IEEE P802.11
Wireless LANs

Proposed Protocol Identification updates to IEEE Std 802

Date: 2024-02-28

Author(s):

Name	Affiliation	Address	Phone	email
Roger Marks	EthAirNet Associates	4040 Montview Blvd Denver, CO 80207	+1 802 capable	roger@emarks.net
Mark Hamilton	Ruckus/CommScope	350 W Java Dr. Sunnyvale, CA 94089	+1-303-818-8472	mark.hamilton2152 @gmail.com

Abstract

Proposed updates to IEEE Std 802, on the issue of protocol identification, for consideration in the Working Group Letter Ballot recirculation of P802-REVc/D1.2. A clean version is presented on pages 2-9. The following pages indicate the markup to P802-REVc/D1.2.

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. 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 (PI) of a specific protocol type, associated with the protocol. Three specific types of protocol identifier are supported:

(1) 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 Authority[1]. 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--While every E-Type PI 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.

(2) 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[1].

NOTE--While every L-Type PI 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.

(3) 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.

Since each PI type is a different length, the PI type of a PI follows from its length. The types are also distinguishable by numeric value. The largest valid L-type value is 0xFE (254 decimal). Valid E-type values are within the range 0x0600 (1536 decimal) to 0xFFFF (65535 decimal). The O-type value is always greater than 0xFFFF.

Further detail regarding these three PI types is provided below.

In IEEE 802 networks, the PI 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 PIF. 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.

[1] 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.

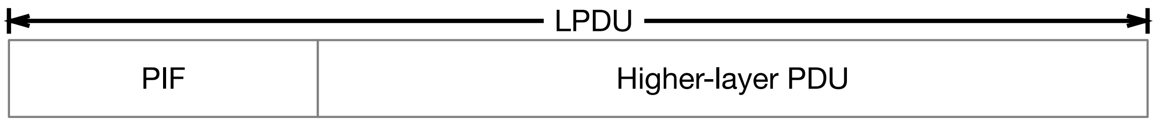


Figure PIF— LPDU including prepended PIF

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 PI type; and (3) identify the PI. 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 PI.

PIF encoding forms, each of which allows the HLDPE to parse the PIF for any of the three PI types, are specified. 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 PI 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[2]. 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 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[3]. 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

[2] 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.

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

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: standards.ieee.org/products-programs/regauth/ethertype/ and https://regauth.standards.ieee.org/standards8 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[4].
- **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 18 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 Subprotocol 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 SPIF 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.

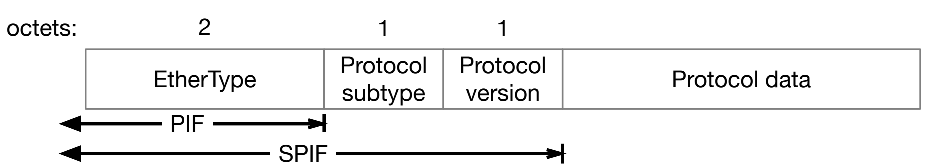


Figure SPIF—Example of subprotocol encoding

9.2.4 Local Experimental EtherTypes

The EtherType identifier space is a finite resource. Obtaining an EtherType assignment is a time-consuming application, and an assignment is not guaranteed. In order to allow users to conveniently operate E-Type protocol

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

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 6.

Within the local 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:

Table 6— Local Experimental EtherType values

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

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 SPIF.

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 42, 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, as indicated above, 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*[5].

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 PI is created under the authority of an OUI, OUI-36, or CID assignee by appending bits to the OUI, CID, or OUI-36 assignment.

An O-Type identifier created by the assignee of an OUI or CID is illustrated in Figure O24.

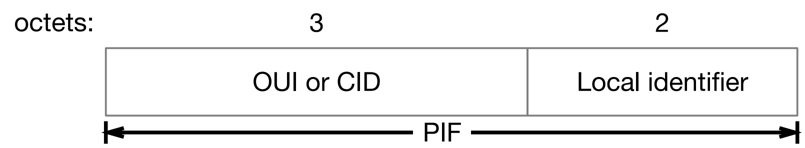


Figure O24—Protocol identifier created from an OUI or CID

Both the OUI and the CID are three octets in length. The assignee of the OUI or CID is exclusively authorized to create O-Type identifiers by appending arbitrary two-octet values to the OUI or CID.

An O-Type identifier created by the assignee of an OUI-36is illustrated in Figure O36.

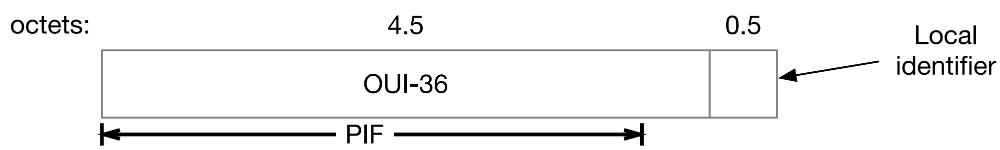


Figure O36—Protocol identifier created from an OUI-36

In this case, since the OUI-36 is 36 bits in length, the assignee of the OUI-36 is exclusively authorized to create O-Type identifiers by appending arbitrary four-bit values to the OUI-36.

9.5 PIF Encoding

The two PIF encoding forms, known as Type 2 PIF encoding and Type 3 PIF encoding, are specified herein. The encoding of PI does not change the meaning of the PI 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. Bridges may transform 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 PI in a three-octet PIF per Fig. L-L.

[5] 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

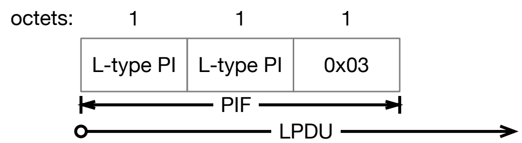


Figure L-L— Type 2 PIF encoding of an L-Type protocol identifier

Here (and in similar figures) the length of the field in bits is indicated parenthetically. Note here that the one-octet PI is duplicated in the PIF.

NOTE—The special case of L-type PI 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 PI in an eight-octet PIF per Fig. L-E.

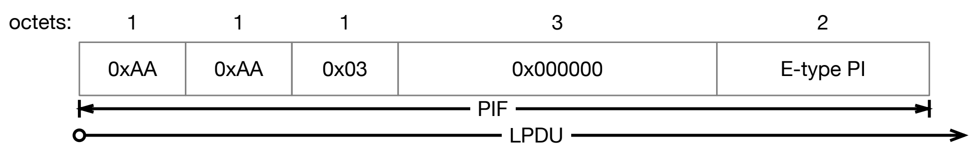


Figure L-E— Type 2 PIF encoding of an E-Type protocol identifier

Note that the one-octet value 0xAA is never assigned as the L-type PI 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 PI (See 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 PI entails embedding the PI in an eight-octet PIF per Fig. L-O.

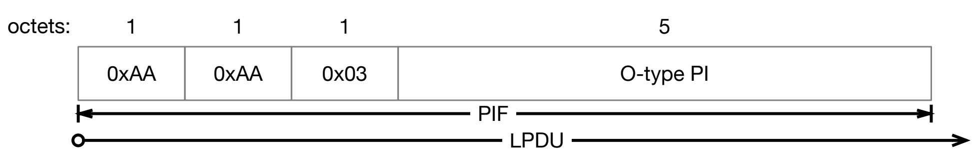


Figure L-O— Type 2 PIF encoding of an O-Type protocol identifier

The content of the O-Type PI is not arbitrary; the details of its format are provided below. In particular, the O-Type PI is never permitted to begin with 0x000000. This allows the HLPDE to distinguish the PIF with respect to the Type 2 PIF encoding of an E-Type PI.

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 L-E 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 PI in a two-octet PIF per Fig. LT-E.

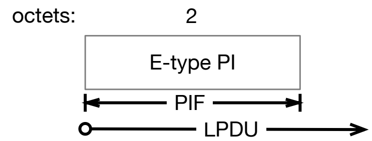


Figure LT-E— Type 3 PIF encoding of an E-Type protocol identifier

The PIF contains only the EtherType.

NOTE—The EtherType is uniquely distinguishable from any possible value of the Length field, per 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 PI in a five-octet PIF per Fig. LT-E.

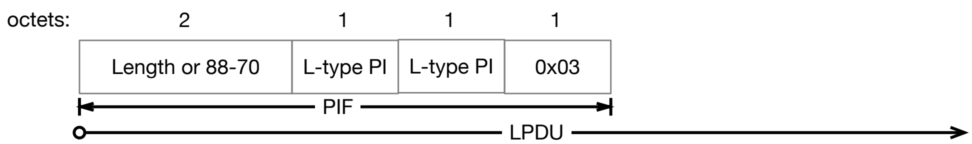


Figure LT-E— Type 3 PIF encoding of an L-Type protocol identifier

The initial field is typically a Length, which takes a value no greater than 0x05DC. Since the minimum EtherType value is 0x0600, the HLPDE can distinguish this encoding with respect to the Type 3 PIF encoding of an E-Type PI. 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 PI and does not indicate a length. This allows, for example, Type 3 PIF encoding of an L-Type PI 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 PI entails embedding the PI in a seven-octet PIF per Fig. LT-O.

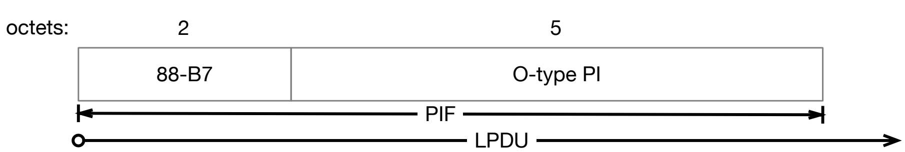


Figure LT-OE— Type 3 PIF encoding of an O-Type protocol identifier

The initial field is the OUI Extended EtherType (value 88-B7), which is never used as an E-Type PI. This allows the HLPDE to distinguish this encoding with respect to the Type 3 PIF encodings of the E-Type and L-Type PI. Details of the O-Type PI are provided below.

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 PI due to a smaller PIF: 2 octets vs. 8 octets. E-type PIs 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 PI: 7 octets vs. 8 octets. Type 2 PIF encoding is more efficient than Type 3 PIF encoding for carrying the L-type PI: 3 octets vs. 5 octets.

12 Context-dependent identifiers

An 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.

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

9.1 Introduction

~~This clause describes methods that allow~~A key function of the LLC sublayer is to support the multiplexing and demultiplexing of 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~~

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 to~~the transmitting and receiving HLPDEs of the ~~receiving~~ LLC sublayer ~~determine which~~ cooperate to identif~~y~~ies the network layer protocol ~~is~~ to be invoked ~~to process~~~~and passes to it the 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 (PI)~~,~~ of a specific protocol type, associated with the protocol. Three specific types of protocol identifier are supported:

(1) 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 Authority[1]. 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 ~~a~~ protocols.

NOTE--While every E-Type PI 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.

(2) 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[1].

NOTE--While every L-Type PI 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.

(3) 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.

Since each PI type is a different length, the PI type of a PI follows from its length. The types are also distinguishable by numeric value. The largest valid L-type value is 0xFE (254 decimal). Valid E-type values are within the range 0x0600 (1536 decimal) to 0xFFFF (65535 decimal). The O-type value is always greater than 0xFFFF.

Further detail regarding these three PI types is provided below.

Commented [RBM1]: Deleted "(often referred to as a "tag")" to avoid controversy.

Commented [RBM2]: I deleted your final comment (and the words that it proposed to delete)), and I added this paragraph instead.

[1] 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 PI 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 PIF. 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.

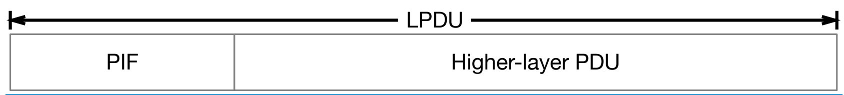


Figure PIF— LPDU including prepended PIF

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 PI type; and (3) identify the PI. 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 PI.

Commented [HM3]: I still think we should delete this, if we're not (now) going to get into how the PI type has anything to do with protocol identification.

PIF encoding forms, each of which allows the HLDPE to parse the PIF for any of the three PI types, are specified. 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 PI 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[2]. 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 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.

Commented [RBM4]: I deleted the VLAN EtherType 81-00 from the examples here, in order to avoid controversy about tags. We really ought to solve the tag problem, but bypassing it for now seems prudent.

[2] 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.

9.2.2 Public EtherType assignments subset

The IEEE Registration Authority (RA) provides a public listing of EtherType assignments[3]. 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: standards.ieee.org/products-programs/regauth/ethertype/ and https://regauth.standards.ieee.org/standards8 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[4].
- **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 18 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 Subprotocol 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 SPIF 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.

---

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

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

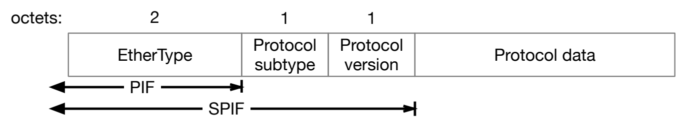


Figure SPIF—Example of subprotocol encoding

Commented [RBM5]: Adjusted arrows in figure to address your emailed comment.

9.2.4 Local Experimental EtherTypes

The EtherType identifier space is a finite resource. Obtaining an EtherType assignment is a time-consuming application, and an assignment is not guaranteed. 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 6.

Within the local 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:

Table 6— Local Experimental EtherType values

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

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 SPIF.

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 42, 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, as indicated above, 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[*5].

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 PI is created under the authority of an OUI, OUI-36, or CID assignee by appending bits to the OUI, CID, or OUI-36 assignment.

An O-Type identifier created by the assignee of an OUI or CID is illustrated in Figure O24.

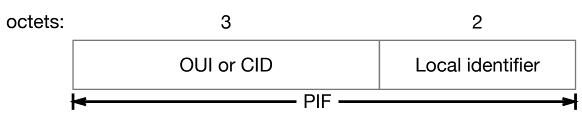


Figure O24—Protocol identifier created from an OUI or CID

Both the OUI and the CID are three octets in length. The assignee of the OUI or CID is exclusively authorized to create O-Type identifiers by appending arbitrary two-octet values to the OUI or CID.

An O-Type identifier created by the assignee of an OUI-36is illustrated in Figure O36.

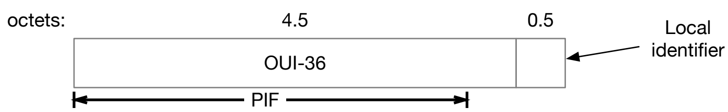


Figure O36—Protocol identifier created from an OUI-36

In this case, since the OUI-36 is 36 bits in length, the assignee of the OUI-36 is exclusively authorized to create O-Type identifiers by appending arbitrary four-bit values to the OUI-36.

[5] 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

9.5 PIF Encoding

The two PIF encoding forms, known as Type 2 PIF encoding and Type 3 PIF encoding, are specified herein. The encoding of PI does not change the meaning of the PI 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. Bridges may transform 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 PI in a three-octet PIF per Fig. L-L.

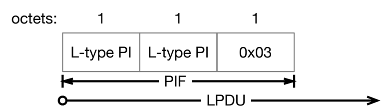


Figure L-L— Type 2 PIF encoding of an L-Type protocol identifier

Here (and in similar figures) the length of the field in bits is indicated parenthetically. Note here that the one-octet PI is duplicated in the PIF.

NOTE—The special case of L-type PI 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 PI in an eight-octet PIF per Fig. L-E.

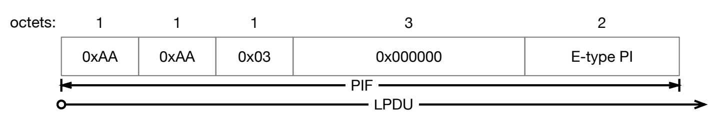


Figure L-E— Type 2 PIF encoding of an E-Type protocol identifier

Note that the one-octet value 0xAA is never assigned as the L-type PI 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 PI (See 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 PI entails embedding the PI in an eight-octet PIF per Fig. L-O.

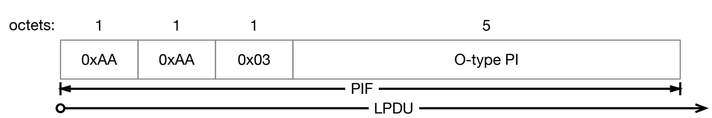


Figure L-O— Type 2 PIF encoding of an O-Type protocol identifier

The content of the O-Type PI is not arbitrary; the details of its format are provided below. In particular, the O-Type PI is never permitted to begin with 0x000000. This allows the HLPDE to distinguish the PIF with respect to the Type 2 PIF encoding of an E-Type PI.

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 L-E 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 PI in a two-octet PIF per Fig. LT-E.

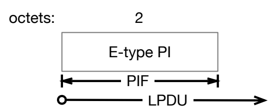


Figure LT-E— Type 3 PIF encoding of an E-Type protocol identifier

The PIF contains only the EtherType.

NOTE—The EtherType ~~must be~~is uniquely distinguishable from any possible value of the Length field, per 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 PI in a five-octet PIF per Fig. LT-E.

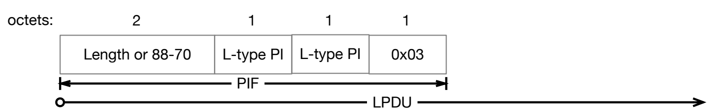


Figure LT-E— Type 3 PIF encoding of an L-Type protocol identifier

The initial field is typically a Length, which takes a value no greater than 0x05DC. Since the minimum EtherType value is 0x0600, the HLPDE can distinguish this encoding with respect to the Type 3 PIF encoding of an E-Type PI. 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 PI and does not indicate a length. This allows, for example, Type 3 PIF encoding of an L-Type PI 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 PI entails embedding the PI in a seven-octet PIF per Fig. LT-O.

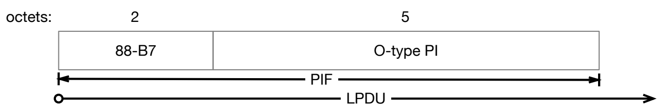


Figure LT-OE— Type 3 PIF encoding of an O-Type protocol identifier

The initial field is the OUI Extended EtherType (value 88-B7), which is never used as an E-Type PI. This allows the HLPDE to distinguish this encoding with respect to the Type 3 PIF encodings of the E-Type and L-Type PI. Details of the O-Type PI are provided below.

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 PI due to a smaller PIF: 2 octets vs. 8 octets. E-type PIs 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 PI: 7 octets vs. 8 octets. Type 2 PIF encoding is more efficient than Type 3 PIF encoding for carrying the L-type PI: 3 octets vs. 5 octets.

~~Various network and higher layer protocols have been assigned reserved LPD addresses or EtherTypes, as recorded by the IEEE RA[6]. 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[7], excluding amendments to existing standards.~~

~~9.2 EtherTypes~~

~~9.2.1 Format, function, and administration~~

~~EtherType protocol identification values are assigned by the IEEE RA[8] 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, interpreted as a 16-bit numeric~~

~~[6] More information can be found at https://standards.ieee.org/products-programs/regauth/.~~

~~[7] IEEE Std 802.2™-1989 (reaffirmed 2003) was administratively withdrawn as an IEEE standard on January 2011 in deference to the stabilized standard ISO/IEC 8802-2:1998 where the same material continues to be available.~~

~~[8] 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.~~

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.

Examples of 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 9.2.4 is recommended.

9.2.2 Public EtherType assignments subset

The IEEE Registration Authority (RA) provides a public listing of EtherType assignments[9]. 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 2 developers of such standards, Annex F identifies some well-known EtherTypes and the protocols they 3 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: standards.ieee.org/products-programs/regauth/ethertype/ and https://regauth.standards.ieee.org/standards8 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[10].
- **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 18 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 EtherTypes for prototype and vendor-specific protocol development

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 9.2.4, assigned, as the name implies, for experimental use within a local area

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

b) A single EtherType value, known as the OUI Extended EtherType, as specified in 9.2.5, assigned for the identification of vendor-specific protocols

The values of the Local Experimental EtherTypes and the OUI Extended EtherType are listed in Table 6.

9.2.4 Local Experimental EtherTypes

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:

Table 6—Assigned EtherType values

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

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 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.

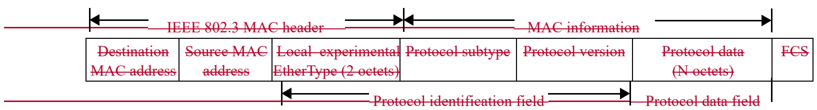


Figure 14—Example usage of Local Experimental EtherType

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 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 [2] provides the protocol identifier for the experimental protocol. The values assigned to the protocol subtype [3] 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.

9.2.5 OUI Extended EtherType

The OUI Extended EtherType provides a means of protocol identification similar to that offered by the SNAP identifier described in 9.5.1. Like the SNAP identifier, the OUI Extended EtherType allows an organization to use protocol identifiers, as described in 9.5. 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 15.

Octets: 2	3	2
EtherType	OUI or CID	Local identifier

Figure 15—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 9.5, 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 9.5; however, the OUI Extended EtherType is used instead of the LPD method.

Figure 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.

IEEE 802.3 MAC Header			MAC Information		
Destination MAC address	Source MAC address	OUI Extended EtherType (2 octets)	Protocol identifier (5 octets)	Protocol data (N octets)	FCS
		Protocol Identification field		Protocol Data field	

Figure 16—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 2 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 9.4 to carry its value within an LPD PDU, using a SNAP identifier with the IETF RFC 1042 [B9] OUI. Figure 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 17); however, this is a valid encoding of the OUI Extended EtherType that can result from the application of the encapsulation described in 9.4.

IEEE 802.3 MAC header			LPD PDU header and SNAP identification field						SNAP data field	
Destination MAC address	Source MAC address	Length	SNAP "AA"	SNAP "AA"	UI "03"	RFC 1042 OUI 00-00-00	OUI Extended EtherType	SNAP identifier	Protocol data	FCS
							Protocol Identification		Protocol Data fieldfield	

Figure 17—IEEE 802.3 frame with the OUI Extended EtherType encoded in an LPD PDU

9.3 OUI, CID and OUI-36 as 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 of 8.2.2) for a CID is illustrated in Figure and for an OUI-36 in Figure 19. 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.

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 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.

Hexadecimal representation: AC-DE-48

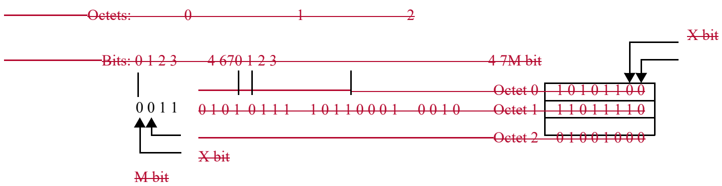


Figure 18—Format of an OUI or CID used as a protocol identifier

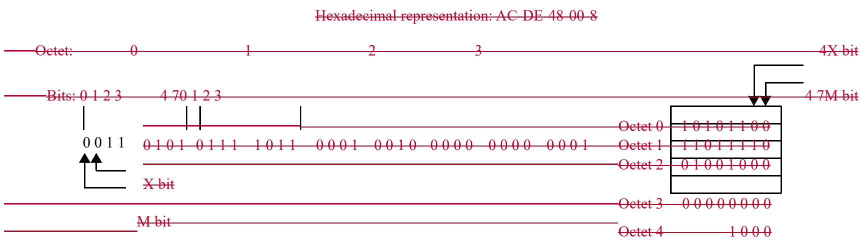


Figure 19—Format of an OUI-36 used as a protocol identifier

9.4 LSAP encoding of EtherType protocol identifier

This subclause specifies the standard method for conveying EtherType protocol identifiers using LSAP[3] 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
   1) The first three octets each take the value zero.
   2) 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].

9.5 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.

9.5.1 SNAP identifier

The SNAP identifier is 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 (see NOTE of 8.2.2), the OUI or CID is contained in octets 0, 1, and 2 with octets 3 and being assigned by the assignee of the OUI or CID.

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

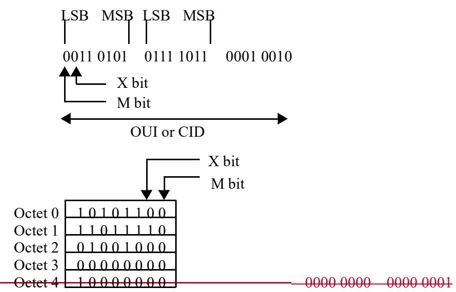


Figure 20—SNAP identifier

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.

9.5.2 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.

9.5.3 SNAP data unit format

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

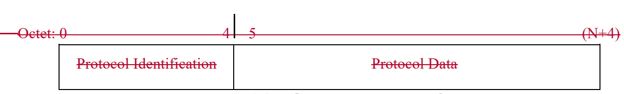
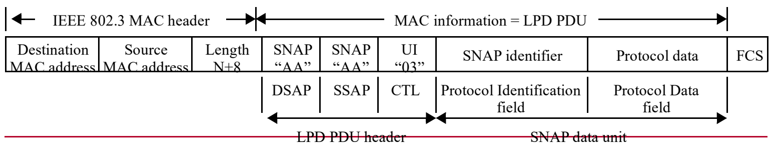


Figure 21—SNAP data unit format

In Figure 21, the Protocol Identification field contains a SNAP identifier whose format and administration are as described in 9.5.1. The Protocol Data field is a field whose length, format, and content are specified by a public or private protocol specification.

Figure 22 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 22—SNAP data unit in IEEE 802.3 MAC frame~~

~~9.6~~ 12 Context-dependent identifiers

~~The~~ An 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.