2	0xAAAA03	O Identifier					
3	L3	Length	DSAP/ SSAP	0x03				
3	E3	EtherType						
3	E3S	Length	0xAAAA03	0x000000	EtherType			
3	O3	0x88B7	O Identifier					
3	O3S	Length	0xAAAA03	O Identifier				

AIF/LIF: tagged

LIF style	AIF/LIF format							
2	L2T	0xAAAA03	0x000000	0x8100	TCI	Length	DSAP/ SSAP	0x03
2	E2T	0xAAAA03	0x000000	0x8100	TCI	EtherType		
2	O2T	0xAAAA03	0x000000	0x8100	TCI	Length	0xAAAA03	O Identifier
3	L3T	0x8100	TCI	len	DSAP/ SSAP	0x03		
3	E3T	0x8100	TCI	EtherType				
3	E3ST	0x8100	TCI	len	0xAAAA03	0x000000	EtherType	
3	O3T	0x8100	TCI	0x88B7	O Identifier			
3	O3ST	0x8100	TCI	len	0xAAAA03	O Identifier		

Per 802.1Q G.3 and 802.1AC 12.3

Note regarding L3T, E3ST, and O3ST: Per IEEE Std 802.3, the "Length" field serves for depadding padded frames. However, when a tag is used, the Length/Type field is a Type, and no length information in the frame is used for depadding. Therefore, in a tagged frame, the value of the field is ignored in the Ethernet MAC. Here it is called "len," not "Length," to indicate that fact.

Some applications may believe that LLC protects against padding if LIF style 2 is used, but Ethernet cannot if the frame is tagged. So, should depadding be explicitly NOT promised? Or should that function be incorporated in the LLC (as 802.1Q specifies depadding in the bridge)? E3ST and O3ST are required for cases in which depadding functionality is needed. No alternative to L3T is available, even if depadding is not required.

Possible LLC Role: Invalid Frame Check

LIF style	AIF/LIF format	AIF/LIF: untagged
2	L2	DSAP/SSAP | 0x03
2	E2	0xAAAA03 | 0x000000 | EtherType
2	O2	0xAAAA03 | O Identifier
3	L3	Length | DSAP/SSAP | 0x03
3	E3	EtherType
3	E3S	Length | 0xAAAA03 | 0x000000 | EtherType
3	O3	0x88B7 | O Identifier
3	O3S	Length | 0xAAAA03 | O Identifier

The LLC could be made responsible to filter invalid frames sent by the LLC Client.

Consider E3S, where the LLC Client sets EtherType=0x8100 and the data begins TCI|EtherType

E3S

Length	0xAAAA03	0x000000	0x8100	Client TCI	EtherType	data

When tagged and translated to LIF style 2, this becomes:

LiF style 2 tag

E2T

0xAAAA03	0x000000	0x8100	Bridge TCI	0x8100	TCI	EtherType	data

This double-tagged format could lead to unexpected results.

Possible solution: specify valid frame formats and have the LLC drop invalid frame. VLAN tagging is the responsibility of the LLC and could be prohibited by the client.

LIF style	AIF/LIF format	AIF/LIF: tagged
2	L2T	0xAAAA03 | 0x000000 | 0x8100 | TCI | Length | DSAP/SSAP | 0x03
2	E2T	0xAAAA03 | 0x000000 | 0x8100 | TCI | EtherType
2	O2T	0xAAAA03 | 0x000000 | 0x8100 | TCI | Length | 0xAAAA03 | O Identifier
3	L3T	0x8100 | TCI | len | DSAP/SSAP | 0x03
3	E3T	0x8100 | TCI | EtherType
3	E3ST	0x8100 | TCI | len | 0xAAAA03 | 0x000000 | EtherType
3	O3T	0x8100 | TCI | 0x88B7 | O Identifier
3	O3ST	0x8100 | TCI | len | 0xAAAA03 | O Identifier

Next topic: MSAP Multiplicity

Per IEEE Std 802-2014:

The MAC sublayer provides one or more MAC service access points (MSAPs) as interfaces to the LLC sublayer in an end station.

What is this MSAP multiplicity per end station? [Note: No indication that this is VLAN multiplexing.]

How is the MSAP characterized? Is an MSAP identified by one and only one MAC address?

- –MAC Client can send frames from multiple SAs; the SA of each frame is a MAC service parameter.
- –MAC Client can receive frames to multiple DAs.
  - –the DA and SA are exposed to the MAC Client, so it can discriminate based on them

-LSAP "source_address" and "destination_address" parameters provide … logical concatenation of the MAC address field (SA and/or DA) and… (SSAP/DSAP)

- • So: the MSAP is characterized by the list of all MAC addresses that it is configured to receive.
  - –this is not a multiplicity of MSAPs
- –We do not need multiple MSAPs per end station.
- –A hardware device could have multiple "end stations," each with an MSAP.