5. Reference models (RMs)

5.2 RM descriptions for end stations

5.2.2 LLC sublayer

The LLC sublayer is between the MAC sublayer and the Network layer (Layer 3) in the OSI model. In IEEE 802, the functions in the LLC are defined in IEEE 802.1 standards. Among the functions that can be present in the LLC are:

1) IEEE Std 802.1AE™ provides MAC security with connectionless user data confidentiality, frame data integrity, and data origin authenticity by media access independent protocols and entities that operate transparently to MAC clients.Link Aggregation
2) IEEE Std 802.1AX™ provides the ability to aggregate two or more links together to form a single logical link at a higher data rate.
3) IEEE Std 802.1X provides authentication, authorization, and cryptographic key agreement mechanisms to support secure communication between end stations connected by IEEE 802 networks.

NOTE: Prior to the 2014 revision of this standard, the LLC layer was defined as only the functions described in IEEE Std 802.2. However, this did not allow for the other end-station functions developed by various IEEE 802.1 standards. In addition, IEEE Std 802.2 has been withdrawn. Since 2014, the LLC layer has been defined to include all of the relevant functions that occur between the MAC sublayer and the Network Layer.

9. Protocol identifiers and protocol multiplexing

9.1 Introduction

A key function of the LLC sublayer is to support the multiplexing and demultiplexing of multiple network layer protocols over an IEEE 802 network.

Within the network layer, entities can exchange data by a mutually agreed protocol mechanism. A pair of entities that do not support a common protocol cannot communicate with each other. For multiple network layer protocols to operate over an IEEE 802 network, the transmitting and receiving HLPDEs of the LLC sublayer cooperate to identify the network layer protocol to be invoked for each service data unit delivered by the lower layer.

A network-layer protocol is identified within the LLC sublayer by means of a protocol identifier of a specific protocol type, associated with the protocol. Three specific types of protocol identifier are supported:

a) E-Type: The E-Type protocol identifier is an EtherType, which is a two-octet identifier, in the range from 06-00 through FF-FF[24], that is uniquely assigned to a protocol. Assignments are made and recorded by the IEEE Registration Authority[25]. Two EtherType values, known as the Local Experimental EtherTypes, do not reflect global protocol assignments but instead are assigned for use by local administrators who decide on their local mapping to protocols.

NOTE 1—While every E-Type protocol identifier is an EtherType, not all EtherTypes are E-Type PIs. For example, some EtherType values are assigned to indicate specific Layer 2 functionality rather than a network-layer protocol; in these cases, a network-layer PDU is typically encapsulated and carried later in the frame.

b) L-Type: The L-Type protocol identifier is an LSAP address, which is a one-octet identifier that is uniquely assigned to a protocol. LSAP address assignments are made and recorded by the IEEE Registration Authority.

NOTE 2—While every L-Type protocol identifier is an LSAP address, not all LSAP address are L-Type PIs. For example, some LSAP address values are assigned to indicate specific Layer 2 functionality rather than a network-layer protocol; in these cases, a network-layer PDU is typically encapsulated and carried later in the frame.

c) O-Type: The O-Type protocol identifier is created under the authority of an OUI, OUI-36, or CID assignee by appending bits to the OUI, OUI-36, or CID assignment. The O-Type identifier allows the OUI, OUI-36, or CID assignee to derive globally-unique protocol identifiers without an external registration authority.

Because each protocol identifier type is a different length, the protocol identifier type of a protocol identifier follows from its length. The types are also distinguishable by numeric value. The largest valid L-Type value is 0xFE (254). Valid E-Type values are within the range 0x0600 (1536) to 0xFFFF (65 535). The O-Type value is always greater than 0xFFFF.

In IEEE 802 networks, the protocol identifier is encoded into a protocol identification field (PIF) that is incorporated as the initial octets of the LPDU, prepended to the higher-layer protocol data unit, as shown in Figure 1. In principle, the LPDU is carried as a MAC service data unit and is opaque to the MAC; use of the LPDU structure is limited to the LLC endpoints of the IEEE 802 network. Some exceptions to this opaqueness are specified in IEEE 802 standards; for example, the first two octets of the LPDU are exposed to the Ethernet MAC of IEEE Std 802.3.

[24]By convention, EtherTypes are represented as two hexadecimal numbers followed by a dash followed by two hexadecimal numbers with no prefix. In this standard, hexadecimal numbers other than EtherTypes are prefixed with 0x.

[25]More information can be found at https://standards.ieee.org/products-programs/regauth/ and https://regauth.standards.ieee.org/standards-ra-web/pub/view.html#registries.

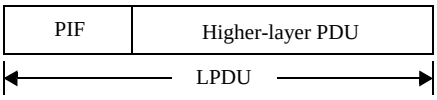

Figure 1—LPDU including prepended PIF

Two forms of encoding a protocol identification field are specified. With either of these two encoding forms, the encoding includes sufficient information for the receiving HLDPE to: a) identify the protocol identification field; b) determine the protocol identifier type; and c) identify the protocol identifier. The HLDPE is then enabled to strip the PIF from the data payload and forward the resulting payload to the network-layer protocol that is associated with the protocol identifier.

Three PIF encoding forms are specified, each of which allows the HLDPE to parse the PIF for any of the three protocol identifier types. These are:

1) Type 1 PIF encoding, which is reserved;
2) Type 2 PIF encoding, which does not use a Length/Type field; and
3) Type 3 PIF encoding, which makes use of a Length/Type field.

While the two PIF encoding forms are each capable of supporting all protocol identifier types, no provision is made herein for the HLDPE to ascertain which of the two encoding forms was applied at the source. Without such information, the HLDPE cannot parse the data payload to identify the PIF. This standard presumes that the HLDPE is aware of the encoding form used.

9.2 EtherTypes and E-Type protocol identifiers

9.2.1 Format, function, and administration

EtherType values are assigned by the IEEE RA[26]. An EtherType is a sequence of 2 octets, interpreted as a 16-bit numeric value with the first octet containing the most significant bits and the second octet containing the least significant bits. Values in the 0–1535 range are not available for use.

Some EtherTypes are assigned as E-Type protocol identifiers and are associated with higher-layer protocols, typically network-layer protocols. Examples of such EtherTypes are 08-00 and 86-DD, which are used to identify IPv4 and IPv6, respectively.

Some EtherTypes not assigned as E-Type protocol identifiers but are instead used within Layer 2. Examples of such EtherType are the OUI Extended EtherType 88-B7 and the LLC Encapsulation EtherType 88-70. The specifications associated with Layer 2 EtherTypes provides guidance as to how to parse the remainder of the data field to extract the protocol identifier.

9.2.2 Public EtherType assignments subset

The IEEE Registration Authority (RA) provides a public listing of EtherType assignments[27]. Many of these are for private or proprietary purposes. However, others are incorporated into well-known standards. In some cases, the IEEE RA Public Listing for an EtherType identifies an assignee without explicitly identifying the standards in which the use of that EtherType is specified. For ready reference by users and developers of such standards, Annex F identifies some well-known EtherTypes and the protocols they

[26]More information on EtherTypes can be found on the IEEE RA web site, https://standards.ieee.org/products-programs/regauth/ether-type/.and https://regauth.standards.ieee.org/standards-ra-web/pub/view.html#registries.

[27]The EtherType public listing is the public view of the EtherType registry managed by the Registration Authority (see https://standards.ieee.org/regauth).

identify. This subset is derived by combining the EtherTypes listed in the ietf-ethertypes YANG module specified in IETF RFC 8519 [B19] with the subset of EtherTypes defined by IEEE 802 Standards (e.g., IEEE 802.1Q, 802.3, etc.) and as provided by participants that developed this standard. Information on products released after that date can be found on the IEEE SA Registration Authority web site: https://standards.ieee.org/products-programs/regauth/ethertype/ and https://regauth.standards.ieee.org/standards-ra-web/pub/view.html#registries. The subset in Table F.1 and in F.3 is provided solely for the convenience of users of this standard and does not constitute an endorsement by IEEE of the listed protocols.

The EtherType public listing includes the following fields, specified by the EtherType assignee:

— **Assignment** — The hexadecimal representation of the EtherType.
— **Assignment Type** — The type is EtherType[28].
— **Company Name** — The registrant of the Assignment.
— **Company Address** — The address of the registrant.
— **Protocol** — A brief protocol description, as provided by the registrant.

This Standard includes the following fields in Table F.1 for use by the YANG module:

a) **Friendly Name** — A short alphanumeric name for the Assignment that is unique within the YANG module in F.2 and is used to enumerate the entry.
b) **Short Description** — A short description of the assigned protocol per its typical usage.
c) **Reference** — A reference to a standard associated with the EtherType assignment.

A YANG model representation can be found in F.3.2.

9.2.3 EtherType sub-protocol encoding

The EtherType identifier space is a finite resource. When the IEEE Registration Authority assigns an EtherType to an organization, it specifies that the usage should be extensible to alternative variations of the protocol and to new versions. This protects the resource against premature exhaustion due to repeat assignment requests from a single user. Such usage also benefits the assignee, since attaining an assignment requires time, effort, and funds.

In order to allow for a single EtherType to multiplex various sub-protocols and versions, a protocol subtype and a protocol version identifier should be used. Figure 2 is an example of the EtherType in a PIF, or at the end of a PIF. As shown, the PIF is followed by additional fields that, together with the PIF, form the sub-protocol information field (SPIF). While the contents of the PIF are sufficient to identify the protocol sufficiently for the HLPDE to direct to the frame to the correct higher-layer protocol, the contents of the protocol subtype and protocol version identifier are intended to be used within the higher-layer protocol to direct the frame to the correct sub-protocol. The lengths of the protocol subtype and the protocol version identifier fields, as well as their order of appearance within the frame, are not constrained by this standard but are determined by the user. The IEEE 802 network has no visibility into this structure.

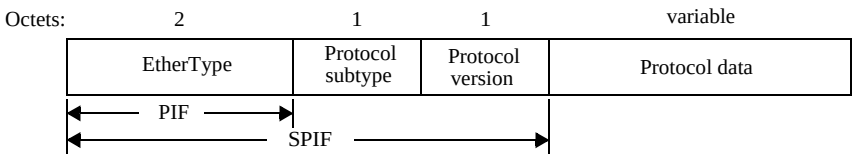

Figure 2—Example of sub-protocol encoding

[28]EtherType is the only assignment type for the records in the EtherType public listing.

9.2.4 Local Experimental EtherTypes

In order to allow users to conveniently operate E-Type protocol identification without a unique assignment, two EtherType values, known as the Local Experimental EtherTypes, are assigned use within a locally administered network. The values of the Local Experimental EtherTypes are listed in Table 1.

Table 1—Assigned EtherType values

Name	Value
Local Experimental EtherType 1	88-B5
Local Experimental EtherType 2	88-B6
OUI Extended EtherType	88-B7

Within that network, a local administrator is free to use a Local Experimental EtherType and to assign subtypes for protocol development purposes. However, by virtue of the way these EtherTypes are intended to be used, the following practical and administrative constraints apply to their use:

a) Since the format for protocols using the Local Experimental EtherTypes does not contain a means to identify the administrative domain, it might not be possible to identify the protocol of a frame if protocols developed within different administrative domains using Local Experimental EtherTypes are used in the same network. Hence, the use of these EtherTypes to identify protocols can only be achieved reliably if all uses of the EtherTypes are within the control of a single administrative domain. Therefore, these EtherTypes shall not be used in protocols or products that are to be released for use in the wider networking community, as freeware, shareware, or any part of a company's commercial product offering. Products shall be transitioned to a product EtherType before it is deployed in an environment outside the developing organization's administrative control, for example, when deployed with a customer or any other connected environments for testing.

b) Local Experimental EtherType shall not be permanently assigned for use with a given protocol or protocols.

c) End stations that bound any administrative domain should be configured to prevent frames containing a Local Experimental EtherType from passing either into or out of a domain in which its contents can be misinterpreted. For example, the default configuration of any firewall should be to not pass this EtherType.

A Local Experimental EtherType is processed by the HLPDE in the same manner as other E-Type identifiers, using either Type 3 PIF encoding or Type 2 PIF encoding. However, in order to allow for a single Local Experimental EtherType to multiplex various experimental protocols, sub-protocols, and versions within the same experimental network, a protocol subtype and a protocol version identifier shall be used in conjunction with the Local Experimental EtherType value, as illustrated in Figure 2.

9.3 LSAP addresses and L-Type protocol identifiers

LSAP addresses values are assigned by the IEEE RA. An LSAP addresses is a sequence of 8 bits, interpreted as a numeric value. The least significant bit is set to 0 for individual identifiers. All LSAP addresses are individual identifiers.

Some LSAP addresses are assigned as L-Type protocol identifiers and associated with higher-layer protocols. An example is 0x42, which is used to identify the bridge protocol data unit of IEEE Std 802.1Q.

Some LSAP addresses are not assigned as L-Type protocol identifiers but are instead used within Layer 2. An example of such an LSAP address is 0xAA which is used in SNAP encoding.

LSAP address 0xFE is the basis of an extensible identifier format, as specified in ISO/IEC TR 9577:1999. One use of that extensible protocol identification is the IS-IS protocol of IEEE Std 802.1Q.

The IEEE Registration Authority (RA) provides a public listing of LSAP addresses[29].

9.4 O-Type protocol identifiers

The O-Type protocol identifier is a five-octet value that, while not directly assigned by a registration authority, is nevertheless intended to allow a globally-unique association to a protocol. The O-Type protocol identifier is created under the authority of an OUI, OUI-36, or CID assignee by appending bits to the OUI, CID, or OUI-36 assignment. The assignee is exclusively authorized to create O-Type protocol identifiers using their OUI, OUI-36 or CID.

An O-Type protocol identifier created by the assignee of an OUI or CID is illustrated in Figure 3.

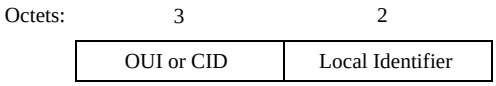

Figure 3—Protocol identifier composed of an OUI or CID

An O-Type protocol identifier created by the assignee of an OUI-36 is illustrated in Figure 4.

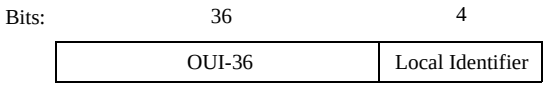

Figure 4—Protocol identifier composed of an OUI-36

9.5 PIF Encoding

The encoding of protocol identifier does not change the meaning of the protocol identifier or its association to a protocol. For example, the protocol identified by a particular E-Type EtherType is identical, regardless of its PIF encoding. The same is true of L-Type and O-Type identifiers. If a bridge transforms the PIF encoding of a frame while relaying, the receiving end station will nevertheless be able to ascertain the destination protocol as long as it knows the final PIF encoding form.

9.5.1 Type 2 PIF encoding

9.5.1.1 Type 2 PIF encoding of an L-Type protocol identifier

Type 2 PIF encoding of an L-Type protocol identifier entails embedding the protocol identifier as shown in Figure 5.

Note that for this encoding, the protocol identifier is duplicated in the PIF.

[29]The LSAP address public listing (https://standards.ieee.org/products-programs/regauth/llc/public/) is the public view of the LSAP address registry managed by the IEEE Registration Authority

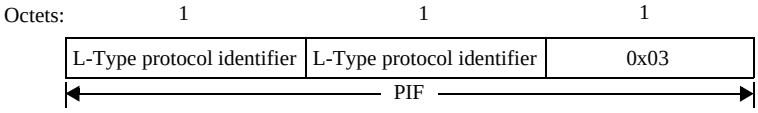

Figure 5—Type 2 PIF encoding of an L-Type protocol identifier

NOTE—The special case of L-Type protocol identifier value of 0xAA is disallowed in this encoding.

9.5.1.2 Type 2 PIF encoding of an E-Type protocol identifier

Type 2 PIF encoding of an E-Type protocol identifier entails embedding the protocol identifier as illustrated in Figure 6.

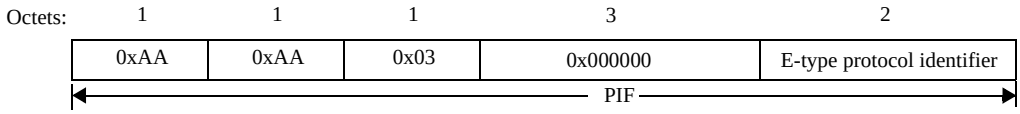

Figure 6—Type 2 PIF encoding of an E-Type protocol identifier

Note that the one-octet value 0xAA is never assigned as the L-Type protocol identifier of a network-layer protocol. This allows the HLPDE to distinguish the PIF with respect to the Type 2 PIF encoding of an L-Type protocol identifier, 9.5.1.1.

9.5.1.3 Type 2 PIF encoding of an O-Type protocol identifier

Type 2 PIF encoding of an O-Type protocol identifier entails embedding the protocol identifier as illustrated in Figure 7.

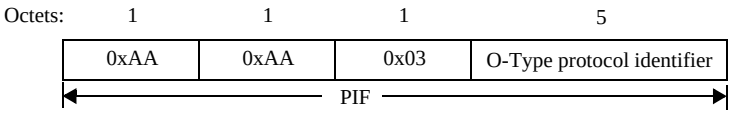

Figure 7—Type 2 PIF encoding of an O-Type protocol identifier

The O-Type protocol identifier is shall not be set to begin with 0x000000. This allows the HLPDE to distinguish the PIF with respect to the Type 2 PIF encoding of an E-Type protocol identifier.

9.5.1.4 SNAP encoding

Both the Type 2 PIF encoding of an E-Type protocol identifier and the Type 2 PIF encoding of an O-Type protocol identifier are also known as Subnetwork Access Protocol (SNAP) encoding. SNAP encoding of an EtherType per Figure 6 was first described in RFC 1042 and is known as the RFC 1042 form of SNAP.

9.5.2 Type 3 PIF encoding

9.5.2.1 Type 3 PIF encoding of an E-Type protocol identifier

Type 3 PIF encoding of an E-Type protocol identifier entails embedding the protocol identifier as illustrated in Figure 8.

The PIF contains only the EtherType.

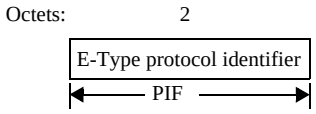

Figure 8—Type 3 PIF encoding of an E-Type protocol identifier

NOTE—The EtherType is uniquely distinguishable from any possible value of the Length field, 9.5.2.2.

9.5.2.2 Type 3 PIF encoding of an L-Type protocol identifier

Type 3 PIF encoding of an L-Type protocol identifier entails embedding the protocol identifier as illustrated in Figure 8.

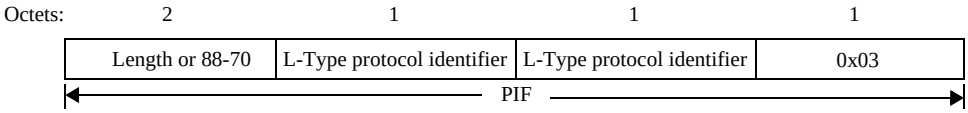

Figure 9—Type 3 PIF encoding of an L-Type protocol identifier

The initial field is typically a Length, which takes a value no greater than 0x05DC. Since the minimum EtherType value is 06-00, the HLPDE can distinguish this encoding with respect to the Type 3 PIF encoding of an E-Type protocol identifier. When using a Length, the value of the Length field assigned by the LLC indicates the length of the LLC service data unit in octets, plus 3, but never exceeding 0x05DC. Some MAC sublayers (in particular, that of IEEE Std 802.3) specify that the LLC service data unit may be padded to meet a minimum length, with the Length field unchanged. In this case, the length and the Length field are temporarily inconsistent during transmission; however, the Length field is then used to remove the padding prior to delivery to the LLC.

In lieu of a Length, Type 3 PIF encoding of an L-Type protocol identifier alternatively uses the LLC Encapsulation EtherType (value 88-70), which is never used as an E-Type protocol identifier and does not indicate a length. This allows, for example, Type 3 PIF encoding of an L-Type protocol identifier even when the LLC service data unit is too long to be expressed in the limited range of the Length field.

The LLC Encapsulation EtherType does not allow depadding of padded short frames. Likewise, Type 3 PIF encoding of an E-Type protocol identifier does not provide a Length for depadding. In either case, the higher-layer procotol might need to provide a depadding service for short frames. If the LLC service data unit is sufficiently long that MAC padding is not added, then the Length value is not used by the MAC and the LLC Encapsulation EtherType functions identically to a Length value.

9.5.2.3 Type 3 PIF encoding of an O-Type protocol identifier

Type 3 PIF encoding of an O-Type protocol identifier entails embedding the protocol identifier in a seven-octet PIF per Fig. LT-O.

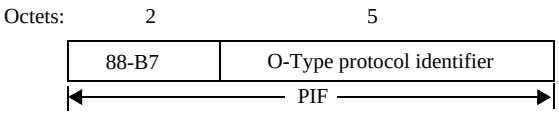

Figure 10—Type 3 PIF encoding of an O-Typeprotocol identifier

The initial field is the OUI Extended EtherType which is never used as an E-Type protocol identifier. This allows the HLPDE to distinguish this encoding with respect to the Type 3 PIF encodings of the E-Type and L-Type protocol identifier.

9.5.3 Encoding type and PIF length

Type 3 PIF encoding is more efficient than Type 2 PIF encoding for carrying the E-Type protocol identifier due to a smaller PIF: 2 octets vs. 8 octets. E-Type protocol identifiers are typically the most common in use. Type 3 PIF encoding is also more efficient than Type 2 PIF encoding for carrying the O-Type protocol identifier: 7 octets vs. 8 octets. Type 2 PIF encoding is more efficient than Type 3 PIF encoding for carrying the L-Type protocol identifier: 3 octets vs. 5 octets.

9.6 Context-dependent identifiers

The IEEE RA tutorial [B2] explains the creation of context dependent identifiers. Just as the OUI is extended to create EUI-48 and EUI-64 identifiers, or a CID can be extended to create a locally administered MAC address, other extended identifiers can be created from an OUI or CID assignment. Such extended identifiers are referred to as context-dependent identifiers. These identifiers are not necessarily globally unique, but are intended to only be unique within a well specified context.

In some cases, the context of a context-dependent identifier is the IEEE 802 LAN. Since this is the same context in which local identifiers operate, the SLAP of Clause 8 provides a basis to assign unique context-dependent identifiers, such as NUI-48 and NUI- 64, within that context.