9. Protocol identifiers and ~~context-dependent identifiers~~protocol multiplexing

9.1 Introduction

~~This clause describes methods that allow multiple network layer protocols to be carried over an IEEE 802 network. These methods provide for the following:~~

— ~~The operation of multiple network layer protocols~~

— ~~The migration of existing networks to future standard protocols~~

— ~~The accommodation of future higher layer protocols~~

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 ~~a given~~ the network layer, entities can exchange data by a mutually agreed ~~upon~~ 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 ~~within a layer~~over an IEEE 802 network, ~~it is necessary~~ the transmitting and receiving HLPDEs of the LLC sublayer cooperate to ~~determine which~~ identify the network layer protocol ~~is~~ to be invoked ~~to process a~~ 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, that is uniquely assigned to a protocol. Assignments are made and recorded by the IEEE Registration Authority1. 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[24].

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 identifier: This five-octet 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.

[24]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.

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.

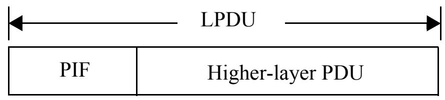

Figure 1—LPDU including prepended PIF

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.

Various network and higher layer protocols have been assigned reserved LPD addresses or EtherTypes, as recorded by the IEEE RA[25]. These addresses permit multiple protocols to operate over a single MAC entity.

This clause describes the protocol identifiers used for the LPD and EPD methods as well as a protocol identifiers based on OUI-36.

The EPD method shall be the primary specified means for protocol identification at the LLC sublayer in IEEE 802 standards developed after January 2011[26], excluding amendments to existing standards.

9.2 ~~EtherTypes~~

Four types of Protocol Identifier that have been standardized for use in IEEE 802 LANs:

a) EtherTypes, 9.3
b) OUI Extended EtherType Identifiers, 9.5

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

[25]More information can be found at https://standards.ieee.org/products-programs/regauth/.
[26]IEEE Std 802.2™-1989 (reaffirmed 2003) was administratively withdrawn as an IEEE standard on 11 January 2011 in deference to the stabilized standard ISO/IEC 8802-2:1998 where the same material continues to be available.

c) LLC addresses, 9.9

d) SNAP Identifiers, 9.8

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

Two MAC 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 Network:

1) Ethernet Protocol Discrimination (EPD, 1.7) as used by Ethernet (IEEE Std 802.3).

NOTE 3—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 4—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.

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 5—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.

9.3 EtherTypes and E-type protocol identifiers

9.3.1 Format, function, and administration

EtherType ~~protocol identification~~ values are assigned by the IEEE ~~RA~~RA[27] ~~and are used to identify the protocol that is to be invoked to process the user data in the frame~~. An EtherType is a sequence of 2 octets,

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

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 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 0x08-00 and 0x86-DD, which are used to identify IPv4 and IPv6, respectively.

~~It is strongly recommended when designing new protocols to be identified by an EtherType, that fields are defined to provide for subtyping. The format used for subtyping in a protocol described in 9.3.4 is recommended.~~

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 0x88-B7 and the LLC Encapsulation EtherType 0x88-70. 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.

9.3.2 Public EtherType assignments subset

The IEEE Registration Authority (RA) provides a public listing of EtherType assignments[28]. 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 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[29].
— **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.

[28]The EtherType public listing is the public view of the EtherType registry managed by the Registration Authority (see https://standards.ieee.org/regauth).
[29]EtherType is the only assignment type for the records in the EtherType public listing.

9.3.3 ~~EtherTypes for prototype and vendor~~EtherType sub-~~specific~~ protocol ~~development~~en-coding

~~The EtherType identifier space is a finite resource. The vendor-specific protocol identifier is a means whereby protocol developers may assign permanent protocol identifier values without consuming type values from this limited resource. This can be useful for prototype, experimental, and private/proprietary protocols to be developed without impacting the rest of the EtherType namespace.~~

~~These objectives are supported by the following EtherType assignments and associated rules for their use:~~

a) ~~Two EtherType values, known as the Local Experimental EtherTypes, as specified in 9.3.4, assigned, as the name implies, for experimental use within a local area~~
b) ~~A single EtherType value, known as the OUI Extended EtherType, as specified in 9.5, assigned for the identification of vendor-specific protocols~~

~~The values of the Local Experimental EtherTypes and the OUI Extended EtherType are listed in Table 2.~~

Table 1—Assigned EtherType values

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

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.

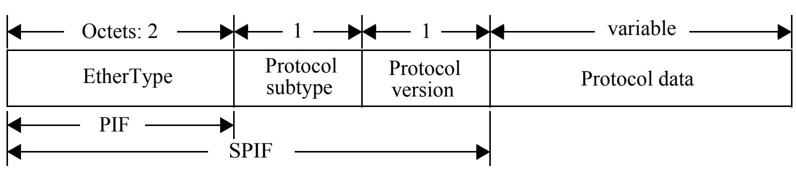

Figure 2—Example of sub-protocol encoding

9.3.4 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 2.

Table 2—Assigned EtherType values

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

The Local Experimental EtherTypes are only intended for use in conjunction with experimental protocol development within a privately administered development network, for example, within an experimental network that has no wide area connectivity. 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 EtherType values.

In order to allow for different experimental protocols, sub-protocols, and versions to coexist within the same experimental network, a protocol subtype and a protocol version identifier shall be used in conjunction with the Local Experimental EtherType value. Figure 3 shows the format of an IEEE 802.3 frame carrying a Local Experimental EtherType. 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.

Two Local Experimental EtherType values are provided to allow protocols that need more than one distinct EtherType value, or two distinct protocols, to be developed within a single administrative domain. In particular, the provision of two Local Experimental EtherTypes allows for cases where it is necessary to be

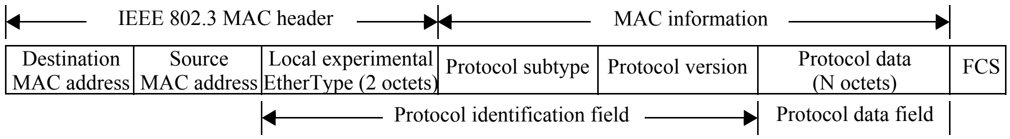

Figure 3—Example of an IEEE 802.3 frame carrying the Local Experimental EtherType

~~able to distinguish protocols or sub-protocols at the EtherType level in order to facilitate the use of filtering actions in bridges.~~

~~The combination of the Local Experimental EtherType value, the protocol subtype, and the protocol version provides the protocol identifier for the experimental protocol. The values assigned to the protocol subtype and protocol version are locally administered; their meaning cannot, therefore, be correctly interpreted outside of the administrative domain within which the value was allocated.~~

~~NOTE—The use of this format provides for a simple migration path to the use of a distinct EtherType permanently assigned to the protocol. The routine examination of proposals made to the IEEE RA for the allocation of EtherTypes includes a check that the proposed protocol format has sufficient subtype capability to withstand enhancement by the originator without the need for the assignment of a further EtherType in the future, and inclusion of the subtype and version values could be deemed to meet this requirement. While the existence of such a mechanism in the protocol specification is not in itself sufficient to ensure that an application for an EtherType succeeds, its existence is a necessary element of an acceptable protocol design. The subtyping mechanism described here offers one way that this requirement may be met.~~

9.3.5 ~~OUI Extended EtherType~~

~~The OUI Extended EtherType provides a means of protocol identification similar to that offered by the SNAP identifier described in 9.8.1. Like the SNAP identifier, the OUI Extended EtherType allows an organization to use protocol identifiers, as described in 9.8. An organization allocates protocol identifiers to its own protocols in a manner that ensures that the protocol identifier is globally unique. An illustration of a protocol identifier created with an OUI or CID is illustrated in Figure 9.~~

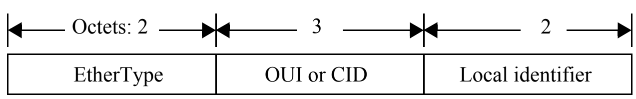

Figure 4—Protocol identifier composed of an OUI or CID

~~The EtherType field shall cointain the Vendor Specific EtherType value.~~

~~The OUI field shall contain the OUI or CID assigned to the entity.~~

~~The Local Identifier field shall contain a 2 octet numeric value assigned by the entity identified by the OUI or CID.~~

~~NOTE 1—The requirement for global uniqueness of protocol identifiers means that if protocol identifier X has been allocated for use by an organization’s protocol, then that protocol identifier can be used with either the SNAP identifier or the OUI Extended EtherType to identify that protocol. Conversely, it means that protocol identifier X cannot be used to identify any other protocol.~~

~~The OUI Extended EtherType is processed by the HLPDE. Immediately following the EtherType value is a protocol identifier, as described in 9.8, consisting of a 3-octet OUI or CID value followed by 2 octets administered by the OUI or CID assignee. The OUI or CID value provides an administrative context within which the assignee can allocate values to a 16-bit protocol subtype. This approach is closely similar to the~~

LPD-based SNAP identifier mechanism specified in 9.8; however, the OUI Extended EtherType is used instead of the LPD method.

Figure 5 shows the format of an IEEE 802.3 frame carrying the OUI Extended EtherType in the Length/Type field. The value used for the OUI component of the protocol identifier is an OUI or CID value assigned to the organization that has developed the vendor-specific protocol. The combination of the OUI Extended EtherType, the OUI or CID value, and the 16-bit value administered by the OUI or CID assignee provides a unique protocol identification field for the vendor-specific protocol. The 16-bit values are administered by the organization to which the OUI or CID has been assigned; their meaning can, therefore, be correctly interpreted only by reference to the organization that owns the OUI or CID concerned.

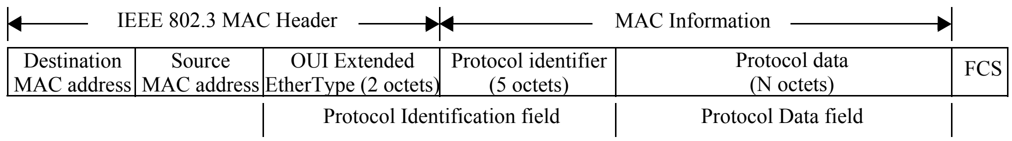

Figure 5—IEEE 802.3 frame with the OUI Extended EtherType encoded in the Length/Type field

NOTE 2—As the protocol designer is free to specify the structure of the Protocol Data field, pad octets can be included in the definition of this field, for example, for the purposes of alignment with 4-octet or 8-octet boundaries.

Good protocol development practice is to use a protocol subtype, along with a protocol version identifier in order to avoid having to allocate a new protocol identifier when a protocol is revised or enhanced. Users of the OUI Extended EtherType are, therefore, encouraged to include protocol subtype and version information in the specification of the protocol data for their protocols.

This method of protocol identification is intended to be used in products or protocols that are planned to be released into multi-vendor environments outside of the control of the administration that assigns the protocol identifier. The use of this mechanism allows such protocols to be developed and distributed without the need for a specific EtherType to be assigned for the use of each protocol.

As the OUI Extended EtherType is a normal EtherType value, it is possible to use the encoding described in 9.9 to carry its value within an LPD PDU, using a SNAP identifier with the IETF RFC 1042 [B9] OUI. Figure 6 shows the format of an IEEE 802.3 frame carrying the OUI Extended EtherType encoded in this way. In this case, it would be more appropriate to use the SNAP identifier directly (i.e., omit the RFC 1042 OUI and OUI Extended EtherType fields shown in Figure 6); however, this is a valid encoding of the OUI Extended EtherType that can result from the application of the encapsulation described in 9.9.

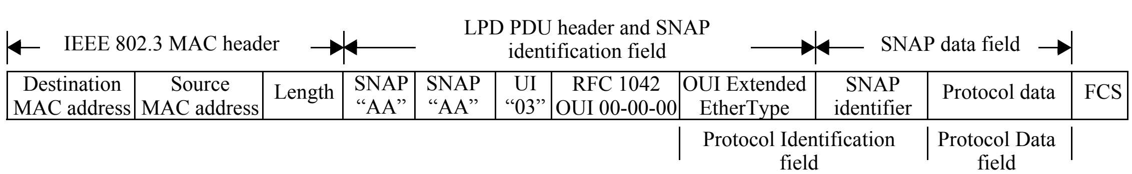

Figure 6—IEEE 802.3 frame with the OUI Extended EtherType encoded in an LPD PDU

aA 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.

9.4 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[30].

9.5 ~~OUI, CID and OUI~~O~~-36 as~~ type protocol identifiers

~~An organization that has an OUI, CID or OUI-36 assigned to it may use its OUI, CID or OUI-36 to assign universally unique protocol identifiers (potentially with additional octets as part of the identifier) to identify its own protocols, and to use in protocols described in IEEE 802 standards.~~

~~The position of the M bit (see NOTE 4 of 8.2.2) for a CID is illustrated in Figure 7 and for an OUI-36 in Figure 8. All OUI, CID and OUI-36 identifiers assigned by the IEEE RA have the M bit set to zero; values with the M bit set to one are reserved.~~

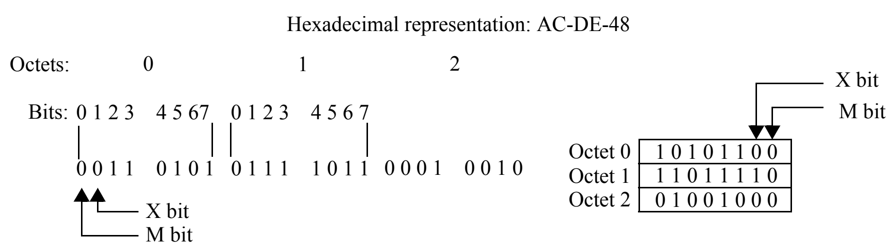

Figure 7—Format of an OUI or CID used as a protocol identifier

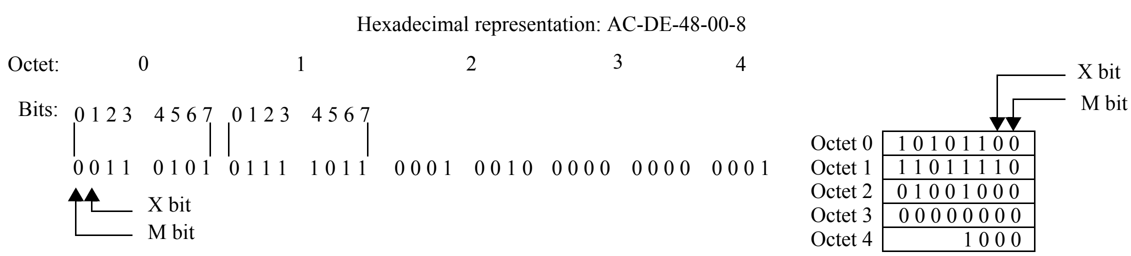

Figure 8—Format of an OUI-36 used as a protocol identifier

~~The X bit of a protocol identifier is the bit of the first octet adjacent to the M bit. All OUI and OUI-36 identifiers assigned by the IEEE RA with the X bit set to zero may be used as OUI or OUI-36 protocol~~

[30] 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

identifiers, respectively, and may also be used to create EUI-48 and EUI-64 addresses. All CIDs assigned by the IEEE RA have the X bit set to one and may be used as a protocol identifier.

9.6 LSAP encoding of EtherType protocol identifier

This subclause specifies the standard method for conveying EtherType protocol identifiers using LSAP encoding.

An EtherType protocol identifier conveyed on an IEEE 802 network using LSAP encoding shall be encapsulated in a SNAP data unit contained in an LPD PDU of type UI, as follows:

a) The Protocol Identification field of the SNAP data unit shall contain a SNAP identifier in which

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 identifiers using their OUI, OUI-36 or CID.

An O-type identifier created by the assignee of an OUI or CID is illustrated in Figure 9.

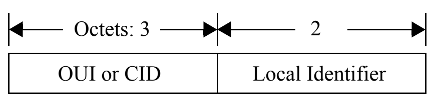

Figure 9—Protocol identifier composed of an OUI or CID

An O-type identifier created by the assignee of an OUI-36 is illustrated in Figure 9.

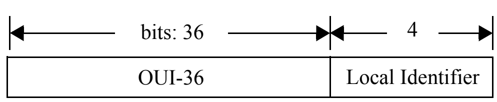

Figure 10—Protocol identifier composed of an OUI or CID

9.7 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.

9.7.1 Type 2 PIF encoding

9.7.1.1 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 11.

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

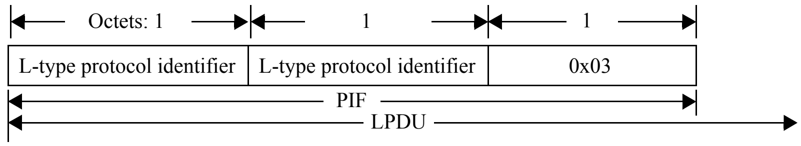

Figure 11—Type 2 PIF encoding of an L-type protocol identifier

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

9.7.1.2 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 12.

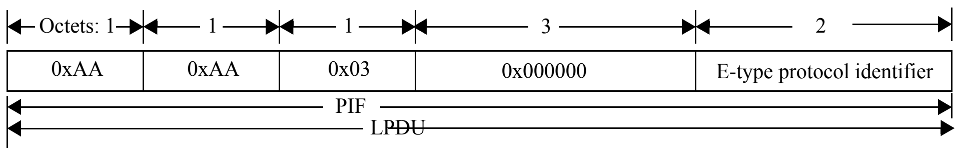

Figure 12—Type 2 PIF encoding of an E-type protocol identifier

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, 9.7.1.1.

9.7.1.3 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 13.

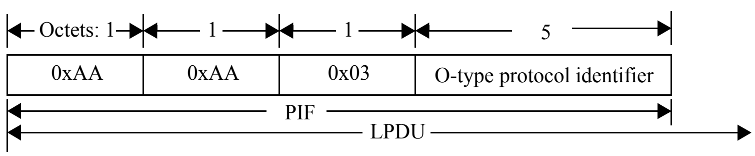

Figure 13—Type 2 PIF encoding of an O-type protocol identifier

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.

9.7.1.4 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 12 was first described in RFC 1042 and is known as the RFC 1042 form of SNAP.

9.7.2 Type 3 PIF encoding

9.7.2.1 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 14.

The ~~first three octets each take~~ PIF contains only the ~~value zero~~EtherType.

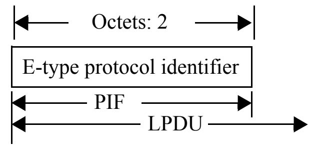

Figure 14—Type 3 PIF encoding of an E-type protocol identifier

4) ~~The two remaining octets take the values, in the same order, of the 2 octets of the EtherType.~~

b) ~~The Protocol Data field of the SNAP data unit shall contain the user data octets, in order.~~

~~NOTE—This method was originally specified in IETF RFC 1042 [B9], which contains recommendations relating to its use. Further recommendations are contained in IETF RFC 1390 [B10].~~

9.8 ~~SNAP~~

~~SNAP provides a method for multiplexing and demultiplexing of private and public protocols among multiple users of the LLC sublayer. An organization that has an OUI or CID assigned to it may use its OUI or CID to assign universal protocol identifiers to its own protocols, for use in the protocol identification field of SNAP data units.~~

9.8.1 ~~SNAP identifier~~

~~The SNAP identifier is 5 octets in length and follows the LPD header in a frame. The first 3 octets of the SNAP identifier consist of the OUI or CID. The remaining 2 octets are administered by the assignee. In the SNAP identifier, an example of which is shown in Figure 15 (see NOTE 4 of 8.2.2), the OUI or CID is contained in octets 0, 1, and 2 with octets 3 and 4 being assigned by the assignee of the OUI or CID.~~

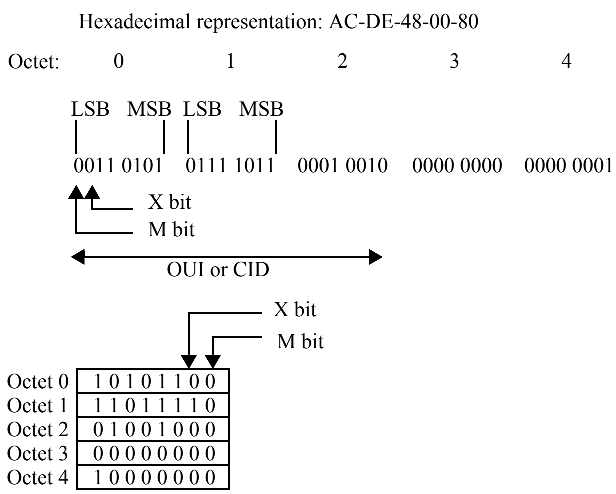

Figure 15—SNAP identifier

~~The standard representation of a SNAP identifier is as a string of 5 octets using the hexadecimal representation.~~

~~The LSB of the first octet of a SNAP identifier is referred to as the M bit. All SNAP identifiers derived from OUIs or CIDs assigned by the IEEE RA shall have the M bit set to zero; values with the M bit set to one are reserved.~~

~~SNAP identifiers may be assigned universally or locally. The X bit of a SNAP identifier is the bit of the first octet adjacent to the M bit. All universally assigned SNAP identifiers derived from OUIs have the X bit set to zero. All universally assigned SNAP identifiers derived from CIDs have the X bit set to one.~~

9.8.2 ~~SNAP address~~

~~The reserved LPD address for use with SNAP is called the SNAP address. It is specified to be the bit pattern (starting with the LSB) Z1010101, in which the symbol Z indicates that either value 0 or 1 can occur, depending on the context in which the address appears (as specified in ISO/IEC 8802-2). The two possible values have hexadecimal representation AA.~~

~~The SNAP address identifies, at each MSAP, a single LSAP for standard, public, and private protocol usage. To permit multiple public and private network layer protocols to coexist at one MSAP, each public or private protocol using SNAP shall employ a protocol identifier that enables SNAP to discriminate among these protocols.~~

9.8.3 ~~SNAP data unit format~~

~~Each SNAP data unit shall conform to the format shown in Figure 16 and shall form the entire content of the LPD information field.~~

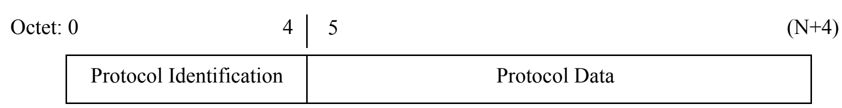

Figure 16—SNAP data unit format

~~In Figure 16, the Protocol Identification field contains a SNAP identifier whose format and administration are as described in 9.8.1. The Protocol Data field is a field whose length, format, and content are specified by a public or private protocol specification.~~

~~Figure 17 illustrates how a SNAP data unit appears in a complete MAC frame (the IEEE 802.3 MAC format is used for the example). The LPD control field (CTL) is shown for PDU type UI, Unnumbered Information, which is the most commonly used PDU type in this context; however, other information-carrying LPD PDU types may also be used with SNAP identifiers.~~

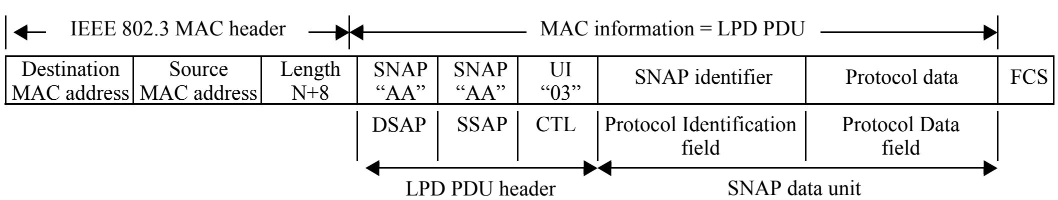

Figure 17—SNAP data unit in IEEE 802.3 MAC frame

NOTE—The EtherType is uniquely distinguishable from any possible value of the Length field, 9.8.3.1.

9.8.3.1 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 14.

The initial field is typically a Length, which takes a value no greater than 0x05DC. Since the minimum EtherType value is 0x0600, the HLPDE can distinguish this encoding with respect to the Type 3 PIF

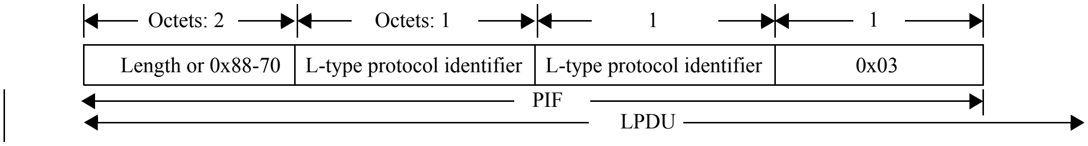

Figure 18—Type 3 PIF encoding of an L-type protocol identifier

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.

9.8.3.2 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.

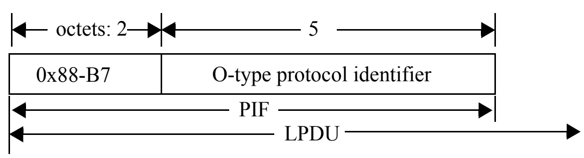

Figure 19—Type 3 PIF encoding of an O-typeprotocol identifier

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. Details of the O-Type protocol identifier are provided below.

9.8.4 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.

9.9 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.