EXHIBIT A

FINNEGAN
HENDERSON
FARABOW
GARRETT &
DUNNER LLP

901 New York Avenue, NW ▪ Washington, DC 20001-4413 ▪ 202.408.4000 ▪ Fax 202.408.4400
www.finnegan.com

JOHN M. WILLIAMSON
202.408.4282
john.williamson@finnegan.com

July 13, 2005

Clement J. Naples, Esq.
Kenyon & Kenyon
One Broadway
New York, NY 10004-1050

via email

Telcordia v. Alcatel
C.A. No. 04-874-GMS

Sasha D. Mayergoyz, Esq.
Latham & Watkins, LLP
Sears Tower, Suite 5800
233 S. Wacker Dr.
Chicago, IL 60606

via email

Telcordia v. Lucent
C.A. No. 04-875-GMS

Edward R. Reines, Esq.
Weil, Gotshal & Manges LLP
201 Redwood Shores Parkway
Redwood Shores, CA 94065

via email

Telcordia v. Cisco
C.A. No. 04-876-GMS

Dear Counsel:

Thank you again for discussing several outstanding discovery and protective order issues with us during our recent respective teleconferences. During those teleconferences, each of the defendants alleged that Telcordia's definitions of products subject to discovery are ambiguous and overly broad. While we continue to believe that Telcordia's definitions appropriately identify the relevant products, nevertheless, in a good faith effort to move discovery forward, we provide the following clarification of those definitions. This clarification serves only as a starting point for your answers to interrogatories and production of documents, and in no way limits the global scope of discovery in these actions. Rather, as these cases progress, Telcordia

may discover through depositions or otherwise that other products beyond those identified in this letter are relevant and should become the subject of further discovery. At this overdue stage, none of the defendants has produced any product-related discovery at all. As a starting point, we expect that each defendant will promptly produce documents and other discovery consistent with the product identifications provided in this letter as to the patents asserted in each suit.

The products and components relevant to the claims and defenses implicated by the '306 patent at this time include at least:

1. All products that map Packetized Data into SONET frames.

   "Packetized Data" is data formatted with a header field and an information field (*see* '306 patent, 7:2-13) including, without limitation:

   (1) Frame Relay frames (*see, e.g.*, Broadband Telecommunications Handbook, p. 159, Fig. 10-4)(attached to this letter as Exhibit A);

   (2) Ethernet frames (*see, e.g.*, IEEE WG 802.3; Broadband Telecommunications Handbook, p. 160, Fig. 10-5)(attached to this letter as Exhibit B);

   (3) ATM cells (*see, e.g.*, ITU-T I.361 Figs. 1 and 2)(attached to this letter as Exhibit C);

   (4) RFC 1662 frames (*see, e.g.*, IETF RFC 1662 at Section 3.1)(attached to this letter as Exhibit D);

   (5) ITU-T X.86 LAPS frames (*see, e.g.*, ITU-T X.86 Figs. 6 and 7)(attached to this letter as Exhibit E);

   (6) ITU-T G.7041 GFP frames (*see, e.g.*, ITU-T G.7041 at Figs. 6-1 to 6-5)(attached to this letter as Exhibit F);

   (7) MPLS packets (*see, e.g.* IETF RFC 3032 at Sections 1 to 2.1)(attached to this letter as Exhibit G).

   Please note that these specific textbook and standards references are provided merely as examples, to clarify what is meant by "Packetized Data," only for purposes of providing context for the scope of discovery in these cases at this time.

2. All integrated circuits used in the products identified in No. 1 above to map Packetized Data into SONET frames.

The products and components relevant to the claims and defenses implicated by the '633 patent at this time include at least:

3. All products that practice and/or are capable of practicing the "Synchronous Residual Time Stamp" technology disclosed in the ATM Forum Standard (*see, e.g.*, af-vtoa-0078.000 Section 3.4.1), the ITU-T Standard I.363.1 (*see, e.g.*, Section 2.5.2.2.2), and/or the ANSI T1.630 standard (*see, e.g.*, Section 5 referencing I.363.1).

4. All integrated circuits used in the products identified in No. 3 above to perform the synchronous residual time stamp function.

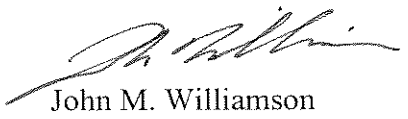
The products and components relevant to the claims and defenses implicated by the '763 patent at this time include at least:

5. All products that that perform unidirectional path switched ring automatic protection switching and/or that comply with GR-1400 (*see, e.g.*, GR-1400 Issue 1, Revision 1, Section 3).

6. All integrated circuits used in the products identified in No. 5 above to perform automatic protection switching.

We intend to move forward with discovery consistent with these product identifications and trust that each of the defendants will promptly respond to Telcorida's outstanding discovery requests, including Telcordia Interrogatory No. 1. By no later than July 20, 2005, please confirm that your respective clients will produce documents and supplement their responses to Telcordia's Interrogatories consistent with the product identifications provided in this letter, and please provide a date when we can expect this production and supplementation.

Very Truly Yours,



John M. Williamson

Enclosures

Exhibit A

Exhibit A

Broadband Telecommunications Handbook

Regis J. Bates

McGraw-Hill
New York • San Francisco • Washington, D.C.
Auckland • Bogotá • Caracas • Lisbon • London
Madrid • Mexico City • Milan • Montreal • New Delhi
San Juan • Singapore • Sydney • Tokyo • Toronto

Library of Congress Cataloging-in-Publication Data on File

McGraw-Hill
A Division of The McGraw-Hill Companies

Copyright © 2000 by The McGraw-Hill Companies, Inc. All rights reserved. Printed in the United States of America. Except as permitted under the United States Copyright Act of 1976, no part of this publication may be reproduced or distributed in any form or by any means, or stored in a data base or retrieval system, without the prior written permission of the publisher.

1 2 3 4 5 6 7 8 9 0 AGM/AGM 9 0 4 3 2 1 0 9

ISBN 0-07-134648-1

This book was set in Century Schoolbook by D&G Limited, LLC.

Printed and bound by Quebecor/Martinsburg.

McGraw-Hill books are available at special quantity discounts to use as premiums and sales promotions, or for use in corporate training programs. For more information, please write to Director of Special Sales, McGraw-Hill, 11 West 19th Street, New York, NY 10011.

Information contained in this work has been obtained by The McGraw-Hill Companies, Inc. ("McGraw-Hill") from sources believed to be reliable. However, neither McGraw-Hill nor its authors guarantees the accuracy or completeness of any information published herein and neither McGraw-Hill nor its authors shall be responsible for any errors, omissions, or damages arising out of use of this information. This work is published with the understanding that McGraw-Hill and its authors are supplying information but are not attempting to render engineering or other professional services. If such services are required, the assistance of an appropriate professional should be sought.

This book is printed on recycled, acid-free paper containing a minimum of 50 percent recycled de-inked fiber.

the FRAD used here is through a CSU/DSU on one end and a high-speed T3 multiplexer on the far end. The choices vary for the end user, so the flexibility of the access is one of the strong points for the Frame Relay network.

The Frame

When Frame Relay was developed, the important part of the data-carrying capacity was the use of the frame to carry the traffic and not have the same overhead as an older technology (such as X.25). The frame was filled with data as necessary, but it handled the speed and throughput via the high-speed communications and lower overhead.

In Figure 10-4, the frame is shown. The frame is an HDLC-framed format, as shown in this figure. The beginning of the frame (as with most HDLC formats) starts with an opening flag. Next, a two-byte sequence defines the addressing of the frame. This is called the *Data Link Connection Identifier* (DLCI). By very nature of the title (DLCI), we can assume that Frame Relay works at the data link layer. The DLCI is comprised of several pieces of information, shown later, but is normally a two-byte sequence. Provisions have been made to enable the DLCI to expand to up to four bytes, but very few implementations occur using more than the two-byte address. Following the DLCI is the information field. This is a variable length field. The initial standard allowed for a variable amount of data is up to 1,610 bytes. This is sufficient for most installations, but change always occurs when things are stable. The reason for the 1,610-byte field is to handle a frame from a LAN using a full frame of Ethernet traffic.

The Ethernet frame can be as large as 1,518 bytes (with overhead) and some subnet access protocol overhead (snap); the full frame should therefore accommodate the 1,610 bytes. The Ethernet frame is shown in Figure 10-5. This frame is the same size for an 802.3 IEEE frame or for a DIX Ethernet frame. Therefore, the variable data frame is sufficient to carry the traffic loads necessary.

Following the data field in the Frame Relay frame is a CRC used only to check for corruption. The CRC determines if the frame or the address

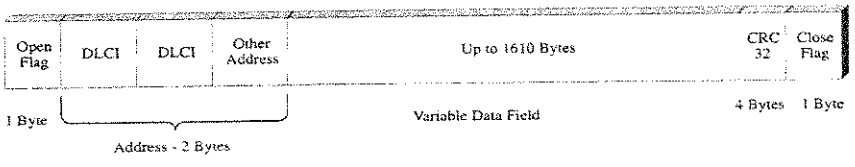
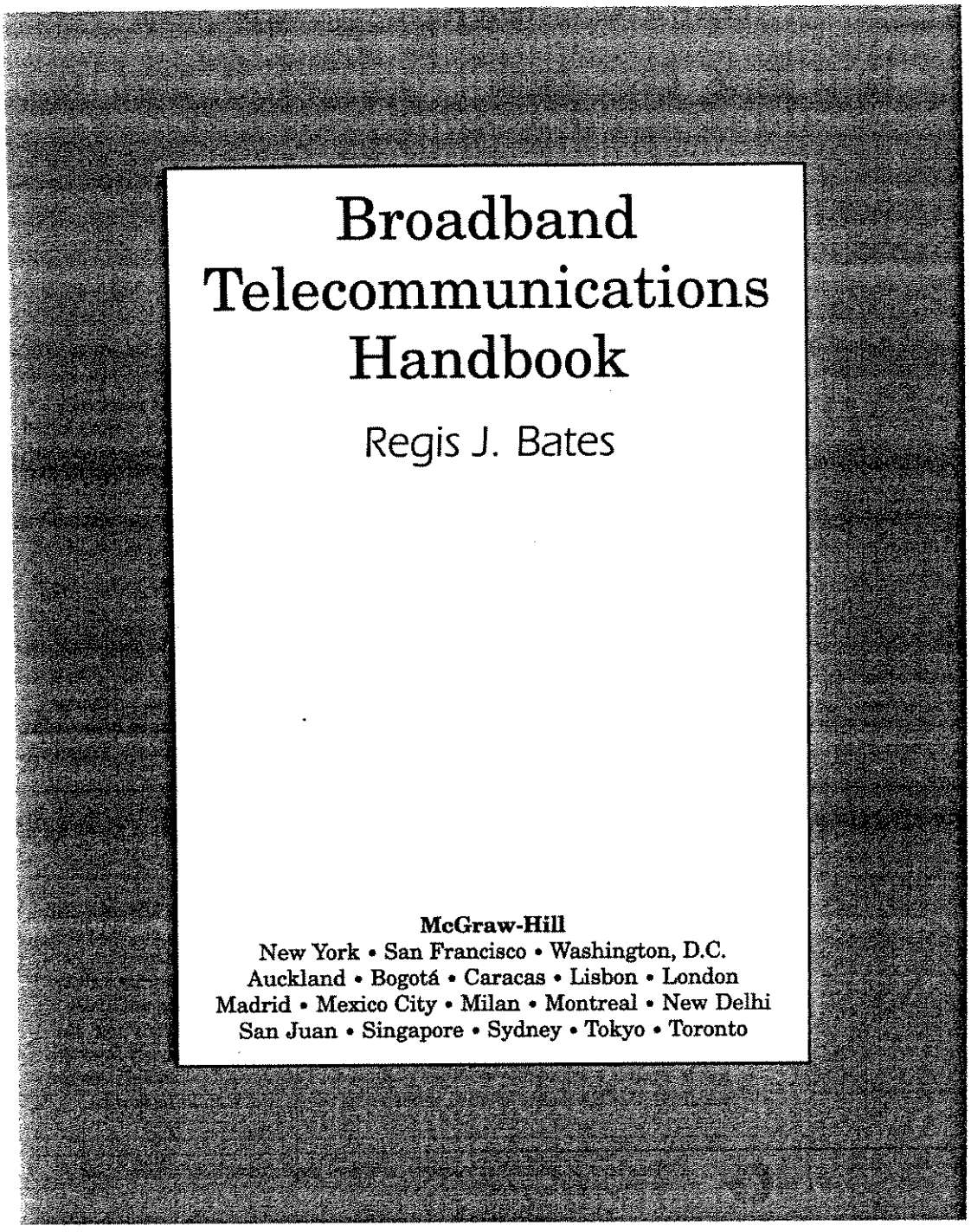


Figure 10-4
A Frame Relay frame

Exhibit B

Exhibit B


Broadband
Telecommunications
Handbook
Regis J. Bates
McGraw-Hill
New York • San Francisco • Washington, D.C.
Auckland • Bogotá • Caracas • Lisbon • London
Madrid • Mexico City • Milan • Montreal • New Delhi
San Juan • Singapore • Sydney • Tokyo • Toronto

Library of Congress Cataloging-in-Publication Data on File

McGraw-Hill
A Division of The McGraw-Hill Companies

Copyright © 2000 by The McGraw-Hill Companies, Inc. All rights reserved. Printed in the United States of America. Except as permitted under the United States Copyright Act of 1976, no part of this publication may be reproduced or distributed in any form or by any means, or stored in a data base or retrieval system, without the prior written permission of the publisher.

1 2 3 4 5 6 7 8 9 0 AGM/AGM 9 0 4 3 2 1 0 9

ISBN 0-07-134648-1

This book was set in Century Schoolbook by D&G Limited, LLC.

Printed and bound by Quebecor/Martinsburg.

McGraw-Hill books are available at special quantity discounts to use as premiums and sales promotions, or for use in corporate training programs. For more information, please write to Director of Special Sales, McGraw-Hill, 11 West 19th Street, New York, NY 10011.

Information contained in this work has been obtained by The McGraw-Hill Companies, Inc. ("McGraw-Hill") from sources believed to be reliable. However, neither McGraw-Hill nor its authors guarantees the accuracy or completeness of any information published herein and neither McGraw-Hill nor its authors shall be responsible for any errors, omissions, or damages arising out of use of this information. This work is published with the understanding that McGraw-Hill and its authors are supplying information but are not attempting to render engineering or other professional services. If such services are required, the assistance of an appropriate professional should be sought.

This book is printed on recycled, acid-free paper containing a minimum of 50 percent recycled de-inked fiber.

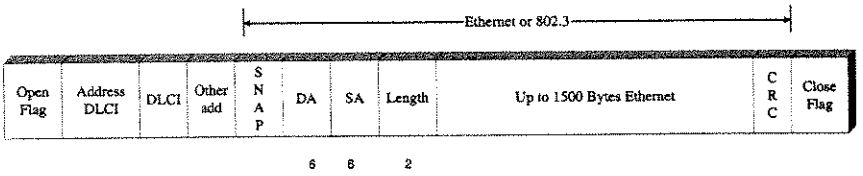


Figure 10-5
Ethernet and IEEE 802.3 frames fit into the Frame Relay frame

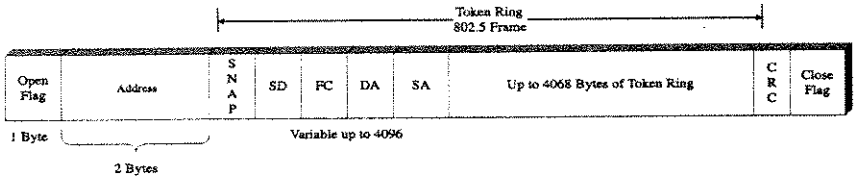


Figure 10-6
Modified frame size of the Frame Relay information field

information is corrupt. If so, the frame is discarded; if not, the frame is forwarded. There is no ACK or NAK in the Frame Relay transmission along the route. Lastly, there is a closing flag on the frame, indicating that the transmission of the frame is ended and the switching system can then process the entire frame. In many cases when a variable data field is used, the switches must allocate enough buffer space to hold a full frame, regardless of how full each frame is. This is somewhat wasteful across the WAN but does provide the necessary flexibility to handle the traffic.

Shortly after Frame relay was introduced with the 1610-byte information field, a new issue cropped up. What about the clients who use an IBM token ring? The token ring LANs can carry a variable amount of information up to 4,068 bytes. This means that a frame in the Frame Relay world is not large enough to carry a full token and the data must be truncated into three tokens to accommodate the token ring. To solve this problem, the frame was expanded to accommodate up to 4,096 bytes in the information field. Not all suppliers supported this change, but the two major suppliers of Frame Relay products (Cascade and Nortel) both adjusted their systems to accommodate the larger frame. This frame is shown in Figure 10-6 with a variable frame size of up to 4,096 bytes.

Figure
OSI con
Frame
stacks

The OSI Protocol Stack and Frame Relay

When we discuss the use of the data link protocols, one always compares the OSI to whatever other protocol is being discussed. This book is no different,

Exhibit C

Exhibit C



INTERNATIONAL TELECOMMUNICATION UNION

ITU-T

TELECOMMUNICATION
STANDARDIZATION SECTOR
OF ITU

I.361

(02/99)

SERIES I: INTEGRATED SERVICES DIGITAL NETWORK

Overall network aspects and functions – Protocol layer requirements

---

B-ISDN ATM layer specification

ITU-T Recommendation I.361

(Previously CCITT Recommendation)

Recommendation I.361

B-ISDN ATM LAYER SPECIFICATION

(revised in 1999)

1 Introduction

This Recommendation specifically addresses:

a) the cell structure and the ATM cell coding;

b) the ATM protocol procedures.

2 Cell structure coding

Two different coding schemes have been adopted: the UNI format and the NNI format. They are described in 2.2 and 2.3.

2.1 Cell structure

The cell consists of a five-octet header and a 48-octet information field as shown in Figure 1.

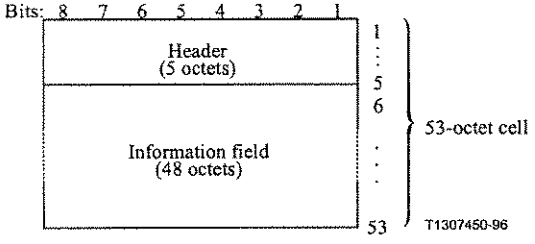


Figure 1/I.361 – Cell structure at the UNI/NNI

NOTE – The header will be sent first followed by the information field.

When a field within the header is contained within a single octet, the lowest bit number of the field represents the lowest order value.

When a field spans more than one octet, the order of bit values within each octet progressively decreases as the octet number increases; the lowest bit number associated with the field represents the lowest order value.

This leads to the following conventions:

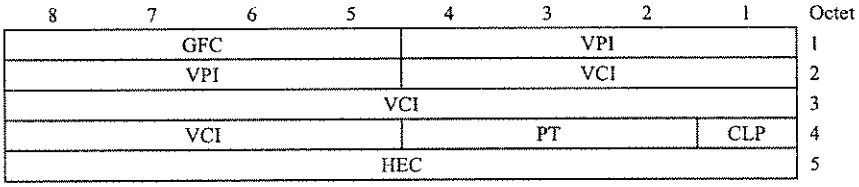
– bits within an octet are sent in decreasing order, starting with bit 8;

– octets are sent in increasing order, starting with octet 1;

– for all fields, the first bit sent is the Most Significant Bit (MSB).

2.2 Cell header format and encoding at UNI

The structure of the header is shown in Figure 2. The fields contained in the header and their encoding are described in the following subclauses.



CLP Cell Loss Priority
GFC Generic Flow Control
PT Payload Type
HEC Header Error Control
VPI Virtual Path Identifier
VCI Virtual Channel Identifier

Figure 2/I.361 – Header structure at UNI

2.2.1 Pre-assigned values of the physical cell header

Physical cells are reserved for use by the Physical layer. These cells are not passed from the Physical layer to the ATM layer.

Pre-assigned values of the cell header (to differentiate cells for the use of the ATM layer from cells for the use of the Physical layer) are given in Table 1. In the case of Physical cells, the bit in the location of the CLP indication is not used for the CLP mechanism as specified in 3.4.2.3.2/I.150. All other values which are described in Tables 2 and 4 are for the use of the ATM layer.

Table 1/I.361 – Pre-assigned Physical cell header values (excluding the HEC field)

	Octet 1	Octet 2	Octet 3	Octet 4
Idle cell identification	00000000	00000000	00000000	00000001
Reserved for use of the Physical layer (Note)	PPPP0000	00000000	00000000	0000PPP1

P Indicates the bit is available for use by the Physical layer.

Values assigned to these bits have no meaning with respect to the fields occupying the corresponding bit positions at the ATM layer.

NOTE – Specific pre-assigned physical layer cell header values are given in Recommendation I.432-series and in other Recommendations related to Physical layer (see Appendix I).

2.2.2 Generic Flow Control (GFC) field

The GFC field contains 4 bits.

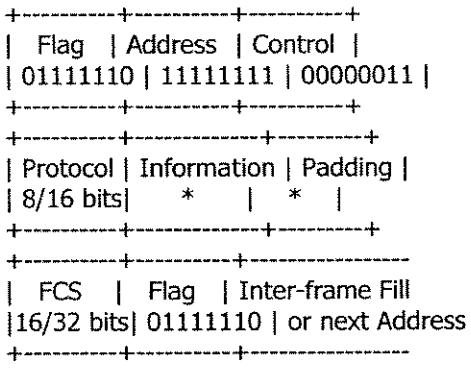
The following provides an overview of the GFC functions for the valid codings of the GFC field (see 4.1.1).

Uncontrolled equipment will always set the GFC field to 0000. Coding of this field by controlling and controlled equipment is given in 4.1.1. The controlled equipment default mode provides for a single queue for controlled ATM connections and allows uncontrolled ATM connections. The two-queue model provides for two queues for controlled ATM connections and allows uncontrolled ATM connections. At any given instant of time, controlled equipment which may not have any controlled ATM connections active will continue to respond to the HALT command.

Exhibit D

3.1. Frame Format

A summary of the PPP HDLC-like frame structure is shown below. This figure does not include bits inserted for synchronization (such as start and stop bits for asynchronous links), nor any bits or octets inserted for transparency. The fields are transmitted from left to right.



The Protocol, Information and Padding fields are described in the Point-to-Point Protocol Encapsulation [1].

Flag Sequence

Each frame begins and ends with a Flag Sequence, which is the binary sequence 01111110 (hexadecimal 0x7e). All implementations continuously check for this flag, which is used for frame synchronization.

Only one Flag Sequence is required between two frames. Two consecutive Flag Sequences constitute an empty frame, which is silently discarded, and not counted as a FCS error.

Address Field

The Address field is a single octet, which contains the binary sequence 11111111 (hexadecimal 0xff), the All-Stations address. Individual station addresses are not assigned. The All-Stations address MUST always be recognized and received.

The use of other address lengths and values may be defined at a later time, or by prior agreement. Frames with unrecognized Addresses SHOULD be silently discarded.

Control Field

The Control field is a single octet, which contains the binary sequence 00000011 (hexadecimal 0x03), the Unnumbered Information (UI) command with the Poll/Final (P/F) bit set to zero.

The use of other Control field values may be defined at a later time, or by prior agreement. Frames with unrecognized Control field values SHOULD be silently discarded.

Frame Check Sequence (FCS) Field

The Frame Check Sequence field defaults to 16 bits (two octets). The FCS is transmitted least significant octet first, which contains the coefficient of the highest term.

A 32-bit (four octet) FCS is also defined. Its use may be negotiated as described in "PPP LCP Extensions" [5].

The use of other FCS lengths may be defined at a later time, or by prior agreement.

The FCS field is calculated over all bits of the Address, Control, Protocol, Information and Padding fields, not including any start and stop bits (asynchronous) nor any bits (synchronous) or octets (asynchronous or synchronous) inserted for transparency. This also does not include the Flag Sequences nor the FCS field itself.

When octets are received which are flagged in the Async-Control-Character-Map, they are discarded before calculating the FCS.

For more information on the specification of the FCS, see the Appendices.

The end of the Information and Padding fields is found by locating the closing Flag Sequence and removing the Frame Check Sequence field.

Exhibit E

Exhibit E



INTERNATIONAL TELECOMMUNICATION UNION

ITU-T X.86/Y.1323

(02/2001)

TELECOMMUNICATION
STANDARDIZATION SECTOR
OF ITU

SERIES X: DATA NETWORKS AND OPEN SYSTEM COMMUNICATIONS

Public data networks – Transmission, signalling and switching

SERIES Y: GLOBAL INFORMATION INFRASTRUCTURE AND INTERNET PROTOCOL ASPECTS

Internet protocol aspects – Transport

Ethernet over LAPS

ITU-T Recommendation X.86/Y.1323

The LAPS is a physical coding sublayer, which provides point-to-point transferring over SDH virtual containers and interface rates. The supported UITS is a connectionless-mode service. The rate adaptation between LAPS and SDH is applied. It provides a mechanism that adjusts the rate of Ethernet MAC MII to SDH VC rate, and also prevents MAC frame going to SDH VC from being written to the SDH overhead since SDH and MAC are operated in the way of period and burst respectively.

7 Service facilities and protocol specifications of LAPS

The default maximum frame size of LAPS shall be capable of supporting an information field of 1600 octets (at least) for Ethernet over LAPS. The SAPI of MAC is assigned to 0xfe01 (hexadecimal). The associated service facilities and protocol specifications of LAPS are included in Annex A of ITU-T X.85/Y.1321.

NOTE – It is needed to replace "Layer 3 or network layer or IP based network", "IP packet" and "Layer 2 or data link layer" with "MAC layer", "MAC frame" and "LAPS" respectively in Annex A of ITU-T X.85/Y.1321.

8 Encapsulation

LAPS link entity accepts frames from the MAC layer through the reconciliation sublayer and an equivalent MII (Media Independent Interface). No address filtering function is used here. The format of LAPS information field is defined in the shaded region of Figure 6. Figure 7 presents the format of LAPS frame after encapsulating MAC field. The order of those octets and bits (shaded area as shown in Figure 7) is kept intact. The FCS computations of LAPS and MAC refer to ITU-T X.85/Y.1321 and IEEE 802.3 standard respectively. The function unit of Ethernet over LAPS forwards all incoming LAPS information field to its peer connected link except the originating link port, and is permitted to buffer one or more incoming frames before forwarding them. Figure 8 shows the relationship between the reconciliation sublayer/MII and LAPS/SDH.

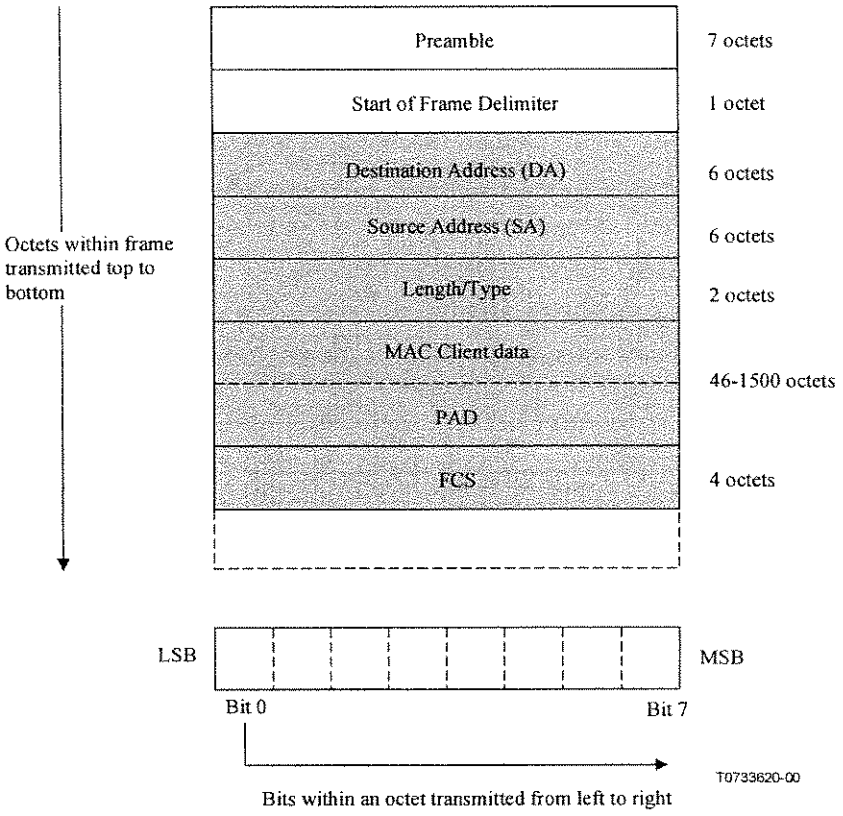


Figure 6/X.86/Y.1323 – The format of IEEE 802.3 Ethernet MAC frame, LAPS information field as defined in the shaded region

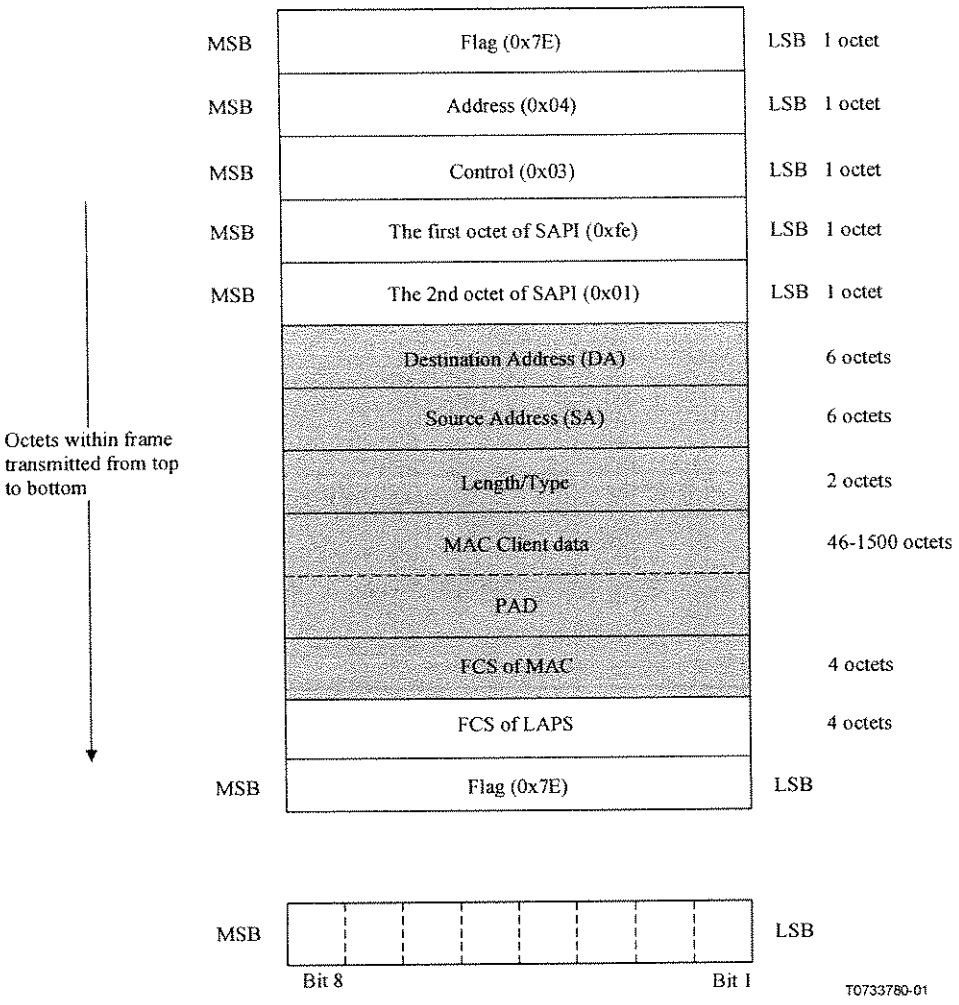


Figure 7/X.86/Y.1323 – The format of LAPS frame after encapsulating MAC field

9 Functional elements of Gigabit Ethernet over LAPS

The full-duplex is used only. The functional elements of IEEE 802.3 Ethernet, along with LAPS/SDH, are illustrated in Figure 8.

10 Rate adaptation

If the Rate Adaptation is needed in the LAPS transmit processing, transmit entity adds the rate-adaptation octet(s) "0xdd" within the frame by sending sequence(s) of {0x7d, 0xdd}. This function is performed just after transparency processing and before the end flag is added. In receive direction, receive entity will remove the Rate Adaptation octet(s) "0xdd" within the LAPS frame when detecting sequence(s) of {0x7d, 0xdd}. This function will be done just before transparency processing and after the end flag is detected.

Exhibit F

Exhibit F



INTERNATIONAL TELECOMMUNICATION UNION

ITU-T G.7041/Y.1303

(12/2001)

TELECOMMUNICATION
STANDARDIZATION SECTOR
OF ITU

SERIES G: TRANSMISSION SYSTEMS AND MEDIA, DIGITAL SYSTEMS AND NETWORKS

Digital terminal equipments – General

SERIES Y: GLOBAL INFORMATION INFRASTRUCTURE AND INTERNET PROTOCOL ASPECTS

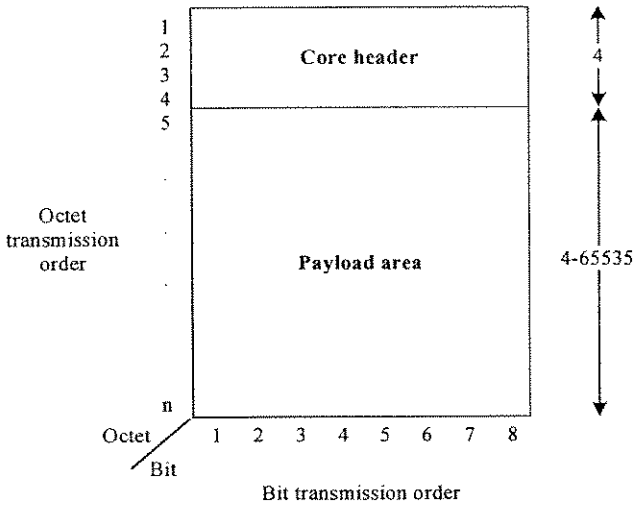
Internet protocol aspects – Transport

Generic framing procedure (GFP)

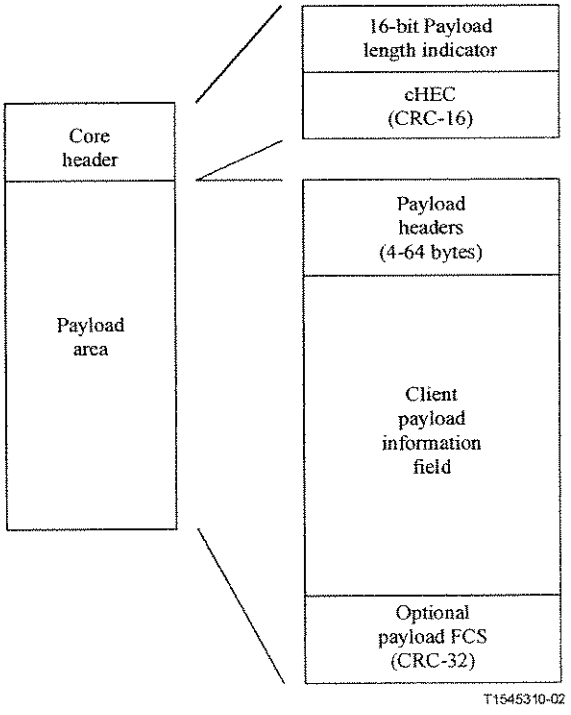
ITU-T Recommendation G.7041/Y.1303

6.1 Basic signal structure for GFP client frames

The format for GFP frames is shown in Figure 6-1. GFP frames are octet-aligned and consist of a GFP Core Header and, except for GFP Idle frames, a GFP Payload Area.



a) Frame size and transmission order



b) Fields constituting a GFP client frame

Figure 6-1/G.7041/Y.1303 – Frame format for GFP client frames

6.1.1 GFP Core Header

The GFP Core Header format is shown in Figure 6-2. The four octets of the GFP Core Header consist of a 16-bit PDU Length Indicator field and a 16-bit Core Header Error Check (cHEC) field. This header allows GFP frame delineation independent of the content of the higher layer PDUs.

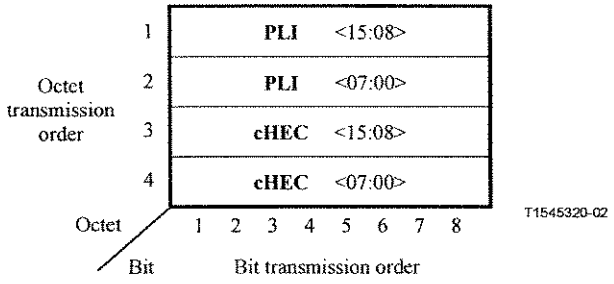


Figure 6-2/G.7041/Y.1303 – GFP Core Header format

6.1.1.1 PDU Length Indicator (PLI) field

The two-octet PLI field contains a binary number representing the number of octets in the GFP Payload Area. The absolute minimum value of the PLI field in a GFP client frame is 4 octets. PLI values 0-3 are reserved for GFP control frame usage (see 6.2).

6.1.1.2 Core HEC (cHEC) field

The two-octet Core Header Error Control field contains a CRC-16 error control code that protects the integrity of the contents of the Core Header by enabling both single-bit error correction and multi-bit error detection. The cHEC sequence is calculated over the octets of the Core Header as defined in 6.1.1.2.1.

6.1.1.2.1 HEC processing

The HEC generating polynomial is $G(x) = x^{16} + x^{12} + x^5 + 1$, with an initialization value of zero, where x^{16} corresponds to the MSB and x^0 corresponds to the LSB.

The cHEC field is generated by the source adaptation process using the following steps (see Appendix I/V.41):

1) The first two octets of the GFP frame are taken in network octet order, most significant bit first, to form a 16-bit pattern representing the coefficients of a polynomial $M(x)$ of degree 15.

2) $M(x)$ is multiplied by x^{16} and divided (modulo 2) by $G(x)$, producing a remainder $R(x)$ of degree 15 or less.

3) The coefficients of $R(x)$ are considered to be a 16-bit sequence, where x^{15} is the most significant bit.

4) This 16-bit sequence is the CRC-16 where the first bit of the CRC-16 to be transmitted is the coefficient of x^{15} and the last bit transmitted is the coefficient of x^0.

The sink adaptation process performs steps 1-3 in the same manner as the source adaptation process. In the absence of bit errors, the remainder shall be 0000 0000 0000 0000.

This single error correction shall be performed on the Core Header. The GFP sink adaptation process shall discard any of those GFP frames where multi-bit errors are detected. The sink adaptation process also updates any relevant system records for performance monitoring purposes.

6.1.1.3 Core header scrambling

The Core Header is scrambled for DC balanced by an exclusive-OR operation (modulo 2 addition) with the hexadecimal number B6AB31E0. This number is the maximum transition, minimum side-lobe, Barker-like sequence of length 32. The scrambling of the GFP Core Header improves the robustness of the GFP frame delineation procedure and provides a sufficient number of 0-1 and 1-0 transitions during idle transmission periods.

6.1.2 GFP Payload Area

The GFP Payload Area, which consists of all octets in the GFP frame after the GFP Core Header, is used to convey higher layer specific protocol information. This variable length area may include from 4 to 65 535 octets. As shown in Figure 6-3, the GFP Payload Area consists of two common components: a Payload Header and a Payload Information field. An optional Payload FCS (pFCS) field is also supported.

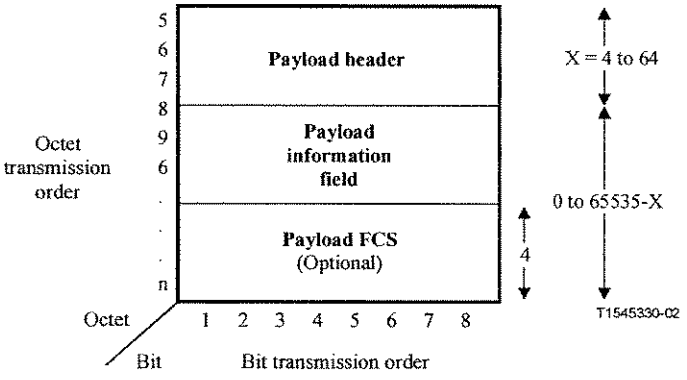


Figure 6-3/G.7041/Y.1303 – GFP Payload Area format

Practical GFP MTU sizes for the GFP Payload Area are application specific. An implementation should support transmission and reception of GFP frames with GFP Payload Areas of at least 1600 octets. By prior arrangement, consenting GFP implementations may use other MTU values.

6.1.2.1 Payload Header

The Payload Header is a variable-length area, 4 to 64 octets long, intended to support data link management procedures specific to the higher-layer client signal. The structure of the GFP Payload Header is illustrated in Figure 6-4. The area contains two mandatory fields, the Type and the tHEC fields, and a variable number of additional payload header fields. This group of additional payload header fields are referred to as the Extension Header. The presence of the Extension Header, and its format, and the presence of the optional Payload FCS are specified by the Type field. The tHEC protects the integrity of the Type field.

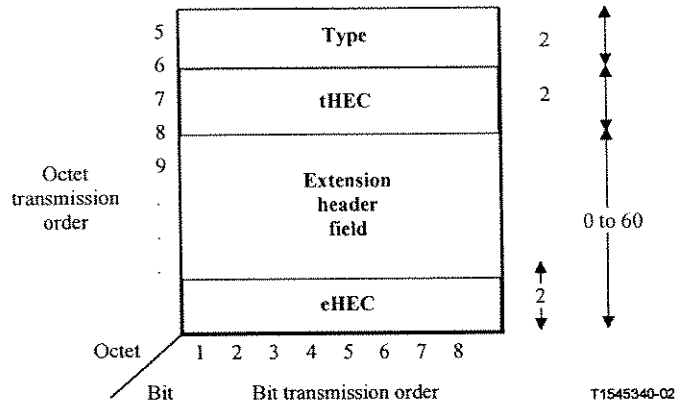


Figure 6-4/G.7041/Y.1303 – GFP Payload Header format

An implementation shall support reception of a GFP frame with a Payload Header of any length in the range 4 to 64 octets.

6.1.2.1.1 GFP Type field

The GFP Type field is a mandatory two-octet field of the Payload Header that indicates the content and format of the GFP Payload Information field (see 6.1.2.2). The Type field distinguishes between GFP frame types and between different services in a multi-service environment. As shown in Figure 6-5, the Type field consists of a Payload Type Identifier (PTI), a Payload FCS Indicator (PFI), a Extension Header Identifier (EXI) and a User Payload Identifier (UPI).

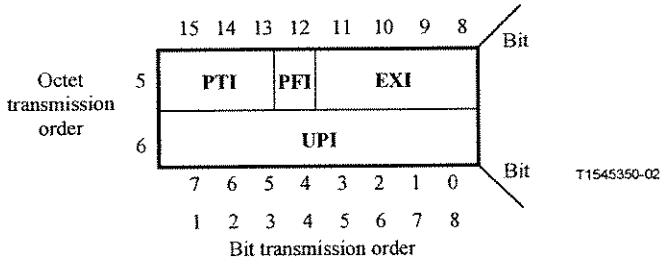


Figure 6-5/G.7041/Y.1303 – GFP Type Field format

The interpretation of the UPI field for PTI values different from 000 or 100 is for further study. Sample Type field values are illustrated in Appendix II.

6.1.2.1.1.1 Payload Type Identifier

A 3-bit subfield of the Type field identifying the type of GFP client frame. Two kinds of client frames are currently defined, User Data frames (PTI = 000) and Client Management frames (PTI = 100). PTI codepoints are given in Table 6-1.

Exhibit G

Exhibit G

Network Working Group
Request for Comments: 3032
Category: Standards Track

E. Rosen
D. Tappan
G. Fedorkow
Cisco Systems, Inc.
Y. Rekhter
Juniper Networks
D. Farinacci
T. Li
Procket Networks, Inc.
A. Conta
TranSwitch Corporation

January 2001

MPLS Label Stack Encoding

Status of this Memo

This document specifies an Internet standards track protocol for the Internet community, and requests discussion and suggestions for improvements. Please refer to the current edition of the "Internet Official Protocol Standards" (STD 1) for the standardization state and status of this protocol. Distribution of this memo is unlimited.

Copyright Notice

Copyright (C) The Internet Society (2001). All Rights Reserved.

Abstract

"Multi-Protocol Label Switching (MPLS)" [1] requires a set of procedures for augmenting network layer packets with "label stacks", thereby turning them into "labeled packets". Routers which support MPLS are known as "Label Switching Routers", or "LSRs". In order to transmit a labeled packet on a particular data link, an LSR must support an encoding technique which, given a label stack and a network layer packet, produces a labeled packet. This document specifies the encoding to be used by an LSR in order to transmit labeled packets on Point-to-Point Protocol (PPP) data links, on LAN data links, and possibly on other data links as well. On some data links, the label at the top of the stack may be encoded in a different manner, but the techniques described here MUST be used to encode the remainder of the label stack. This document also specifies rules and procedures for processing the various fields of the label stack encoding.

Table of Contents

1 Introduction 2
1.1 Specification of Requirements 3
2 The Label Stack 3
2.1 Encoding the Label Stack 3
2.2 Determining the Network Layer Protocol 5
2.3 Generating ICMP Messages for Labeled IP Packets 6
2.3.1 Tunneling through a Transit Routing Domain 7
2.3.2 Tunneling Private Addresses through a Public Backbone .. 7
2.4 Processing the Time to Live Field 8
2.4.1 Definitions 8
2.4.2 Protocol-independent rules 8
2.4.3 IP-dependent rules 9
2.4.4 Translating Between Different Encapsulations 9
3 Fragmentation and Path MTU Discovery 10
3.1 Terminology 11
3.2 Maximum Initially Labeled IP Datagram Size 12
3.3 When are Labeled IP Datagrams Too Big? 13
3.4 Processing Labeled IPv4 Datagrams which are Too Big 13
3.5 Processing Labeled IPv6 Datagrams which are Too Big 14
3.6 Implications with respect to Path MTU Discovery 15
4 Transporting Labeled Packets over PPP 16
4.1 Introduction 16
4.2 A PPP Network Control Protocol for MPLS 17
4.3 Sending Labeled Packets 18
4.4 Label Switching Control Protocol Configuration Options . 18
5 Transporting Labeled Packets over LAN Media 18
6 IANA Considerations 19
7 Security Considerations 19
8 Intellectual Property 19
9 Authors' Addresses 20
10 References 22
11 Full Copyright Statement 23

1. Introduction

"Multi-Protocol Label Switching (MPLS)" [1] requires a set of procedures for augmenting network layer packets with "label stacks", thereby turning them into "labeled packets". Routers which support MPLS are known as "Label Switching Routers", or "LSRs". In order to transmit a labeled packet on a particular data link, an LSR must support an encoding technique which, given a label stack and a network layer packet, produces a labeled packet.

This document specifies the encoding to be used by an LSR in order to transmit labeled packets on PPP data links and on LAN data links. The specified encoding may also be useful for other data links as well.

This document also specifies rules and procedures for processing the various fields of the label stack encoding. Since MPLS is independent of any particular network layer protocol, the majority of such procedures are also protocol-independent. A few, however, do differ for different protocols. In this document, we specify the protocol-independent procedures, and we specify the protocol-dependent procedures for IPv4 and IPv6.

LSRs that are implemented on certain switching devices (such as ATM switches) may use different encoding techniques for encoding the top one or two entries of the label stack. When the label stack has additional entries, however, the encoding technique described in this document MUST be used for the additional label stack entries.

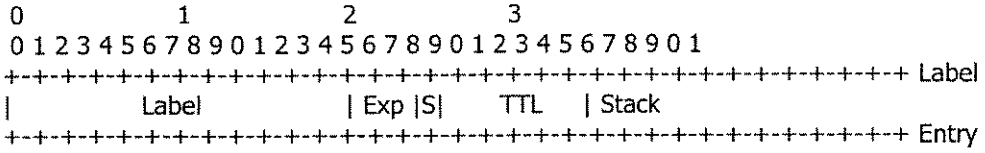
1.1. Specification of Requirements

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [2].

2. The Label Stack

2.1. Encoding the Label Stack

The label stack is represented as a sequence of "label stack entries". Each label stack entry is represented by 4 octets. This is shown in Figure 1.



Label: Label Value, 20 bits
Exp: Experimental Use, 3 bits
S: Bottom of Stack, 1 bit
TTL: Time to Live, 8 bits

Figure 1

The label stack entries appear AFTER the data link layer headers, but BEFORE any network layer headers. The top of the label stack appears earliest in the packet, and the bottom appears latest. The network layer packet immediately follows the label stack entry which has the S bit set.

Each label stack entry is broken down into the following fields:

1. Bottom of Stack (S)

   This bit is set to one for the last entry in the label stack (i.e., for the bottom of the stack), and zero for all other label stack entries.

2. Time to Live (TTL)

   This eight-bit field is used to encode a time-to-live value. The processing of this field is described in section 2.4.

3. Experimental Use

   This three-bit field is reserved for experimental use.

4. Label Value

   This 20-bit field carries the actual value of the Label.

   When a labeled packet is received, the label value at the top of the stack is looked up. As a result of a successful lookup one learns:

   a) the next hop to which the packet is to be forwarded;

   b) the operation to be performed on the label stack before forwarding; this operation may be to replace the top label stack entry with another, or to pop an entry off the label stack, or to replace the top label stack entry and then to push one or more additional entries on the label stack.

   In addition to learning the next hop and the label stack operation, one may also learn the outgoing data link encapsulation, and possibly other information which is needed in order to properly forward the packet.

There are several reserved label values:

i. A value of 0 represents the "IPv4 Explicit NULL Label". This label value is only legal at the bottom of the label stack. It indicates that the label stack must be popped, and the forwarding of the packet must then be based on the IPv4 header.

ii. A value of 1 represents the "Router Alert Label". This label value is legal anywhere in the label stack except at the bottom. When a received packet contains this label value at the top of the label stack, it is delivered to a local software module for processing. The actual forwarding of the packet is determined by the label beneath it in the stack. However, if the packet is forwarded further, the Router Alert Label should be pushed back onto the label stack before forwarding. The use of this label is analogous to the use of the "Router Alert Option" in IP packets [5]. Since this label cannot occur at the bottom of the stack, it is not associated with a particular network layer protocol.

iii. A value of 2 represents the "IPv6 Explicit NULL Label". This label value is only legal at the bottom of the label stack. It indicates that the label stack must be popped, and the forwarding of the packet must then be based on the IPv6 header.

iv. A value of 3 represents the "Implicit NULL Label". This is a label that an LSR may assign and distribute, but which never actually appears in the encapsulation. When an LSR would otherwise replace the label at the top of the stack with a new label, but the new label is "Implicit NULL", the LSR will pop the stack instead of doing the replacement. Although this value may never appear in the encapsulation, it needs to be specified in the Label Distribution Protocol, so a value is reserved.

v. Values 4-15 are reserved.

2.2. Determining the Network Layer Protocol

When the last label is popped from a packet's label stack (resulting in the stack being emptied), further processing of the packet is based on the packet's network layer header. The LSR which pops the last label off the stack must therefore be able to identify the packet's network layer protocol. However, the label stack does not contain any field which explicitly identifies the network layer