Protocol identification in 802 LANs

Mick Seaman

This note follows up on discussions in the 802.1 Maintenance Task Group. Roger Marks has pointed out inconsistencies between the description of Ethernet Protocol Discrimination (EPD) in various current IEEE 802 standards. Norm Finn made a crucial point in the last Maintenance TG teleconference—we should distinguish between a protocol identifier (potentially identifying a given protocol used between peer end stations attached to LANs of different types, but necessarily known by those stations to be the same identifier) and the encoding of that identifier (regretably, but now irremediably different on some media). A bridge that connects media using different protocol identifier encodings has the job of translating between the two. There are many devils in the details of any restructuring of a description in a standard, so I thought it best to make a thorough attempt at the task before advocating such a group activity. In recent years I have only consulted IEEE Std 802-2014 for specific point detail, and was surprised to find that Clause 9 (Protocol identifiers) makes no mention of LLC Addresses at all and LLC Protocol Discrimination (LPD) is confined to carrying EtherTypes. That's hardly useful when we still have significant protocols (RSTP/MSTP and IS-IS - including SPB when used in the context of IEEE Std 802.1Q) that use assigned LLC Addresses for protocol identification. It means that IEE Std 802-2014 is not useful, and indeed positively misleading when used as a reference point for EPD/LPD discussions. IEEE Std 802c-2017 goes further by reintroducing the SNAP SAP encapsulation of EtherTypes in 802.3 frames - an alternative that we spent considerable effort advising against in IEEE Std 802.1H-1995.

Apart from apparently writing off 802.1Q, there is much in IEEE Std 802-2014[1] that I believe requires significant rework. This note includes the beginning of a proposed complete rewrite of Clause 9. At present it focuses on clearly identifying the Protocol Identifiers that are in current use and their encoding (following up on Norm's suggestion) on LPD and EPD media. I have included detailed references that identify the necessary historical background. I plan to add text on protocol versioning and interoperability (which have specified elsewhere, in much more useful form) and subtyping, but I believe there is enough for discussion on the EPD/LPD documentation issue.

The remainder of the introduction to this note discusses various topics that I have found of interest while working on the subject. At present these should be regraded simply as reminders, place holders for future work. They may not be focused.

ISS Length/Type encoded?

Clause 12 (Protocol discrimination and media) of 802.1AC-2016 (pg 30, ~line 12) states that the ISS is Length/Type encoded. I don't believe it, and it is not a useful statement. In fact it makes what we have to describe somewhat harder. Clause 11 (Service primitives) of 802.1AC-2016 specifies the mac_service_data_unit (MSDU) as a parameter of the ISS and places no restriction on what is in the data octets of that parameter.

What is true is that a bridge that relays frames should know the length of each frame that it is relaying. It is true that (in the absence of further protocol specific information, not usually available to a bridge's MAC Relay Entity) that 802.3 frames with an EtherType Protocol Identifier have an MSDU of 48 octets or greater when received. Some future medium-access method might well wish to use an EtherType identified protocol with shorter frames, particularly if applicable to very low bandwidth scenarios. However if there is any possibility of such frames being bridged to and from 802.3 media it would be wise to either (a) use an LLC Address as a protocol identifier, and accept the overall length limitation for the relevant frames or (b) include a length identifer in the protocol. Note that (for 802.3 at present, and probably for 802.3 for all time) it would be sufficient to carry an indication of the MSDU being less than 48 octets, with the length in that case. The length indication field need, therefore, only take 6 bits so can be encode in a single octet, can even coexist with a small amount of subtyping

[1] I do need to revise my proposed text to take into account 802c-2017.

information in that octet, or can coexit with subtyping and versioning information in a pair of octets. 802.1AE encodes such a ‘short length’ field in the second octet following the MACsec EtherType so it can locate the ICV in framea that might be:

a) MACsec protected and then tagged with some additional tag before being padded by 802.3; or

b) MACsec protected prior to transmission on a media without the 802.3 padding requirement, but subsequently bridged (without terminating the MACsec protection) to 802.3 for subsequent reception and validation by an 802.3 station.

1. Protocol Identification

1.1 Introduction

Many different protocols can coexist in an IEEE 802 network, and a given station can participate in the operation of more than one protocol. Different protocols can address quite different requirements (e.g. for communication over the network as a whole, control over the network's configuration, authentication of network users, and allocation of resources) and a complete list is beyond the scope of this specification. The wide range of these requirements and their evolution to meet the needs of new and improved network applications and control paradigms means that the protocols used by given individual stations have been developed, and will continue to be developed, by different groups of experts working in parallel or at different times with limited opportunity for inter-group communication. Differing protocol requirements often call for differences in the encoding of the different data sets communicated. However, to allow a LAN-attached station to direct any given received frame to the higher-layer protocol entity responsible for processing that frame (or to identify the frame as associated with a protocol not supported by that station), the initial octets of each frame encode a Protocol Identifier.

Correct protocol identification for received frames, and the development of implementation-dependent frame reception procedures that allow the deployment of new protocols to existing stations without disrupting their handling of existing protocols, are supported by specification of the following:

a) The use of a small number of Protocol Identifier types.
b) Encoding rules that allow the identification of each Protocol Identifier type and recovery of its value from each received frame
c) Protocol Identifier assignment procedures.

This specification (Clause 1) is concerned with those protocol identifiers used by peer station functions in conjunction with the MAC Service as specified by IEEE Std 802.1AC. Those protocol identifiers are commonly assigned for use by protocol entities independently of the particular medium-access method (MAC type) used by a station to transmit frames to its peer(s). A protocol frame transmitted by an end station function, and conveying such a protocol identifier, can be bridged from one LAN to another along the path to its destination, and those LANs can use different media access methods. The protocol identifiers themselves thus need to be defined and assigned independently of the detailed operation of particular media access method.

Individual medium-access methods may, as part of the operation of an individual LAN, need to identify other frame types, sub-types, and identifiers. The definition of these types and identifiers, and the assignment and registration of their values, is part of the specification of individual medium-access method. They are not discussed further in this clause, apart from noting the need for each such medium-access method to be capable of clearly distinguishing between such local frames and those used to support the MAC Service.

1.2 Overview

This clause (Clause 1) describes and specifies the following:

a) Four types of Protocol Identifier that have been standardized for use in IEEE 802 LANs:
   1) EtherTypes (1.3)
   2) OUI Extended EtherType Identifiers (1.4)

      NOTE 1—OUI Extended EtherType Identifiers were first standardized in IEEE Std 802-2014.

   3) LLC addresses (1.5)
   4) SNAP Identifiers (1.6)

      NOTE 2—IEEE Std 802-2014 introduced the term SNAP identifier, derived from prior descriptions of the encoding of those identifiers.

b) Two media-access method dependent Protocol Identifier encoding formats, each capable of encoding any one of the four Protocol Identifier types in the initial octets of any given user data frame transmitted on an IEEE 802 LAN:
   1) Ethernet Protocol Discrimination (EPD, 1.7) as used by Ethernet (IEEE Std 802.3).

      NOTE 2—EPD can also be identified as Length/Type encoding.

   2) LLC Protocol Discrimination (LPD, 1.8) as used, e.g., by Token Ring (IEEE Std 802.5).

      NOTE 3—IEEE Std 802.5 [An] has been withdrawn, but provides useful background for the development and maintenance of this document as there is no prospect of changes to that standard. The provisions of this document do not depend on IEEE Std 802.5.

c) Requirements for conversion between EPD and LPD encoding in end stations and bridges, and for consistent use of EPD encoding in the protocol data of that frames that can be bridged ().

   NOTE 4—A bridge that relays a frame between LANs that use the same protocol identifier encoding is not required to modify a relayed frame. The transformation between EPD and LPD encodings is specified by reference to Clause 12 of IEEE Std 802.1AC-2016. IEEE Std 802.1Q Bridges and Bridged Networks mandates use of the MAC Internal Sublayer Service (ISS) as specified by IEEE Std 802.1AC.

The use of both EtherTypes and LLC Addresses and different encodings (EPD, LPD) reflects, in part, their use by different organizations to meet different requirements prior to the standardization of IEEE 802 LANs. Two additional types [1.2 a.2), a.4)] allow protocol specification and protocol identifier assignment by organizations that have been assigned an OUI or OUI-36, without the need for an additional registration authority assignment. Specific EtherType values have also been reserved for experimental use (1.3.1).

Future protocols that are intended for widespread use should be identified by an EtherType, subject to meeting the criteria for EtherType assignment (). These criteria include the availability of the protocol specification, and protocol procedures that accommodate enhancements to the protocol without requiring the assignment of additional EtherTypes (see n.n for guidelines).

Future IEEE 802 media-access method standards shall support EPD encoding of EtherTypes, LLC Addresses, and OUI Extended EtherType Identifiers. Amendments to existing standards may continue to support LPD encoding. Amendments to an existing standard that currently support only LPD may add EPD support as an additional capability, selectable on a per LAN, per station, per peer, or per frame basis as determined by the interoperability requirements for that standard. Annex F discusses the standardization of such capabilities.

1.3 EtherTypes

An EtherType is a sequence of 2 octets, interpreted as a 16-bit numeric value with the first octet containing the most significant 8 bits and the second octet containing the least significant 8 bits. Values in the range 0-1535 range are not used, permitting EPD encoding of LLC Addresses (Figure 1.7.3).

Two human-readable representations of EtherType values are in widespread use:

a) Each octet is represented as a hexadecimal number, with the more significant digit to the left. The first octet is shown to the left, followed by a dash or hyphen, with the second octet to the right, e.g., 86-DD.

b) The 16-bit numeric value of the EtherType is shown as a hexadecimal number, with the most significant digit to the left, preceded by ‘0x’ to indicate the use of a hexadecimal representation thus guarding against a possible misinterpretation if all digits are in the range 0-9, e.g. 0x86DD.

NOTE—The EtherType 0x86DD (alternative representation 86-DD) has been assigned to identify IPv6.

1.3.1 Local Experimental EtherTypes

The EtherType identifier space is limited. To avoid its unnecessary depletion, two EtherType values have been assigned to allow experimental, and prototype protocol development within a network controlled by a single administration, with the permission of that administration. See Table 1-1.

Table 1-1—Local Experimental EtherTypes

Name	Value
Local Experimental EtherType 1	88-B5
Local Experimental EtherType 2	88-B6

The local network administrator is free to use these EtherTypes and to assign subtypes for protocol development. However the following considerations apply to their use:

a) To allow the widest possible experimental use of these EtherTypes they have been assigned without constraints on the data carried in the remaining octets of protocol frames. There is no guarantee that independently designed protocols using the same Local Experimental EtherType will coexist. Local Experimental EtherTypes shall not be used in protocols or products released for use in any environment outside the developing organization’s complete administrative control, as freeware, shareware, for testing, or as part of a product. A specifically assigned EtherType or other Protocol Identifier shall be substituted prior to any such deployment.

b) If eventual protocol deployment using an assigned EtherType is a possibility, the protocol design needs to include sufficient subtyping or versioning capability to accommodate future development without the assignment of a further EtherType ().

1.4 OUI Extended EtherType Identifiers

An organization that has been assigned an OUI or an OUI-36 may assign universally unique OUI Extended EtherType Identifiers to identify its own protocols. Each OUI Extended EtherType Identifier comprises a sequence of 7 octets. The first two octets encode the OUI Extended EtherType (Table 1-2). The next five octets are encoded as follows:

a) If an OUI is used as the basis of the OUI Extended EtherType Identifier, the next three octets encode that OUI, in the same order as specified for use in an EUI-48 assignment. The values of the final two octets (the sixth and seventh octets of the identifier) are assigned by the OUI assignee.

b) If an OUI-36 is used as the basis of the OUI Extended EtherType Identifier, the next three octets and the four least-significant bits of the following octet (the sixth octet) encode that OUI-36, in the same order as specified for use in an EUI-48 assignment. The values of the four most-significant bits of the sixth octet and the final (seventh) octet are assigned by the OUI-36 assignee.

The human-readable representation of an OUI Extended EtherType Identifier comprises the hexadecimal representation of each octet, with the octets shown in order separated by a dash or hyphen. Each octet is represented as a hexadecimal number, with the more significant digit to the left. The assignee of the OUI AC-DE-48 might, for example, assign the OUI Extended EtherType Identifier 88-B7-AC-DE-48-10-80 to a protocol developed by that assignee. Alternatively, if the original assignment had been of an OUI-36 AC-DE-48-09 that could be used to assign the OUI Extended EtherType Identifier 88-B7-AC-DE-48-19-80.

Table 1-2—OUI Extended EtherType

Name	Value
OUI Extended EtherType	88-B7

If an OUI is used to make both SNAP Identifier and OUI Extended EtherType Identifiers, the values used in the final octets of those identifiers should be distinct. While this document specifies different encodings for those identifiers, there remains the possibility of confusion and non-standard translation between these identifiers.

The next to least-significant bit of the first octet of universally unique OUI Extended EtherType Identifiers (the position of the universal/local (U/L) bit when an OUI is used to assign universal MAC address, see 8.2.2 of IEEE Std 802-2014) is always 0, indicating a universal assignment. Locally-significant SNAP Identifiers can be assigned, with this bit set to 1, but have no relationship to an OUI or other IEEE Registration Authority assignment.

1.5 LLC Addresses

LLC Addresses are specified by ISO/IEC 8802-2:1998 Logical Link Control. An LLC Address identifies an Link Service Access Point (LSAP) within a station. In principle, the association between a protocol entity that uses LLC procedures to support a higher-layer protocol and an LLC Address value could be dynamic. However no protocols or procedures for making such a dynamic association have been standardized for use with IEEE 802 LANs, and LLC Addresses have been assigned to identify specific protocols.

LLC Address values are defined by 3.3.1 of ISO/IEC 8802-2:1998 by reference to their encoding in the Destination Service Access Point (DSAP) and Source Service Access Point (SSAP) address fields of LLC PDUs. The least significant bit of the DSAP field is used to distinguish Individual and Group DSAPs, and the least significant bit of the SSAP field is a Command/Response (C/R) bit used by the LLC Procedures specified in Clauses 6, 7, and 8. Individual (as opposed to Group) LLC Address values are always recorded as 8-bit numbers with the least significant bit set to zero.

To avoid any ambiguity as to bit-ordering or number base , the human readable representation of an assigned LLC Address should be its 8-bit numeric value shown as a hexadecimal number (with the most significant digit to the left), e.g., 0x42 for the Bridge spanning tree protocol specified by IEEE Std 802.1Q, and 0xFE assigned for use in conjunction with ISO/IEC TR 9577 and used in the identification of IS-IS (Intermediate System to Intermediate System routing protocol) PDUs. If an LLC Address is represented by a binary string, all eight bits should be shown and the bit significance convention stated.

NOTE 1—The LLC address value 0x42 is recorded as assigned to ISO/IEC 10038, a precursor of IEE Std 802.1Q.

In addition to the Individual LLC Address values that have been assigned or reserved for future assignment both Individual and Group LLC values are available for unreserved use. The meaning and use of reserved Group LLC Address values in the DSAP field is for future study.

NOTE 2—As of 30th March 2020, the public listing of LLC address values maintained by the IEEE Registration Authority shows the range of unreserved Individual LLC Address values as 0000 0001 to 0011 1111 (least-significant bit to the left). This range is also described as 80 to FC, but it is hopefully clear from examining the table of assigned and reserved values that the unreserved range comprises values 0000 0001, 0000 0010, 0000 0011, … i.e. 0x80, 0x40, 0xC0, … and not all successive values in the numeric range 0x80 to 0xFC. Values 0xXY, where X is any hexadecimal digit and Y is the hexadecimal digit 0, 4, 8, or C are included , with the exception of 0x00.

It is expected, but not required, that the use of assigned LLC Address values will result in the encoding of the same value in both DSAP and SSAP address fields, representing communication between peer protocol entities.

NOTE 3—The technical background of LLC included HDLC and SDLC, where the Link-layer addresses could be used to represent one station or another (‘this end of the wire’, ‘the other’) or to address a particular station or a group of stations in a multi-drop communication prior to the use of MAC Addresses to identify each stations.

1.6 SNAP Identifiers

An organization that has been assigned an OUI may assign universally unique SNAP Identifiers to identify its own protocols. Each SNAP Identifier comprises a sequence of 5 octets. The first three octets are those of the OUI, in same order as specified for use in an EUI-48 assignment. The values of following two octets are assigned by the OUI assignee.

The human-readable representation of a SNAP Identifier comprises the hexadecimal representation of each octet, with the octets shown in order separated by a dash or hyphen. Each octet is represented as a hexadecimal number, with the more significant digit to the left. The assignee of the OUI AC-DE-48 might, for example, assign the SNAP Identifier AC-DE-48-00-80 to a protocol developed by that assignee.

When assignment of an EtherType is not appropriate, new protocols should be identified by an OUI Extended EtherType Identifier (1.4) rather than a SNAP Identifier.

The next to least-significant bit of the first octet of universally unique SNAP Identifiers (the position of the universal/local (U/L) bit when an OUI is used to assign universal MAC address, see 8.2.2 of IEEE Std 802-2014) is always 0, indicating a universal assignment. Locally-significant SNAP Identifiers can be assigned, with this bit set to 1, but have no relationship to an OUI or any other IEEE Registration Authority assignment.

1.7 Ethernet Protocol Discrimination

This clause (1.7) specifies the encoding, in the initial octets of IEEE Std 802.3 frames and other media-access methods using Ethernet Protocol Discrimination (EPD), of the following Protocol Identifiers:

a) EtherTypes (1.7.1)
b) OUI Extended EtherType Identifiers (1.7.2)
c) LLC Addresses (1.7.3)
d) SNAP Identifiers (1.7.4)

IEEE Std 802.1AC Media Access Control (MAC) Service Definition defines a MAC Service and an Internal Sublayer Service (ISS) common to all IEEE 802 media-access methods.The media-access method independent ISS is used by MAC Bridges (IEEE Std 802.1Q) to relay frames between LANs and is specified in terms of two primitives, one for frame reception and another for transmission, and their parameters. The MAC Service uses the subset of those parameters relevant to end station use of the MAC: destination MAC address, source MAC address, MAC Service Data Unit (MSDU), and priority. EPD encodes Protocol Identifiers in the initial octets of each MSDU.

Media-access method dependent information is also encoded in transmitted frames, and can vary by MAC type. IEEE Std 802.3, for example, encodes a Start of Frame Delimiter (SFD) prior to the first octet of the destination MAC address. A MAC of a particular type might encode additional information specific to the operation of that MAC elsewhere in the frame, but that additional information is not relevant to the specification or use of EPD.

The EPD encoding of EtherTypes and LLC Addresses is also referred to as Length/Type encoding. The MAC Service supported by IEEE Std 802.3 requires transmitted frames to have a 64-octet minimum length (13.1 of IEEE Std 802.1AC-2016), and so does not provided the length of the MA-UNITDATA data parameter as assumed by the LLC procedures specified by ISO/IEC 8802-2. However when an LSDU (Link Service Data Unit) specified by IEEE Std 802.2 is conveyed by an IEEE 802.3 MAC the encoding specified in 2.1 b) above is used, prepending the Length. In this case the MAC sublayer service specified by ISO /IEC 8802-2 (2.3 of ISO/IEC 8802-2:1998) is provided by a protocol shim that uses the MAC Service and ISS specified in Clause 11 and 13.1 of IEEE Std 802.1AC-2016, adding a length field on transmission and removing that length field on reception. The term MSDU in this specification always refers to the MAC Service Data Unit as specified by IEEE Std 802.1AC.

When EPD is used, the initial two octets (the Type/Length field) of the MSDU encode a 16-bit number with the first octet comprising the most significant bits. If the value of the Type/Length field is 1536 (or greater) it is interpreted as a Type and the initial octets of the MSDU encode an EtherType (1.7.1) or an OUI Extended EtherType Identifier (1.7.2). If the value of the Type/Length field is 1500 (decimal) or less it is interpreted as a Length and the initial octets of the MSDU encode LLC Addresses (1.7.3) or a SNAP Identifier (1.7.4).

1.7.1 EPD encoding of EtherTypes

An EtherType is encoded in the first two octets of an MSDU (Figure 1-1).

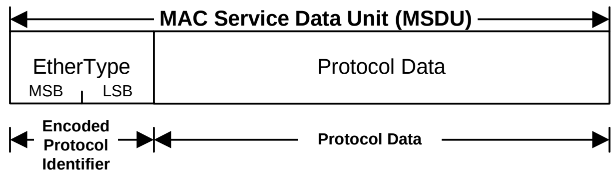


Figure 1-1—EPD Encoding with an EtherType

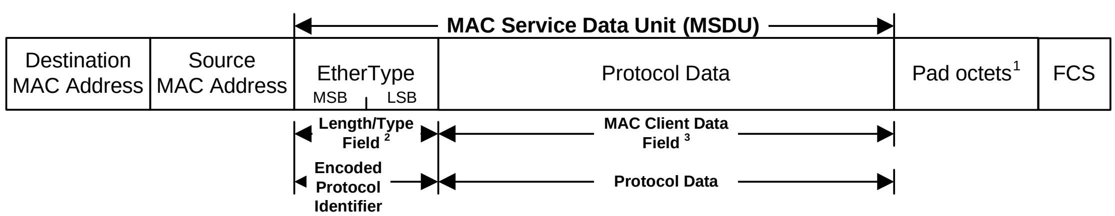
NOTE—The criteria for EtherType assignment require the inclusion of sufficient subtyping and versioning capability within the Protocol Data field to accommodate future development without the assignment of a further EtherType ().

Figure 1-2 shows the MSDU, conveying an EtherType and Protocol Data, within an IEEE 802.3 frame. The media access method-dependent format of frames used by other media-access methods that use EtherType protocol identification with EPD encoding may differ, subject to the following constraints:

a) The media-access method is capable of conveying MSDUs comprising any integral number of octets between three and some maximum as specified for a particular media-access method (see, for example, 3.2.7 of IEEE Std 802.3-2018).

b) The media access method is capable of providing the length of the received MSDU on receipt, if that length (including the encoded EtherType) is greater than or equal to 48 octets.

A protocol that is identified by an EtherType and uses MSDUs of less than 48 octets in length shall be capable of recovering Protocol Data from such an MSDU without relying on an indication of that length from the MAC Service.



[1] Pad field (3.2.8 of IEEE Std 802.3-2018). Present if required to ensure a minimum frame length (DA. SA, MSDU, FCS) of 64 octets.
[2] Length/Type Field (3.2.6 of IEEE Std 802.3-2018).
[3] MAC Client Data Field (3.2.7 of IEEE Std 802.3-2018).

Figure 1-2—EPD encoding with an EtherType in an IEEE 802.3 frame

1.7.2 EPD encoding of OUI Extended EtherType Identifiers

An OUI Extended EtherType Identifier is encoded in the first seven octets of an MSDU (Figure 1-3 provides an example).

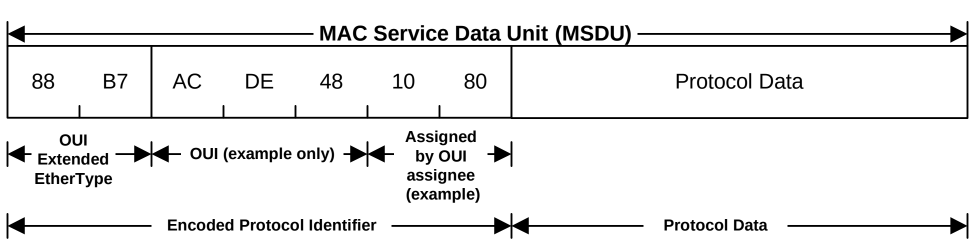
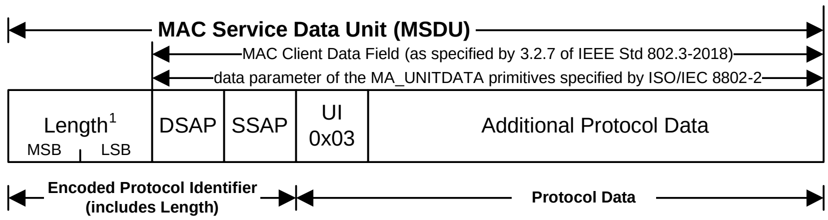


Figure 1-3—EPD Encoding with an OUI Extended EtherType Identifier

1.7.3 EPD encoding of LLC Addresses

The first two octets of the MSDU encode the length of the remainder of the MSDU, as a 16-bit number with the most significant octet first. Length values of between 1 and 1500 (decimal) octets can be encoded. The LLC Address of the Destination Service Access Point (DSAP) is encoded in the third octet, and the LLC Address of the Source Service Access Point (SSAP) is encoded in the fourth octet.

The fifth octet can contain the value 0x03, specified by ISO/IEC 8802-2 as identifying an Unnumbered Information (UI) frame. LLC does not define a relationship between different UI frames, so a protocol that sends UI frames is free to establish its own relationship between those frames, using the protocol data that follows that fifth octet. See Figure 1-4.



[1] Length of the following octets of the MSDU i.e. total length of the MSDU in octets minus two.

Figure 1-4—EPD Encoding with LLC Addresses and an Unnumbered Information frame

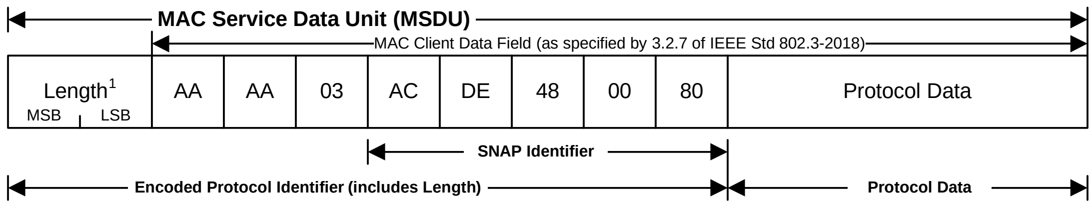
ISO/IEC 8802-2 specifies the use of different frame types (different values of the fifth octet, 0x03 in Unnumbered Information frames) in conjunction with other frame fields to support additional LLC Type 1 procedures and LLC Type 2 and Type 3 procedures. A station that uses the three octet encoding (DSAP, SSAP, UI) specified here (1.7.3) as a Protocol Identifier does not need to implement those additional LLC procedures. However an implementation that is also the subject of a claim of conformance to ISO/IEC 8802-2 needs (at a minimum) to support LLC Type 1 operation with UI, XID, and TEST capability (4.2 of ISO/IEC 8802-2:1998).

NOTE—The Protocol Data field following the fifth octet (0x03 in Figure 1-4) can convey additional protocol identifier information specified by the LLC Address assignee, to provide protocol subtyping and versioning. See, for example, 14.1.2 of IEEE Std 802.1Q-2018.

The EPD encoding of LLC Addresses specified in this document does not permit the transmission of frames whose MSDUs are more than 1502 octets in length. An EtherType, C9-D1, was specified by 12.2 of IEEE Std 802.1AC-2016 to allow the encapsulation of those frames. Where specified by the protocol specification associated with the assignment of an LLC Address, this EtherType may be used by end stations to encapsulate frames whose following octets are those of an LPD encoded LLC Address (). However the protocol specification should note that a peer protocol participant attached to a LAN might receive such a frame with its initial octets encoded as AA-AA-03-C9-D1-Length(MSB)-Length(LSB)-DSAP-SSAP rather than the expected DSAP-SSAP, as a result of being relayed by between EPD and LPD media by a bridge conformant with earlier editions of IEEE 802 standards.

1.7.4 EPD encoding of SNAP Identifiers

The first two octets of the MSDU encode the length of the remainder of the MSDU, as a 16-bit number with the most significant octet first. Length values of between 1 and 1500 (decimal) octets can be encoded. The third, fourth, and fifth encode the octets string AA-AA-03 and are followed by the five octets of the SNAP Identifier.



[1] Length of the following octets of the MSDU i.e. total length of the MSDU in octets minus two.

Figure 1-5—EPD Encoding with a SNAP Identifier

NOTE 1—The LLC Address value 0xAA has been assigned so that this encoding does not conflict with a possible use of that value with an EPD encoding of that LLC Address with protocol data beginning in the sixth octet of the MSDU. However this encoding does not require end station support of ISO/IEC 8802-2 LLC XID and TEST procedures.

The three octet sequence 00-00-00 is not a valid OUI, and is not used to EPD encode SNAP Identifiers.

NOTE 2—IEEE Std 802.1H-1995, Recommended Practice for Media Access Control (MAC) Bridging of Ethernet V2.0 in IEEE 802 Local Area Networks was created to clarify that EPD encoding of SNAP Identifiers does not include use of 00-00-00 in the OUI portion of the SNAP Identifier. This Recommended Practice was withdrawn at a time when it was felt no longer necessary to address possible confusion on this point.

1.8 LLC Protocol Discrimination

This clause (1.8) specifies the encoding, in the initial octets of media-access methods using LLC Protocol Discrimination (LPD), of the following Protocol Identifiers:

a) LLC Addresses (1.8.1)

b) SNAP Identifiers (1.8.2)

c) EtherTypes (1.8.3)

d) OUI Extended EtherType Identifiers (1.8.4)

IEEE Std 802.1AC Media Access Control (MAC) Service Definition defines a MAC Service and an Internal Sublayer Service (ISS) common to all IEEE 802 media-access methods.The media-access method independent ISS is used by MAC Bridges (IEEE Std 802.1Q) to relay frames between LANs and is specified in terms of two primitives, one for frame reception and another for transmission, and their parameters. The MAC Service uses the subset of those parameters relevant to end station use of the MAC: destination MAC address, source MAC address, MAC Service Data Unit (MSDU), and priority. LPD encodes Protocol Identifiers in the initial octets of each MSDU.

Media-access method dependent information is also encoded in transmitted frames, and can vary by MAC type. IEEE Std 802.5, for example, encodes an Access Control (AC) octet prior to the first octet of the destination MAC address and an optional Routing Information (RI) field between the last octet of the source MAC Address and the first octet of the INFO field that conveys the MSDU. A MAC of a particular type might encode additional information specific to the operation of that MAC elsewhere in the frame, but that additional information is not relevant to the specification or use of EPD. A media access method-dependent that uses LPD encoding shall be capable of:

a) Conveying MSDUs comprising any integral number of octets between three and some maximum as specified for that media-access method.

b) Providing the length of the received MSDU on receipt.

NOTE 1—IEEE Std 802.5 [An] has been withdrawn, but provides useful background for the development and maintenance of this document as there is no prospect of changes to that standard. The provisions of this document do not depend on IEEE Std 802.5.

1.8.1 LPD encoding of LLC Addresses

The LLC Address of the Destination Service Access Point (DSAP) is encoded in the first octet, and the LLC Address of the Source Service Access Point (SSAP) is encoded in the second octet. The third and subsequent octets of the MSDU convey values as specified by ISO/IEC 8802-2. See Figure 1-6.

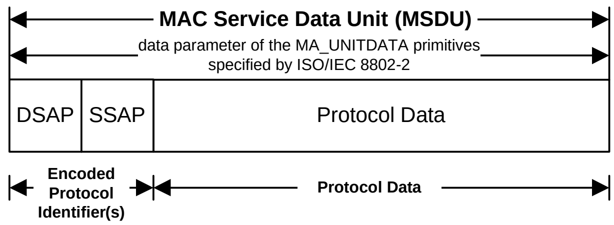
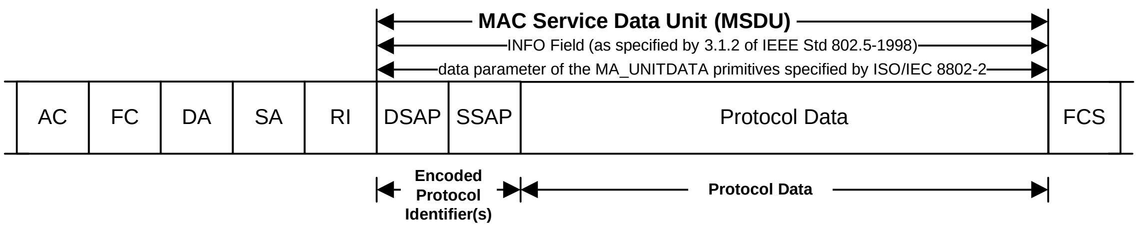


Figure 1-6—LPD Encoding with LLC Addresses

The LLC Address value 0xAA has been assigned to allow its use in the first two octets of LPD encoded frames to encode a SNAP Identifier, an EtherType, or an OUI Extended EtherType Identifier. Figure 1-2 shows the MSDU, conveying LLC Addresses and Protocol Data, within an IEEE 802.5 frame.



Only part of the 802.5 frame is shown (see 3.1.2 of IEEE Std 802.15-1998)

Figure 1-7—LPD Encoding with LLC Addresses in an IEEE 802.5 frame

1.8.2 LPD encoding of SNAP Identifiers

The first, second, and third octets of the MSDU encode the octet string AA-AA-03 and are followed by the five octets of the SNAP Identifier.

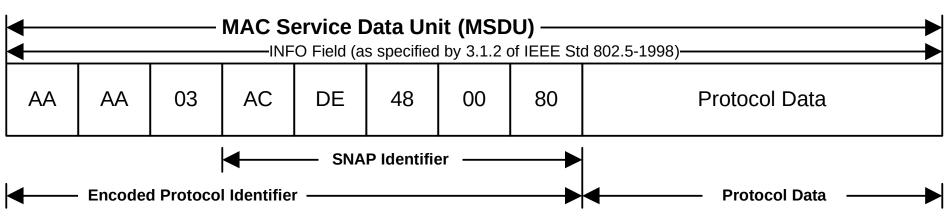


Figure 1-8—LPD Encoding with a SNAP Identifier

1.8.3 LPD encoding of EtherTypes

The first through fifth octets of the MSDU encode the octet string AA-AA-03-00-00-00 and are followed by the EtherType encoded in two octets.

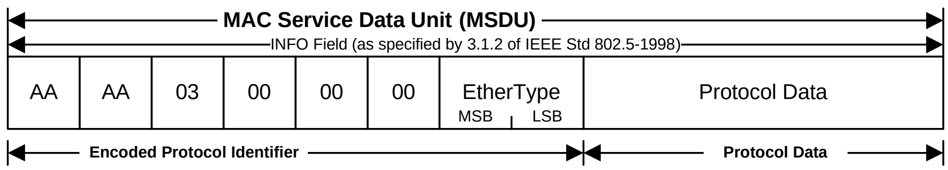


Figure 1-9—LPD Encoding with an EtherType

1.8.4 LPD encoding of OUI Extended EtherType Identifiers

The first through fifth octets of the MSDU encode the octet string AA-AA-03-00-00 and are followed by the OUI Extended EtherType Identifier encoded in seven octets.

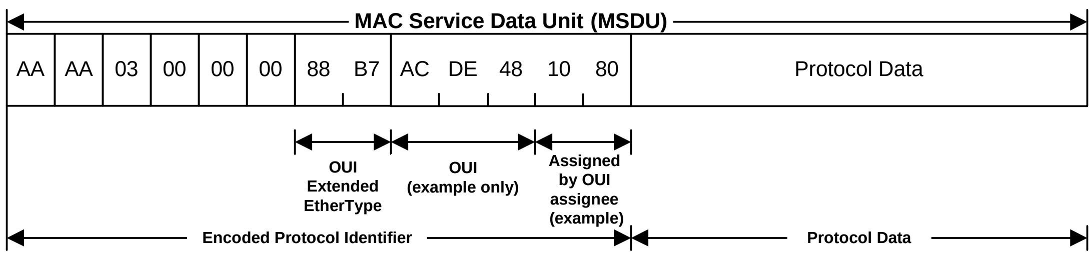


Figure 1-10—LPD Encoding with an OUI Extended EtherType Identifier

1.9 Communicating and using protocol identifiers

IEEE 802 stations that communicate using the MAC Service (IEEE Std 802.1AC) need a shared understanding of the format of the data and applicable procedures (i.e. the protocol) associated with each transmitted and received frame. In principle that shared understanding could be limited to each pair of communicating stations, identified by the source and destination MAC Addresses of the frames they exchange. In practice the need for independent specification and development of protocols for many stations, the need to support management though observation of frames, and the need to support discovery of potential participants in a protocol with frame exchanges using group destination MAC Addresses, requires the use of protocol identifiers and protocol identifier formats of the widest possible scope and applicability.

This clause () describes how protocol identifiers are communicated and used by the following:

a) Bridges that interconnect LANs using the same or different protocol identifier formats (1.9.1)
b) Protocol entities within interface stacks that
   1) Recognize and demultiplex control frames (1.9.2)
   2) Convert received protocol identification information to a common internal format to simplify specification and management (1.9.3)
c) Bridges and protocol entities that add and remove frame tags, including VLAN tags (1.9.3)

NOTE—The operation of bridges and other systems that manipulate protocol identifer encoding formats can be modelled by an interface stack shim that converts from any media-dependendent encoding format to EPD. Standards conformance of any such system is purely with respect to external observable behavior.()

1.9.1 Bridge conversion and validation of initial MSDU octets

The basic requirement is for protocol entities in communicating stations to be able to exchange both protocol identifiers and accompanying protocol data (Figure 1-11).

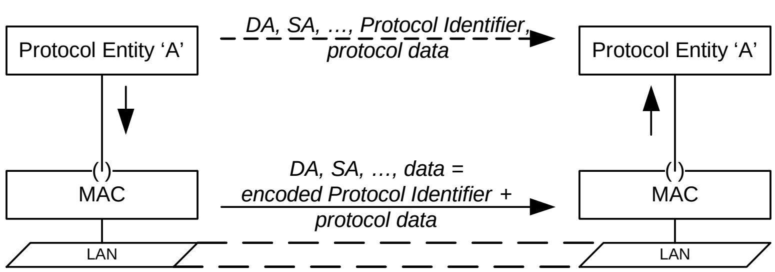


Figure 1-11—Transmission with an encoded Protocol Identifier

Where the communicating stations are attached to the same individual LAN they are obliged to encode each Protocol Identifier in a mutually comprehensible way, e.g. they both use EPD encoding or both use LPD. Where the path between the passes through a bridge, that bridge can convert from one protocol identifier encoding format to another (see Figure 1-11). That conversion applies to all frames transiting the bridge. IEEE Std 802.1Q does not specify selective protocol identifier encoding or decoding for frames with a given DA, SA, or DA and SA pairing, so all stations attached to a given individual LAN and transmitting frames that can be bridged support the ISS (see IEEE Std 802.1AC) with the same protocol identifier encoding format or a subset of that encoding.

NOTE 1—Many different types of device can be attached to the same LAN, and not all those devices include protocol entities that require the use of multiple types of protocol identifier.

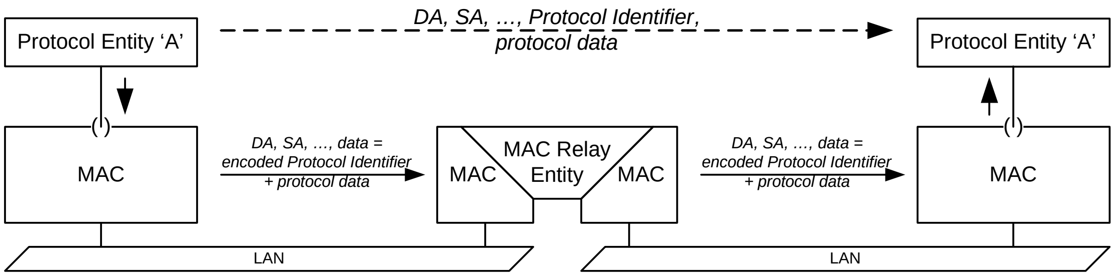


Figure 1-12—Bridge can re-encode protocol id for stations attached to different LANs

A bridge that relays frames between individual LANs that use the same protocol identifier encoding format is not required to convert or validate the protocol identifier encoding in the initial octets of each relayed frame. For example, a bridge implementation that relays frames between two IEEE 802.3 LANs might relay without modification a frame whose initial octets begin with a Length in the Length/Type field followed by AA-AA-03-00-00-00 and an EtherType (see 1.7.4). However a bridge or a series of bridges that connect LANs that use different protocol identifier encoding formats can convert the protocol identifier encoding in the initial octets of all relayed frames to a single format, replacing a Length-AA-AA-03-00-00-00-EtherType sequence with the EtherType in the Type/Length field when relaying a frame to an IEEE 802.3 LAN.

NOTE—IEEE Std 802.1H-1995 [B2] noted that a frame bridged from an IEEE 802.3 LAN to FDDI (LPD) and back to IEEE 802.3 (EPD) could remove a Length-AA-AA-03-00-00-00 prepended to an EtherType, which is why the prepending sequence is not used with EtherTypes and medium--access methods that specify the use of EPD.

1.9.2 Protocol recognition and demultiplexing within network interface stacks

The functionality provided within a network interface stack can require demultiplexing of frames to addressable protocol entities that control that functionality, as shown, e.g., for IEEE Std 802.1AX Link Aggregation and the Link Aggregation Control Protocol (LACP) in Figure 1-13. A frame is received by a protocol entity if, and only if, the frame's MAC destination address and the Protocol Identifier both identify the entity. The MAC destination address can be an individual address or a group address.

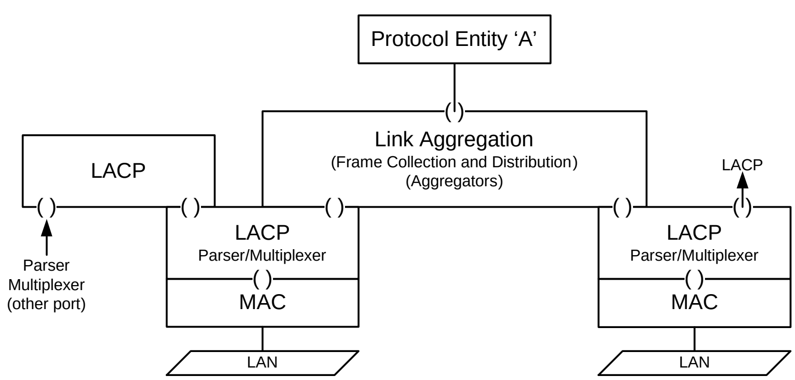


Figure 1-13—Protocol identification within an interface stack (LACP example)

NOTE 2—Figure 1-13 is simplified from Figure 6-2 of IEEE Std 802.1AX-2020. For a definitive specification consult that standard. IEEE Std 802.1AX uses the Slow Protocols EtherType assigned to IEEE Std 802.3 and a subtype to identify LACP, and a separate DRNI EtherType (with subtypes) for protocols that control Distributed Relay operation.

1.9.3 Protocol format conversion within an interface stack

The specification of a protocol that supports network interface stack functionality can assume a particular Protocol Identifier encoding format. IEEE Std 802.1AX, for example, assumes the use of EPD. The management of other aspects of system functionality, e.g. the stream identification functions specified by IEEE Std 802.1CB, can also make implicit or explicit assumptions about Protocol Identifier encoding. Any such function should support EPD. The consistent use of a single encoding format within a system or set of related systems simplifies management, and can be supported by a protocol shim that translates between the encoding used by a particular media-access method and the encoding used within the system(s). Figure show such a shim within interface stacks that support Link Aggregation and LPD media.

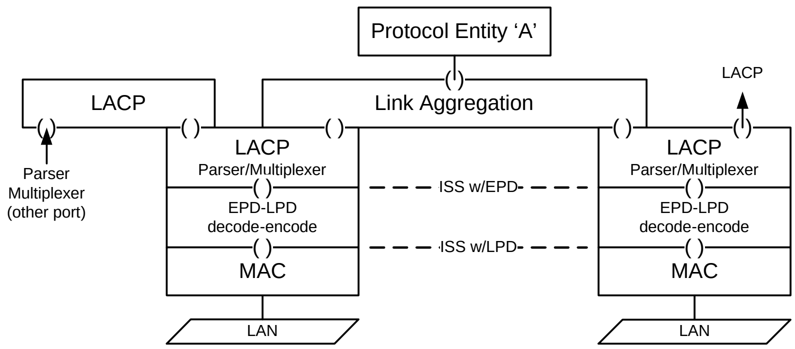


Figure 1-14—Using Link Aggregation LACP over LPD media

NOTE 3—A shim is a protocol entity that uses the same service as it provides (in this case the ISS), while adding functionality. Other shims include Link Aggregation and the MAC Security Entity specified in IEEE Std 802.1AE. Clause 7 of IEEE Std 802.1AC-2016 describes shims and other basic architectural concepts and terms.

The changes performed by a shim that provides the ISS with the initial octets of each frame conveying a protocol identifier using EPD encoding, and uses the ISS with LPD encoding was specified in 12.1 and 12.2 of IEEE Std 802.1AC-2016. The effect of that specification is to provide the conversion between the EPD and LPD encodings as described in 1.7 and 1.8 of this document, and to clarify that no particular result can be relied upon when the initial octets do not encode a valid protocol identifier—as is the case when the Length/Type field of a frame that is assumed to conveying an EPD encoding has a value between 1500_{10} and 1536_{10}. IEEE Std 802.1AC-2016 adds a further conversion, to allow protocols identified by LLC Addresses (1.8) to use the LLC encapsulation EtherType 0x8870 followed by those LLC Addresses. Use of that EtherType allows frames of those protocols to be longer than 1500_{10} on some EPD media, and is typically used to support the IS-IS routing protocol [Bn]. This document recommends the use of the LLC encapsulation EtherType by end stations, but recommends against the addition or removal of that prepending EtherType by bridges (). The LLC Addresses that follow the encapsulation EtherType can be treated as subtypes, and the initial octets of the frame follow the common rules for EtherType encoding (1.7.1, 1.8.3).

1.9.4 Tag addition and removal

One or more tags, such as the C-VLAN Tag specified by IEEE Std 802.1Q, can be prepended to or removed from a frame on the path from the originating protocol entity to its destination. Intermediate systems that can bridge, e.g., from LPD to EPD media, cannot be expected to understand the format of all possible tags. A requirement for such an understanding would inhibit future protocol development. In an intermediate system, a protocol entity that adds or removes a particular type of tag can also not be expected to know the type of the media where a following tag (of a potentially different type) was originally added. These potential difficulties are resolved by encoding a protocol identifier in the initial octets of each tag, by limiting the choice of that protocol identifier to be an EtherType or an OUI Extended EtherType Identifier, and by requiring any protocol entity that adds a tag to a frame whose initial octets are not EPD encoded to convert those octets to EPD encoding prior to addition of the tag (1.9.3). Tags other than an outermost tag (in the initial octets of frame data) are thus always EPD encoded. The encoding format of the outermost tag always matches the format used by the local medium-acccess method. A tag can only be added or removed by a protocol entity that is specified for handling tags of that type, and is therefore aware of the tag format. If the tag is removed from the initial octets of a frame prior to transmission, the following octets are converted to use the encoding format specified for the transmit medium-access method. Figure 1-15 illustrates these points for VLAN tags and a combination of LPD and EPD media.

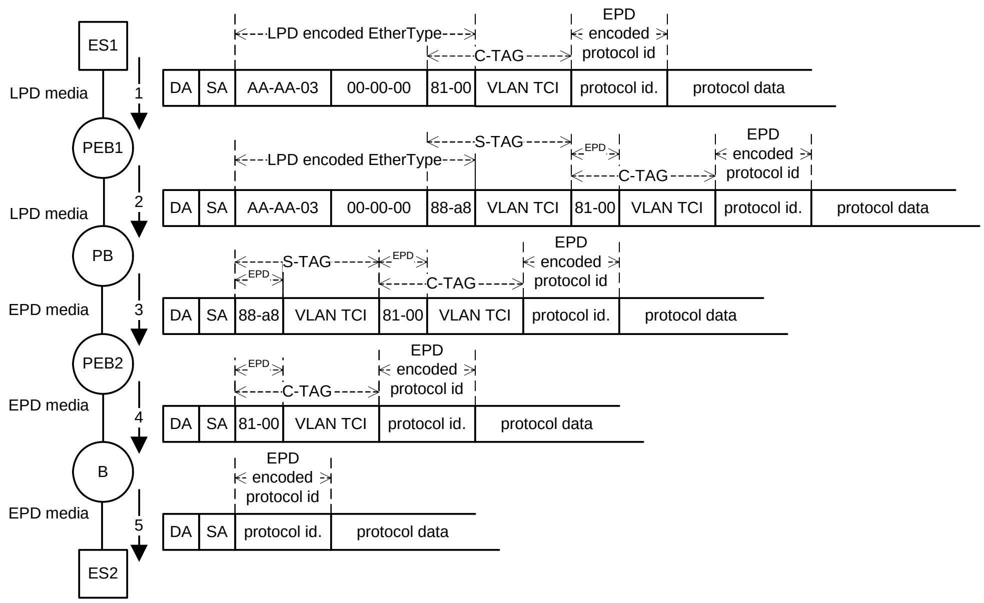


Figure 1-15—Inner tags always EPD encoded, outermost tag is media-dependent

In Figure 1-15 an end station (ES1) on an LPD medium transmits a C-VLAN with the destination address of an endstation ES2. An S-VLAN tag is added by a Provider Edge Bridge (PEB1) that forwards the frame on LPD media. A Provider Bridge (PB) then forwards the frame on to EPD media, and a further Provider Bridge, PEB2, removes the S-VLAN tag before forwarding the frame to a Customer VLAN Bridge (B) that forwards it (untagged) to ES2. Since PEB2 is attached only to EPD media, it does not convert between protocol discriminator encoding formats when removing the S-TAG. Frames received from other end stations (ES3, ES4, …) might only traverse EPD media, and PEB2 has no way of determining which frames would contain EPD encoded C-TAGs and which LPD encoded C-TAGs, if the latter were permitted.

Note also that when ES1 (attached to LPD media) transmits the original C-VLAN tagged frame, it has already encoded the protocol identifier following the C-TAG in EPD format, so bridge B can remove C-TAGs before transmitting to ES2 without having to know that ES1 is attached to LPD media.