

# **HIGH-PERFORMANCE PARALLEL INTERFACE - Encapsulation of ISO 8802-2 (IEEE Std 802.2) Logical Link Control Protocol Data Units (802.2 Link Encapsulation)**

## **(HIPPI-LE)**

maintenance copy of  
American National Standard  
for Information Systems  
X3.218-1993

August 6, 1993

Secretariat:

Computer & Business Equipment Manufacturers Association

ABSTRACT: The described Specification provides an encapsulation of ISO 8802-2 (IEEE Std 802.2) Logical Link Control Protocol Data Units over the High-Performance Parallel Interface (HIPPI).

**NOTE:**

*This document is an X3T11 maintenance copy of American National Standard X3.218-1993. A list of the proposed changes since approval of the standard is included. Some or all of these changes may be included in a future erratum, amendment, interpretation, or revision to the standard. The committee lists included in the approved standard, and some other boilerplate, have been omitted in this draft. For current information on the status of this document contact the individuals shown below.*

**POINTS OF CONTACT:**

Roger Cummings (X3T11 Chairman)  
Storage Technology Corporation  
2270 South 88th Street  
Louisville, CO 80028-0268  
(303) 661-6357, FAX (303) 684-8196  
E-mail: Roger\_Cummings@Stortek.com

Carl Zeitler (X3T11 Vice-Chairman)  
IBM Corporation, MS 9440  
11400 Burnet Road  
Austin, TX 78758  
(512) 838-1797, FAX (512) 838-3822  
E-mail: zeitler@ausvm6.vnet.ibm.com

Don Tolmie (HIPPI-LE Technical Editor)  
Los Alamos National Laboratory  
C-5, MS-B255  
Los Alamos, NM 87545  
(505) 667-5502, FAX (505) 665-7793  
E-mail: det@lanl.gov

**Proposed changes to ANSI X3.218-1993**– (and which have been incorporated in this draft)

Proposed technical changes are preceded with (T), proposed editorial changes are preceded with (E).

Rev 3.4, August 6, 1993 version - The following changes were the result of comments from the ISO editor.

1. (E) Change the Introduction by removing the sentence "The information in the document is organized as follows:" and the seven bullets that follow.
2. (E) Change the last sentence of 4.5, page 4, by starting the word "Clause" with a lowercase "c".
3. (E) Change the second word of 7.2, page 8, from "section" to "subclause".
4. (E) Change the subitem numbering of 7.2.2, page 10, from "1." to "a)", "2." to "b)", and "3." to "c)".

## Contents

	Page
Foreword .....	ii
Introduction .....	iii
<b>1</b> Scope .....	1
<b>2</b> Normative references .....	1
<b>3</b> Definitions and conventions .....	1
<b>3.1</b> Definitions .....	1
<b>3.2</b> Editorial conventions .....	1
<b>4</b> HIPPI-LE services to 802.2 LLC .....	2
<b>4.1</b> Service primitives types .....	2
<b>4.2</b> Sequences of primitives .....	2
<b>4.3</b> HIPPI-LE service primitive summary .....	2
<b>4.4</b> HIPPI-LE transfer service primitives .....	3
<b>4.5</b> HIPPI-LE address resolution primitives .....	4
<b>4.6</b> HIPPI-LE station management service primitives .....	5
<b>5</b> HIPPI-FP data framing services used by HIPPI-LE .....	6
<b>6</b> HIPPI-LE data formats .....	6
<b>6.1</b> HIPPI-LE PDU format .....	6
<b>6.2</b> IEEE 48-bit address format .....	6
<b>7</b> HIPPI Address Resolution .....	8
<b>7.1</b> Overview .....	8
<b>7.2</b> Address Resolution in Detail .....	8

## Annexes

<b>A</b> Example HIPPI packets containing HIPPI-LE PDUs .....	10
<b>B</b> Bibliography .....	14
Alphabetical index .....	15

## Figures

<b>1</b> Protocol hierarchy .....	v
<b>2</b> HIPPI-LE service relationship .....	2
<b>3</b> Data transfer service primitives .....	3
<b>4</b> HIPPI-LE Address resolution service primitives .....	4
<b>5</b> Station management service primitives .....	5
<b>6</b> HIPPI-LE header format .....	7
<b>7</b> Mapping of 6-byte IEEE universal LAN MAC address to two 32-bit HIPPI words in LE_Header. ....	7

## **Foreword** (This Foreword is not part of American National Standard X3.218-1993)

This High-Performance Parallel Interface, Encapsulation of ISO 8802-2 (IEEE STD 802.2) Logical Link Control Protocol Data Units (HIPPI-LE) standard defines the HIPPI-LE Protocol Data Unit (PDU) format and interface for transporting (via HIPPI) PDUs conforming to ISO 8802-2 (IEEE Std 802.2) Logical Link Control.

This standard was developed by Task Group X3T9.3 of Accredited Standards Committee X3 during 1990. The standards approval process started in 1991.

This document includes an annex which is informative and is not considered part of the standard.

Requests for interpretation, suggestions for improvement or addenda, or defect reports are welcome. They should be sent to the X3 Secretariat, Computer and Business Equipment Manufacturers Association, 1250 Eye Street, NW, Suite 200, Washington, DC 20005.

This standard was processed and approved for submittal to ANSI by Accredited Standards Committee on Information Processing Systems, X3. Committee approval of the standard does not necessarily imply that all committee members voted for approval. At the time it approved this standard, the X3 Committee had the following members:

*(List of X3 Committee members included in published standard)*

Subcommittee X3T9 on Computer Input/Output Interfaces, which reviewed this standard, had the following members:

*(List of X3T9 committee members included in published standard)*

Task Group X3T9.3 on Device Level Interfaces, which developed this standard, had the following participants:

*(List of X3T9.3 committee participants included in published standard)*

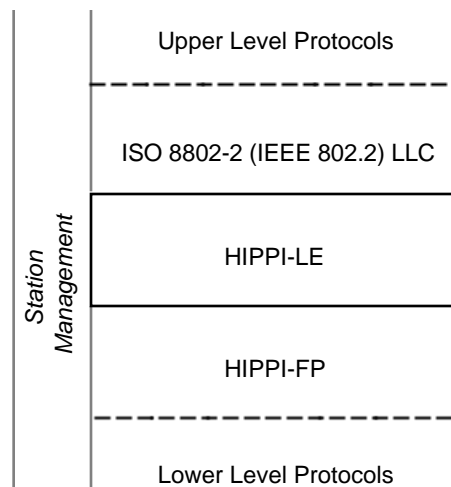
## Introduction

This High-Performance Parallel Interface, Encapsulation of ISO 8802-2 (IEEE Std 802.2) Logical Link Control Protocol Data Units (HIPPI-LE) standard defines the HIPPI-LE Protocol Data Unit (PDU) format and interface for transporting ISO 8802-2 (IEEE STD 802.2) Logical Link Control PDUs over HIPPI.

Characteristics of this HIPPI-LE include:

- Encapsulation of arbitrary Protocol Data Units that conform to ISO 8802-2 (IEEE Std 802.2) Logical Link Control;
- Provision for 48-bit source and destination addresses conforming to IEEE 802.1A;
- Provision for eight forwarding classes to distinguish, for example, among ordinary data PDUs, PDUs for services that require bandwidth guarantees such as packet video, etc.

Figure 1 shows the relationship of this American National Standard (in the solid rectangle) with the other entities shown.



**Figure 1 – Protocol hierarchy**



American National Standard  
for Information Systems –

High-Performance Parallel Interface –  
Encapsulation of ISO 8802-2 (IEEE Std 802.2) Logical  
Link Control Protocol Data Units (HIPPI-LE)

## 1 Scope

This American National Standard provides a common method for encapsulating ISO 8802-2 (IEEE Std 802.2) Logical Link Control Protocol Data Units (PDU) on HIPPI.

## 2 Normative references

The following standards contain provisions which, through reference in this text, constitute provisions of this American National standard. At the time of publication, the editions indicated were valid. All standards are subject to revision, and parties to agreements based on this American National standard are encouraged to investigate the possibility of applying the most recent editions of the standards listed below.

ANSI X3.210-1992, *High-performance parallel interface, –framing protocol (HIPPI-FP)*.

ANSI X3.222-1993, *High-performance parallel interface, – physical switch control (HIPPI-SC)*.

ANSI/IEEE Std 802-1990, *Local and metropolitan area networks: Overview and architecture*.

ISO 8802-2:1989 (ANSI/IEEE 802.2-1989) *Information processing systems – Local area networks – Part 2: Logical link control*.

## 3 Definitions and conventions

### 3.1 Definitions

For the purposes of this standard, the following definitions apply.

**3.1.1 802.2 LLC PDU:** A PDU formatted according to ISO 8802-2 (IEEE Std 802.2) Logical Link Control.

**3.1.2 Address Resolution:** The general process of discovering the low-level (or switch) address associated

with a known higher level address for a particular remote entity. This involves sending a query message, containing the known higher level address, to a well-known switch address. The requester receives a reply containing the associated switch address.

**3.1.3 Address Resolution Protocol (ARP):** the particular protocol for address resolution used by the Internet.

**3.1.4 HIPPI-LE PDU:** A PDU that includes a HIPPI-LE header and an 802.2 LLC PDU and is formatted according to clause 6.

**3.1.5 optional:** A term referring to features that are not required by the standard. However, if any optional feature defined by the standard is implemented, it shall be implemented according to the standard.

**3.1.6 protocol data unit (PDU):** A formatted data set that is a logical unit of interchange between entities.

**3.1.7 service data unit (SDU):** A formatted data set that includes a PDU and optional information for the convenience of an implementer.

**3.1.8 station management (SMT):** The supervisory entity that monitors and controls HIPPI.

**3.1.9 Switch Address:** A value in the switch hardware domain which selects the route or end point of a HIPPI connection. The range and format depend on switch hardware. HIPPI-SC specifies two formats: Logical Address (12 bits) and Source Route (24 bits).

### 3.2 Editorial conventions

In this standard, certain terms that are proper names of service primitives, or similar terms are printed in uppercase to avoid possible confusion with other uses of the same words (e.g., TRANSFER). Any lowercase uses of these words have the normal technical English meaning.

The first letter of each word of service parameters, service primitive types, and similar terms is capitalized (e.g., Destination\_Address, Status, Request). Any lowercase uses of these words have the normal technical English meaning.

#### 4 HIPPI-LE services to 802.2 LLC

This clause specifies the services provided by HIPPI-LE. The intent is to allow 802.2 LLC to operate correctly with HIPPI-LE. How many of the services described in this clause are chosen for a given implementation is up to the implementer; however, the set of HIPPI-LE services supplied shall be sufficient to satisfy the higher level protocol(s) being used. Service parameters are also optional and additional parameters not shown may be used as necessary. The services as defined in this standard do not imply any particular implementation, or any interface.

In this standard the 802.2 LLC and station management protocol (SMT) are service users, and the HIPPI-LE is the service provider to the 802.2 LLC and SMT. The primitives to 802.2 LLC are prefixed with LE\_ and the SMT primitives are prefixed with LESM\_.

The HIPPI-LE is also the service user of HIPPI-FP services, which are prefixed with FP\_.

Figure 2 shows the relationship of the services provided by and used by HIPPI-LE.

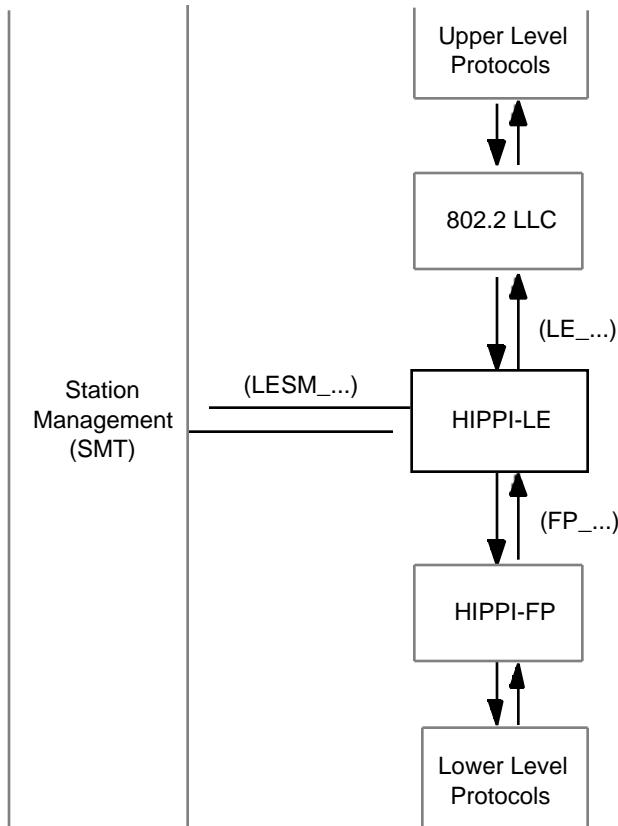


Figure 2 – HIPPI-LE service relationship

#### 4.1 Service primitives types

HIPPI service primitives are of four types:

- *Request primitives* are issued by a service user to initiate a service from the service provider;
- *Confirm primitives* are issued by the service provider to acknowledge a Request;
- *Indicate primitives* are issued by the service provider to notify the service user of a local event. This primitive is similar in nature to an unsolicited interrupt. Note that the local event may have been caused by a service Request;
- *Response primitives* are issued by a service user to acknowledge an Indicate.

#### 4.2 Sequences of primitives

The order of execution of service primitives is not arbitrary. Logical and time sequence relationships exist for all described service primitives. Time sequence diagrams are used to illustrate a valid sequence. Other valid sequences may exist. The sequence of events between peer users across the user/provider interface is illustrated. In the time sequence diagrams, the HIPPI-LE users are depicted on either side of the vertical bars while the service provider is in the center.

#### 4.3 HIPPI-LE service primitive summary

802.2 LLC Data Transfer:

LE\_TRANSFER.Request ( Destination\_Address, LE\_SDU)

LE\_TRANSFER.Confirm (Transmission\_Status)

LE\_TRANSFER.Indicate ( Destination\_Address, Source\_Address, LE\_SDU, Reception\_Status)

LE\_TRANSFER.Response

Address Resolution Protocol:

LE\_ADDRESS\_RESOLUTION.Request ( Destination\_Address, Destination\_ULA, Destination\_Switch\_Address, Source\_ULA, Source\_Switch\_Address, Message\_Type, LE\_SDU)

LE\_ADDRESS\_RESOLUTION.Confirm (Transmission\_Status)



```
LE_ADDRESS_RESOLUTION.Indicate (
    Destination_ULA,
    Destination_Switch_Address,
    Source_ULA,
    Source_Switch_Address,
    Message_Type,
    LE_SDU,
    Reception_Status)
```

```
LE_ADDRESS_RESOLUTION.Response
```

Local Station Management:

```
LESM_STATUS_CTRL.Request (Control)
```

```
LESM_STATUS_CTRL.Confirm (Status)
```

```
LESM_STATUS_CTRL.Indicate (Status)
```

```
LESM_STATUS_CTRL.Response
```

#### 4.4 HIPPI-LE transfer service primitives

These primitives, as illustrated in figure 3, are used to transfer an 802.2 LLC PDU from source 802.2 LLC to destination 802.2 LLC.

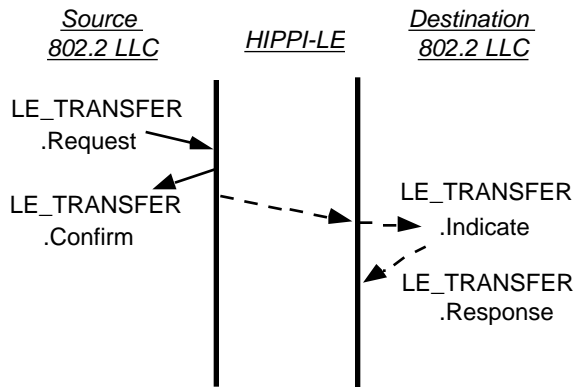


Figure 3 – Data transfer service primitives

##### 4.4.1 LE\_TRANSFER.Request

This primitive is issued by the source 802.2 LLC to request a data transfer to the destination 802.2 LLC. The HIPPI-LE PDU format is defined in 6.1.

```
LE_TRANSFER.Request (
    Destination_Address,
    LE_SDU)
```

Destination\_Address is the information necessary to forward the HIPPI-LE PDU to its desired destination.

LE\_SDU is the service data unit which includes the 802.2 LLC PDU that is to be sent.

NOTE – The Requested\_Service\_Class parameter of 802.2 LLC is not used by HIPPI-LE.

Issued – The source 802.2 LLC issues this primitive to request a transfer of an 802.2 LLC PDU to a destination 802.2 LLC.

Effect – The Source HIPPI-LE shall accept the 802.2 LLC PDU for transmission. The HIPPI-LE shall build an LE\_Header and D2\_Area, as specified in 6.1, and invoke FP\_TRANSFER.Request to send the HIPPI-LE PDU to the destination.

NOTE – Significant performance loss may occur if the D2\_Offset in the HIPPI-FP header is non zero, although receivers must be prepared to accept any D2\_Offset allowed by HIPPI-FP.

##### 4.4.2 LE\_TRANSFER.Confirm

This primitive acknowledges the LE\_TRANSFER.Request from the Source ULP.

```
LE_TRANSFER.Confirm (Transmission_Status)
```

Transmission\_Status is used to pass status information back to the local requesting LLC entity. The content of the Transmission\_Status is implementation dependent.

NOTE – The Provided\_Service\_Class parameter of 802.2 LLC is not used by HIPPI-LE.

Issued – The HIPPI-LE issues this primitive to the Source 802.2 LLC to acknowledge the LE\_TRANSFER.Request.

Effect – Unspecified.

##### 4.4.3 LE\_TRANSFER.Indicate

This primitive indicates to the 802.2 LLC that a HIPPI-LE PDU addressed to the HIPPI-LE entity at this particular destination has been received.

```
LE_TRANSFER.Indicate (
    Destination_Address,
    Source_Address,
    LE_SDU,
    Reception_Status)
```

Destination\_Address includes the IEEE 48-bit address as contained in the Destination\_IEEE\_Address field of the received HIPPI-LE PDU.

Source\_Address includes the IEEE 48-bit address as contained in the Source\_IEEE\_Address field of the received HIPPI-LE PDU.

LE\_SDU is the service data unit that includes the encapsulated 802.2 LLC PDU as contained in the D2\_Data\_Set of the received HIPPI-LE PDU.

Reception\_Status shall denote whether the received HIPPI-LE PDU was received with errors. Reception\_Status may include other information.

NOTE – The Requested\_Service\_Class parameter of 802.2 LLC is not used by HIPPI-LE.

Issued – The destination HIPPI-LE issues this primitive to the destination 802.2 LLC when a PDU for the 802.2 LLC has been received.

Effect – Unspecified.

NOTE – This standard does not protect against errors that may be introduced by intermediate devices that interconnect HIPPI-PHs, e.g., re-ordered HIPPI bursts.

#### 4.4.4 LE\_TRANSFER.Response

This primitive acknowledges the LE\_TRANSFER.Indicate from the Destination HIPPI-LE.

LE\_TRANSFER.Response

Issued – The destination 802.2 LLC issues this primitive to acknowledge receipt of the LE\_TRANSFER.Indicate.

Effect – Unspecified.

### 4.5 HIPPI-LE address resolution primitives

These primitives, as illustrated in figure 4, are used for dynamic discovery of a remote destination's hardware (switch) address. Two HIPPI-LE message types, AR\_Request and AR\_Response, accomplish the task of discovering a remote destination's switch address given its Universal LAN MAC Address (ULA). Support of these primitives is an implementation option. See clause 7 for a discussion of HIPPI address resolution.

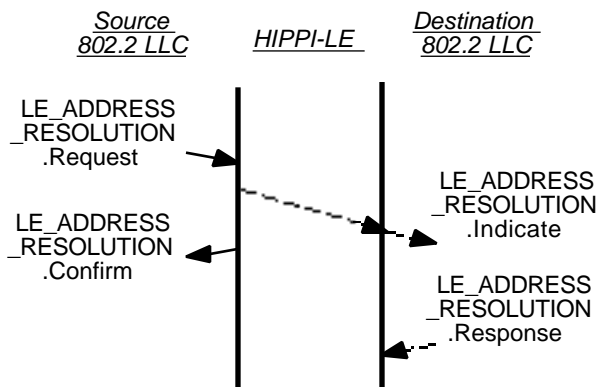


Figure 4 - HIPPI-LE Address resolution service primitives

#### 4.5.1 LE\_ADDRESS\_RESOLUTION.Request

This primitive is issued by the source 802.2 LLC to request an address resolution PDU (with optional data) to be sent. The HIPPI-LE PDU format is defined in 6.1.

LE\_ADDRESS\_RESOLUTION.Request(  
 Destination\_Address,  
 Destination\_ULA,  
 Destination\_Switch\_Address,  
 Source\_ULA,  
 Source\_Switch\_Address,  
 Message\_Type,  
 LE\_SDU)

Destination\_Address is the information necessary to forward the HIPPI-LE PDU to the desired destination. The HIPPI-LE layer forms the Connection Control Information (CCI) for the HIPPI-FP layer from this.

Destination\_ULA supplies the information to form the Destination\_IEEE\_Address field in the outgoing HIPPI-LE PDU.

Destination\_Switch\_Address supplies the information to form the Destination\_Switch\_Address field in the outgoing HIPPI-LE PDU.

Source\_ULA supplies the information to form the Source\_IEEE\_Address field in the outgoing HIPPI-LE PDU.

Source\_Switch\_Address supplies the information to form the Source\_Switch\_Address field in the outgoing HIPPI-LE PDU.

Message\_Type specifies information for the Message\_Type field of the header. (see 6.1.1)

LE\_SDU is the optional IEEE 802.2 LLC PDU to be sent.

Issued - The source 802.2 LLC issues this primitive to request the transfer of a HIPPI-LE Address Resolution Request or Response PDU, with or without an 802.2 LLC PDU, to a destination 802.2 LLC.

Effect - The Source HIPPI-LE shall accept the 802.2 LLC PDU for transmission. The HIPPI-LE shall build an LE\_Header and optional D2\_Area as specified in 6.1, and invoke an FP\_TRANSFER.Request to send the HIPPI-LE PDU to the destination.

#### 4.5.2 LE\_ADDRESS\_RESOLUTION.Confirm

This primitive acknowledges the LE\_ADDRESS\_RESOLUTION.Request from the Source ULP.

LE\_ADDRESS\_RESOLUTION.Confirm (Transmission\_Status)

Transmission\_Status is used to pass status information back to the local requesting LLC entity. The content of the Transmission\_Status is implementation dependent.

Issued - The HIPPI-LE issues this primitive to the Source 802.2 LLC to acknowledge the LE\_ADDRESS\_RESOLUTION.Request.

Effect - Unspecified.

**4.5.3 LE\_ADDRESS\_RESOLUTION.Indicate**

This primitive indicates to the destination 802.2 LLC that an address resolution PDU with optional data was received.

```
LE_ADDRESS_RESOLUTION.Indicate (
    Destination_ULA,
    Destination_Switch_Address,
    Source_ULA,
    Source_Switch_Address,
    Message_Type,
    LE_SDU,
    Reception_Status)
```

Destination\_ULA indicates the Destination\_IEEE\_Address from the received HIPPI-LE PDU.

Destination\_Switch\_Address indicates the Destination\_Switch\_Address from the received HIPPI-LE PDU.

Source\_ULA indicates the Source\_IEEE\_Address from the received HIPPI-LE PDU.

Source\_Switch\_Address indicates the Source\_Switch\_Address from the received HIPPI-LE PDU.

Message\_Type indicates the kind of message. (see 6.1.1)

LE\_SDU is the received IEEE 802.2 LLC PDU, if present.

Reception\_Status indicates whether the received HIPPI-LE PDU was received with errors. Reception\_Status may include other information.

Issued - The destination HIPPI-LE issues this primitive to the destination 802.2 LLC when a HIPPI-LE Address Resolution PDU for the 802.2 LLC has been received.

Effect - Unspecified.

**4.5.4 LE\_ADDRESS\_RESOLUTION.Response**

This primitive acknowledges the LE\_TRANSFER.Indicate from the Destination HIPPI-LE.

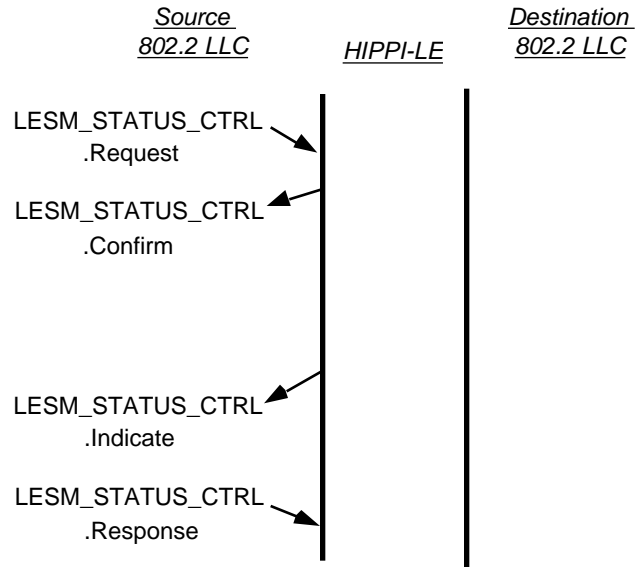
```
LE_ADDRESS_RESOLUTION.Response
```

Issued - The destination 802.2 LLC issues this primitive to acknowledge receipt of the LE\_ADDRESS\_RESOLUTION.Indicate.

Effect - Unspecified.

**4.6 HIPPI-LE station management service primitives**

These primitives, as illustrated in figure 5, are used to manage the local HIPPI-LE. These primitives may be used at either a source or destination HIPPI-LE.



**Figure 5 – Station management service primitives**

**4.6.1 LESM\_STATUS\_CTRL.Request**

This primitive is issued to set HIPPI-LE parameters or to read the value of HIPPI-LE parameters. The parameters are implementation dependent and are not specified.

LESM\_STATUS\_CTRL.Request (Control)

Control represents the commands, parameters, and values to be acted on.

Effect – Unspecified.

**4.6.2 LESM\_STATUS\_CTRL.Confirm**

This primitive replies to the associated LESM\_STATUS\_CTRL.Request with status information.

LESM\_STATUS\_CTRL.Confirm (Status)

Status represents the parameters and values that were set or read as a result of the LESM\_STATUS\_CTRL.Request.

Effect – Unspecified.

#### 4.6.3 LESM\_STATUS\_CTRL.Indicate

This primitive informs the SMT entity that an important event has occurred that affects the operation of the HIPPI-LE.

LESM\_STATUS\_CTRL.Indicate (Status)

Status represents the parameters and values that describe the event.

Effect – Unspecified.

#### 4.6.4 LESM\_STATUS\_CTRL.Response

This primitive acknowledges the LESM\_STATUS\_CTRL.Indicate.

LESM\_STATUS\_CTRL.Response

Effect – Unspecified.

### 5 HIPPI-FP data framing services used by HIPPI-LE

HIPPI-LE is dependent on the capabilities of the lower layer, e.g., HIPPI-FP, for both data delivery and error detection.

The following are the HIPPI-FP primitives that are logically used by HIPPI-LE. The complete specification of these primitives is contained in the HIPPI Framing Protocol (HIPPI-FP) document. The exact mapping from the HIPPI-LE services to HIPPI-FP services is implementation dependent.

HIPPI-LE Data Transfer:

FP\_TRANSFER.Request (CCI, ULP-id, D1\_Size, D1\_Data\_Set, D2\_Size, D2\_Data\_Set, Keep\_Connection, Start\_D2\_on\_Burst\_Boundary)

FP\_TRANSFER.Confirm

FP\_TRANSFER\_D1.Indicate (ULP-id, CCI, Status, D2\_Size, D2\_Offset, D1\_Area\_Size, D1\_Data\_Set)

FP\_TRANSFER\_D2.Indicate (ULP-id, CCI, Status, D2\_Size, D2\_Offset, D2\_Data\_Set)

FP\_TRANSFER.Response

### 6 HIPPI-LE data formats

The data transferred between the HIPPI-FP and HIPPI-LE is an ordered byte stream formatted as specified in the HIPPI-FP document.

#### 6.1 HIPPI-LE PDU format

A HIPPI-LE PDU shall comprise two areas, (1) a D1\_Area containing the LE\_Header, and (2) a D2\_Area containing the 802.2 LLC PDU.

##### 6.1.1 LE\_Header (D1\_Data\_Set)

The LE\_Header shall comprise the D1\_Data\_Set.

The LE\_Header shall be formatted as shown in figure 6. The image is shown as it would appear in 32-bit HIPPI words. The most-significant bit of the individual fields shown in figures 6 and 7 shall be at the left end of the field. The reserved field in figure 6 shall be transmitted as zeros.

FC is the 3-bit Forwarding Class for this 802.2 LLC PDU; the value ranges from 0 to 7. The value 0 shall be used as the default. The meaning of the other values is not specified.

Double\_Wide (W) (1 bit) may contain one if the Destination whose address is given in the Source\_Switch\_Address field supports 64 bit HIPPI operation. Otherwise it shall contain zero.

NOTE - The W bit is a hint to the receiver that it may make a return connection in 1600 Mbit/s mode.

Message\_Type (M\_Type) (4 bits) contains a code identifying the type of HIPPI-LE PDU. Defined values (binary) are:

- 0000 Data PDU
- 0001 Address Resolution Request PDU (AR\_Request)
- 0010 Address Resolution Response PDU (AR\_Response)
- 0011 Self Address Resolution Request PDU (AR\_S\_Request)
- 0100 Self Address Resolution Response PDU (AR\_S\_Response)
- 0101-1011 Reserved
- 1100-1111 Locally Assigned

Destination\_Address\_Type (D\_A\_T) (4 bits) and Source\_Address\_Type (S\_A\_T) (4 bits) contain a code identifying the type of addresses in the Destination\_Switch\_Address and Source\_Switch\_Address fields, respectively. Defined values (binary) are:

- 0000 Address not present
- 0001 HIPPI-SC Source Route (24 bits)
- 0010 HIPPI-SC Logical Address (12 bits)
- 0011-1011 Reserved
- 1100-1111 Locally Assigned

Destination\_Switch\_Address is a 24-bit field containing the Switch Address of the Destination. If the address consists of less than 24 bits, it shall be right justified (occupying the least significant bits) in the field.

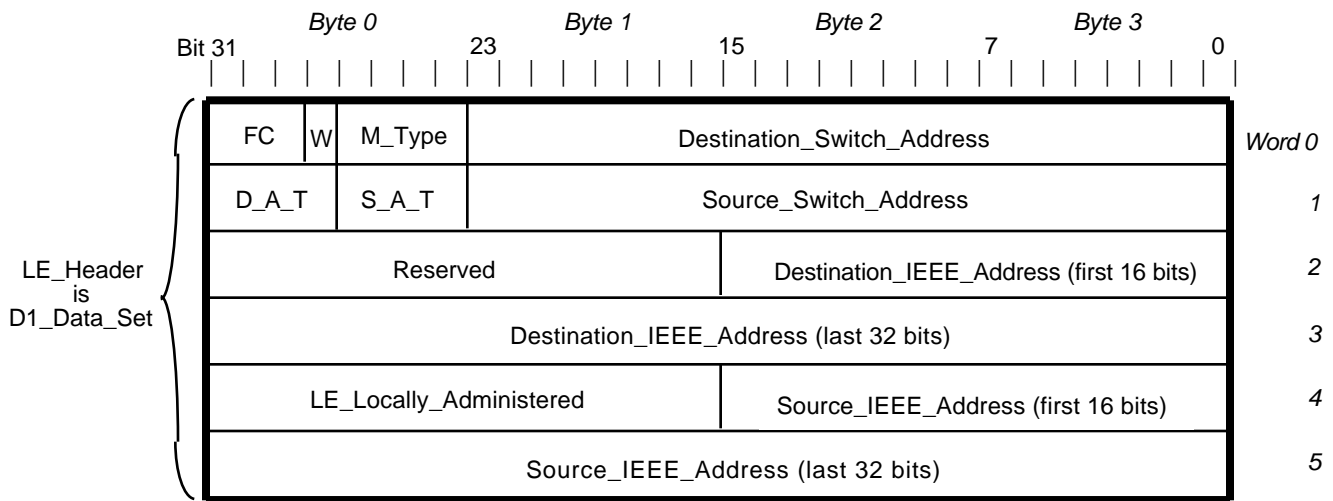


Figure 6 – HIPPI-LE header format

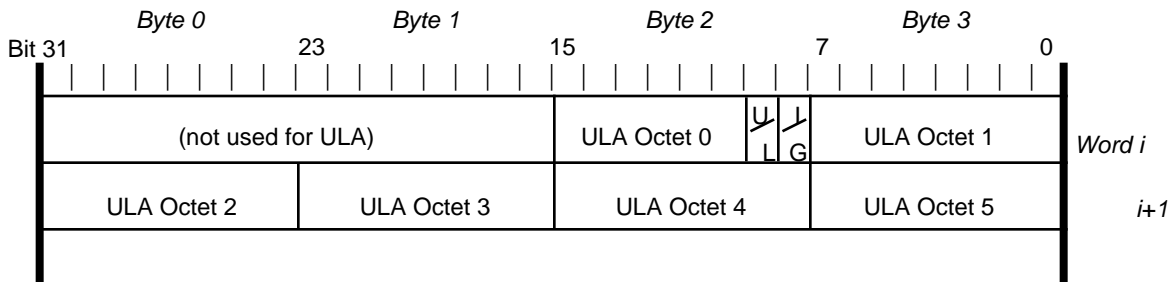


Figure 7 – Mapping of 6-byte IEEE universal LAN MAC address to two 32-bit HIPPI words in LE\_Header

Source\_Switch\_Address is a 24-bit field containing the Switch Address of the Source. If the address consists of less than 24 bits, it shall be right justified (occupying the least significant bits) in the field.

NOTE - The Double\_Wide, Switch\_Address, and Address\_Type fields are of primary interest for the Address Resolution Request and Response PDU types. Whether or not they are used with Data PDUs is implementation dependent.

Destination\_IEEE\_Address is the 48-bit Universal LAN MAC Address of the destination. This address space is specified by IEEE Standard 802.1A and the globally unique part of the space is administered by the IEEE. If the Destination\_IEEE\_Address is not known, then the value 0 shall be used.

NOTE – Some network devices, e.g., 802 LANs, FDDI, and MAC-level bridges, require a non zero 48-bit Universal LAN MAC destination address in order to deliver a PDU.

LE\_Locally\_Administered is a 16-bit field which is locally administered for each HIPPI-LE network. The values in this field are not specified although the value 0 is suggested as a default.

Source\_IEEE\_Address is the 48-bit Universal LAN MAC Address of the source of this 802.2 LLC PDU. This address space is specified by IEEE Standard 802.1A and the globally unique part of the space is administered by the IEEE. It is recommended that every source of HIPPI-LE PDUs be assigned a Universal LAN MAC source address according to the method specified in IEEE Standard 802.1A. However, if the Source\_IEEE\_Address is not known or has not been assigned, then the value 0 shall be used.

NOTE – Some network devices, e.g., 802 LANs, FDDI, and MAC-level bridges, require a non zero 48-bit Universal LAN MAC source address in order to provide two-way communication.

### 6.1.2 802.2 LLC PDU (D2\_Data\_Set)

The contiguous bytes of the 802.2 LLC PDU shall comprise the D2\_Data\_Set. See annex A for examples.

## 6.2 IEEE 48-bit address format

The IEEE Standard 802.1A specifies that a 48-bit Universal LAN MAC Address (ULA) is represented as an ordered string of six bytes numbered from 0 to 5. The least significant bit of byte 0 is the Individual/Group Address (I/G) bit. The next least significant bit is the Universally or Locally Administered Address (U/L) bit. The IEEE Standard 802.1A further specifies that the bytes be transmitted in the order 0 through 5.

Figure 7 shows how the bytes of an ULA (for either source or destination address) shall be mapped to two 32-bit HIPPI words in the LE\_Header.

## 7 HIPPI Address Resolution

### 7.1 Overview

HIPPI-FP describes a unidirectional link between a single source and a single destination which can be configured in groups of two or more to build bi-directional links between computer systems. HIPPI-SC describes switch hardware which can be used as a hub for the interconnection of multiple systems using bi-directional pairs of HIPPI links; other kinds of switches are possible. In a switched HIPPI network, each system must have an address which allows another system to direct a packet to it. This address takes one of several forms, depending on the switch hardware, and is called the Switch Address. Address resolution is the process of discovering a system's Switch Address.

A system's Switch Address can change depending on its location on the switch and other factors, such as address mappings within the switch or the configuration of routes between switches in multi-switch environments. When it changes, it must be rediscovered before other systems can resume communication with the re configured system. This is only possible if the target system (the one whose address must be discovered) has another address that remains constant, independent of its Switch Address. This address is the IEEE Universal LAN MAC address (ULA).

Address Resolution works by asking "what is the Switch Address associated with this target ULA?" The question is in the form of a HIPPI-LE Address Resolution Request PDU, and the answer is a HIPPI-LE Address Resolution Response PDU. In both packets, the Source\_Address\_Type and Destination\_Address\_Type fields indicate the form of the Switch Addresses. (HIPPI-SC defines two forms, 24 bit Source Route and 12 bit Logical Address.)

The Address Resolution Request and Response PDUs use the Source\_Switch\_Address, Destination\_Switch\_Address, Source\_IEEE\_Address and Destination\_IEEE\_Address fields of the HIPPI-LE Header. In the AR\_Request PDU, the Destination\_IEEE\_Address field specifies the target for

address resolution, and the Source\_Switch\_Address and Source\_IEEE\_Address fields are correct for the sender of the PDU. In the AR\_Response PDU the Source and Destination fields are exchanged and the Source\_Switch\_Address field contains the target's switch address.

These HIPPI-LE PDUs can also carry an IEEE 802.2 LLC PDU containing "piggy-backed" address resolution protocol for a higher layer, such as Internet ARP. If this is the case, the Destination\_IEEE\_Address field in the AR\_Request PDU may contain zero, since the target ULA will be resolved by the higher layer's address resolution entity. Both layers of addresses can be resolved from the higher layer's target address in the same exchange of HIPPI-LE PDUs.

The AR\_S\_Request and AR\_S\_Response PDUs are for "self address discovery," in which a HIPPI-LE entity is trying to discover its own Switch Address. In the request, both Source\_IEEE\_Address and Destination\_IEEE\_Address are set to the correct ULA for the sender, and the Source\_Switch\_Address and Destination\_Switch\_Address contain zero. After a host sends the request, two positive outcomes are possible:

- the host receives its own request, and obtains its own Switch Address from the CCI passed up from the underlying HIPPI-FP layer, or
- the host receives an AR\_S\_Response with the Destination\_Switch\_Address filled in.

In the first case the host should not respond to its own request.

The address to which to send AR\_Request and AR\_Response PDUs is outside the scope of this document. For example, HIPPI-SC allocates a specific logical address for HIPPI-LE Address Resolution and other configuration-related messages.

HIPPI-LE address resolution is not mandatory, since Switch Addresses may be known by other implementation dependent means. Whether or not a particular HIPPI-LE responds to an AR\_Request PDU is an implementation decision, since address resolution does not depend on the identity of the HIPPI-LE which actually transmits the AR\_Response PDU. It can be sent by the target system itself or by a third party that responds on behalf of the target.

NOTE - If the interconnection network is a HIPPI-SC switch, the appropriate multicast address must result in the transmission of the packet to all participating targets, or it may be mapped to a server which responds on the targets behalf. Specification of these mechanisms is beyond the scope of HIPPI-LE.

### 7.2 Address Resolution in Detail

This subclause describes in detail what is done by a HIPPI-LE entity during transmission or receipt of HIPPI-LE Address Resolution PDUs. This discussion presumes that two or more HIPPI-LE entities can communicate through HIPPI interfaces that are attached to some interconnection mechanism. A crosspoint switch that supports multicast addressing, as described in HIPPI-SC, is one such

mechanism. Others, either standard or vendor proprietary, may exist and support address resolution.

Each HIPPI-LE entity keeps a list of remote HIPPI-LE entities it is aware of, giving the destination Switch Address and ULA for each.

### 7.2.1 Transmission

The following steps are used to transmit HIPPI-LE Address Resolution PDUs.

1. The HIPPI-LE entity receives an LE\_TRANSFER.Request or LE\_ADDRESS\_RESOLUTION.Request from the Source ULP.

If the HIPPI-LE entity knows the correct value for the Source\_Switch\_Address and Source\_Address\_Type fields of the HIPPI-LE header, then go to step 4.

Otherwise, the source Switch Address must be discovered before the ULP's message can be sent.

2. Construct a HIPPI-LE PDU, setting the following fields:

```
FC = [implementation dependent]
Double_Wide = 0
Message_Type = 0011 (binary), AR_S_Request
Destination_Switch_Address = 0
Source_Switch_Address = 0
Destination_Address_Type = 0
Source_Address_Type = 0
Destination_IEEE_Address = ULA of this HIPPI-LE
Source_IEEE_Address = ULA of this HIPPI-LE
```

Invoke an FP\_TRANSFER.Request for this PDU using a CCI value appropriate for transmission of this type of PDU on the interconnection hardware in use. On a HIPPI-SC compliant switch, this should be the address reserved for HIPPI-LE Address Resolution requests, i.e., hexadecimal FE0.

3. Issue the LE\_TRANSFER.Confirm or LE\_ADDRESS\_RESOLUTION.Confirm primitive to the ULP, with Transmission\_Status indicating that the transmission was not performed. Request handling is complete.

4. In this case the source Switch Address is known. If the ULP requested a Data or AR\_Response PDU, update the table entry or add a new entry for the remote HIPPI-LE, using the Destination\_Address information given by the ULP in the LE\_TRANSFER.Request or LE\_ADDRESS\_RESOLUTION.Request. This information should include the remote HIPPI-LE's ULA, Switch Address and Switch Address type.

Construct a HIPPI-LE PDU containing the 802.2 LLC PDU. HIPPI-LE Header contents depend on the type of request and the Message\_Type parameter in a LE\_ADDRESS\_RESOLUTION.Request from the ULP. Invoke the FP\_TRANSFER.Request primitive to send the PDU.

- If the ULP requested a Data PDU:

```
FC = [implementation dependent]
Double_Wide = (optional) 1 if this HIPPI-LE can
accept double-wide requests.
Message_Type = 0000 (binary), Data PDU
Destination_Switch_Address = (optional) the
destination's Switch Address
Source_Switch_Address = (optional) this HIPPI-LE's
Switch Address
Destination_Address_Type = value appropriate to
Destination_Switch_Address
Source_Address_Type = value appropriate to
Source_Switch_Address
Destination_IEEE_Address = ULA of the destination
if known, otherwise zero
Source_IEEE_Address = ULA of this HIPPI-LE if
known, otherwise zero
The CCI parameter of the FP_TRANSFER.Request
contains the destination's Switch Address.
```

- If the ULP requested an Address Resolution Request:

```
FC = [implementation dependent]
Double_Wide = (optional) 1 if this HIPPI-LE can
accept double-wide requests.
Message_Type = 0001 (binary), AR_Request
Destination_Switch_Address = 0
Source_Switch_Address = this HIPPI-LE's Switch
Address
Destination_Address_Type = 0
Source_Address_Type = value appropriate to
Source_Switch_Address
Destination_IEEE_Address = ULA of the destination
if known, otherwise zero.
Source_IEEE_Address = ULA of this HIPPI-LE if
known, otherwise zero
The CCI parameter to the FP_TRANSFER.Request
primitive contains the Switch Address appropriate
for transmission of Address Resolution Request
PDUs on the interconnection hardware in use. On
a HIPPI-SC compliant switch, this should be the
address reserved for HIPPI-LE Address Resolution
requests, i.e., hexadecimal FE0.
```

- If the ULP requested an Address Resolution Response:

```
FC = [implementation dependent]
Double_Wide = (optional) 1 if this HIPPI-LE can
accept double-wide requests.
Message_Type = 0010 (binary), AR_Response
Destination_Switch_Address = the destination's
Switch Address
Source_Switch_Address = this HIPPI-LE's Switch
Address
Destination_Address_Type = value appropriate to
Destination_Switch_Address
Source_Address_Type = value appropriate to
Source_Switch_Address
Destination_IEEE_Address = ULA of the destination
if known, otherwise zero.
Source_IEEE_Address = ULA of this HIPPI-LE if
known, otherwise zero.
The CCI parameter to the FP_TRANSFER.Request
primitive contains the destination's Switch Address.
```

### 7.2.2 Receipt of normal Address Resolution PDUs

When the HIPPI-LE entity receives a FP\_TRANSFER.Indicate primitive from the underlying HIPPI-FP, the action for normal address resolution messages (Message\_Type 0001 or 0010) is:

- a) If the received message has a Source\_IEEE\_Address equal to one already in the table, copy the Source\_Switch\_Address and Source\_Address\_Type fields to the table, updating the entry for the remote HIPPI-LE.
- b) If the Destination\_IEEE\_Address field contains the ULA of the local HIPPI-LE entity and the table was not updated in step 1, add a new entry using the Source\_Switch\_Address, Source\_Address\_Type and Source\_IEEE\_Address field contents.

NOTE - The checks in this step prevent the creation of table entries for remote hosts with which the local host is not communicating.

- c) Report the receipt of the message to the ULP with a LE\_ADDRESS\_RESOLUTION.Indicate primitive.

### 7.2.3 Receipt of Self Address Resolution PDUs

AR\_S\_Request: If the HIPPI-LE entity does not know its own Switch Address and type, use the CCI from the FP\_TRANSFER.Indicate primitive to construct it, and otherwise ignore the received message.

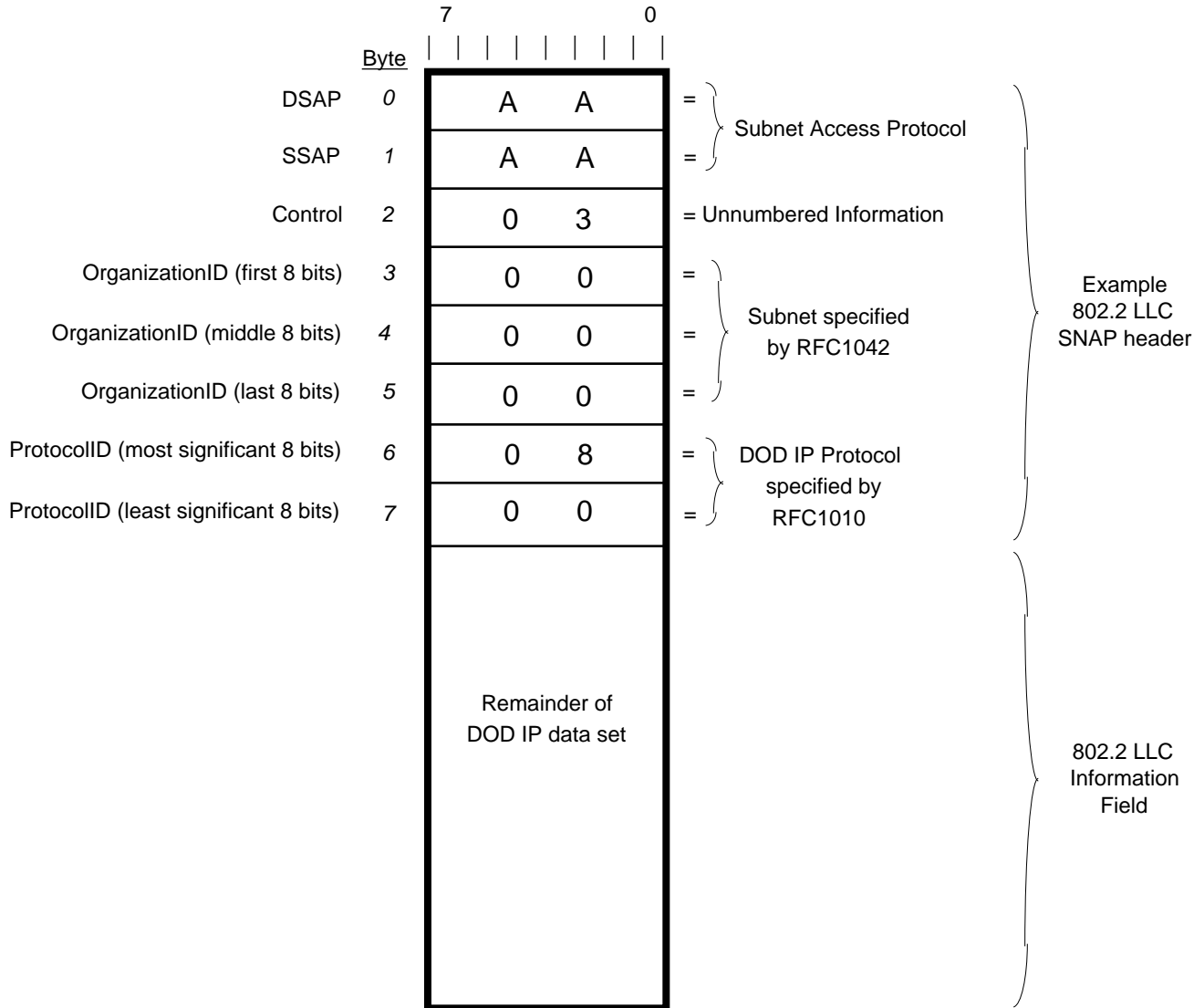
AR\_S\_Response: If the Destination\_IEEE\_Address is the same as the HIPPI-LE entity's ULA, the HIPPI-LE entity takes the contents of the Destination\_Switch\_Address field as its new Switch Address, also saving the value of the Destination\_Address\_Type field. Otherwise, it ignores the message.

No LE\_ADDRESS\_RESOLUTION.Indicate primitive is given for receipt of self address resolution messages.



## Annex A (informative)

### Example HIPPI packets containing HIPPI-LE PDUs



**Figure A.1 – Example IEEE 802.2 LLC PDU as an ordered hex byte stream with information field = DOD IP data set as specified by the Internet Task Force in RFC1010 and RFC1042**

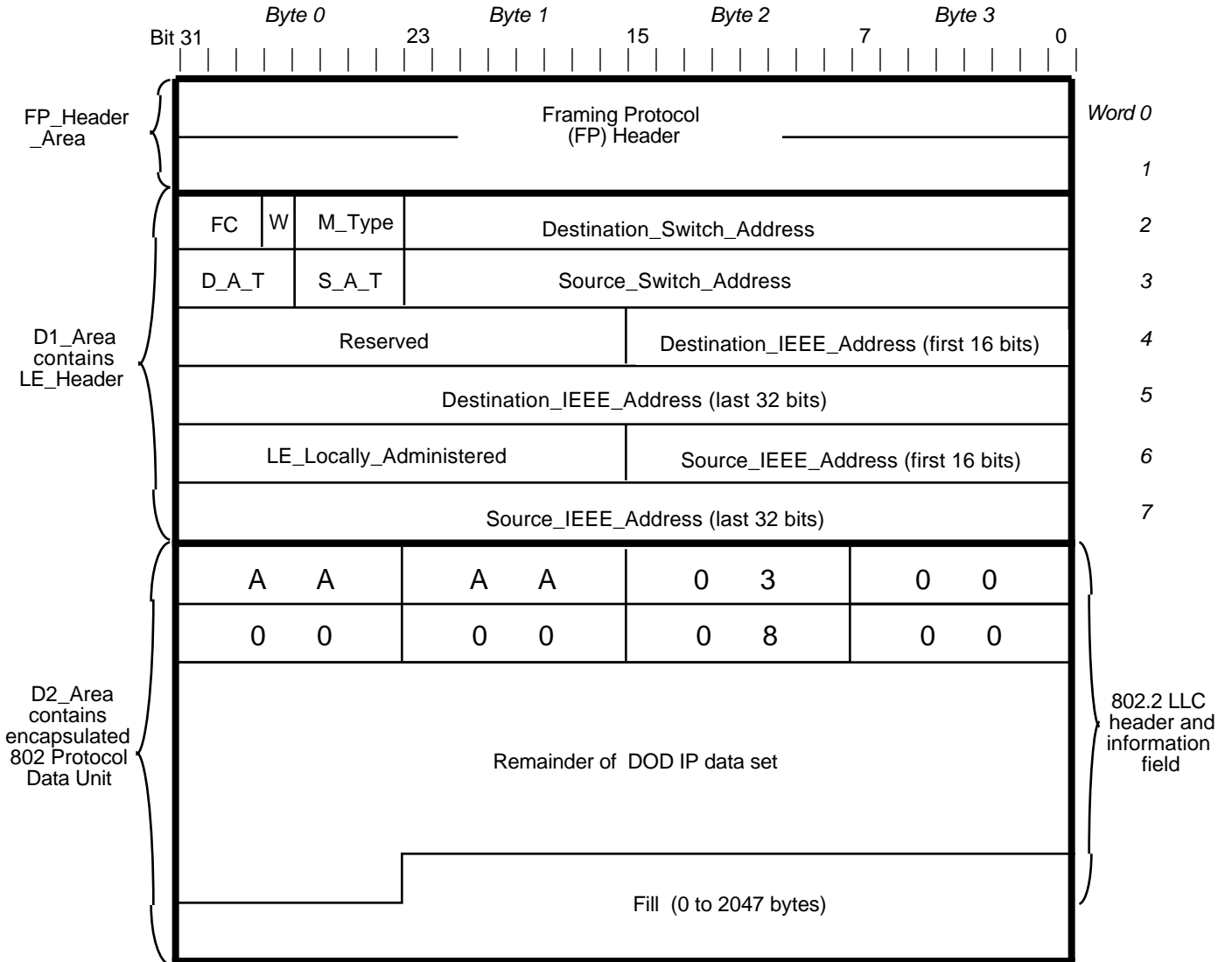


Figure A.2 – Example encapsulation of figure A.1 LLC PDU within a HIPPI packet with *suggested* fills and alignment, i.e., D1\_Fill = 0 bytes and D2 offset = 0

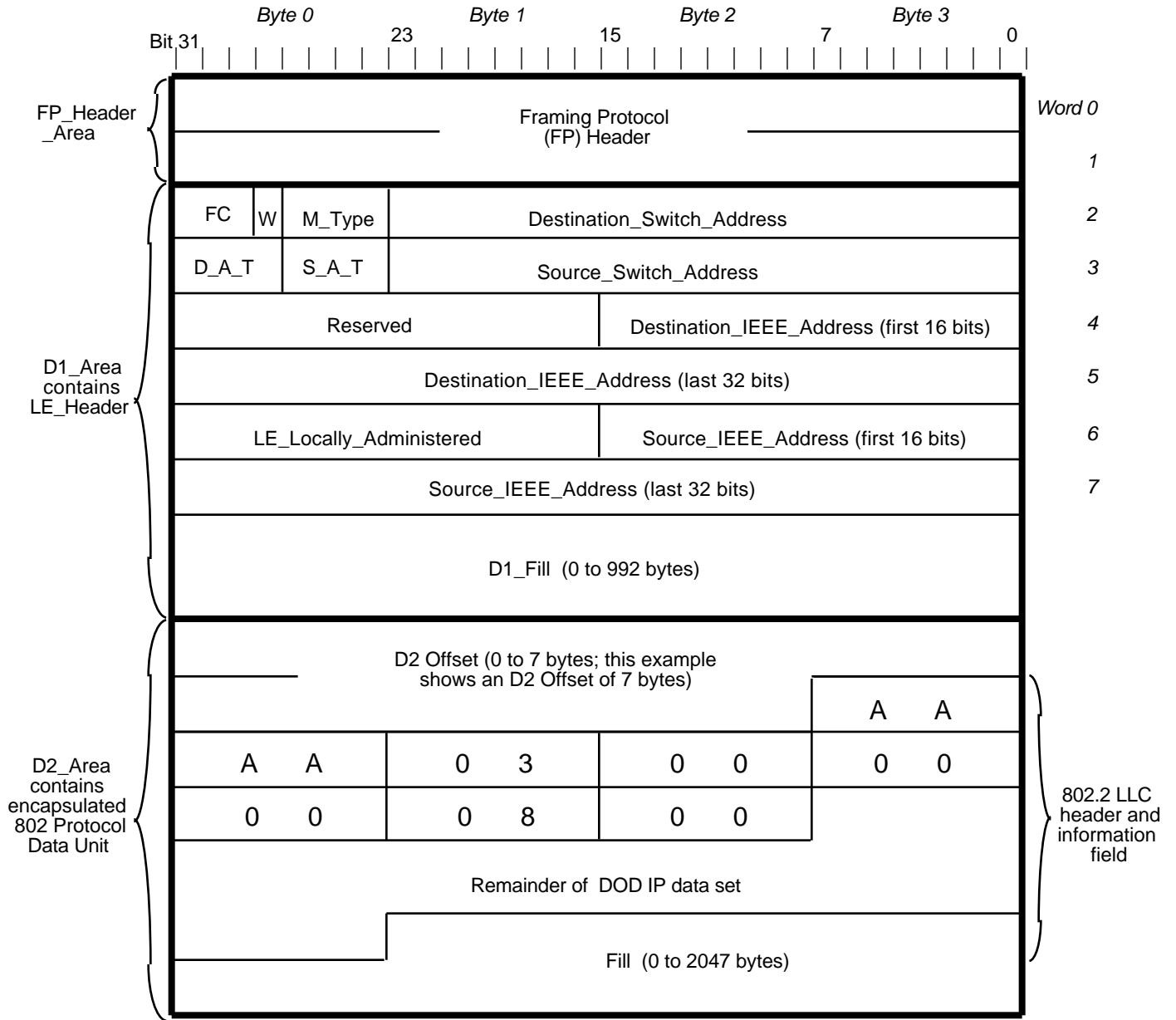


Figure A.3 – Example encapsulation of figure A.1 LLC PDU within a HIPPI packet with D2 offset = 7 and a possibly non-empty D1\_Fill

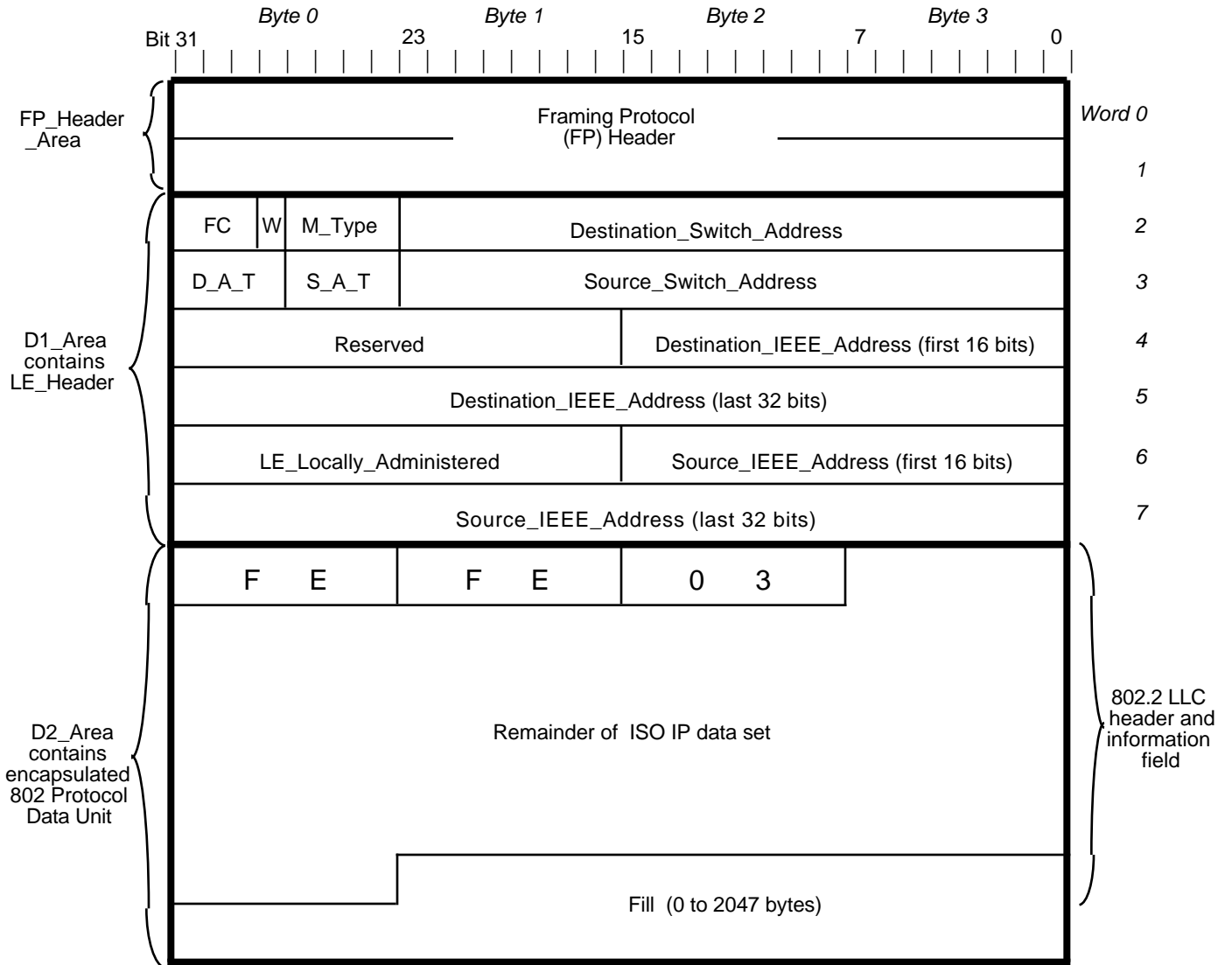


Figure A.4 – Example encapsulation of ISO IP PDU within a HIPPI packet with D1\_Fill = 0 bytes and D2 offset = 0

## **Annex B** (informative)

### **Bibliography**

The following documents are the basis for address resolution:

RFC 826<sup>1)</sup> , "An Ethernet Address Resolution Protocol - or -  
Converting Network Protocol Addresses to 48.bit Ethernet  
Address for Transmission on Ethernet Hardware"

---

<sup>1)</sup> RFC (Request For Comment) documents are working standards documents for the TCP/IP internetworking community. Copies of these documents are available from numerous electronic sources or by writing to Network Information Systems, SRI International, 333 Ravenswood Ave Room BJ291, Menlo Park, CA 94025.

## Alphabetical index

Address Resolution .....	3.1.2, 7, 7.2	Requested_Service_Class ....	4.4.1, 4.4.3
Address Resolution Protocol ..	3.1.3, 7.1, B	Reserved .....	6.1.1, A.2, A.3, A.4
Address Resolution Request ..	4.5.1, 6.1.1, 7.1, 7.2.1	SDU .....	3.1.5
Address Resolution Response..	4.5.1, 6.1.1, 7.1, 7.2.1	Self Address Resolution .....	6.1.1, 7.1, 7.2.3
CCI .....	4.5.1, 5, 7.1, 7.2.1, 7.2.3	SMT .....	3.1.6, 4
D1_Area .....	6.1, A.2, A.3, A.4	Source_Address .....	4.3, 4.4.3
D1_Area_Size .....	5	Source_Address_Type .....	6.1.1, 7.1, 7.2.1, 7.2.2
D1_Data_Set .....	5, 6.1.1	Source_IEEE_Address .....	4.4.3, 4.5.3, 6.1.1, 7.2.1, 7.2.2, A.2, A.3, A.4
D1_Size .....	5	Source_ULA .....	4.3, 4.5.1, 4.5.3
D2_Area .....	4.4.1, 6.1, A.2, A.3, A.4	Start_D2_on_Burst_Boundary..	5
D2_Data_Set .....	4.4.3, 5, 6.1.2	Status	
D2_Offset .....	4.4.1, 5	Reception_Status .....	4.3, 4.4.3, 4.5.3
D2_Size .....	5	Transmission_Status .....	4.3, 4.4.2, 4.5.2, 7.2.1
Destination_Address .....	3.2, 4.3, 4.4.1, 4.4.3, 4.5.1, 7.2.1, 7.2.3	ULP-id .....	5
Destination_Address_Type ...	6.1.1, 7.1, 7.2.3	Universal LAN MAC Address..	4.5, 6.1.1, 6.2, 7.1
Destination_IEEE_Address ...	4.5.1, 4.5.3, 6.1.1, 7.1, 7.2.1, 7.2.2, 7.2.3, A.2, A.3, A.4		
Destination_ULA .....	4.3, 4.5.1, 4.5.3		
DOD IP data set .....	A.1, A.2, A.3		
FC .....	6.1.1, 7.2.1, A.2, A.3, A.4		
Forwarding class .....	Introduction, 6.1.1		
HIPPI-LE PDU .....	Introduction, 3.1.4, 4.4.1, 4.4.3, 4.5.1, 4.5.3, 6.1, 6.1.1, 6.2, A.1		
HIPPI-FP services .....	4, 5		
IEEE 48-bit address .....	4.4.3, 6.2		
Individual/Group Address .....	6.2		
ISO IP PDU data set .....	A.4		
Keep_Connection .....	5		
LE_Header .....	4.4.1, 6.1, 6.1.1		
LE_Locally_Administered ....	6.1.1, A.2, A.3, A.4		
LE_SDU .....	4.3, 4.4.1, 4.4.3, 4.5.1, 4.5.3		
LLC PDU .....	3.1.1, 3.1.4, 4.4, 4.4.1, 4.4.3, 4.5.1, 4.5.3, 6.1, 6.1.1, 6.1.2, 7.1, 7.2.1, A.1, A.2, A.3		
Locally Administered Address..	6.2		
Message_Type .....	4.3, 4.5.1, 4.5.3, 6.1.1, 7.2.1, 7.2.2		
Multicast .....	3.1.2, 7.1, 7.2		
PDU .....	3.1.4		
Provided_Service_Class .....	4.4.2		