Infrared Data Association

Serial Infrared Link Access Protocol (IrLAP)

Version 1.1

June 16, 1996

IBM Corporation
Hewlett-Packard Company
Apple Computer, Inc.
Counterpoint Systems Foundry, Inc.

Authors:

Timothy Williams, Peter Hortensius, Frank Novak (IBM Corporation)
Kevin Smith (Hewlett-Packard Company)
David Suvak (Hewlett-Packard Company, Counterpoint Systems Foundry, Inc.)
Mike Cremer (Apple Computer, Inc.)

Significant Contributors:

Darrell Sell (Connexus)

Document Status

Version 1.0: This document has been submitted to the IrDA committee for a vote of approval. It was unanimously approved. The authors reserve the right to make small editing changes and to complete section 0 8.
Appendix B (Point-To-Multipoint).

Version 1.1: This version has been approved by the IrDA. It incorporates all errata submitted for version 1.0 including the changes necessary for adding the higher speeds 576kbps, 1.152Mbps and 4.0Mbps.

INFRARED DATA ASSOCIATION (IrDA) - NOTICE TO THE TRADE -

SUMMARY:

Following is the notice of conditions and understandings upon which this document is made available to members and non-members of the Infrared Data Association.

- Availability of Publications, Updates and Notices
- Full Copyright Claims Must be Honored
- Controlled Distribution Privileges for IrDA Members Only
- Trademarks of IrDA - Prohibitions and Authorized Use
- No Representation of Third Party Rights
- Limitation of Liability
- Disclaimer of Warranty
- Certification of Products Requires Specific Authorization from IrDA after Product Testing for IrDA Specification Conformance

IrDA PUBLICATIONS and UPDATES:

IrDA publications, including notifications, updates, and revisions, are accessed electronically by IrDA members in good standing during the course of each year as a benefit of annual IrDA membership. Electronic copies are available to the public on the IrDA web site located at irda.org. IrDA publications are available to non-IrDA members for a pre-paid fee. Requests for publications, membership applications or more information should be addressed to: Infrared Data Association, P.O. Box 3883, Walnut Creek, California, U.S.A. 94598; or e-mail address: info@irda.org; or by calling John LaRoche at (510) 943-6546 or faxing requests to (510) 934-5600.

COPYRIGHT:

1. Prohibitions: IrDA claims copyright in all IrDA publications. Any unauthorized reproduction, distribution, display or modification, in whole or in part, is strictly prohibited.

2. Authorized Use: Any authorized use of IrDA publications (in whole or in part) is under NONEXCLUSIVE USE LICENSE ONLY. No rights to sublicense, assign or transfer the license are granted and any attempt to do so is void.

DISTRIBUTION PRIVILEGES for IrDA MEMBERS ONLY:

IrDA Members Limited Reproduction and Distribution Privilege: A limited privilege of reproduction and distribution of IrDA copyrighted publications is granted to IrDA members in good standing and for sole purpose of reasonable reproduction and distribution to non-IrDA members who are engaged by contract with an IrDA member for the development of IrDA certified products. Reproduction and distribution by the non-IrDA member is strictly prohibited.

TRANSACTION NOTICE to IrDA MEMBERS ONLY:

Each and every copy made for distribution under the limited reproduction and distribution privilege shall be conspicuously marked with the name of the IrDA member and the name of the receiving party. Upon reproduction for distribution, the distributing IrDA member shall promptly notify IrDA (in writing or by e-mail) of the identity of the receiving party.

A failure to comply with the notification requirement to IrDA shall render the reproduction and distribution unauthorized and IrDA may take appropriate action to enforce its copyright, including but not limited to, the termination of the limited reproduction and distribution privilege and IrDA membership of the non-complying member.

TRADEMARKS:

1. Prohibitions: IrDA claims exclusive rights in its trade names, trademarks, service marks, collective membership marks and certification marks (hereinafter collectively "trademarks"), including but not limited to the following trademarks: INFRARED DATA ASSOCIATION (wordmark alone and with IR logo), IrDA (acronym mark alone and with IR logo), IR logo, IR DATA CERTIFIED (composite mark), and MEMBER IrDA (wordmark alone and with IR logo). Any unauthorized use of IrDA trademarks is strictly prohibited.

2. Authorized Use: Any authorized use of a IrDA collective membership mark or certification mark is by NONEXCLUSIVE USE LICENSE ONLY. No rights to sublicense, assign or transfer the license are granted and any attempt to do so is void.

NO REPRESENTATION of THIRD PARTY RIGHTS:

IrDA makes no representation or warranty whatsoever with regard to IrDA member or third party ownership, licensing or infringement/non-infringement of intellectual property rights. Each recipient of IrDA publications, whether or not an IrDA member, should seek the independent advice of legal counsel with regard to any possible violation of third party rights arising out of the use, attempted use, reproduction, distribution or public display of IrDA publications.

IrDA assumes no obligation or responsibility whatsoever to advise its members or non-members who receive or are about to receive IrDA publications of the chance of infringement or violation of any right of an IrDA member or third party arising out of the use, attempted use, reproduction, distribution or display of IrDA publications.

LIMITATION of LIABILITY:

BY ANY ACTUAL OR ATTEMPTED USE, REPRODUCTION, DISTRIBUTION OR PUBLIC DISPLAY OF ANY IrDA PUBLICATION, ANY PARTICIPANT IN SUCH REAL OR ATTEMPTED ACTS, WHETHER OR NOT A MEMBER OF IrDA, AGREES TO ASSUME ANY AND ALL RISK ASSOCIATED WITH SUCH ACTS, INCLUDING BUT NOT LIMITED TO LOST PROFITS, LOST SAVINGS, OR OTHER CONSEQUENTIAL, SPECIAL, INCIDENTAL OR PUNITIVE DAMAGES. IrDA SHALL HAVE NO LIABILITY WHATSOEVER FOR SUCH ACTS NOR FOR THE CONTENT, ACCURACY OR LEVEL OF ISSUE OF AN IrDA PUBLICATION.

DISCLAIMER of WARRANTY:

All IrDA publications are provided "AS IS" and without warranty of any kind. IrDA (and each of its members, wholly and collectively, hereinafter "IrDA") EXPRESSLY DISCLAIM ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE AND WARRANTY OF NON-INFRINGEMENT OF INTELLECTUAL PROPERTY RIGHTS.
IrDA DOES NOT WARRANT THAT ITS PUBLICATIONS WILL MEET YOUR REQUIREMENTS OR THAT ANY USE OF A PUBLICATION WILL BE UN-INTERRUPTED OR ERROR FREE, OR THAT DEFECTS WILL BE CORRECTED. FURTHERMORE, IrDA DOES NOT WARRANT OR MAKE ANY REPRESENTATIONS REGARDING USE OR THE RESULTS OR THE USE OF IrDA PUBLICATIONS IN TERMS OF THEIR CORRECTNESS, ACCURACY, RELIABILITY, OR OTHERWISE. NO ORAL OR WRITTEN PUBLICATION OR ADVICE OF A REPRESENTATIVE (OR MEMBER) OF IrDA SHALL CREATE A WARRANTY OR IN ANY WAY INCREASE THE SCOPE OF THIS WARRANTY.

LIMITED MEDIA WARRANTY:

IrDA warrants ONLY the media upon which any publication is recorded to be free from defects in materials and workmanship under normal use for a period of ninety (90) days from the date of distribution as evidenced by the distribution records of IrDA. IrDA's entire liability and recipient's exclusive remedy will be replacement of the media not meeting this limited warranty and which is returned to IrDA. IrDA shall have no responsibility to replace media damaged by accident, abuse or misapplication. ANY IMPLIED WARRANTIES ON THE MEDIA, INCLUDING THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE, ARE LIMITED IN DURATION TO NINETY (90) DAYS FROM THE DATE OF DELIVERY. THIS WARRANTY GIVES YOU SPECIFIC LEGAL RIGHTS, AND YOU MAY ALSO HAVE OTHER RIGHTS WHICH VARY FROM PLACE TO PLACE.

CERTIFICATION and GENERAL:

Membership in IrDA or use of IrDA publications does NOT constitute IrDA compliance. It is the sole responsibility of each manufacturer, whether or not an IrDA member, to obtain product compliance in accordance with IrDA rules for compliance.

All rights, prohibitions of right, agreements and terms and conditions regarding use of IrDA publications and IrDA rules for compliance of products are governed by the laws and regulations of the United States. However, each manufacturer is solely responsible for compliance with the import/export laws of the countries in which they conduct business. The information contained in this document is provided as is and is subject to change without notice.

TABLE OF CONTENTS

1. Introduction

1.1 Scope and Purpose

This specification is one of a family of specifications intended to facilitate the interconnection of computers and peripherals using a directed half duplex serial infrared physical communications medium such as that provided by the IrDA serial infrared physical layer.

This specification describes the functions, features, protocol and services for interconnection between computers at the data link layer (OSI layer 2). Interconnection at other layers is described in other specifications.

The data link layer protocol specified here is based on pre-existing standard asynchronous HDLC and SDLC half duplex protocols as used on multi-drop links (see references). The major modifications to these standard protocols are as follows:

- Addressing is extended to account for the mobile, ad-hoc nature of the medium
- Since the various wrapping frame methods are independent of the payload protocols, the wrapping frame protocols have been segregated into an appendix. As wrapping frame implementations develop, the changes can be appended to the IrLAP protocols without affecting the main document
- A dynamic address conflict resolution procedure is introduced
- Recovery mechanisms are extended to account for the mobile, ad-hoc nature of the medium
- A dynamic station discovery/identification procedure is introduced
- Connection setup is extended to include a negotiation framework which stations use to establish the best connection characteristics that both connecting parties can support
- Any station can contend to become a primary station
- Medium access rules are extended to resolve contention between stations competing for control of the medium and to prevent hidden node transmissions.

This data link protocol will be referred to as IrLAP.

IrLAP constitutes one layer in a hierarchical stack of communication protocol layers. It uses services provided by the physical layer and provides services to the layer above it -- referred to as “The Upper Layer” and “The Service User (Layer)” in this document.

A complete protocol specification must define the following five elements:

- The *services* the protocol will provide.
- The *assumptions* made about the environment in which the protocol will be executed.
- The *vocabulary* of messages used to implement the protocol.
- The *encoding* (format) of each message in the vocabulary.
- The *procedure rules* that guard the consistency of message exchanges.

In this specification these elements are addressed as follows:

- Section 2.0 *Service Specifications* specifies the services provided by IrLAP to the service user layer.

- Section 3.0 *Environmental Characteristics* specifies the assumptions that IrLAP makes about the IrDA physical layer.

- Section 4.0 *Frame Structure* specifies the general encoding rules for all IrLAP frames.

- Section 5.0 *Elements of Procedure* specifies all the valid types of IrLAP frames (the vocabulary of frames).

- Section 6.0 *Description of Procedures* specifies the procedural rules that govern all IrLAP frame exchanges.

1.2 References

[ISO4335] ISO 4335 High Level Data Link Control (HDLC) Procedures - Elements of Procedures 1991-09-15

[ISO8885] ISO 8885 High Level Data Link Control (HDLC) Procedures - General Purpose XID Frame Information Field Content and Format 1991-06-01

[ISO3309] ISO 3309 High Level Data Link Control (HDLC) Procedures - Frame Structure 1991-06-01

[ISO3309-2] ISO 3309 Amendment 2 High Level Data Link Control (HDLC) Procedures - Frame Structure 1991-06-01

[ISO8886] ISO 8886 Information technology - Telecommunications and information exchange between systems - Data link service definition for Open Systems Interconnection 1992-06-15

1.3 Acronyms and Definitions

A = Address byte in IrLAP frame which is the first byte after the BOF
Abort = prematurely terminating a frame
API = Application Program Interface
BOF = Beginning Of Frame
bps = bits per second
C = Control byte in IrLAP frame which is the second byte after the BOF
C/R bit = Command/Response bit
CCITT = International Telegraph & Telephone Consultative Committee
CE byte = Control Escape byte (X'7D')
CRC = Cyclic Redundancy Check
CRC-16 = CRC with polynomial equal to $X^{16} + X^{15} + X^2 + 1$
CRC-CCITT = CRC with polynomial equal to $X^{16} + X^{12} + X^5 + 1$
EOF = End Of Frame
FCS = Frame Check Sequence
FI = Format Identifier
GI = Group Identifier
HDLC = High-level Data Link Control
ID = Identification
IEC = International Electrotechnical Commission
I-frames = Information frames
ISO = International Organization for Standardization
LSB = Least Significant Bit
MAC = Media Access Control
ms = millisecond
MSB = Most Significant Bit
NDM = Normal Disconnect Mode
Nr = sequence number of next frame expected
NRM = Normal Response Mode

Ns = sequence number of frame sent
NULL = having all zero elements
OSI = Open Systems Interconnection
S-frames = Supervisory frames
SDLC = Synchronous Data Link Control
SIR = Serial Infrared
STA = Start flag
STO = End flag
us = microsecond
U-frames = Unnumbered frames

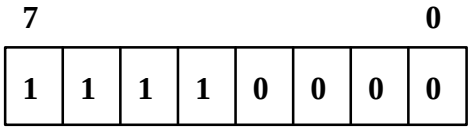
1.4 Bit and Byte Ordering

This document represents frames as collections of octets (bytes). Each byte is composed of 8 bits numbered 0 - 7 where 0 is always the least significant bit (LSB) and 7 is always the most significant bit (MSB). In some cases frames contain larger components that are composed of multiple bytes. These larger components are represented as *n* * 8 bits where *n* is the number of bytes. Usually the least significant bit is numbered bit 0 of byte 0 while the most significant bit is numbered bit 7 of byte (*n*-1). Sometimes the least significant bit is numbered 0 while the most significant bit is numbered (8 * *n)* - 1. The least significant byte of a multiple byte component is defined to be the byte that contains bits 0 - 7. Bytes are represented throughout the document in the following forms:

- Diagram form - a byte is represented as a rectangle with slots for each bit. The leftmost slot contains the most significant bit and rightmost slot contains the least significant bit.
- Binary form - a byte is represented as a sequence of 8 digits (1 or 0) with least significant bit on the right and most significant bit on the left.
- Hex form - a byte is represented with two hex digits with least significant nibble on the right and most significant nibble on the left.
- Multiple byte from - is represented as a rectangle with slots for each byte. The least significant byte is on the left and the most significant byte is on the right.

Examples of each representation for the hex value X'F0' is shown below. The multiple byte example shows a three bytes sequence of X'F0', X'F1', and X'F2':

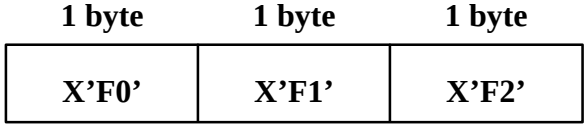
Diagram form



Binary form - B'11110000'

Hex form - X'F0'

Multiple byte form



2. Data Link Layer Service Specifications

This section describes the services provided by the data link layer to the upper layer. The services are specified in terms of service primitives and parameters. The service primitives are an abstraction in that they specify only the service provided rather than the means by which the service is provided. This definition of service is independent of any specific interface implementation. These primitives do not constitute an API. This document does not provide a summary of service primitive time-sequence diagrams. IrLAP adheres closely to the time-sequence diagrams given in [ISO 8886]. See that document for further details.

IrLAP provides two general types of services:

- Connectionless Services
- Connection-oriented Services

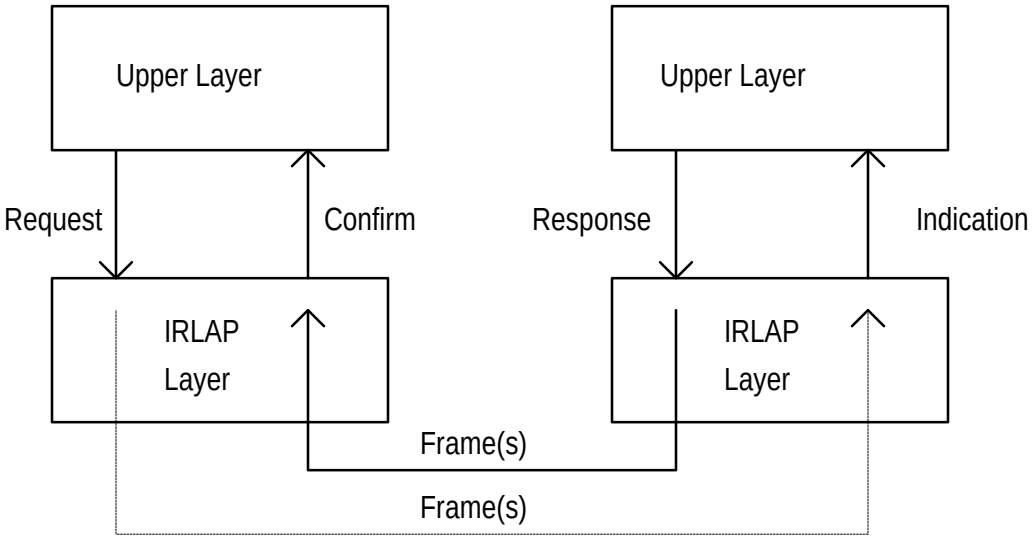
2.1 IrLAP Service Definitions

IrLAP employs four generic types of service primitive:

1. Request: Passed from the Upper Layer to invoke a service.
2. Indication: Passed from IrLAP to the Upper Layer to indicate an event or to notify the Upper Layer of an IrLAP initiated action.
3. Response: Passed from the Upper Layer to acknowledge some procedure invoked by an indication primitive.
4. Confirm: Passed from IrLAP to the Upper Layer to convey the results of the previous service request.

IrLAP uses these primitives to communicate with the upper layer in order to manage the communications processes on the link between devices.

These primitives are shown graphically here.



2.2 Connectionless Services

2.2.1 **Discovery Services**

- IrLAP_DISCOVERY.request
- IrLAP_DISCOVERY.indication (*Discovery-Log*)
- IrLAP_DISCOVERY.confirm (*List-of-Discovery-Logs*)

Description: The *request* primitive is used to find out what, if any, devices are within communication range and are available for connections. A list of the available devices is returned with the matching *confirm* primitive. A device that is discovered by another device's request primitive issues an unsolicited *indication* primitive with information about the device that issued the request primitive.

Parameters:

Discovery-Log	= *solicited* + *sniff* + *device-address* + *IrLAP-version* + *discovery-info*
List-of-Discovery-Logs	= { *Discovery-Log* }
Solicited	= [true | false] * Information about other devices can be learned in two ways, solicited and unsolicited. Solicited discovery occurs when a request primitive is issued Unsolicited discovery occurs since devices that initiate discovery also provide information about themselves. This flag indicates the manner by which the device information was learned.*
Sniff	= [true | false] * This lets the upper layer know if the discovered device is a sniffing device.*
Device Address	= *This is an IrLAP 32 bit device address*
IrLAP-version	= [0..255] *Version number of responder's IrLAP layer*
Discovery-Info	= *This is a field, up to 32 bytes long, whose content is specified by the service user layer.*

2.2.2 **Address Conflict Services**

- IrLAP_NEW_ADDRESS.request (*Device-Adr*)
- IrLAP_NEW_ADDRESS.confirm (*List-of-Discovery-Logs*)

Description: The Address Conflict services are used resolve device address conflicts. Following a discovery operation if the discovery log contains entries for more than one device with the same device address, this request service primitive may be invoked in order to cause the IrLAP layers of the conflicting devices to select new non-conflicting device addresses.

The confirm Discovery-Logs are as described in discovery services except only devices with the conflicting address will respond, they contain the new device addresses.

Parameters:

Device-Adr	=	*This is an IrLAP 32-bit device address*
List-of-Discovery-Logs	=	*See Discovery (sec. 2.2.1) above*

2.2.3 Unit Data Services

- IrLAP_UNITDATA.request (*User-Data*)
- IrLAP_UNITDATA.indication (*User-Data*)

Description: The UNITDATA service primitives provide a way to transmit data outside of a connection. This data transmission is unreliable. All data is sent "broadcast" and cannot be directed to a specific device address. The request primitive is passed to IrLAP to cause data to be transmitted; the indication primitive is passed from IrLAP to indicate that data has been received.

Parameters:

User-Data = * Up to 384 bytes of data*

2.3 Connection Oriented Services

2.3.1 Connect Services

- IrLAP_CONNECT.request (*Target-Device-Adr, Requested-QOS, Sniff*)
- IrLAP_CONNECT.indication (*Source-Device-Adr, Connection-Handle, Returned-QOS*)
- IrLAP_CONNECT.response (*Source-Device-Adr, Connection-Handle, Requested-QOS*)
- IrLAP_CONNECT.confirm (*Connection-Handle, Returned-QOS*)

Description: The *request* primitive is used to request that an IrLAP connection be established to a station with device address *Target-Device-Adr* and quality of service *Requested-QOS.* If the *Sniff* flag is set true then the connection is being attempted to a device that is using a special mode called "Sniffing". Both the *Target-Device-Adr* and the sniffing requirement are determined from the log returned by the discovery services. The *indication* primitive to the Upper Layer of the target device provides the Device Address, *Source-Device-Adr*, of the station requesting the connection and a connection handle and quality of service parameter, both of which become valid if the station chooses to accept the connection by issuing the affirmative *response* primitive. The *confirm* primitive is returned on successful establishment of the connection. After this, all primitives refer to the established connection by the Connection Handle.

Parameters:

Target-Device-Adr	= **An IrLAP 32-bit device address**
Source-Device-Adr	= **An IrLAP 32-bit device address**
Connection-Handle	= *An IrLAP 7-bit connection handle*
Sniff	= [true | false]
Requested-QOS	= Baud-Rate + Max-Turn-Around-Time + Disconnect-Threshold + Data-Size
Returned-QOS	= Baud-Rate + Data-Size + Disconnect-Threshold
Max-Turn-Around-Time	= *See Negotiation section*
Disconnect-Threshold	= *See Negotiation section*
Baud-Rate	= [9600 | 19200 | 38400 | 57600 | 115200 | 576000 | 1152000 | 4000000]
Data-Size	= [64 | 128 | 256 | 512 | 1024 | 2048]

2.3.2 Sniffing Services

- IrLAP_SNIFF.request (*Cancel*)

Description: This Sniff *request* primitive is used to initiate or cancel the special low power connect procedure (sniffing). A sniff request can be canceled by issuing a request primitive with the *Cancel* flag set to true. The IrLAP_CONNECT.indication primitive is returned by IrLAP when a connection is successfully established.

Parameters:

Cancel = [true | false]

2.3.3 Data Services

- IrLAP_DATA.request (*Connection-Handle, User-Data, Expedited-Unreliable-Flag*)
- IrLAP_DATA.indication (*Connection-Handle, User-Data, Expedited-Unreliable-Flag*)

Description: Data can either be sent as reliable, sequenced data or as unreliable, expedited, unsequenced data. This is differentiated through the *Expedited-Unreliable-Flag*. Note that no confirmation primitives are returned to the sender even for reliable data. The IrLAP layer will deliver reliable data error-free and in the proper order. The request primitive is used to request IrLAP to transmit the supplied user data. IrLAP uses the indication primitive to pass received user data to the upper layer.

Parameters:

Connection-Handle	= *An IrLAP 7-bit Connection Handle*
User-Data	= *The number of bytes of user data may not exceed the Data-Size Quality of Service parameter returned for this connection handle*
Expedited-Unreliable-Flag	= [true | false] * true indicates data to be sent unreliably*

2.3.4 Status Services

- IrLAP_STATUS.request(*Connection-Handle*)
- IrLAP_STATUS.indication(*Connection-Handle, Quality-of-Link*)
- IrLAP_STATUS.confirm(*Connection-Handle, Unacked-Data-Flag*)

Description: IrLAP uses the status indication to inform the upper layer that the quality of the link is suspect. Either the link is experiencing high levels of noise or all connection activity has ceased. If the link quality does not improve then a spontaneous IrLAP_DISCONNECT.indication is likely. IrLAP uses the request and indication primitives to provide the upper layer information about unacknowledged “send” data. If there is any unacknowledged data that hasn’t yet been successfully transmitted, the Unacked Data Flag is set true. It is set false otherwise. This does not affect the transmission of the data. This is just a “peek” mechanism for the upper layer.

Parameters:

Connection-Handle	= *An IrLAP 7-bit connection handle*
Quality-of-Link	= [no-activity | noisy]
Unacked-Data-Flag	= [true | false] * true indicates IrLAP layer has unacked data to be sent

2.3.5 Reset Services

- IrLAP_RESET.request (*Connection-Handle*)
- IrLAP_RESET.indication (*Connection-Handle*)
- IrLAP_RESET.response (*Connection-Handle, accept*)
- IrLAP_RESET.confirm (*Connection-Handle, accept*)

Description: A reset causes all unacknowledged data units to be discarded. All counters and timers are reset. A reset only occurs if both ends of the connection agree to it. If the response primitive indicates the reset is NOT accepted then the reset has no effect on the connection.

Parameters:

Connection-Handle	=	*An IrLAP 7-bit connection handle*
accept	=	[true | false] *if false, the reset does not occur*

2.3.6 Disconnection Services

- IrLAP_DISCONNECT.request(*Connection Handle*)
- IrLAP_DISCONNECT.indication(*Connection Handle, Unacked-Data*)

Description: A disconnect request terminates the logical connection and all outstanding data units are discarded. No confirm primitive is needed since the disconnect is always successful. The Unacked-Data parameter in the indication primitive contains information about any data that was unacknowledged when the disconnection occurred.

Parameters:

Connection-Handle	=	*An IrLAP 7-bit connection handle*
Unacked-Data	=	*implementation specific information regarding un-sent data*

3. Environmental and Operational Characteristics

3.1 Configurations and Operating Characteristics

The IrDA infrared physical layer characteristics affect the rules needed to access it. These characteristics are given below.

- point to point, point to multipoint
- half duplex
- hidden nodes
- narrow infra-red cone (15 degree half angle)
- It is assumed that stations will be able to detect the presence of data transmissions even if they are transmitted at a baud rate other than that for which the receiver is currently set. This "detection" may be as framing errors, overrun errors or characters. This detection is used as crude Carrier Sense.
- no collision detection

3.2 Data Link States

A data link channel can be in one of two basic states:

- Connection state
- Contention state

The data link channel is said to be in the connection state when two or more nodes have an established connection and are exchanging control and/or information frames.

The data link channel is said to be in the contention state any time it is not in the connection state. This happens as soon as a connection is disconnected or when no connection ever existed.

3.3 Unbalanced Data Link

The IrLAP protocol treats the IrDA SIR medium as an unbalanced data link due to its half duplex nature, lack of collision detection, variable speed, and various other characteristics.

An unbalanced data link involves two or more participating data stations. For control purposes one station on the data link assumes responsibility for the organization of data flow and for unrecoverable data link error conditions. The data station assuming these responsibilities is known as the primary station, and the frames it transmits are known as command frames. The other stations on the data link are known as secondary stations, and the frames they transmit are known as response frames.

All transmissions over an unbalanced data link go to or from the primary station. There is always one and only one primary station; all other stations must be secondary stations. Not all stations must have primary capability, but those which do not can only communicate with stations which have primary capability. The preferred implementation is that all stations have the capability to play the primary station role.

3.4 **Modes**

IrLAP data stations can be in either of two modes: Normal Response Mode (NRM) or Normal Disconnect Mode (NDM). These correspond to the Connection state and the Contention state respectively. Each station after entering NRM knows which role it is to play: primary station or secondary station. When in NRM stations are operational and connected, and when in NDM they are operational and disconnected.

When in NRM a secondary station will initiate transmission only as the result of receiving explicit permission to do so from the primary station. After receiving permission, the secondary will initiate a response transmission. The response transmission will consist of one or more frames. The last frame of the response transmission will be explicitly indicated by the secondary station. Following indication of the last frame, the secondary station will stop transmitting until explicit permission is again received from the primary station.

Communications in NDM are contention based. As a result, stations that wish to transmit while in NDM must use caution and follow the NDM media access rules fully.

4. IrLAP Frame Structure

4.1 General

This section defines the general format of frames exchanged by the IrLAP peer layers.

All data and control transmissions on an IrLAP data link are organized in a specific format called a frame. This format carries control information and user data between a transmitting station and a receiving station and allows a receiving station:

- To determine where the frame begins and ends
- To determine whether the frame is intended for that station
- To determine what actions to perform with the information received
- To detect the occurrence of transmission errors in received frames
- To acknowledge its receipt of frames to the transmitting station

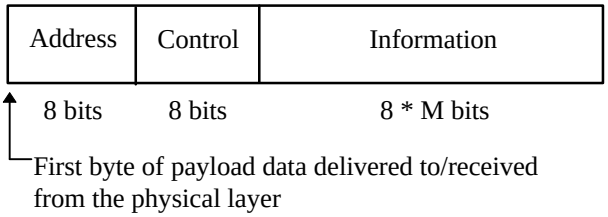
4.2 IrLAP Frame

4.2.1 Frame Format

Each IrLAP frame has a specific format:
- An address (A) field that identifies a secondary station connection address[1]
- A control (C) field that specifies the function of the particular frame
- An optional information (I) field that contains the information data

Each of these fields contains either 8 bits or a multiple of 8 bits. Together, the A, C, and I fields are referred to as the payload data.

IrLAP Frame Payload Data



4.2.2 Wrapping Layer

Each IrLAP frame is preceded and succeeded by fields which constitute the wrapping layer. The wrapping layer implements a physical layer scheme that serves to reliably transmit the payload data. The wrapping layer fields serve to mark the beginning and end of the frame and to check for the reliable transmission of

[1]Note: connection handles and connection addresses are related but are not the same entity.

data. The format of the wrapper fields will vary according to the particular physical layer scheme used, but every frame wrapper will include at least three components:

- A start flag, BOF, or STA that marks the beginning of the frame
- A frame check sequence field that allows the receiving station to check the transmission accuracy of the frame
- A stop flag, EOF, or STO that signals the end of the frame.

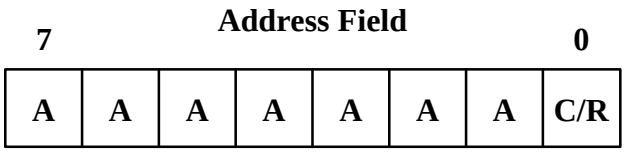
A wrapper's design and function are independent of the payload frame's function. Therefore, the two are treated separately. The descriptions of different IrLAP frame wrappers are included in section 0, 10. Appendix D (IrLAP Frame Wrappers).

4.3 Elements of the IrLAP Frame

4.3.1 Address

If the primary station is transmitting the frame then the address tells for which secondary station the frame is intended. If the secondary station is transmitting the frame, the address tells at which secondary station the frame originated.

4.3.1.1 Address Field Representation



The address field contains 7 bits of actual address (the A bits) and a command/response identifier bit (C/R bit).

If the C/R bit is one ("1") it indicates that the frame is a command frame (sent by a primary station if a connection is established or by an initiator when in the contention state). If the C/R bit is zero ("0") it indicates that the frame is a response frame (sent by a secondary station if a connection is established or a responder when in the contention state).

4.3.1.2 Address Usage

The following special addresses are defined:

- The address B'0000000' (binary) is reserved as the NULL connection address. No secondary station is assigned this address.
- The address B'1111111' is reserved as the global, or broadcast, address. This address designates a group consisting of all stations that receive the transmission. No secondary station is assigned this address. Only the primary station or a non-connected station can use the broadcast address.

4.3.2 Control Field

The control field (C) follows the address field. It defines the function of the frame. The C field has three formats: unnumbered (U) format, supervisory (S) format, or information (I) format. The corresponding frame is similarly named.

4.3.3 Information Field

Following the control field, there may or may not be an information field. Supervisory frames do not contain an information field.

Data to be transferred on the data link is contained in the information field of a frame. The information field does not have a set length, but must be a multiple of 8 bits.

5. Elements of Procedure

5.1 General

This section defines the vocabulary of the IrLAP frame types.

A frame's type or function is determined from the content of its C field. There are three general C field formats: U or unnumbered format, I or information format and S or supervisory format.

5.2 Unnumbered Format (U)

Unnumbered frames are used for such functions as:

- Establishing and disconnecting the data link
- Reporting procedural errors
- Transferring data (when the location of the data in a sequence of frames is not to be checked)

Command and response frames having a C field of this format are used for data link management. Data link management includes discovering, activating and initializing secondary stations, controlling the response mode of secondary stations, and reporting procedural errors (not recoverable by retransmission). Data may also be transmitted, in an I field, using a frame with a C field of the unnumbered format. Frames with an unnumbered format C field are not counted in the Nr or Ns counts (see Section 0 5.4 Information Transfer Format (I)).

IrLAP unnumbered format frames differ from those of standard HDLC in that some of them contain an extended attribute field in place of, or in addition to, the information field. The extended attribute field contains a source and destination device address and optional control parameters. Those frames with extended attribute fields are identified in the detailed descriptions below.

5.3 Supervisory Format (S)

Supervisory frames assist in the transfer of information, though they do not carry information themselves. They are used to acknowledge received frames, to convey ready or busy conditions, and to report frame sequencing errors.

This format is an adjunct to the information transfer format. Frames containing a C field of the supervisory format convey ready or busy conditions and may be used to report sequence errors (thus requesting retransmission). Such frames may be interspersed with frames having a C field of the information transfer format.

Whether or not a primary station has information data to transmit, it may use a frame having a C field of the supervisory format to poll a secondary station; a secondary station may use the supervisory format to respond to a request for confirmation. Frames with a supervisory format C field are not counted in the Nr or Ns counts (see Section 0 5.4 Information Transfer Format (I)).

5.4 Information Transfer Format (I)

Information frames transfer information. IrLAP procedures are designed as a vehicle for data contained in the I field. The I field contains data that is moved, via the data link, from place to place in the system. The I field is unrestricted in content. Besides indicating the format, the control field contains send and receive counts (*Ns* and *Nr*, respectively). IrLAP procedures use the Ns count to ensure that these frames

are received in their proper order; they use the Nr count to confirm that received information frames are accepted.

The Ns count indicates the number of the information frame within the sequence of information frames transmitted. The Nr count transmitted in a frame is the number (Ns) of the information frame that the station transmitting the Nr count expects to receive next. For more details refer to frame sequencing below.

Note: The Ns count is only present in a C field of the I format. An Nr count is present in C fields of both I and S format frames. Neither Ns or Nr appear in the C field of U format frames.

The I field length is a multiple of eight bits. An information field is normally included with every frame having a C field of the information transfer format. These information transfer frames are the only ones that are sequenced (counted for Nr or Ns counts). There are provisions for an I field in frames with an unnumbered format C field, but these are not supported by sequence checking.

5.5 **Frame Sequencing**

Two levels of information grouping are incorporated in IrLAP procedures. The basic level, called a “frame”, is checked for transmission errors. The frame is the vehicle for every command, every response, and all information that is transmitted using the procedures. The higher level of grouping, a frame sequence, is checked for missing or duplicated frames. Each station maintains two state variables Vs and Vr. Vs denotes the sequence number of the next sequenced frame to be transmitted. Vr denotes the sequence number of the next sequenced frame expected to be received. Sequenced frames contain two sequence numbers Ns and Nr. Ns represents the sequence number of the transmitted frame. Nr represents the sequence number of the next expected sequenced frame.

Vs is placed in the Ns field of a each frame before it is transmitted. Vs is incremented after transmission of an I-frame. The Vr count advances when an incoming I-frame is received in sequence and found to be error-free. Vr then becomes the count of the next expected frame and should agree with the next incoming Ns count. Vr is placed in the Nr field of outgoing frames. If the incoming Ns does not agree with Vr, the frame is out of sequence and Vr does not advance. Error-free out-of-sequence frames may be rejected or saved, at the option of the using system. The receiver does, however, accept the incoming Nr count (confirmation) if the out-of-sequence frame is otherwise error free.

The counting capacity for Nr or Ns is 8 using the digits 0 through 7. 7 wraps around to 0. Depending on the capabilities of the devices, up to 7 frames may be sent before the receiver reports its Nr count to the transmitter. The number of I-frames that can be sent before acknowledgment is referred to as the window size, and is described further in section 6.6, Negotiation. All unconfirmed frames must be retained by the transmitter because it may be necessary to re-send some or all of them. The reported Nr count is the sequence number of the next frame that the receiver expects to receive, so if, at a checkpoint, it is not the same as the transmitter’s next sequence number, some of the frames already sent must be repeated. The Vr and Vs counts of both stations are initialized to 0 at connect establishment and during a reset by the primary station. At other times the counts advance as sequenced frames are sent and received.

5.6 **The Poll/Final (P/F) bit**

The P/F bit occupies bit 4 of the control field in U, S, and I format frames. The P/F bit is used to control the two-way alternative access to the link when in a connection (NRM). This bit takes on two meanings based on the sending station. When sent from a primary station, it is the poll (P) bit. This is used by the primary station to solicit a response or sequence of responses from the secondary station. When sent from a secondary station, it is the final (F) bit. This is used by the secondary station to indicate the final frame transmitted as the result of the previous soliciting (poll) command.

This bit may be viewed as a mechanism for giving transmit permission on the link when in NRM. The secondary station is not allowed to transmit until it receives a command frame with the P bit set to “1”. The secondary may then send multiple frames to the primary station. The secondary sets the F bit to “1” when sending the last frame of its response transmission. This gives transmit permission back to the primary station. At that point, the secondary no longer has permission to transmit on the link.

Only the primary shall re-transmit a frame with a P bit set to “1”. The frame with the F bit set to “1” is not re-transmitted.

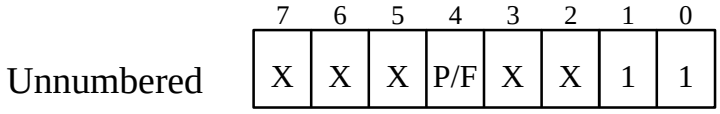
5.7 **Commands and Responses**

This section defines the commands and associated responses that are encoded into the C field. Unassigned bit configurations are reserved for future use. When one of these configurations is received by a secondary station it is a command; when it is received by a primary station it is a response.

A response may be expected to a given command, but the transmission of that response is permitted only when the secondary station is polled (when it receives a frame with the P-bit on).

5.7.1 **U (unnumbered) Format**

The C field in this format has bits 0 and 1 set to 1. These are the first C-field bits sent. Unnumbered communications are not sequence checked and do not use Nr or Ns. The mode setting non-sequenced command, SNRM resets Nr and Ns to 0. Excluding the P/F bit, the other five bits are available for encoding the commands and responses listed below:



7	6	5	4	3	2	1	0	
1	0	0	P	0	0	1	1	SNRM command
0	1	0	P	0	0	1	1	DISC command
0	0	0	P	0	0	1	1	UI command
0	0	1	P	1	1	1	1	XID command
1	1	1	P	0	0	1	1	TEST command
1	0	0	F	0	0	1	1	RNRM response
0	1	1	F	0	0	1	1	UA response
1	0	0	F	0	1	1	1	FRMR response
0	0	0	F	1	1	1	1	DM response
0	1	0	F	0	0	1	1	RD response
0	0	0	F	0	0	1	1	UI response
1	0	1	F	1	1	1	1	XID response
1	1	1	F	0	0	1	1	TEST response

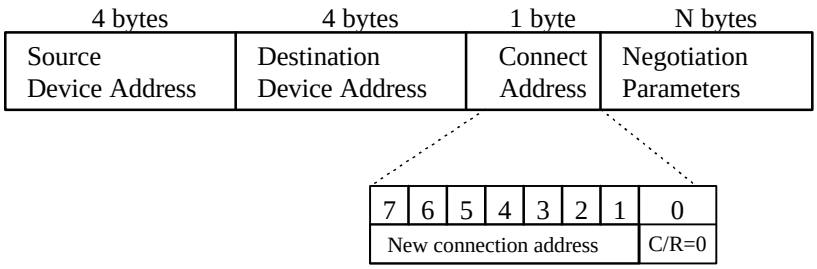
5.7.1.1 **SNRM (Set Normal Response Mode):**

This command is used to establish (or reset) a connection. Upon receipt of a UA response a connection is established (reset). When a connection is established (reset) the station that sent the SNRM command enters NRM as the primary; the station that responded with UA enters NRM as a secondary.

The A field of a SNRM frame used to establish a connection is set to X'FF', identifying it as a command (C/R bit = 1) with the broadcast address. When SNRM is used to reset a connection, the A field is set to the connection address.

When SNRM is used to establish a connection, the I field contains a source and destination device address, the new connection address, and the negotiation parameters. The order and meaning of the negotiation parameters is specified in section 6.6, Negotiation. The new connection address is in the byte following the destination device address (ninth byte of the I field). This connection address will be used in the A field of all future frames on this connection. The C/R bit of this new connection address is set to zero, but should be ignored by the receiver. When SNRM is used to reset a connection it does not contain an I-Field.

Information Field Format for SNRM Frame



5.7.1.2 DISC (Disconnect):

This command terminates a connection and places the NRM secondary station that receives it in NDM mode. The secondary station confirms the disconnection by sending the UA response. No I field is permitted with the DISC command.

5.7.1.3 UI (Unnumbered Information):

Unnumbered information frames can be sent both within a connection and outside a connection. UI frames sent outside of a connection require that the connection address contain the broadcast address (B'1111111') and the C/R bit set to 1 (command). UI frames used within a connection will contain the connection address for that connection. The I field contains information bytes only.

The maximum size of the information field of a UI frame in NRM is equal to the negotiated frame size. In NDM it is governed by the maximum amount of data that can be sent at 9600bps, which is 400bytes (see the max capacity table in section 0, 6.6.11 Negotiation Procedure). The maximum size of the information field is 384 bytes (400 bytes minus overhead for 11 BOFs, the address byte, the control byte, 2 CRC bytes and the EOF which is 400 minus 16). (The number of BOFs needed is discussed in section 6.6.7).

5.7.1.4 XID (exchange station identification):

The XID frame may be used as a command or a response. It is used for device discovery, address conflict resolution, and sniffing.

Both XID commands and responses use the same general form. The main difference between the two is the use of the command/response (C/R) bit in the address byte and the control field. The two frames are shown here.

Command XID frame

1 byte	1 byte	1 byte	
C/R = 1 Addr = X'FE'	XID Command	Format Identifier	Format Specific

Response XID frame

1 byte	1 byte	1 byte	
C/R = 0 Addr = X'FE'	XID Response	Format Identifier	Format Specific

5.7.1.4.1 Definition of XID Frame Fields

The major fields of each frame are discussed below.

5.7.1.4.1.1 Address

If the XID frame is being sent over a connection the address field contains the connection address otherwise, it is set to the broadcast address B'1111111'. The B'1111111' address distinguishes these frames from XID frames used inside a connection. The command XID frame has the C/R bit set to 1 and the response has the C/R bit set to 0. The response XID frame is also used for Sniffing (discussed later in this section).

5.7.1.4.1.2 Control Field

The control field is the standard unnumbered control byte (see sec. 5.7.1) for an XID command and response. The P/F bit should be set to 1 in both the command and the response.

5.7.1.4.1.3 Format Identifier

The format identifier differentiates between possible XID formats. The only value currently used by IrLAP is X'01' which is the Discovery XID format. All other values are reserved.

5.7.1.4.1.4 Format Specific

This field varies depending on the use of the XID frame. There are three main uses for XID frames: device discovery, address conflict resolution, and sniffing.

5.7.1.4.1.4.1 Discovery and Address Conflict Resolution Format Specific Information

Discovery and address conflict resolution frames utilize the same format as shown below. The only difference is the use of the "generate new device address" bit in the discovery flags.

Discovery and Address Conflict Resolution Command XID Format

1 byte	4 bytes	4 bytes	1 byte	1 byte	1 byte	32 bytes
FI X'01'	Source Device Address	Destination Device Address	Discovery Flags	Slot Number	Version Number	Discovery Info (final slot only)

Discovery and Address Conflict Resolution Response XID Format

1 byte	4 bytes	4 bytes	1 byte	1 byte	1 byte	32 bytes
FI X'01'	Source Device Address	Destination Device Address	Discovery Flags	Slot Number	Version Number	Discovery Info

5.7.1.4.1.4.1.1 *Source Device Address*

The source device address is the 32 bit address of the sender of the frame. This number should never be 0 or all 1's (X'FFFFFFFF').

5.7.1.4.1.4.1.2 *Destination Device Address*

The destination device address is the 32 bit address of the intended receiver of the frame. A destination address of X'FFFFFFFF' is used to refer to any device. All devices need to respond to the X'FFFFFFFF' address. The device address is also used to direct XID commands to only those devices with the given destination device address. XID responses should have the destination device address set to the address of the initiator. The exception is Sniffing. A device performing Sniffing (see section 0 5.7.1.4.1.4.2 Sniffing Format Specific Information) uses a destination device address of X'FFFFFFFF'.

5.7.1.4.1.4.1.3 *Discovery Flags*

The discovery flags are used in the command frame when the format identifier is set to X'01' to control the discovery process and resolve address conflicts. Response frames use the flags to indicate the parameters of the command frame to which they are responding and to indicate status. Bits 0 and 1 are used to indicate the number of slots. The bits have the following meaning:

Bit 1	Bit 0	meaning
0	0	**1 slot**
0	1	**6 slots**
1	0	**8 slots**
1	1	**16 slots**

Bit 2 is the "generate a new device address" indication. When set in the command frame, it indicates that all devices with the destination device address found in this frame should generate a new device address (this is the mechanism used for address conflict resolution). When set in the response frame it indicates that the device has generated a new address.

Bit 3-7 are reserved for future use. These bits must be set to 0. Devices receiving the discovery flags should ignore these bits.

5.7.1.4.1.4.1.4 *Slot Number*

Slot number is used in a discovery command frame to indicate the number of the current discovery slot. The initial discovery XID frame contains a slot number of 0. This frame starts the discovery process and marks the beginning of slot 0. Subsequent slots are marked by discovery command XID frames where the slot number is set to the corresponding slot it marks. Discovery command frames are also called "Beginning of Slot" (BOS) frames. A slot number value of X'FF' indicates the end of the discovery process. The slot number field is undefined in discovery XID response frames.

5.7.1.4.1.4.1.5 *IrLAP Version Number*

This field is set to the version number of the IrLAP layer that transmits it. IrLAP layers that conform to this document (revision 1.1) set this field to X'00'. In future IrLAP revisions the content of this field only need be changed if significant functionality that needs identification has been added (This is in order to conserve the 256 values available).

5.7.1.4.1.4.1.6 *Discovery Info*

The discovery info is a field up to 32 bytes long whose content is specified by the service user layer. The IrLAP layer simply transports this information from one station to another during the discovery procedure.

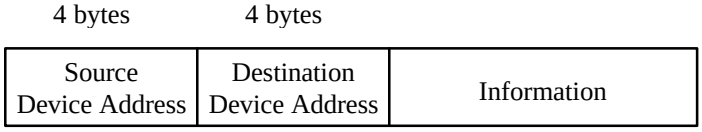
5.7.1.4.1.4.2 Sniffing Format Specific Information

A sniffing frame is identical to an XID discovery response frame except that the destination address is set to the broadcast address (X'FFFFFFFF'). This frame can be uniquely recognized by its use of the broadcast address.

5.7.1.5 TEST

As a command, a TEST frame may be sent to a station in disconnected mode (NDM) or to a connected secondary station to solicit a TEST response. If an information field is included with the command, it is returned in the response. If the secondary station has insufficient buffering available for the I field, a TEST response with no I field is returned. The information field of a TEST frame with a broadcast A field (connection address) is always preceded by an eight byte field containing the source and destination device addresses.

Information Field Format for TEST Frame with Broadcast A field



The maximum size of the information field of a TEST frame in NRM is equal to the negotiated frame size. In NDM it is governed by the maximum amount of data that can be sent at 9600bps which is 400bytes (see the max capacity table in section 0, 6.6.11 Negotiation Procedure). The maximum size of the information field is 376 bytes (400 bytes minus overhead for 11 BOFs, the address byte, the control byte, the source device address, the destination device address, 2 CRC bytes and the EOF which is 400 minus 24).

5.7.1.6 RNRM (request normal response mode):

This response is only used by a secondary station to solicit a reset of the connection identified in the A field by the primary station. SNRM is the expected reply.

5.7.1.7 UA (unnumbered acknowledgment):

This is the affirmative response to an SNRM or DISC command. The UA frame has three formats. The UA response to a SNRM used to establish a connection contains an extended attribute field in place of the information field as shown below. Even with the extended attribute field the A field of the UA frame is set to the connection address contained in the SNRM frame (not broadcast).

Information Field Format for UA Frame response to SNRM Connect Frame

4 bytes	4 bytes	
Source Device Address	Destination Device Address	Negotiation Parameters

The UA response to a SNRM used to reset a connection and the UA response to a DISC does not contain negotiations parameters. Furthermore, the Source and Destination addresses are optional.

5.7.1.8 FRMR (Frame Reject):

The frame reject response is used by connected secondary stations to report problems that cannot be corrected by retransmission of the identical frame. It is sent when one of the following conditions results from a frame without an FCS error:

1. receipt of a command that is undefined or not implemented.
2. receipt of an I/UI, TEST or XID command with an information field that exceeded the maximum supported (or negotiated if a connection is in effect). A station is also allowed to ignore this condition instead of sending FRMR.
3. receipt of an invalid Nr count, *i.e.*, one which identifies an I frame which has previously been transmitted and acknowledged or has not been transmitted. This error must be handled.
4. receipt of a frame containing an information field when no I field is permitted for that frame type. A station is allowed to ignore this condition instead of sending FRMR.
5. receipt of other unexpected frame that violates the protocol to be defined.

The secondary station will transmit the FRMR response immediately when it next receives the poll bit. After sending FRMR the station will stop sending I frames if condition (3) occurred.

Upon receipt of a FRMR response, a primary station is responsible for initiating the appropriate corrective action. For example, this may involve initializing one or both directions of transmission using SNRM or DISC if condition (3) occurred.

The FRMR I field will be arranged as follows:

1 Byte	1 Byte				1 Byte				
Bits 0 - 7	5-7	4	1 - 3	0	4-7	3	2	1	0
Rejected frame Control field	N(R)	C/R	N(S)	0	0000	z	y	x	w

Rejected frame control field: control field of frame which caused the FRMR condition.

N(S): the current value of the Ns variable at the secondary station that sends the FRMR response.

C/R: if set to “1” indicates the rejected frame was a response frame. If set “0” indicates the rejected frame was a command frame.

N(R): the current value of the Nr variable at the secondary station that sends the FRMR response.

w: when set to “1” indicates the rejected control field is undefined or not implemented.

x: when set to “1” indicates the rejected control field was invalid because it contained a non permitted I field. Bit w will also be set to “1” if this bit is set.

y: when set to “1” indicates the received I field exceeded the maximum negotiated for the existing connection or exceeded the maximum this station supports if no connection exists.

z: when set to “1” indicates the rejected control field contained an invalid Nr count.

The *w, x, y* and *z* bits of the FRMR response may all be set zero to indicate an unspecified rejection of the frame for one or more of the conditions cited above.

The *w, x, y* and *z* bits of the FRMR response are all be set zero when sending a FRMR for an invalid Ns.

5.7.1.9 DM (disconnected mode):

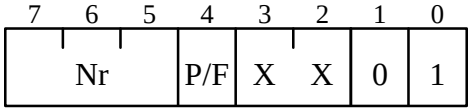
A station sends this response to indicate that it is in the disconnected (NDM) mode.

Outside of a connection, the A field contains the broadcast address and is optionally followed by Source and Destination addresses. During connection establishment or when a connection exists, the A field contains the current connection address and no I-field.

5.7.1.10 RD (Request disconnect):

A secondary station sends this response to indicate that it wishes to be placed in the disconnected (NDM) mode. The A field contains the current connection address, and there is no required I field.

5.7.2 S (supervisory) Format



7	6	5	4	3	2	1	0	
Nr	Nr	Nr	P/F	0	0	0	1	RR command/response
Nr	Nr	Nr	P/F	0	1	0	1	RNR command/response
Nr	Nr	Nr	P/F	1	0	0	1	REJ command/response
Nr	Nr	Nr	P/F	1	1	0	1	SREJ command/response

5.7.2.1 RR (receive ready):

Sent by either a primary or a secondary station, RR confirms numbered frames through Nr-1 and indicates that the originating station is ready to receive additional I frames.

5.7.2.2 RNR (receive not ready):

Sent by either a secondary or a primary station, RNR indicates a temporary busy condition caused by unavailability of buffers or other internal constraints.

As a command or response, RNR confirms numbered information frames through Nr-1 and indicates that frame Nr is expected next.

A secondary station reports the clearing of the RNR condition by transmitting an RR frame with the F bit on in response to an RR (with P bit on) from the primary.

A primary station indicates that an RNR condition has been cleared by transmitting an RR frame with the P bit on.

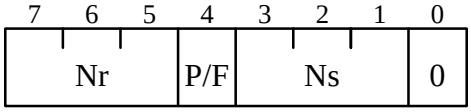
5.7.2.3 REJ (reject):

This command or response may be transmitted to request retransmission of numbered I frames. REJ confirms frames through Nr-1 and requests the retransmission of numbered information frames starting at the Nr count contained in the REJ frame. The reject condition is cleared when the requested frame or mode setting command has been correctly received.

5.7.2.4 SREJ (selective reject):

This command or response may be transmitted to request retransmission of a particular I-frame specified by the Nr count contained in the SREJ frame. This Nr count also acknowledges all frames through Nr -1. The reject condition is cleared when the requested frame or mode setting command has been correctly received.

5.7.3 I (information) Format



Only frames with an I format C field are sequenced. The Nr and Ns counts provide for numbering the frame being sent and the frame expected to be received next. Confirmation must be requested if the maximum count of outstanding unconfirmed frames (seven) is reached. Retransmission, as required, is requested by an appropriate S-frame. A primary station concludes the transmission of sequenced I frames with a frame that has the P (poll) bit on; a secondary station concludes with a frame that has the F (final) bit on. The expected acknowledgment is an S or I format frame whose Nr count confirms correctly received frames or, conversely, indicates which frames should be retransmitted. Frames of the S format may be interspersed with I format frames as needed.

6. IrLAP Description of Procedures

6.1 Introduction

This section specifies in detail the IrLAP operating procedures. These procedures define the behavior of the IrLAP layer during each phase of operation. The operating procedures are: link startup and shutdown, address discovery, address conflict resolution, connection establishment, sniff-open, information exchange, connection reset and disconnection.

Link Startup/Shutdown. These procedures govern the behavior of the IrLAP layer when its operation is enabled and disabled.

Address Discovery. This procedure is used to determine the device addresses and some other key attributes of all stations (with active enabled IrLAP layers) that are within communication range.

Address Conflict Resolution. This procedure is used when two or more stations that responded to the address discovery procedure are determined to have selected the same device address. The stations are informed of the conflict and guided in selection of new addresses that do not conflict.

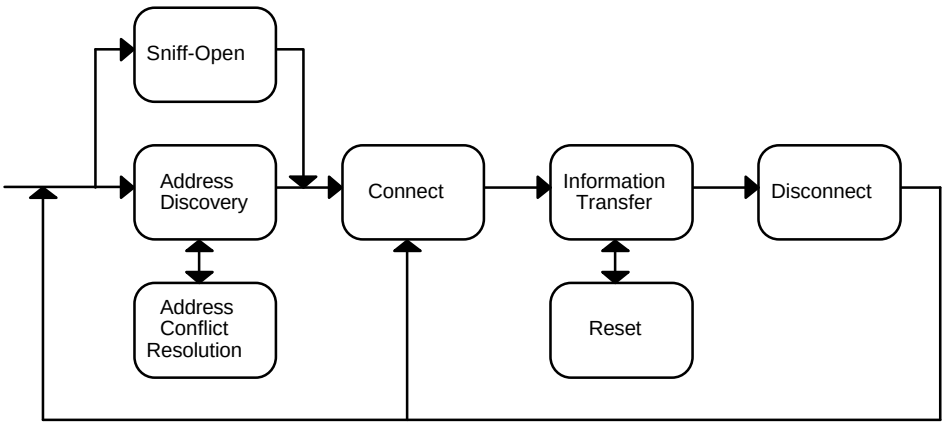
Connection Establishment. This procedure is used to establish an IrLAP connection to a station whose device address has been determined using the address discovery procedure.

Sniff-Open. This procedure allows a device to broadcast its desire to connect in a way that conserves power.

Information Exchange. This procedure governs how IrLAP layers exchange information frames over an established IrLAP connection.

Connection Reset. This procedure is used to reset an established IrLAP connection.

Disconnection. This procedure is used to terminate an established IrLAP connection.



6.2 General Rules for all State Machines

Throughout this section, precise descriptions of the IrLAP procedures are specified using state machines. The following notes apply to all the state machines. Notes specific to a particular state machine and/or procedure are included with that procedure's text.

1. The state machines included in the precise description of operation are supplied in order to clearly specify the behavior of the protocol. Designers and implementers may choose any design / implementation technique they wish, provided it behaves externally in a manner identical to the external behavior of the specified state machines.

2. Flag variables are used to limit the number of states by maintaining the state of particular conditions. Specific flags are defined with the descriptions of the state machines in which they are included.

3. In the state machine events, events of the form *Recv x:x:x:x* are sometimes used, where the *x*'s indicate "don't care" conditions on the various fields of a received frame. These events indicate the reception of any command or response frame not specifically listed for the state, including unknown or invalid frames.

4. For some combinations of state and events, the state tables provide alternative groups of actions. These alternatives are separated by horizontal lines in the **Actions** and **Next State** columns. The alternatives are mutually exclusive; selection of an alternate is done based upon (i) local status, (ii) a layer management action, or (iii) an implementation decision. There is no relationship between the order of the alternatives between events, nor is it implied that the same alternative must be selected every time the event occurs.

5. The state tables use timers. Any *Start Timer* action restarts the specified timer from zero, even if the timer is already running. When the timer reaches its limit the appropriate **timer expired** event is set and the timer stops. The *stop timer* actions stops a timer if it is running.

6. Events not recognized in a particular state are assumed to remain pending until any masking flag is modified or a transition is made to a state where they can be recognized.

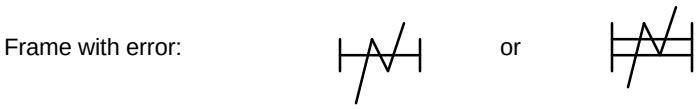
7. When events in which a frame arrives have an action that includes sending a frame back immediately, the responder should still wait the minimum turnaround delay - "immediately" means as soon as data can reasonably be expected to get through to the other side. However, earlier implementations that do not wait shall not be considered non-compliant, but simply in danger of requiring re-transmissions.

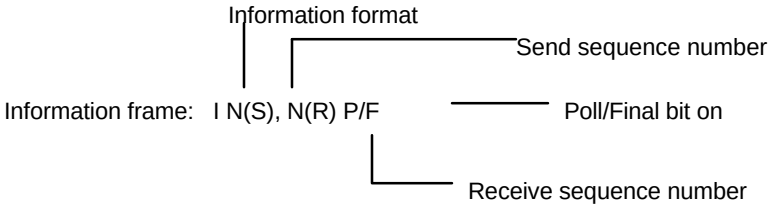
8. Ill-formed frames are either ignored, or have unspecified behavior. Ill-formed frames include those that are missing required fields and those with improper field values. Examples include U frames with the 32 bit address set to 0 or to broadcast, SNRM frames with proposed connection address of 0, 1, 0XFF or 0XFE, or FRMR frames with a missing I field. In general, reserved portions of fields should be ignored (masked out) so they are not included when checking for an improper field. Extra fields or extra data in a frame should be ignored and does not constitute an ill formed frame. The behavior when receiving most ill-formed frames is unspecified and left to the implementer to decide what will be best for a particular system; implementations are not required to check the validity of all fields in frames they receive.

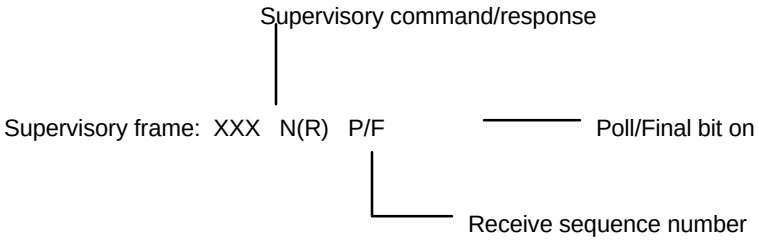
9. Stations may ignore received data when in a XMIT state (*i.e.*, receiving may be disabled while transmitting, a situation often required by the hardware).

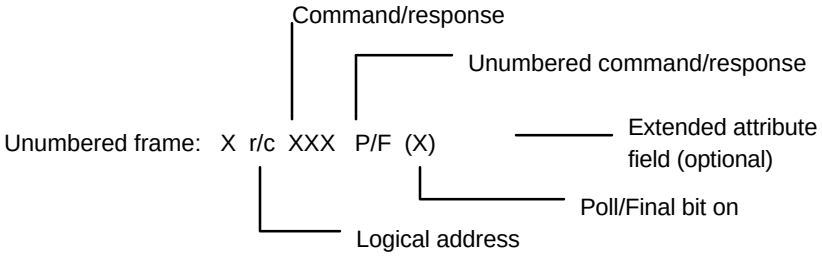
6.3 Notation Used for Examples

Throughout this section examples are given for each procedure in order to better illustrate how each procedure operates in practice. The notation used in these examples is described below:









6.4

Modes

Once initialized, IrLAP link layers can be in either of two defined modes: operational mode or non-operational mode.

6.4.1 Operational Mode

The one operational mode is the Normal Response Mode (NRM). NRM is an unbalanced operational mode where a data link connection has been established between two stations. When in the NRM a station must play either the primary or the secondary role. The role each station is to play in NRM is determined during connection establishment. The stations are able to exchange frames according to their role.

Primary Role. The primary station has responsibility for controlling the data link: it issues commands to the secondary stations and gives them permission to transmit.

Secondary Role. Secondary stations will initiate transmission only as the result of receiving explicit permission to do so from the primary station. After receiving permission, the secondary will initiate a response transmission. The response transmission will consist of one or more frames. The last frame of the response transmission will be explicitly indicated by the secondary station. Following indication of the last frame, the secondary station will stop transmitting until explicit permission is again received from the primary station.

6.4.2 Non-Operational Mode

The one non-operational mode is the Normal Disconnected Mode (NDM). NDM differs from NRM in that no connection is established to another station on the physical medium. No user information can be sent or accepted while in NDM. The discovery and address conflict resolution procedures are carried out entirely by stations in NDM, and the connection procedure is initiated from NDM. Since all NDM communications are contention based, stations that wish to transmit while in NDM must use caution and follow the NDM media access rules fully.

Some conditions that cause a link layer to enter the NDM mode are covered in this specification, for example, receipt of a DISC command frame. Others are beyond its scope, for example: power is turned on or the data link layer logic is reset.

An IrLAP layer in NDM is required to monitor received frames for the purpose of generating responses as specified in the procedure definitions. For example, responding to discovery XID command frames or sending a DM response frame at the appropriate time.

6.5

Addressing

IrLAP utilizes two classes of addressing information:

- *Handles*: these are allocated by IrLAP and passed to the upper layer. The upper layer uses handles to refer to various IrLAP resources (e.g. connections) when making service requests (see *Data Link Layer Service Specifications*).
- *Addresses* these are allocated and utilized by IrLAP peer layers when they communicate with one another.

IrLAP peer layers use two types of address information in their communications: device addresses and connection addresses. Two corresponding types of handle are presented to the upper layers: device handles and connection handles.

Device Address: a 32-bit value that is used to uniquely identify an IrLAP layer. The device address is generated and maintained internally by the IrLAP layer. Whenever the IrLAP layer is initialized it will generate a 32-bit random number that it will use as its device address. If an address conflict is detected by another node an IrLAP layer will be requested to change its device address. IrLAP layers will honor such requests if they are in disconnected (NDM) mode. Such requests will not be honored by layers that are in NRM. Device addresses are included in the device address fields of most frames transmitted by stations in NDM. Device Addresses are transmitted least significant byte first (little endian).

Connection Address: A 7-bit value used in the A field of all IrLAP frames that uniquely identifies a secondary station that is connected to a primary station when in NRM and is set either NULL (B‘0000000’) or broadcast (B’1111111’) by stations in NDM. Whenever a connection is established the primary station allocates a 7-bit value at random (that does not conflict with any existing connection addresses it has active) and assigns it as the connection address.

Device handle: a value generated by the IrLAP layer and returned to the upper layer that the IrLAP layer can use to lookup a corresponding device address.

Connection handle: a value generated by the IrLAP layer and returned to the upper layer that the IrLAP layer can use to identify an existing connection between two devices. A connection handle is only valid for the duration of the related connection.

In NDM stations are only required to handle U-frames that contain a broadcast connection address and a broadcast device address or their own device address. I and S frames should be silently ignored which means that DM response is not required.

In NRM stations are only required to handle frames that contain the connection address of the connection in which they are participating. They are allowed to ignore all other frames including frames containing a broadcast connection address (even if the frame contains a broadcast device address or their device address).

6.6

Negotiation

6.6.1 Introduction

Negotiation is the process by which two stations agree on seven basic connection parameters: baud rate, maximum turn around time, data size, window size, additional BOFs, minimum turn around time and link disconnect/threshold time. These parameters are negotiated by exchange of SNRM/UA unnumbered frames. SNRM/UA frames are used to bring up a connection and to negotiate the initial connection parameters.

6.6.2 Negotiation Field Parameters

SNRM and UA frames can have a negotiation field containing negotiation parameters. SNRM/UA frame negotiation fields should contain all 7 parameters but if a parameter is missing a default value is assumed.

Each parameter in a negotiation field is defined by a tuple containing three fields: Parameter Identifier (PI), Parameter Length (PL), and Parameter Value (PV). The PI and PL fields are one byte each. The length of the PV field is PL bytes. Currently all SNRM & UA parameters except (potentially) Baud Rate have a PL field equal to one; if the 4Mbps rate is supported, a second byte is required for PV and PL will equal 2. The total number of bytes in the negotiation field of SNRM and UA frames is therefore 21 bytes (3 bytes * 7 parameters) if 4Mbps is not supported, or 22 bytes if it is supported.

Each bit in a parameter's PV field represents a specific value of the parameter. When set to one, the bit indicates that the specific value of the parameter is supported. When set to zero, the bit indicates that that specific value of the parameter is not supported. Parameters are divided into two types. The first type (type 0) are parameters that must be negotiated to the same value for both stations involved in a connection. The second type (type 1) are parameters that are negotiated independently for both stations involved in a connection. The most significant bit of the PI field of type 0 parameters is always zero. The most significant bit of the PI field of type 1 parameters is always one. The format of the PV field for each parameter is described below in the order in which the parameters are transmitted in SNRM/UA frame negotiation fields.

6.6.3 Baud Rate

The baud rate parameter dictates the speed at which both stations will transmit on the data link channel. Both devices must agree on the same baud rate.

Baud Rate parameter format (PI = X'01', type 0)
- First byte of PV field:
  - bit 0 = 2400 bps (lsb, transmitted first)
  - bit 1 = 9600 bps
  - bit 2 = 19200 bps
  - bit 3 = 38400 bps
  - bit 4 = 57600 bps
  - bit 5 = 115200 bps
  - bit 6 = 576000 bps
  - bit 7 = 1152000 bps
- Second byte of PV field (needed only if 4Mbps supported):
  - bit 0 = 4000000 bps
  - bits 1-7 of 2nd byte: reserved and must be set to zero

For example, a station supporting all baud rates would fill the Baud Rate parameter with the binary number B'0000000111111111' (X'01FF'). A station supporting only 9600 bps and 115200 bps would fill the Baud Rate parameter with B'00100010' (X'22'), using only one byte since 4Mbps is not supported.

6.6.4 **Maximum Turn Around Time**

Maximum turn around time is the maximum time that a station can hold the P/F bit. This parameter along with the baud rate parameter dictates the maximum number of bytes that a station can transmit before giving the line to another station by transmitting a frame with the P/F bit set. The maximum turn around time has higher priority than the maximum data size and window size parameters. This parameter is used by one station to indicate the maximum time the other station can send before it must turn the link around. It is negotiated independently for each station. 500ms is the only valid value when the baud rate is less than 115200 bps. Stations acting as primary are not required to turn the link around faster than 500ms though they must honor the maximum turn around parameter as it applies to the actual maximum data size and window size.

Maximum Turn Around Time parameter format (PI = X‘82’, type 1)

bit 0 = 500 ms
bit 1 = 250 ms (only valid at 115200 bps and higher)
bit 2 = 100 ms (only valid at 115200 bps and higher)
bit 3 = 50 ms (only valid at 115200 bps and higher)
bit 4 = is reserved and must be set to 0
bit 5 = is reserved and must be set to 0
bit 6 = is reserved and must be set to 0
bit 7 = is reserved and must be set to 0

For example, the typical station will use 500 ms to minimize the overhead imposed by the protocol therefore, the parameter would be set to the binary number B‘00000001’ (X‘01’). A station that wants to simulate full duplex might use 100 ms, and therefore, the parameter would be set to the binary number B‘00000111’ (X‘07’).

6.6.5 **Data Size**

The data size is the maximum number of data bytes allowed in any received frame for the duration of the connection. Data size is defined as all bytes in the I field of a frame prior to the application of any transparency algorithm (i.e. byte stuffing). The actual maximum frame size for the connection must be adjusted to accommodate the baud rate and maximum turn around time. This parameter is negotiated independently for each station.

Data Size parameter format (PI = X‘83’, type 1)

bit 0 = 64 bytes (lsb, transmitted first)
bit 1 = 128 bytes
bit 2 = 256 bytes
bit 3 = 512 bytes
bit 4 = 1024 bytes
bit 5 = 2048 bytes
bit 6 is reserved and must be set to 0
bit 7 is reserved and must be set to 0

For example, a station capable of receiving any size frame would fill the Data Size parameter with X‘3F’. A station capable of receiving only frames of 128 (or less) bytes would fill this parameter with X‘03’.

6.6.6 **Window Size**

The window size is the maximum number of unacknowledged I frames that a station can receive before an acknowledgment must be sent. This parameter is the maximum possible size for the window but not necessarily the actual window size used. The actual window size must be adjusted to accommodate the

baud rate and maximum turn around time. Also the actual maximum frame size must be taken into account.
Window size indicates the number of buffers a station has for receiving I and UI frames. A device is only required to have “window size” buffers of size “data size” and one extra buffer for receiving a single S-frame. This parameter is negotiated independently for each station.

Window Size format (PI = X‘84’, type 1)
bit 0 = 1 frame window (lsb, transmitted first)
bit 1 = 2 frame window
bit 2 = 3 frame window
bit 3 = 4 frame window
bit 4 = 5 frame window
bit 5 = 6 frame window
bit 6 = 7 frame window
bit 7 is reserved and must be set to 0

For example, a station capable of receiving up to seven frames would set the Window Size parameter to X‘7F’. A station capable of only stop-and-wait would set this parameter to X‘01’.

6.6.7 **Additional BOFs**

The Additional BOFs parameter indicates the number of additional flags needed at the beginning of every frame. The main purpose of the parameter is to provide a delay at the beginning of each frame for devices with long interrupt latency. The delay is based on the time for transmitting a character at 115200 bps (approx. 87 us). Each parameter value is the number of BOFs needed beyond the required one BOF for each frame sent at 115200 bps. The additional number of BOFs needed at baud rates below 115200 bps is calculated by dividing the selected parameter value by a factor equal to 115200/baud rate. For baud rates 576000 bps and 1152000 bps the Additional BOFs parameter indicates whether to use the required number of BOFs (2 STAs) or use 4 STAs (2 additional). If the Additional BOFs parameter is set to 0 then 2 STAs are used otherwise 4 STAs are used (if the hardware is capable of generating additional STAs). For 4000000 bps this parameter is ignored. This parameter is negotiated independently for each station.

Additional BOFs format (PI = X‘85’, type 1)
bit 0 = 48 additional BOFs at 115200 (lsb, transmitted first)
bit 1 = 24 additional BOFs at 115200
bit 2 = 12 additional BOFs at 115200
bit 3 = 5 additional BOFs at 115200
bit 4 = 3 additional BOFs at 115200
bit 5 = 2 additional BOFs at 115200
bit 6 = 1 additional BOFs at 115200
bit 7 = 0 additional BOFs at 115200

The following equations are used to calculate the number of additional BOFs needed for baud rates below 115200 given the number negotiated for the “Additional BOFs” parameter.

2400 bps = Additional BOFs parameter value/48
9600 bps = Additional BOFs parameter value/12
19200 bps = Additional BOFs parameter value/6
38400 bps = Additional BOFs parameter value/3
57600 bps = Additional BOFs parameter value/2
115200 bps = Additional BOFs parameter value/1

The table below shows the additional BOFs needed for the baud rates of 115200 bps and below for all possible parameter values.

Baud Rate	48 BOFs	24 BOFs	12 BOFs	6 BOFs	3 BOFs	2 BOFs	1 BOF	0 BOFs
2400	1	0	0	0	0	0	0	0
9600	4	2	1	0	0	0	0	0
19200	8	4	2	1	0	0	0	0
38400	16	8	4	2	1	0	0	0
57600	24	12	6	3	1	1	0	0
115200	48	24	12	6	3	2	1	0

As an example, a station that requires 12 additional BOFs at 115200 bps will set the parameter to the binary number B'00000100' (X'04'). A station that does not need any additional BOFs at 115200 will set the parameter to B'10000000' (X'80').

6.6.8 Minimum Turn Around Time

The minimum turn around time parameter deals with the time needed for a receiver circuit to recover following saturation by transmissions from the same device (turn around latency). This parameter corresponds to the required time delay between the last byte of the last frame sent by a station and the point at which it is ready to receive the first byte of a frame from another station. This parameter comes into play when the link is turned around and is negotiated independently for each station.

Minimum Turn Around Time format (PI = X'86', type 1)

bit 0 = 10 ms (lsb, transmitted first)
bit 1 = 5 ms
bit 2 = 1 ms
bit 3 = 0.5 ms
bit 4 = 0.1 ms
bit 5 = 0.05 ms
bit 6 = 0.01 ms
bit 7 = 0 ms

There are two methods for creating the turn around delay. The first method is to wait the specified time without any transmission before sending the first frame. The second is to insert a number of BOFs at the beginning of the first frame which occupies the required turn around time. It should be noted that inserted BOFs are in addition to any BOFs added by the Additional of BOFs parameter. Also it is highly recommended that additional BOFs be X'FF'.

As an example, a station that needs 5ms of turn around time would set the parameter to the binary number B'00000010' (X'02').

6.6.9 Link Disconnect/Threshold Time

The link disconnect/threshold time is used to control the time a station will wait without receiving valid frames before it disconnects the link. Associated with this is the time a station will wait without receiving valid frames before it will send a status indication to the service user layer (this can be used to display a warning message to the user). The values listed below are the number of seconds before disconnection. Each of these values implies a threshold time which is also indicated. This parameter must be agreed upon by both the primary and the secondary.

Link Disconnect/Threshold Time format (PI = X’08’, type 0)

bit 0 = 3 seconds (threshold = 0)
bit 1 = 8 seconds (threshold = 3 seconds)
bit 2 = 12 seconds (threshold = 3 seconds)
bit 3 = 16 seconds (threshold = 3 seconds)
bit 4 = 20 seconds (threshold = 3 seconds)
bit 5 = 25 seconds (threshold = 3 seconds)
bit 6 = 30 seconds (threshold = 3 seconds)
bit 7 = 40 seconds (threshold = 3 seconds)

For example, a station that would like to present a warning message to the user and give the user plenty of time to correct the problem might select 30 seconds as its maximum time and set the parameter to the binary number B‘01111111’ (X‘7F’). A station running an application with a session layer protocol that can deal with link disconnections will probably set the value to the minimum which is B‘00000001’ (X‘01’).

6.6.10 Contention State Communication Parameters

In the contention state devices do not yet know one another’s capabilities. Therefore devices must use a “least common denominator” approach: all frames transmitted while the medium is in the contention state, including SNRM, UA and XID frames, are always transmitted using the default parameters of 9600 bps, 8 data bits, 1 start bit, 1 stop bit, window size equal to 1, maximum data size of 64 bytes, maximum turn around time set to 500ms and 11 BOFs on every frame. The 11 BOFs create a 10ms link turnaround time (10 additional BOFs at 9600 bps is about 10ms), which will allow safe connection to devices whose hardware requires the worst case 10ms turnaround time. In previous specifications, only 5 BOFs were required, and while this shall still be considered IrDA compliant, drivers using only 5 BOFs do so at their own risk. Using more than 11 BOFs may be excessive but is permitted. It is highly recommended that the 10 additional BOFs be sent as X‘FF’ instead of the standard X‘C0’

Because they are required for contention state communication, it follows that the above contention state communication parameters must be supported by all IrLAP compliant devices. It is recommended that they be explicitly set in a SNRM or UA negotiation field, but even when they are not explicitly set the parameters are assumed to be valid. Behavior when these default bits are not set is not specified. In general if negotiation parameters are incorrect, the defaults (specified above) are assumed and connection is attempted but cannot be guaranteed. The default value for Link Disconnect/Threshold Time is 40 seconds (X’FF’) and the default value for Additional BOFs is 0 (X’80’).

6.6.11 Negotiation Procedure

As described earlier the negotiation parameters consist of a sequence of parameters where each parameter is divided into 3 fields, PI, PL, and PV. There are two types of parameters, type 0 and type 1. When setting type 0 parameters a station should set all the bits which correspond to parameter values it supports. When setting type 1 parameters it only needs to set the bit for the value it wishes to select. It is recommended that it set all the bits lower in the byte also in cases where these bits are also viable alternative values.

The procedure for negotiating type 0 parameters is as follows.

1. The primary station sets all the bits in the PV field for the values it supports and sends the SNRM frame.
2. When a secondary station receives an SNRM frame, it produces the intersection of its capabilities and the primary's capabilities by logically AND’ing the SNRM negotiation values to the negotiation field representing the parameter values it is able to support.

3. The result of this operation is included in the SNRM-UA frame and specifies the parameter values that will be employed during the connection.
4. The chosen parameter is the most significant bit that is set to a 1. If multiple bytes are present in the PV field it is the most significant bit set in the most significant byte which is selected (this is the last byte received). The least significant bit of the PV field is transmitted first. Except for the 4Mbps setting of Baud rate, the currently defined parameters are part of the least significant byte and will always be the byte that immediately follows the PL byte.

   Note: When parameters are not understood, the parameters should be ignored rather than rejecting the frame.

Algorithm for parsing the parameters:

The procedure for negotiating type 1 parameters is different than that for type 0. Since type 1 parameters are independently negotiated, the PV field is simply scanned for the most significant bit set to a one and this bit is taken as the selected value (like step 4 for type 0 parameters). Bit positions that are not defined for a particular version are ignored when searching for the most significant bit set.

Following the above procedures the negotiated values of baud rate, maximum turnaround time, data size and window size must be checked for consistency this is done by checking that:

requested-line-capacity < *maximum-line-capacity[baud-rate, max-turnaround-time]*

If this relationship does not hold then the window size and/or data size parameters must be decreased until it does.

requested-line-capacity is computed as follows:

requested-line-capacity := *window size* * (*data size* + 6 + *number-of-BOFs)* + *min-turnaround-time-in-bytes*

number-of-BOFs, *window-size* and *data size* are the values negotiated for these parameters.

The *min-turnaround-time-in-bytes* is given in the table below:

Baud Rate	10ms	5ms	1ms	0.5ms	0.1ms	0.05ms	0.01ms
9600	10	5	1	0	0	0	0
19200	20	10	2	1	0	0	0
38400	40	20	4	2	0	0	0
57600	58	29	6	3	1	0	0
115200	115	58	12	6	1	1	0
576000	720	360	72	36	7	4	2
1152000	1440	720	144	72	14	7	1
4000000	5000	2500	500	250	50	25	5

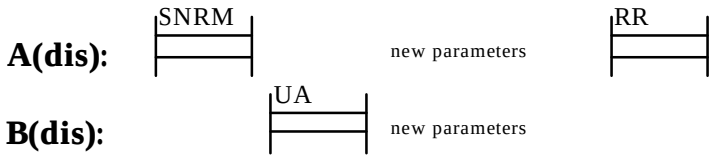
The *maximum-line-capacity* in bytes is given in the table below:

Baud Rate	500ms	250ms	100ms	50ms
9600	400	n/a	n/a	n/a
19200	800	n/a	n/a	n/a
38400	1600	n/a	n/a	n/a
57600	2360	n/a	n/a	n/a
115200	4800	2400	960	480
576000	28800	11520	5760	2880
1152000	57600	28800	11520	5760
4000000	200000	100000	40000	20000

Note for Primaries: Link turnaround with heavy transparency

The line capacity table above allows for some overhead due to transparency (byte or bit stuffing for data that would otherwise look like a control sequence), but the table does not allow for a worst case scenario in which a frame can nearly double in size in the byte stuffing case. Primary implementations need to be forgiving in this respect, and allow more time for a frame to finish even after the F-timer[2] has expired. The worst case time beyond the F-timer is about 300ms. Thus, when the F-timer expires a primary should check to see if it is in the middle of receiving a frame and allow for more time (up to 300ms) if it is.

6.6.12 Example of Initial Negotiation Packet Exchange



6.7

[2] F-timer is the final bit timer used by a primary to bound the time it waits for a frame with the F bit set from the secondary.

Link Initialization and Shutdown Procedures

6.7.1 Purpose

These procedures govern the behavior of the IrLAP layer when its operation is enabled and disabled.

6.7.2 Overview

When an IrLAP layer is enabled, it generates a device address and sets the infrared medium to communicate using the IrLAP default connection parameters.

6.7.3 Precise Description of Link Initialization and Shutdown

If discrepancies appear to exist between the precise description of this procedure and any textual material in this specification the precise description shall be taken as the definitive description.

6.7.3.1 State Chart

Current State	Event	Action(s)	Next State
OFFLINE (entry state)	*link-initialize*	NA := *Generate-Random-Device-Address* *Apply-Default-Connection-Parameters*	ONLINE

ONLINE	*link-shutdown*	for each existing connection ‘c’ do
		begin
		close connection
		Disconnect-Indication(aborted)
		end
			OFFLINE

	physical-layer-down	for each existing connection ‘c’ do
		begin
		close connection
		Disconnect-Indication(aborted)
		end
			OFFLINE

6.7.3.2 State Definitions

OFFLINE. The station is powered off, not initialized and disabled from operating in the infrared physical medium.

ONLINE. The station is powered on, initialized and able to send and receive IrLAP frames.

6.7.3.3 Event Descriptions

Link-Initialize. Station user has initialized/enabled the station.

Link-Shutdown. Station user has disabled the station.

Physical-Layer-Down. The physical layer is has become unusable.

6.7.3.4 Action Descriptions

NA := Generate-Random-Device-Address. Generate a random 32-bit device address and assign it to "NA" for use in other state machines as this station's device address.

Apply-Default-Connection-Parameters. Configure IrLAP layer to use the default connection and transmission parameters, *e.g.*, return to default baud rate (9600 bps).

Close Connection. Perform any cleanup actions prior to aborting a data link connection.

Disconnect-Indication(aborted). Inform the service user that the local IrLAP layer has aborted an established data link connection.

6.8 Discovery Procedure

6.8.1 Purpose

The discovery procedure is used to determine the device addresses and some other key attributes of all stations (with active enabled IrLAP layers) that are within communication range.

6.8.2 Overview

The station performing the discovery procedure is called the *initiator*, and the stations that reply are called the *responding* stations, or *responders*.

There are four pieces of information about each device discovered which are reported in a *discovery log* to the service user layer when the discovery procedure is complete:

1. Solicited/unsolicited - this information indicates whether the a discovered device was found by the initiator of discovery or from the responder (responders can discover the initiator based on information in sends to end the discovery procedure).

2. Sniffer/non-sniffer - indicates whether the discovered device is a sniffer or not.

3. Device address - the discovered device's 32 bit device address.

4. Discovery information - information about key attributes of the discovered device.

The discovery procedure is carried out as follows:

1. The initiator broadcasts a discovery XID command frame indicating a discovery procedure using *n* time slots. This frame also serves as notice of the beginning of time slot zero.

2. All nodes that receive the discovery XID command become responders and each generates a random number between 0 and *n* -1 (inclusive). If the random number generated is 0 the responder transmits a discovery response XID frame immediately. Otherwise, it waits for a discovery XID command frame which contains a slot number field that matches the random number that it generated and at that time it transmits its response XID frame.

3. The initiator times each time slot and sends out a discovery XID command at the beginning of each time slot. The XID command's slot number field indicates the slot number (1 to *n* - 1). Discovery XID frames are sent out at intervals dictated by the rules specified in section 0 6.13.2 Time Slot Rules. After the XID frame with the slot number *n* - 1 is sent, a final XID frame with the slot number set to X'FF' is sent indicating the end of the discovery procedure.

4. The initiator enters information from all discovery XID responses that it receives into the *discovery-log* that is returned to the service user when discovery completes. The initiator may now check the table to see if any entries have duplicate device addresses, or have the same device address as the initiator. If duplicate addresses are detected the address resolution procedure may be used to resolve them.

6.8.3 **Precise Description of Discovery Procedure**

If discrepancies appear to exist between the precise description of this procedure and any textual material in this specification the precise description shall be taken as the definitive description.

(State chart appears on next page)

6.8.3.1 State Chart

Current State	Event	Action(s)	Next State
NDM (entry state)	*Discovery-Request(S)* ∧ mediaBusy = false	maxSlot := (S-1) slotCount := 0 *send Discovery-XID-Cmd*: maxSlot, slotCount *start-slot-timer* log := {∅}	QUERY
	Discovery-Request(S) ∧ mediaBusy = true	*Discovery-Indication(media-busy)* -- see note 1	NDM

	Recv Discovery-XID-Cmd:S,s
	-- see note 2
		slot := *Generate-Random-Time-Slot(S,s)*
		if slot = s then
		Send-Discovery-XID-Rsp:NA,discovery-info
		frameSent := true
		else
		frameSent := false
		start-query-timer
			REPLY

QUERY	*slot-timer-expired* ∧ slotCount < maxSlot	slotCount := slotCount + 1 *send Discovery-XID-Cmd:* maxSlot, slotCount *start-slot-timer*	QUERY
	slot-timer-expired ∧ slotCount ≥ maxSlot	*send End-Discovery-XID-Cmd* *Discovery-Confirm(log)* -- see note 3	NDM
	Discovery-Abort-Condition	*stop-slot-timer* *send End-Discovery-XID-Cmd* *Discovery-Indication(aborted)*	NDM
	RecvDiscovery-XID-Rsp:sa,info	log := log ∪ { <*sa,info*> }	QUERY
	Response-Collision	log := log ∪ {<ϕ, ϕ>}	QUERY
	Recv x:x:x:x	*Empty*	QUERY
REPLY	*Recv Discovery-XID-Cmd:S,s* ∧ (s ≥ slot) ∧ ¬frameSent	*Send Discovery-XID-Rsp: NA,discovery-info* frameSent := true	REPLY
	Recv End-Discovery-XID-Cmd	*stop-query-timer* *Discovery-Indication(remote)*	NDM
	query-timer-expired	*Empty*	NDM
	Recv x:x:x:x	*Empty*	REPLY

6.8.3.2 Notes

1. It is acceptable for implementations to “hold” this event waiting for the mediaBusy flag to become false. However, should some other event occur prior to mediaBusy becoming false then the *Discovery-Indication(media-busy)* action must be executed.

2. Usually when this frame is received while in NDM the current slot number ‘s’ will be zero. However, the station attempts to participate even if ‘s’ is greater than zero. In the case where ‘s’ is received (in NDM) greater than zero it is acceptable for the query timer’s timeout to be reduced by an appropriate factor.

3. It is the responsibility of the service user to determine if address conflicts exist in the *discovery-log* and to decide when/if they need to be resolved using the address resolution discovery procedure.

6.8.3.3 State Definitions

NDM. The station is in the normal disconnected mode. It can initiate or respond to local and remote requests to connect with a remote peer layer, and it can initiate or respond to local and remote discovery and address resolution procedure requests.

QUERY. The local layer is currently executing the discovery procedure. It has transmitted a discovery XID command frame and is currently transmitting the time slot indication XID frames and logging any XID responses that are received within the time slots.

REPLY. An XID discovery command frame has been received from a remote peer layer. A time slot has been selected at random and when the corresponding XID time slot frame is received the local layer will send a discovery response XID frame.

6.8.3.4 Event Descriptions

Discovery-Request(S). The service user has requested a discovery operation be performed using *S* time slots.

Recv Discovery-XID-Cmd:S,s. A discovery command frame has been received from a remote peer layer. *S* indicates the total number of time slots the discovery procedure will use, *s* indicates the number of the current time slot. If *s* is zero then this frame initiates a discovery procedure. Otherwise ($0 < s < S$), the discovery procedure is already in progress and has reached slot number *s*. The *End-Discovery-Cmd* frame is similar to this frame and is described below. The discovery XID command uses the general XID command format described in section 0 5.7.1.4.1 Definition of XID Frame Fields with the X’01’ format identifier (FI). The specific format for a Discovery XID frame is detailed in section 0 5.7.1.4.1.4.1 Discovery and Address Conflict Resolution Format Specific Information.

Slot-timer-expired. The discovery time slot timer has expired.

SlotCount < maxSlot. There are still time slots remaining in the current discovery procedure in progress.

SlotCount ‡ maxSlot. All time slots in the current discovery procedure in progress have expired.

mediaBusy = false. During the preceding 500ms sense period no media activity that would indicate an active connection (at any baud rate) or discovery/address resolution process has been received.

mediaBusy = true. At some time during the preceding 500ms sense period media activity that indicates an active connection or discovery/address resolution process was detected.

Discovery-Abort-Condition. An unspecified condition which requires the immediate termination of a discovery procedure currently in progress has been detected.

Recv-Discovery-XID-Rsp:sa,info. A discovery response frame that identifies a remote station's source device address, *sa*, and its capabilities, *info*, has been received from a remote peer layer. The discovery XID response uses the general XID response format described in section 0 5.7.1.4.1 Definition of XID Frame Fields with the X'01' format identifier (FI).

Response-Collision. A condition where two stations have selected the same time slot in which to transmit their Discovery-XID-Rsp frames is detected.

Slot. When the local layer is participating as a responder in a discovery process (REPLY state) this variable is set to the discovery time slot within which it will send its own discovery response frame.

Slot = s. The discovery time slot selected by the local layer corresponds to that received in the *s* field of a discovery XID command frame.

Recv End-Discovery-XID-Cmd. An end of discovery procedure frame has been received from a remote peer layer. This frame is similar to the general discovery XID command, it uses the general XID command format described in section 0 5.7.1.4.1 Definition of XID Frame Fields with the X'01' format identifier (FI). However, the slot number field is set to X'FF', and an *info* (hints) field that describes the capabilities of the discovery initiator is included.

Query-timer-expired. The timer that times the anticipated duration of a discovery operation carried out by a remote peer layer and participated in by the local layer has expired.

6.8.3.5 Action Descriptions

maxSlot := S. The variable maxSlot records the number of time slots to be used in a discovery procedure initiated by the local layer. *S* is received from the service user in the discovery request.

slotCount := 0. The variable slotCount maintains the number of the current time slot for a discovery procedure initiated by the local layer. The first time slot is always numbered zero.

Send Discovery-XID-Cmd:maxSlot, slotCount. The local layer transmits a discovery XID command frame indicating a discovery procedure that has 'maxSlot' time slots and has reached time slot 'slotCount'.

Start-slot-timer. The discovery time slot duration timer is started from zero.

Discovery-Indication(condition). Inform the service user that a discovery related condition has occurred.

slot := Generate-Random-Time-Slot(S,s). Generate a random time slot number between *s* and *S*-1 and save it in 'slot'.

frameSent := false. When a station is in the REPLY state the 'frameSent' flag indicates whether it has transmitted its discovery XID response frame. 'frameSent' false indicates the response has not been transmitted.

Start-query-timer. Start from zero the timer that times the anticipated duration of a remote discovery operation in which the local layer is participating.

slotCount := slotCount + 1. Increment the number of the current time slot of the discovery process that is in progress.

Send End-Discovery-XID-Cmd. The local layer transmits an “End of discovery” XID command frame indicating the end of the discovery procedure that it has been executing. This frame is similar to the general discovery XID command. It uses the general XID command format described in section 0 5.7.1.4.1 Definition of XID Frame Fields with the X’01’ format identifier (FI). However, the slot number field is set to X‘FF’, and the frame includes a *discovery-info* (hints) field that describes the capabilities of the discovery initiator.

Discovery-Confirm(log). A discovery procedure requested by the service user has completed successfully. The service user is informed, and the log of discovered stations is passed to the service user.

Log. Log is a *bag* (sometimes called a family or multi-set) whose elements are the ordered pairs <*a*,*b*>, where *a* is a device address and *b* is a discovery info string. Bags are similar to sets in that the order of the elements is not significant, but, unlike a set, the number of occurrences of each object in the bag is significant[3] (sets do not hold multiple copies of the same element).

Log := {˜ }. Log is initialized to the “empty bag”.

Log := log ¨ {<sa,string>}. A new device address, discovery info string pair is added to the log. ∪ indicates bag union, *i.e.*, the element is added even if an identical element is already contained in the bag. The special pair <ϕ, ϕ> indicates a detected response collision condition.

Stop-slot-timer. Stop the discovery time slot duration timer.

Send-Discovery-XID-Rsp:NA,discovery-info. The local layer transmits a discovery response XID frame containing its 32-bit device address, NA, and its discovery info string. The discovery XID response uses the general XID response format described in section 0 5.7.1.4.1 Definition of XID Frame Fields with the X’01’ format identifier (FI).

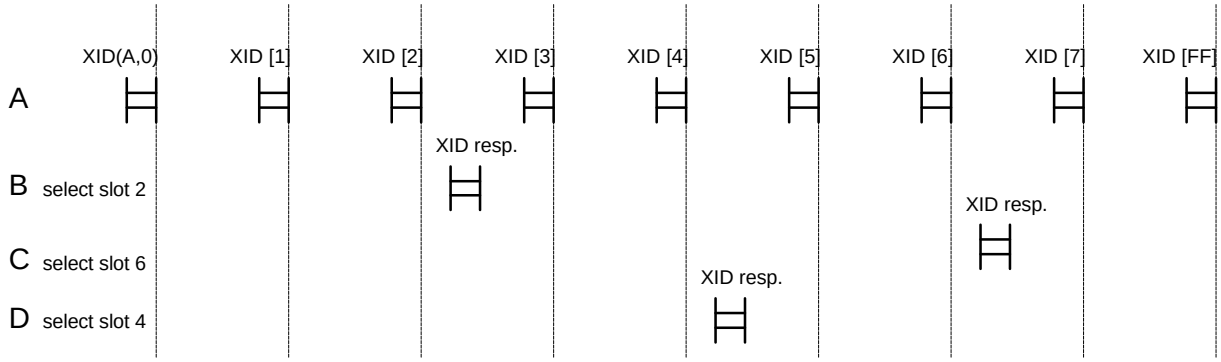
Stop-query-timer. Stop the timer that is timing the anticipated duration of a remote discovery operation in which the local layer is participating.

Empty. No actions.

6.8.4 Discovery Procedure example

Node's A, B, C and D are all in NDM, the physical medium is not in use. Node A initiates discovery to find all nodes that it can communicate with.

[3]See Diller A., "Z An Introduction to Formal Methods", Wiley, 1990, pp.85-94.


XID(A,0)
XID [1]
XID [2]
XID [3]
XID [4]
XID [5]
XID [6]
XID [7]
XID [FF]
A
B select slot 2
C select slot 6
D select slot 4
XID resp.
XID resp.
XID resp.

6.9

Sniff-Open Procedure

6.9.1 Purpose

This procedure allows a device to broadcast its desire to connect in a way that conserves power.

6.9.2 Overview

The basic procedure is as follows:

1. A Sniffing device wakes up and listens for a short period of time (see MAC rules). If it hears traffic it goes back to sleep.

2. If it does not hear traffic it transmits an XID response frame with the destination device address set to X‘FFFFFFFF’. This frame indicates that the device desires to be connected as a secondary. This response is unique from other discovery XID responses via the device address so another devices listening knows that this device is performing sniffing.

3. The device then waits a short period for a message directed to it. Either an XID command frame or a SNRM frame with its device address as the destination device address. In the case of the XID a destination address of X‘FFFFFFFF’ is also a valid message in which to respond. If the frame is an XID discovery frame the sniffer can enter the discovery process, but it must respond with a sniff frame not a discovery response frame. See section 0 5.7.1.4.1.4.2 Sniffing Format Specific Information for the format of a sniff frame.

4. If no frames are sent to it, the Sniffing device goes to sleep (usually 2 - 3 seconds) and starts the procedure again. If it hears traffic not directed to it, it must follow the MAC rules described in section 0 Media Access Control Procedures.

6.9.3 Precise Description of Sniff-Open Procedure

If discrepancies appear to exist between the precise description of this procedure and any textual material in this specification the precise description shall be taken as the definitive description.

(State chart appears on next page).

6.9.3.1 State Chart (Sniffing)

Current State	Event	Action(s)	Next State
NDM (entry state)	*Sniff-Request*	mediaBusy := false *start-sense-timer*	POUT
POUT	*sense-timer-expired* ∧ mediaBusy = false	*Send Sniff-XID-Rsp:NA,discovery-info* *start-sniff-timer*	SNIFF
	sense-timer-expired ∧ mediaBusy = true	disable receiver etc. *start-sleep-timer*	SLEEP
	Recv Discovery-XID-Cmd:S,s	slot := *Generate-Random-Time-Slot(S,s)* *if* slot = *s then* *send-discovery-XID-Rsp:NA, discovery-info* frameSent := true *else* frameSent := false *start-Query-timer*	REPLY
	Recv x:x:x:x	*Empty*	POUT

SNIFF	*Recv Discovery-XID-Cmd:S,s*	slot := *Generate-Random-Time-Slot(S,s)*
		if slot = s *then*
		send-discovery-XID-Rsp:NA, discovery-info
		frameSent := true
		else
		frameSent := false
		start-Query-timer
			REPLY

	Recv u:snrm:cmd:P	dest := *d;* ca := *c* *Connect-Indication*	CONN
	sniff-timer-expired	disable receiver etc. *start-sleep-timer*	SLEEP
	Recv x:x:x:x	*Empty*	SLEEP
SLEEP	*sleep-timer-expired*	mediaBusy := false enable receiver etc. *start-sense-timer*	POUT

6.9.3.2 State Chart (Connect to Sniffer)

Current State	Event	Action(s)	Next State
NDM (entry state)	*Recv-Sniff-XID-Rsp: sa,info*	*Discovery-Indication (sniff)*	NDM
	Connect-Request(sniff)	*Empty*	SCONN
SCONN	*Recv Sniff-XID-Rsp: sa,info*	*Generate-Random-ConnectionAdr(ca)* *send* u:snrm:cmd:P:ca:NA *start-P-timer*	SSETUP

	Recv x:x:x:x	*Empty*	SCONN
SSETUP	*P-timer-expired*	*Disconnect-Indication*	NDM
	recv u:ua:rsp:F	*stop-P-timer* *Initialize-Connection-State* *Negotiate-Connection-Parameters* *Apply-Connection-Parameters* *Connect-Confirm* *send* s:rr:cmd:P *start-P-timer*	NRM(P)
	recv u:dm:rsp:F	*stop-P-timer* *Disconnect-Indication*	NDM
	recv x:x:x:x	*Empty*	SSETUP

6.9.3.3 State Definitions

NDM. The station is in the normal disconnected mode. It can initiate or respond to local and remote requests to connect with a remote peer layer, it can initiate or respond to local and remote discovery and address resolution procedure requests.

POUT. The local layer is currently attempting to connect by sniffing. It has set the 'mediaSense' flag to false and is listening to the medium for the period specified by the media access rules (see section 0 Media Access Control Procedures) to determine if the media is currently busy.

SNIFF. The local layer is currently attempting to connect by sniffing. It has transmitted its 'sniff XID response' frame and is listening for a SNRM frame from a remote layer that wishes to connect to it.

SLEEP. The local layer is currently attempting to connect by sniffing. It has turned of its receiver and is currently idle awaiting its next sniff attempt.

SCONN. A sniff-XID-response frame has been received from a remote peer layer, the service user has been informed and the local layer is awaiting the service user's response as to whether to ignore the frame or attempt to connect to the sniffer.

SSETUP. The local layer has transmitted a SNRM command frame to a remote peer layer that earlier issued a sniff-XID-response frame and is awaiting a reply.

REPLY. Behavior in this state is identical to the REPLY state in the address discovery state machine, with one exception: all transitions from REPLY to NDM become transitions from REPLY to POUT and 'start-sense-timer' is added to the actions in each of these transitions.

6.9.3.4 Event Descriptions

Sniff-Request. The service user has requested that a connection be established to a remote station using the sniff-open procedure. *Sniff-Request* should get *Connect.Indication* and appear as a passive open.

Recv Sniff-XID-Rsp:sa,info. A sniff XID response frame has been received from a remote peer layer. The remote station's device address is 'sa' and its discovery information string is 'info'. The frame advertises that the remote station is "sniffing" for a connection.

Connect-Request(da,sniff). The service user has requested the local layer to establish a connection to a device with device address 'da' that is sniffing for a connection.

P-timer-expired. The poll bit timer has expired.

Recv a:b:c:d:e:f. A frame addressed to this station has been received. Where *a* is the frame format: unnumbered (u), supervisory (s) or information (i); *b* is the frame type e.g. disc, rr; *c* is command (cmd) or response (rsp); *d* indicates P, poll bit set, ¬P, poll bit not set, F, final bit set, ¬F, final bit not set; *e*, if present, is the source device address (*e* is always NA); *f*, if present, is the destination device address. When any of the *a,b,c,d,e,f* fields is set to *x* this indicates the value of the field is unimportant, e.g. *Recv* x:x:x:x indicates the event "receive any frame not specifically enumerated".

Sense-timer-expired. The sniff sense timer has expired. The sense timer times the period wherein a station executing the sniff-open procedure listens to the medium before transmitting its “Sniff XID response” frame.

mediaBusy = false. During the preceding 500ms sense period no media activity that would indicate an active connection (at any baud rate) or discovery/address resolution process has been received.

mediaBusy = true. At some time during the preceding 500ms sense period media activity that indicates an active connection or discovery/address resolution process was detected.

Recv Discovery-XID-Cmd:S,s. A discovery command frame has been received from a remote peer layer. *S* indicates the total number of time slots the discovery procedure will use, *s* indicates the number of the current time slot. If *s* is zero then this frame initiates a discovery procedure. Otherwise ($0 < s < S$), the discovery procedure is already in progress and has reached slot number *s*. The discovery XID command uses the general XID command format described in section 0 5.7.1.4.1 Definition of XID Frame Fields with the X’01’ format identifier (FI).

Sniff-timer-expired. The sniff timer has expired. The sniff timer times the period wherein a station executing the sniff-open procedure is waiting to receive a SNRM frame after advertising its presence.

Sleep-timer-expired. The sleep timer has expired. The sleep timer times the period for a which a station executing the sniff-open procedure turns off its receiver and ignores all medium activity.

6.9.3.5 Actions Descriptions

mediaBusy := false. The mediaBusy flag is reset to False.

Start-sense-timer. The sniff sense timer is started from zero. The sense timer times the period wherein a station executing the sniff-open procedure listens to the medium before transmitting its “Sniff XID response” frame.

Discovery-Indication(sniff). Inform the service user that a “Sniff XID response” frame has been received. The device address “sa” and discovery information string “info” are passed up to the service user.

Generate-Random-Connection-Adr(ca). Generate a new random 7-bit connection address *ca*.

Send a:b:c:d:e:f:g. Transmit an IrLAP frame. Where *a* is the frame format: unnumbered (u), supervisory (s) or information (i); *b* is the frame type e.g. disc, rr; *c* is command (cmd) or response (rsp); *d* indicates P, poll bit set, ¬P, poll bit not set, F, final bit set, ¬F, final bit not set; *e*, if present, is the source device address (*e* is always NA); *f*, if present, is the destination device address. When any of the *a,b,c,d,e,f* fields is set to *x* this indicates the value of the field is unimportant.

Start-P-timer. Start the poll bit cycle timer.

Disconnect-Indication Inform the service user that either a remote peer layer or the local layer has initiated disconnection of the an established or pending data link connection.

Stop-P-timer. Stop the P bit timer.

Initialize-Connection-State. Initialize the connection state variables:

Vr := Vs := 0
window := “negotiated window size”
remoteBusy := false
retryCount := 0

Negotiate-Connection-Parameters. Compare the connection capability bytes of an incoming SNRM or UA frame with the capability bytes for this IrLAP layer and determine the best connection capability that can be supported by both stations. (Algorithm will be provided here).

Apply-Connection-Parameters. Set internal controls to apply the connection and transmission parameters determined by the last *Negotiate-Connection-Parameters* action executed.

Connect-Indication. Inform the service user that a connection has been requested by a remote peer layer.

Send Sniff-XID-Rsp:NA,discovery-info. The local layer transmits a Sniff XID response frame indicating that it is currently listening for a connection request (SNRM frame). The frame includes the stations 32 bit device address, NA, and its discovery information string.

Start-sniff-timer. Start the sniff timer from zero. The sniff timer times the period wherein a station executing the sniff-open procedure is waiting to receive a SNRM frame after advertising its presence.

Disable receiver etc. In order to conserve power a station executing the sniff-open procedure disables the medium reception and transmission equipment.

Start-sleep-timer. Start the sleep timer from zero. The sleep timer times the period for a which a station executing the sniff-open procedure turns off its receiver and ignores all medium activity.

slot := Generate-Random-Time-Slot(S,s). Generate a random time slot number between S-1 and *s* and save it in ‘slot’.

frameSent := false. When a station is in the REPLY state the ‘frameSent’ flag indicates whether it has transmitted its discovery XID response frame, frameSent false indicates the response has not been transmitted.

Start-query-timer. Start from zero the timer that times the anticipated duration of a remote discovery operation in which the local layer is participating.

Connect-confirm. Inform the service user that the remote service user has accepted the requested connection.

Start-WD-timer. Start the NRM(S) watchdog timer from zero.

Enable receiver etc. In order to conserve power a station executing the sniff-open procedure disables the medium reception and transmission equipment, this action re-enables medium reception/transmission.

Empty. No actions.

6.10

Address Conflict Resolution Procedure

6.10.1 Purpose

The address conflict resolution procedure is used when two or more stations that are within communication range of the local IrLAP link layer are determined to have selected identical device addresses. The address conflict resolution procedure is used to inform the stations of the detected conflict and to guide them in the selection of new addresses that do not conflict.

Address conflicts may be detected by a station that performs the address discovery procedure or attempts to connect with a device in an environment where multiple devices have selected the same device address (Note: in both cases such conflicting devices may not be able to "hear" each other's transmissions).

6.10.2 Overview

The initiating station sends an XID command frame with the address conflict flag set true to the "shared" device address of the conflicting nodes, this requests that they select a new address. The XID command also indicates that the replies will be spread over S time slots. Each station that receives the XID command selects a new device address and also selects a time slot (between zero and s-1) at random. The initiator then sends beginning of frame time slot XID command frames at the start of each time slot followed by an end of procedure XID command frame with the slot number set to X'FF'. The responding stations send an XID response with their new address upon receipt of the beginning of slot frame for the slot which they selected.

The new addresses do not become effective until the completion of the entire conflict resolution procedure.

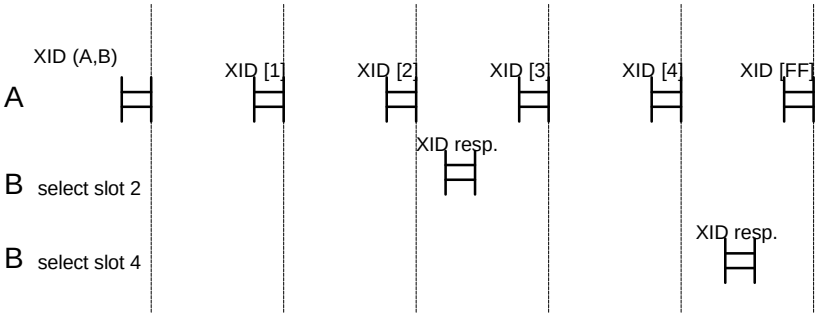
6.10.3 Precise Description of Address Conflict Resolution Procedure

If discrepancies appear to exist between the precise description of this procedure and any textual material in this specification the precise description shall be taken as the definitive description.

6.10.4 Address Conflict Resolution State Machine

The address conflict resolution procedure uses the same state machine as the discovery procedure. The only difference is that the discovery command XID frame is not broadcast it is sent to the conflicting device address, this effectively multi-casts to all conflicting stations. The discovery command XID has its "address conflict" flag set, so that each recipient selects a new address and returns it with its discovery XID response frame.

6.10.5 Address Conflict Resolution Example



6.11

Connection Establishment Procedure

6.11.1 Purpose

This procedure is used to establish an IrLAP connection to a station whose device address has been determined using the address discovery procedure.

6.11.2 Overview

Two stations use this procedure to establish a connection. One or both of the stations may actively try to establish the connection by sending a SNRM frame. The SNRM frame contains fields that indicate the connection parameters that can be supported by the sender (baud rates etc.). Upon receipt of a SNRM frame a station determines if it will accept the attempted connection or not. If it accepts it uses the negotiation procedures to determine a set of mutually acceptable connection parameters and sends a UA frame accepting the connection and indicating these parameters. If it decides not to accept it returns a DM frame.

Contained in the precise description below is the procedures for TEST frames and all other received events in the NDM state.

6.11.3 Precise Description of Connection Procedure

If discrepancies appear to exist between the precise description of this procedure and any textual material in this specification the precise description shall be taken as the definitive description.

6.11.3.1 State Chart

Current State	Event	Action(s)	Next State
NDM (entry state)	*Connect-Request(da)* ∧ mediaBusy = false	*Generate-Random-ConnectionAdr(*ca*)* dest := *da* *send* u:snrm:cmd:P:ca:dest *start-F-timer* retryCount := 0	SETUP
		Disconnect-Indication	NDM
	Connect-Request(da) ∧ mediaBusy = true	*Disconnect-Indication* -- see note 1	NDM
	Recv u:snrm:cmd:P:*c*:*d*	dest := *d;* ca := *c* *Connect-Indication*	CONN
	Recv u:test:cmd:P	*send u:test:rsp:F* -- see note 2	NDM
		Empty	NDM
	recv x:x:cmd:P	*send* u:dm:rsp:F	NDM
		Empty	NDM
	recv x:x:x:x	*Empty*	NDM
CONN	*Connect-Response*	*Negotiate-Connection-Parameters* *Initialize-Connection-State* *send* u:ua:rsp:F – see note 6 *Apply-Connection-Parameters* *start-WD-timer* -- see note 3	NRM(S)
	Disconnect-Request	*send* u:dm:rsp:F	NDM
	recv x:x:x:x	*Empty*	CONN

SETUP	*F-timer-expired* ∧ retryCount < N3	*Perform-Random-Backoff* *send* u:snrm:cmd:P:ca:dest *start-F-timer* retryCount := retryCount + 1	SETUP
	F-timer-expired ∧ retryCount ≥ N3	*Disconnect-Indication*	NDM
	recv u:snrm:cmd:P: *ca:sa* ∧ (sa > *NA*)	*stop-F-timer* *Initialize-Connection-State* *Negotiate-Connection-Parameters* *send* u:ua:rsp:F *Apply-Connection-Parameters* *Connect-Confirm* *start-WD-timer* -- see note 3	NRM(S)
		Empty -- see note 4	SETUP
	recv u:snrm:cmd:P: *sa:da* ∧ (sa < NA)	*Empty*	SETUP
	recv u:ua:rsp:F	*stop-F-timer* *Initialize-Connection-State* *Negotiate-Connection-Parameters* *Apply-Connection-Parameters* *Connect-Confirm* *send* s:rr:cmd:P *start-F-timer* -- see note 5	NRM(P)
	recv u:dm:rsp:x	*stop-F-timer* *Disconnect-Indication*	NDM
	recv u:disc:cmd:x	*stop-F-timer* *Disconnect-Indication*	NDM
	recv x:x:x:x	*Empty*	SETUP

6.11.3.2 Notes

1. It is acceptable for implementations to "hold" this event waiting for the mediaBusy flag to become false. However, should some other event occur prior to mediaBusy becoming false then the *Disconnect-Indication* action must be executed.

2. If the TEST command frame contains an information field it should be returned in the response but it is legal to always return a zero length information field.

3. The WD timer duration is set to the normal duration of the P timer for this case only. The purpose of this is to quickly detect the failure of both stations to negotiate to the new connection parameters (especially the new baud rate). This "quick disconnect" method is valid for implementations following version 1.0. It is recommended that the WD timer be set for a longer time (at least twice the duration of the P timer).

4. If a "contending SNRM" situation arises usually the station with the numerically smaller device address yields and returns a "UA" frame and connects playing the secondary role. However, in some cases a station caught in this situation may not be willing to play the secondary role. When such a situation arises the station simply does not send a "UA" frame.

5. The F timer duration can be set to one half of its regular duration in this case only. The purpose of this is to quickly detect the failure of both stations to negotiate to the new connection parameters (especially the new baud rate). This "quick disconnect" method is valid for implementations

following version 1.0. In some situations devices may take 200ms or more before they are ready to operate at the new baud rate. Thus, it is possible for secondary devices to miss the first RR from the primary. The recommended procedure is for secondary devices to wait longer than the normal P-time such as twice the normal P-time and for primaries to send more than one RR with a period on the order of a normal F-time (400 - 500 ms) between RR's.

6. In noisy environments the first UA may be missed so devices are allowed to send two UAs instead of one.

6.11.3.3 State Definitions

NDM. The station is in the normal disconnected mode. It can initiate or respond to local and remote requests to connect with a remote peer layer, it can initiate or respond to local and remote discovery and address resolution procedure requests.

CONN. A SNRM frame has been received from a remote peer layer, the service user has been informed and the local layer is awaiting the service user's refusal or acceptance of the connection.

SETUP. The local layer has transmitted a SNRM command frame to a remote peer layer and is awaiting a reply.

NRM(P). The station is in the normal response (connected) mode playing the primary role.

NRM(S). The station is in the normal response (connected) mode playing the secondary role.

6.11.3.4 Event Descriptions

Connect-Request(da). The service user has requested that a connection be established to the remote station with device address *da*.

mediaBusy = false. During the preceding 500ms sense period no media activity that would indicate an active connection (at any baud rate) or discovery process has been received.

mediaBusy = true. Either the media has not been continuously sensed for at least 500ms or, at some time during the preceding 500ms sense period media activity that indicates an active connection or discovery process was detected.

Recv a:b:c:d:e:f. A frame addressed to this station has been received. Where *a* is the frame format: unnumbered (u), supervisory (s) or information (i); *b* is the frame type e.g. disc, rr; *c* is command (cmd) or response (rsp); *d* indicates P, poll bit set, ¬P, poll bit not set, F, final bit set, ¬F, final bit not set; *e*, if present, is the (7 bit) connection address; *f*, if present, is the destination device address. When any of the *a,b,c,d,e,f* fields is set to *x* this indicates the value of the field is “don’t care”, e.g. *Recv* x:x:x:x indicates the event “receive any frame addressed to this station that has not been specifically enumerated for this state”.

Connect-Response. The service user has accepted a remote connection request.

Disconnect-Request. The service user has requested that a requested or existing connection be terminated.

F-timer-expired. The final bit timer has expired.

retryCount* < *N3. The number of retried connection attempts is less than the maximum number required.

retryCount* ‡*N3. The number of retried connection attempts has reached or exceeded the maximum number allowed.

sa* > *NA. The device address of the station which transmitted a frame is numerically greater than the device address of this station.

sa* < *NA. The device address of the station which transmitted a frame is numerically smaller than the device address of this station.

6.11.3.5 Action Descriptions

Generate-Random-Connection-Adr(ca). Generate a new random 7-bit connection address, *ca*.

Send a:b:c:d:e:f. Transmit an IrLAP frame. Where *a* is the frame format: unnumbered (u), supervisory (s) or information (i); *b* is the frame type e.g. disc, rr; *c* is command (cmd) or response (rsp); *d* indicates P, poll bit set, ¬P, poll bit not set, F, final bit set, ¬F, final bit not set; *e*, if present, is the 7-bit connection address; *f*, if present, is the destination device address. When any of the *a*,*b*,*c*,*d*,*e*,*f* fields is set to *x* this indicates the value of the field is "don't care".

Start-F-Timer. Start the final bit timer from zero.

retryCount := 0. Reset the number of retry attempts.

Disconnect-Indication. Inform the service user that either a remote peer layer or the local layer has initiated disconnection of the data link connection.

Connect-Indication. Inform the service user that a connection has been requested by a remote peer layer.

Negotiate-Connection-Parameters. Compare the connection capability bytes of an incoming SNRM or UA frame with the capability bytes for this IrLAP layer and determine the best connection capability that can be supported by both stations.

Apply-Connection-Parameters. Set internal controls to apply the connection and transmission parameters determined by the last *Negotiate-Connection-Parameters* action executed.

Initialize-Connection-State. Initialize the connection state variables:

```
Vr := Vs := 0;
window := "negotiated window size";
remoteBusy := false;
retryCount := 0;
```

Refer to the NRM(P) and NRM(S) state machines for usage of these variables.

Start-WD-timer. Start the NRM(S) watchdog timer from zero.

Perform-Random-Backoff. Wait a random number of time units, minimum duration half the time taken to transmit a SNRM frame, maximum duration 1.5 times the time taken to transmit a SNRM frame.

RetryCount := retryCount + 1. Increment the number of retry attempts.

Apply-Default-Connection-Parameters. Set internal controls to return to the default connection and transmission parameters.

Connect-Confirm. The remote service user has accepted the requested connection.

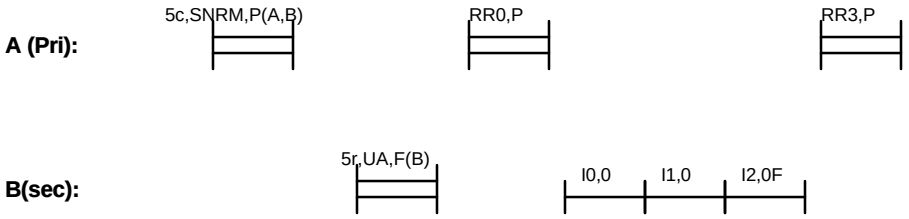
Empty. No actions required.

Stop-F-Timer. Stop the final bit timer.

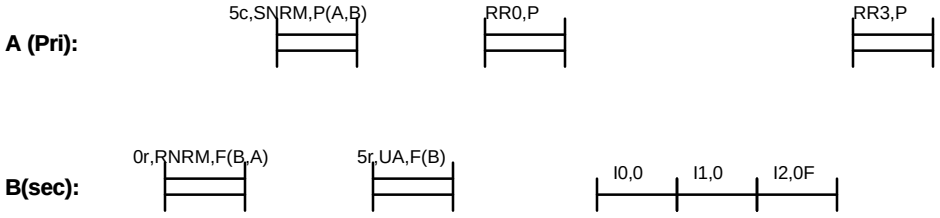
6.11.4 Connection Procedure Examples

Note in the following examples the importance of a random backoff following a time-out on a lost/damaged frame during connection startup.

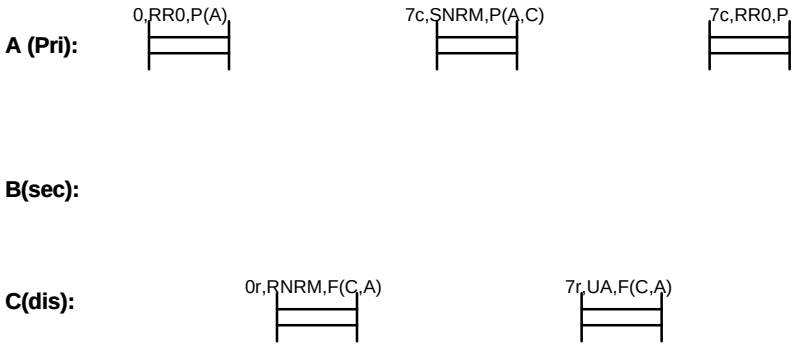
6.11.4.1 Startup Procedure without errors, secondary only information transfer



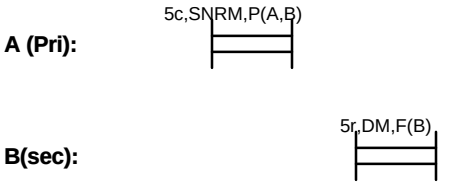
6.11.4.2 Startup Procedure using RNRM



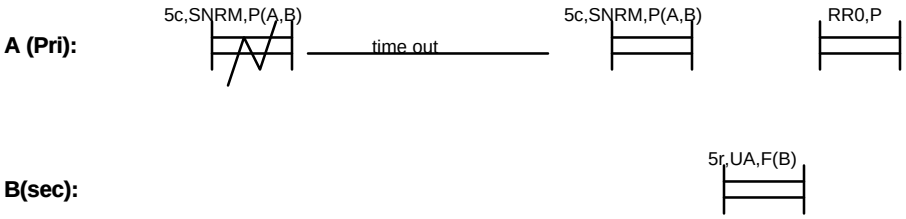
6.11.4.3 Startup Procedure New Node Joins Primary



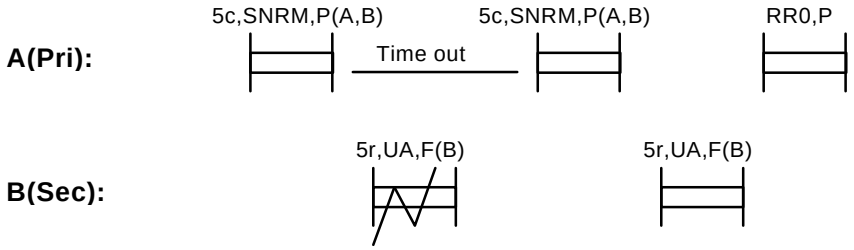
6.11.4.4 Startup procedure Secondary Station Refusal



6.11.4.5 NRM Start-up Command Error



6.11.4.6 NRM Start-up Response Error



6.12

Procedures for Information Exchange, Reset and Disconnection

6.12.1 Purpose

The information exchange procedures govern how IrLAP layers exchange information frames both inside and outside of an IrLAP connection. The reset and disconnect procedures specify how an IrLAP connection can be reset and terminated respectively.

6.12.2 Overview

After having sent the UA response frame to an SNRM command frame or having received the UA response frame to a sent SNRM frame the IrLAP layer shall accept and send UI frames, I format and S format frames according to the state machines given below.

The disconnection procedure is used to close an established connection. The reset procedure is used to reset the state of an established connection. The responsibility for all unacknowledged I-frames reverts to the service user layer when either a reset or a disconnect takes place.

Implementations are allowed to completely ignore all frames that do not contain the proper connection address including frames that contain the broadcast address. It is highly recommended that devices completely ignore XID discovery frames while connected.

6.12.3 Primary Role State Machine NRM(P)

6.12.3.1 State Chart

Current State	Event	Action(s)	Next State
XMIT (note 1)	*Data-Request(data)* ∧ ¬remoteBusy ∧ window > 1	*Send-Data-With-P-Bit-Set*	RECV
		Send-Data-With-P-Bit-Cleared	XMIT
	Data-Request(data) ∧ ¬remoteBusy ∧ window = 1	*Send-Data-With-P-Bit-Set*	RECV
	Reset-Request	*stop-P-timer* *Wait-Minimum-Turnaround-Delay* *Send* u:snrm:cmd:P retryCount := 0 *start-F-timer*	RESET
	Disconnect-Request	*Stop-P-Timer* *Wait-Minimum-Turnaround-Delay* *Send* u:disc:cmd:P *Release-Buffered-Data* *start-F-timer* retryCount := 0	PCLOSE
	Local-Busy-Detected	*Empty*	BUSY
	P-Timer-Expired	*Send* s:rr:Vr:P *start-F-timer*	RECV

RECV (entry state) (note 5)	*Recv* i:rsp:Ns:Nr:¬F	*Data-Indication* Vr := Vr + 1 mod 8 Update Nr Received AckRequired := true	RECV
	Recv i:rsp:Ns:Nr:F	*stop-F-timer* *Data-Indication* Vr := Vr + 1 mod 8 Update Nr Received AckRequired := true *start-P-timer*	XMIT
	Recv u:ui:rsp:¬F	*Unitdata-Indication*	RECV
	Recv u:ui:rsp:F	*stop-F-timer* *Unitdata-Indication* *start-P-timer*	XMIT
	Recv u:xid:rsp:F	*Wait-Minimum-Turnaround-Delay* *Send* s:rr:cmd:P:Vr AckRequired := false *start-F-timer*	RECV
	Recv i:rsp:Ns:Nr:¬F *with-unexpected-Ns*	Update Nr Received	RECV
	Recv i:rsp:Ns:Nr:F *with-unexpected-Ns*	Update Nr Received *Wait-Minimum-Turnaround-Delay* *Send* s:rr:cmd:P:Vr AckRequired := false *start-F-timer*	RECV
		Update Nr Received *Wait-Minimum-Turnaround-Delay* *Send* s:rej:cmd:P:Vr AckRequired := false *start-F-timer*	RECV
	Recv i:rsp:Ns:Nr:F *with-unexpected-Nr*	*Data-Indication* Vr := Vr + 1 mod 8 Update Nr Received resend rejected frames AckRequired := false *start-F-timer*	RECV
	Recv s:rr:rsp:Nr:F *with-unexpected-Nr*	remoteBusy := false Update Nr Received resend rejected frames *start-F-timer*	RECV

	Recv s:rej:rsp:Nr:F	Update Nr Received
		if (remoteBusy is false) then
		  resend rejected frames
		else
		  Wait-Minimum-Turnaround-Delay
		  Send s:rr:cmd:Vr:P
		start-F-timer
			RECV

Recv i:rsp:Ns:Nr:¬F *with-invalid-Ns* (see note 6) ∨ *Recv* i:rsp:Ns:Nr:¬F *with-invalid-Nr* ∨ *Recv* s:x:rsp:Nr:¬F *with-invalid-Nr*	*stop-F-timer* *Reset-Indication(local)* xmitFlag := false	RESET_ WAIT
	Empty	PCLOSE_ WAIT
Recv i:rsp:Ns:Nr:F *with-invalid-Ns* (see note 6) ∨ *Recv* i:rsp:Ns:Nr:F *with-invalid-Nr* ∨ *Recv* s:x:rsp:Nr:F *with-invalid-Nr*	*stop-F-timer* *Reset-Indication(local)* xmitFlag := true	RESET_ WAIT
	Wait-Minimum-Turnaround-Delay *Send* u:disc:cmd:P *Release-Buffered-Data* *start-F-timer* retryCount := 0	PCLOSE
Recv s:rr:rsp:Nr:F	*stop-F-timer* remoteBusy := false Update Nr Received *start-P-timer*	XMIT

Recv s:srej:rsp:Nr:F	Update Nr Received
	if (remoteBusy is false) then
	resend rejected frame
	else
	Wait-Minimum-Turnaround-Delay
	Send s:rr:cmd:Vr:P
	start-F-timer
		RECV

Recv s:rnr:rsp:Nr:F	*stop-F-timer* remoteBusy := true Update Nr Received *start-P-timer*	XMIT
Recv u:frmr:rsp:F	*stop-F-timer* *Reset-Indication(local)* xmitFlag := true	RESET_ WAIT
	stop-F-timer *start-P-timer*	XMIT
	Wait-Minimum-Turnaround-Delay *Send* u:disc:cmd:P *Release-Buffered-Data* *start-F-timer* retryCount := 0	PCLOSE

	Recv u:rd:rsp:F	*Wait-Minimum-Turnaround-Delay* *Send* u:disc:cmd:P *Release-Buffered-Data* *start-F-timer* retryCount := 0	PCLOSE
	Recv u:rnrm:rsp:F	*stop-F-timer* *Reset-Indication(remote)*	RESET_ CHECK
		Wait-Minimum-Turnaround-Delay *Send* u:disc:cmd:P *Release-Buffered-Data* *start-F-timer* retryCount := 0	PCLOSE
	Local-Busy-Detected	*Empty*	BUSY_ WAIT
	F-timer-expired ∧ retryCount < N2 ∧ retryCount ≠ N1 (see note 2)	*Wait-Minimum-Turnaround-Delay* *Send* s:rr:cmd:Vr:P *start-F-timer* retryCount := retryCount + 1	RECV
	F-timer-expired ∧ retryCount = N1 (see note 2)	*Status-Indication* *Wait-Minimum-Turnaround-Delay* *Send* s:rr:cmd:Vr:P *start-F-timer* retryCount := retryCount + 1	RECV
	F-timer-expired ∧ retryCount ≥ N2 (see note 2)	*Apply-Default-Connection-Parameters* *Disconnect-Indication*	NDM
	Recv x:x:x:¬F	*Empty.*	RECV
	Recv x:x:x:F	*stop-F-timer* *start-P-timer*	XMIT
	Recv s:x:cmd:x ∨ *Recv* i:cmd:x:x:x	*stop-F-timer* *Apply-Default-Connection-Parameters* *Disconnect-Indication(PrimaryConflict)*	NDM
PCLOSE_ WAIT	*F-timer-expired* ∨ *Recv* x:x:x:F	*Wait-Minimum-Turnaround-Delay* *Send* u:disc:cmd:P *Release-Buffered-Data* *start-F-timer* retryCount := 0	PCLOSE
	Recv x:x:x:¬F	*Empty.*	PCLOSE_ WAIT
	Recv s:x:cmd:x ∨ *Recv* i:cmd:x:x:x	*stop-F-Timer* *Apply-Default-Connection-Parameters* *Disconnect-Indication(PrimaryConflict)*	NDM
RESET_ WAIT	*Reset-Request* ∧ xmitFlag	*Wait-Minimum-Turnaround-Delay* *Send* u:snrm:cmd:P *start-F-timer*	RESET
	Reset-Request ∧ ¬xmitFlag	*start-F-timer*	RESET
	Disconnect-Request	*Wait-Minimum-Turnaround-Delay* *Send* u:disc:cmd:P *Release-Buffered-Data* *start-F-timer* retryCount := 0	PCLOSE

RESET_ CHECK	*Reset-Response*	*Wait-Minimum-Turnaround-Delay* *Send* u:snrm:cmd:P *Initialize-Connection-State* *start-F-timer*	RESET
	Disconnect-Request	*Wait-Minimum-Turnaround-Delay* *Send* u:disc:cmd:P *Release-Buffered-Data* *start-F-timer* retryCount := 0	PCLOSE
RESET	*Recv* u:ua:rsp:F	*stop-F-timer* *Initialize-Connection-State* *Reset-Confirm* remoteBusy := false *start-P-timer*	XMIT
	Recv u:dm:rsp:F	*stop-F-timer* *Apply-Default-Connection-Parameters* *Disconnect-Indication*	NDM
	Recv x:x:x:x	*Empty*	RESET
	Recv x:x:rsp:F	*Wait-Minimum-Turnaround-Delay* *Send* u:snrm:cmd:P *start-F-timer*	RESET
	F-timer-expired ∧ retryCount < *N3*	*Wait-Minimum-Turnaround-Delay* *Send* u:snrm:cmd:P *start-F-timer*	RESET
	F-timer-expired ∧ retryCount ≥ N3	*Apply-Default-Connection-Parameters* *Disconnect-Indication*	NDM
BUSY	*Data-Request(data)* ∧ window > 1 ∧ ¬remoteBusy	*Send-Data-With-P-Bit-Cleared*	BUSY
	Data-Request(data) ∧ ¬remoteBusy ∧ window = 1	*stop-P-timer* *Send-Data-With-P-Bit-Set* window := windowSize *start-F-timer*	BUSY_ WAIT
	Disconnect-Request	*stop-P-timer* *Wait-Minimum-Turnaround-Delay* *Send* u:disc:cmd:P *Release-Buffered-Data* *start-F-timer* retryCount := 0	PCLOSE
	Local-Busy-Cleared	*stop-P-timer* *Wait-Minimum-Turnaround-Delay* *Send* s:rr:cmd:P *start-F-timer*	RECV
	P-timer-expired	*Wait-Minimum-Turnaround-Delay* *Send* s:rnr:Vr:P *start-F-timer*	BUSY_ WAIT
BUSY_ WAIT	*Recv* i:rsp:Ns:Nr:¬F ∨ *Recv* i:rsp:Ns:Nr:¬F *with-unexpected-Ns*	Update Nr Received	BUSY_ WAIT

Recv i:rsp:Ns:Nr:F ∨ *Recv* i:rsp:Ns:Nr:F *with-unexpected-Ns*	*stop-F-timer* Update Nr Received *start-P-timer*	BUSY
Recv u:ui:rsp:¬F	*Empty*	BUSY_ WAIT
Recv u:ui:rsp:F ∨ *Recv* u:xid:rsp:F	*stop-F-timer* *start-P-timer*	BUSY
Recv s:rr:rsp:F	*stop-F-timer* Update Nr Received remoteBusy := false *start-P-timer*	BUSY
Recv s:rnr:rsp:F	*stop-F-timer* Update Nr Received remoteBusy := true *start-P-timer*	BUSY

Recv s:rej:rsp:F	Update Nr Received
	if (remoteBusy is false) then
	resend rejected frames
	else
	Wait-Minimum-Turnaround-Delay
	Send s:rnr:cmd:Vr:P
	start-F-timer
		BUSY_
		WAIT

Recv s:srej:rsp:Nr:F	Update Nr Received
	if (remoteBusy is false) then
	resend rejected frame
	else
	Wait-Minimum-Turnaround-Delay
	Send s:rnr:cmd:Vr:P
	start-F-timer
		BUSY_
		WAIT

Recv u:rd:rsp:F	*Wait-Minimum-Turnaround-Delay* *Send* u:disc:cmd:P *Release-Buffered-Data* *start-F-timer* retryCount := 0	PCLOSE
Recv u:frmr:rsp:F	*stop-F-timer* *Reset-Indication(local)* xmitFlag := true	RESET_ WAIT
	stop-F-timer *start-P-timer y*	BUSY
	Wait-Minimum-Turnaround-Delay *Send* u:disc:cmd:P *Release-Buffered-Data* *start-F-timer* retryCount := 0	PCLOSE
Recv u:rnrm:rsp:F	*stop-F-timer* *Reset-Indication(remote)*	RESET_ CHECK
	Wait-Minimum-Turnaround-Delay *Send* u:disc:cmd:P *Release-Buffered-Data* *start-F-timer* retryCount := 0	PCLOSE

	Recv x:x:x:¬F	*Empty.*	BUSY_ WAIT
	Recv x:x:x:F	*stop-F-timer* *start-P-timer*	BUSY
	F-timer-expired ∧ retryCount < N2 ∧ retryCount ≠ N1 (see note 2)	*Wait-Minimum-Turnaround-Delay* *Send* s:rnr:cmd:Vr:P *start-F-timer* retryCount := retryCount + 1	BUSY_ WAIT
	F-timer-expired ∧ retryCount = N1 (see note 2)	*Status-Indication* *Wait-Minimum-Turnaround-Delay* *Send* s:rnr:cmd:Vr:P *start-F-timer* retryCount := retryCount + 1	BUSY_ WAIT
	F-timer-expired ∧ retryCount ≥ N2 (see note 2)	*Apply-Default-Connection-Parameters* *Disconnect-Indication*	NDM
PCLOSE	*Recv* u:ua:rsp:F	*Stop-F-timer* *Apply-Default-Connection-Parameters* *Disconnect-Indication*	NDM
	Recv u:dm:rsp:F	*Stop-F-timer* *Apply-Default-Connection-Parameters* *Disconnect-Indication*	NDM
		Empty	PCLOSE
	Recv s:x:cmd:x ∨ *Recv* i:cmd:x:x:x	*stop-F-timer* *Apply-Default-Connection-Parameters* *Disconnect-Indication*	NDM
	F-timer-expired ∧ retryCount < N3	*Wait-Minimum-Turnaround-Delay* *Send* u:disc:cmd:P *start-F-timer* retryCount := retryCount + 1	PCLOSE
	F-timer-expired ∧ retryCount ≥ N3	*Apply-Default-Connection-Parameters* *Disconnect-Indication*	NDM

6.12.3.2 **Notes**

1. Whenever a transition into the XMIT state (from some receiving state) is made the station must wait for the minimum link turnaround time (established by negotiation during connection startup) before transmitting any frames. Also note that a station may acknowledge received I frames with an I frame of its own (if it has data to be sent) - there is no need for a separate S frame if an I frame will be sent.

2. The retry limits N1 and N2 are determined by the negotiated link disconnect/threshold time (see the negotiation section). N1 is set so that N1 * F-timer duration is the time negotiated for the disconnect warning threshold. N2 is set so that N2 * F-timer duration is the time negotiated for the link disconnect.

3. When an in-band SNRM connection reset occurs, the responsibility for all unacknowledged I frames assigned to the data link control reverts to a higher layer. Whether the content of the information fields of such unacked I frames is subsequently retransmitted is decided by the higher layer.

4. For a given window size “n”, a device must be able to receive n I-frames or UI-frames and one S-frame, for a total of n+1 frames, before requiring the link to turn around. This covers the case in

which a device receives some I-frames (and therefore needs to acknowledge them), but sends a window-full of UI frames (recall that UI-frames cannot acknowledge I-frames), and therefore must send an S-frame as well with the acknowledgment.

5. It is permitted to ignore received frames which are too long; an FRMR response is not required.

6. Stations are allowed to treat I-Frames with invalid Ns as if they have unexpected Ns and therefore, are not required to reset the link.

6.12.3.3 State Definitions

XMIT. The primary station has the right to transmit any type of IrLAP command frame. No secondary station has been given permission to transmit frames and so the primary does not expect to receive any transmissions from other stations.

RECV. The primary station has given permission to a secondary station to transmit IrLAP response frames (by sending a frame with the P bit set). The primary station will not transmit any frames and is expecting to receive frames only from the secondary to which transmission permission has been given.

PCLOSE_WAIT. The primary station has received a frame which causes the primary to want to disconnect the IrLAP link (such as an I-frame with an invalid Nr) except the frame does not have the P bit set. The primary is waiting for a frame with the P bit set so it can issues a disconnect.

RESET_WAIT. The IrLAP layer has informed the service user of a local reset condition and is awaiting the service user to indicate a *Reset-Request* or a *Disconnect-Request*.

RESET_CHECK. The IrLAP layer is waiting for the service user to accept or refuse a remote reset request.

RESET. As a result of a service user request the local IrLAP layer has sent a SNRM command to the remote IrLAP peer layer to reset the data link and is awaiting a reply.

BUSY. The primary station is currently able to carry out all the functions that it is capable of when in the XMIT state. However, conditions at the local IrLAP layer make it likely that when the primary gives transmit permission to a secondary (by sending a frame with the P bit set) I frames received from the secondary will have to be discarded.

BUSY_WAIT. The primary station has given permission to a secondary station to transmit IrLAP frames (by sending a frame with the P bit set). The primary station will not transmit any frames and is expecting to receive frames only from the secondary to which transmission permission has been given. Conditions at the local IrLAP layer make it likely that I frames received from the secondary will have to be discarded. S and U frames will be received and processed as usual.

PCLOSE. The station is in the normal response mode playing the primary role, and has transmitted a DISC frame to the remote peer layer in order to close the existing connection. It is currently awaiting a UA response frame from the remote peer layer.

6.12.3.4 Event Descriptions

Data-Request(data). The service user has requested that a data unit be sent over the connection by posting a IrLAP_DATA.request, or IrLAP_UNITDATA.request.

RemoteBusy. This flag is set "true" when an RNR frame has received been from the remote connection component to indicate that I frames should not be sent, It is reset to "false" when an RR frame is received from the remote connection component or the connection is reset. *Data-Request* events are not recognized unless this flag is set to "false".

Window. The number of I frames remaining in the current transmit "window".

Reset-Request. The service user has requested that the data link connection be reset.

Disconnect-Request. The service user has requested that the data link connection be terminated.

Local-Busy-Detected. The local station has detected a busy condition and will not be able to accept I frames over the connection.

P-timer-expired. The P-bit timer has expired indicating that it is time to give transmit permission to the secondary station.

Recv i:rsp:Ns:Nr:ØF. An I frame has been received from the secondary station, both the Ns and Nr fields are valid and the Ns value is the expected sequence number. The F bit was not set so additional frames will follow.

Recv i:rsp:Ns:Nr:F. An I frame has been received from the secondary station, both the Ns and Nr fields are valid and the Ns value is the expected sequence number. The F bit was set indicating the primary may now transmit frames.

Recv ... with-unexpected-Ns. An I frame has been received from the secondary station The Ns field of the frame does not contain the expected sequence number (Vr) but it is within the window size. The Nr field is valid.

Recv ... with-unexpected-Nr. An I or S frame has been received from the secondary station with the F bit set. The Nr field of the frame does not contain the expected sequence number (Vs) but it is within the window size.

Recv s:b:c:Nr:d. A supervisory frame has been received. Where *b* is the frame type, rr, rnr, rej, srej; *c* is command (cmd) or response (rsp). Nr is the value of the Nr field which is valid; *d* indicates P, poll bit set, ¬P, poll bit not set, F, final bit set, ¬F, final bit not set. When any of the *b,c,* or *d* fields is set to *x* this indicates the value of the field is "don't care",

Recv ... with-invalid-Ns. An I frame has been received from the secondary station The Ns field of the frame is invalid. The Nr field is valid. Implementations are allowed to tread invalid-Ns as unexpected-Ns.

Recv ... with-invalid-Nr. An I or S frame has been received from the secondary station where the Nr field is invalid (i.e. the sequence number requests a frame that is not the next frame to send and is not an unacknowledged frame).

Recv u:b:c:d. An unsequenced frame addressed to this connection has been received. Where *b* is the frame type, e.g. disc; *c* is command (cmd) or response (rsp); *d* indicates P, poll bit set, ¬P, poll bit not set, F, final bit set, ¬F, final bit not set. When any of the *b, c,* or *d* fields is set to *x* this indicates the value of the field is "don't care",

F-timer-expired. The final bit timer has expired.

Reset-Response. The service user has accepted a requested reset operation.

retryCount < N2. The number of retried "send" attempts is less than that required to cause a spontaneous disconnect.

retryCount = N1. The number of retried "send" attempts is less than that required to cause a spontaneous disconnect (N2), but has reached the threshold at which the service user should be warned of the problem via a *Status-Indication*.

retryCount ‡ N2. The number of retried "send" attempts has reached or exceeded the maximum number allowed, a spontaneous disconnect will occur.

retryCount < N3. The number of retried reset connection attempts is less that the maximum required.

retryCount ‡ N3. The number of retried reset connection attempts has reached or exceeded the maximum number allowed.

Local-Busy-Cleared. The local station busy condition has ended and it can again accept I frames over the connection. Note: Implementers are expected to avoid "silly window syndrome" i.e. a station that sends an RNR frame should wait until it has accumulated a non-trivial amount of buffer space before sending a RR frame.

6.12.3.5 Action Descriptions

Send-Data-With-P-Bit-Set. This action is carried out as the result of a *Data-Request* event. If this is the first frame in the window it should be sent after waiting the minimum turn around time. If the event is for reliable data the following actions are carried out:

stop-P-timer
Store[Vs] := data; Ack[Vs] := false
Send i:Vr:Vs:P:*data*
Vs := Vs + 1 mod 8
window := windowSize
AckRequired := false
start-F-timer

If the event is for unreliable these actions are carried out:

stop-P-timer
*if (*AckRequired *is* true*)*
then *Send* u:ui:cmd: :¬P:*data*
Send s:rr:cmd:Vr:P
AckRequired := false
else *Send* u:ui:cmd:P:*data*
window := windowSize
start-F-timer

Send-Data-With-P-Bit-Cleared. This action is carried out as the result of a *Data-Request* event. The first frame should be sent after waiting the minimum turn around time. If the event is for reliable data the following actions are carried out:

Store[Vs] := data; Ack[Vs] := false
Send i:Vr:Vs:¬P:*data*
Vs := Vs + 1 mod 8
AckRequired := false
window := window - 1

If the event is for unreliable data these actions are carried out:

Send u:ui:cmd:¬P:*data*
window := window - 1

Stop-P-timer. Stop the P bit timer.

Store[Vs] := data; Ack[Vs] := false. Save the data transmitted in the I frame with sequence number Vs ready for retransmission if requested, record that it has not been acknowledged yet.

Send i:Vr:Vs:a:data. Transmit an I frame with sequence number Vs, piggyback acknowledge I frames received up to Vr. The *a* field indicates P, poll bit set, ¬P, poll bit not set, F, final bit set, ¬F, final bit not set.

Vs := Vs + 1 mod 8. Increment Vs (modulo 8) to get the sequence number of the next I frame to send.

window := windowSize. Set the number of I frames that may still be transmitted in the current transmit window (window) to the maximum allowed on this connection (windowSize). Note: windowSize is determined during the connection establishment negotiation.

Start-F-timer. Start the final bit timer from zero.

window := window-1. Decrement the number of I frames that may still be transmitted in the current transmit window.

Send u: b:c:d. Send an unsequenced frame over the established connection. Where *b* is the frame type, e.g. disc; *c* is command (cmd) or response (rsp); *d* indicates P, poll bit set, ¬P, poll bit not set, F, final bit set, ¬F, final bit not set. When any of the *b, c,* or *d* fields is set to *x* this indicates the value of the field is "don't care",

retryCount := 0. Reset the number of retry attempts.

Release-Buffered-Data. Release buffered copies of unacknowledged I frames held in "Store". Responsibility for these I frames reverts to the service user.

Send s:b:Vr:d. Send a supervisory frame. Where *b* is the frame type, rr, rnr, rej, srej; *c* is command (cmd) or response (rsp). Vr is the sequence number of the next I frame expected by this layer; *d* indicates P, poll bit set, ¬P, poll bit not set, F, final bit set, ¬F, final bit not set. When any of the *b ,c,* or *d* fields is set to *x* this indicates the value of the field is "don't care",

Data-Indication. Pass the information field of a received I frame to the service user.

Update Nr Received. If the Nr field of the received frame acknowledges receipt of one or more previously transmitted I frames, remove those frames from the "Store" buffer and mark them "true" in the "Ack" buffer.

Stop-F-timer. Stop the final bit timer.

Start-P-timer. Start the poll bit timer from zero.

Wait-Minimum-Turnaround-Delay. Whenever a transition into the XMIT state (from some receiving state) is made the station must wait for the minimum link turnaround time (established by negotiation during connection startup) before transmitting any frames.

Resend Rejected Frame(s). If the Nr field of the received I or S frame "not" acknowledges one or more previously transmitted I frames, or one or more previously transmitted I frames is specifically rejected by receipt of a REJ or SREJ frame, retransmit the rejected I frames. The first frame sent should be sent after waiting the minimum turn around time. If more data frames exist then additional data frames can be sent to fill the window.

RemoteBusy. When set to "true" the remoteBusy flag indicates that the remote IrLAP layer is currently unable to accept I frames due to a busy condition. When set "false" the remote IrLAP layer is able to accept I frames.

Reset-Indication. Inform the service user that either the remote station has requested a reset of the data link connection, or that the local IrLAP layer has determined that the data link connection is in need of reinitialization.

XmitFlag. When set true indicates that a SNRM frame may be transmitted when transitioning from the RESET_WAIT to the RESET state. When set false a SNRM frame must not be transmitted when transitioning from RESET_WAIT to RESET.

AckRequired When set true indicates that an I-Frame requires and acknowledgment so if only UI frames are transmitted then an S-Frame (RR) must also be transmitted to acknowledge the I-Frames.

Initialize-Connection-State. Initialize the connection state variables:

```
Vr := Vs := 0;
window := windowSize;
remoteBusy := false;
retryCount := 0;
```

retryCount := retryCount + 1. Increment the number of retry attempts.

Status-Indication. Inform the service user that the link has passed its "retry" threshold and a spontaneous disconnect is likely unless corrective action is taken.

Apply-Default-Connection-Parameters. Configure IrLAP layer to use the default connection and transmission parameters, e.g. return to default baud rate (9600 bps).

Disconnect-Indication(PrimaryConflict). Inform the service user that the local layer has initiated disconnection of the data link connection due to detection of one or more stations behaving as primary stations.

Reset-Confirm. Inform the service user that the remote station has accepted the reset operation.

Unitdata-Indication. Pass the information field of a received UI frame to the service user.

Empty. No actions.

6.12.4 Secondary Role State Machine NRM(S)

6.12.4.1 State Chart

Current State	Event	Action(s)	Next State

XMIT (note 1)	*Data-Request(data)* ∧ *Pending-Data-Requests* ∧ ¬remoteBusy ∧ window > 1	*Send-Data-With-F-Bit-Cleared*	XMIT
	Data-Request(data) ∧ ¬remoteBusy ∧ window > 1	*Send-Data-With-F-Bit-Set*	RECV
	Data-Request(data) ∧ ¬remoteBusy ∧ window = 1	*Send-Data-With-F-Bit-Set*	RECV
	Disconnect-Request	*Send* u:rd:rsp:F *Release-Buffered-Data* *Start-WD-timer*	SCLOSE
	Reset-Request	*Send* u:rnrm:rsp:F retryCount := 0 *Start-WD-timer*	RESET
	Local-Busy-Detected	*Empty*	BUSY
RECV (entry state) (note 5)	*Recv* i:cmd:Ns:Nr:¬P	*Data-Indication* Vr := Vr + 1 mod 8 Update Nr Received AckRequired := true *Start-WD-timer (optional: see note 6)*	RECV
	Recv i:cmd:Ns:Nr:P ∧ *Pending-Data-Requests* ∧ ¬ remoteBusy	*Data-Indication* Vr := Vr + 1 mod 8 Update Nr Received AckRequired := true *Stop-WD-timer*	XMIT
	Recv i:cmd:Ns:Nr:P ∧ (*No-Pending-Data-Requests* ∨ remoteBusy)	*Data-Indication* Vr := Vr + 1 mod 8 Update Nr Received *Wait-Minimum-Turnaround-Delay* *Send* s:rr:rsp:Vr:F *Start-WD-timer*	RECV
	Recv u:ui:cmd:¬P	*Unitdata-Indication* *Start-WD-timer (optional: see note 6)*	RECV
	Recv u:ui:cmd:P ∧ *Pending-Data-Requests* ∧ ¬ remoteBusy	*Unitdata-Indication* *Stop-WD-timer*	XMIT
	Recv u:ui:cmd:P ∧ (*No-Pending-Data-Requests* ∨ remoteBusy)	*Unitdata-Indication* *Wait-Minimum-Turnaround-Delay* *Send* s:rr:rsp:Vr:F AckRequired := false *Start-WD-timer*	RECV
	Recv u:xid:cmd:P	*Wait-Minimum-Turnaround-Delay* *Send* s:rr:rsp:Vr:F AckRequired := false *Start-WD-timer*	RECV
	Recv u:test:cmd:P	*Wait-Minimum-Turnaround-Delay* *send u:test:rsp:F* *Start-WD-timer*	RECV

	Wait-Minimum-Turnaround-Delay *Send* s:rr:rsp:Vr:F *Start-WD-timer*	RECV
Recv i:cmd:Ns:Nr:¬P *with-unexpected-Ns*	Update Nr Received *Start-WD-timer (optional: see note 6)*	RECV
Recv i:cmd:Ns:Nr:P *with-unexpected-Ns*	Update Nr Received *Wait-Minimum-Turnaround-Delay* *Send* s:rr:rsp:F:Vr *Start-WD-timer*	RECV
	Update Nr Received *Wait-Minimum-Turnaround-Delay* *Send* s:rej:rsp:F:Vr *Start-WD-timer*	RECV
Recv i:cmd:Ns:Nr:P *with-unexpected-Nr*	*Data-Indication* Vr := Vr + 1 mod 8 Update Nr Received resend rejected frames *Start-WD-timer*	RECV
Recv s:rr:cmd:Nr:P *with-unexpected-Nr*	remoteBusy := false Update Nr Received resend rejected frames *Start-WD-timer*	RECV

Recv s:rej:cmd:Nr:P	Update Nr Received
	if (remoteBusy is false) then
	  resend rejected frames
	else
	  Wait-Minimum-Turnaround-Delay
	  Send s:rr:rsp:Vr:F
	Start-WD-timer
		RECV

Recv i:cmd:Ns:Nr:¬P *with-invalid-Ns* (see note 7*)* ∨ *Recv* i:cmd:Ns:Nr:¬P *with-invalid-Nr* ∨ *Recv* s:x:cmd:Nr:¬P *with-invalid-Nr*	*Prepare-FRMR-response*	ERROR
Recv i:cmd:Ns:Nr:P *with-invalid-Ns* (see note 7*)* ∨ *Recv* i:cmd:Ns:Nr:P *with-invalid-Nr* ∨ *Recv* s:x:cmd:Nr:P *with-invalid-Nr*	*Wait-Minimum-Turnaround-Delay* *Send u:frmr:rsp:F* *Start-WD-timer*	RECV
Recv s:rr:cmd:Nr:P ∧ *Pending-Data-Requests* ∧ ¬ remoteBusy	remoteBusy := false Update Nr Received *Stop-WD-timer*	XMIT

	Recv s:rr:cmd:Nr:P ∧ (*No-Pending-Data-Requests* ∨ remoteBusy)	remoteBusy := false Update Nr Received *Wait-Minimum-Turnaround-delay* *Send* s:rr:rsp:Vr:F *Start-WD-timer*	RECV

	Recv s:srej:cmd:Nr:P	Update Nr Received
		if (remoteBusy is false) then
		resend rejected frame
		else
		Wait-Minimum-Turnaround-Delay
		Send s:rr:rsp:Vr:F
		Start-WD-timer
			RECV

	Recv s:rnr:cmd:Nr:P	remoteBusy := true Update Nr Received *Wait-Minimum-Turnaround-delay* *Send* s:rr:rsp:Vr:F *Start-WD-timer*	RECV
	Recv u:disc:cmd:P	*Wait-Minimum-Turnaround-Delay* *Send* u:ua:rsp:F *Stop-WD-timer* *Release-Buffered-Data* *Apply-Default-Connection-Parameters* *Disconnect-Indication*	NDM
	Recv Unknown-Frame	*Wait-Minimum-Turnaround-Delay* *Send* u:frmr:rsp:F *Start-WD-timer*	RECV
	Local-Busy-Detected	*Empty*	BUSY_ WAIT
	Recv u:snrm:cmd:P	*Stop-WD-Timer* *Reset-Indication(remote)*	RESET_ CHECK
		Wait-Minimum-Turnaround-Delay *Send* u:rd:rsp:F *Start-WD-timer*	SCLOSE
	WD-timer-expired *(see note 2)*	*Status-Indication* *Start-WD-Timer*	RECV
		Apply-Default-Connection-Parameters *Release-Buffered-Data* *Disconnect-Indication(NoResponse)*	NDM
	Recv s:x:rsp:x ∨ *Recv* i:rsp:x:x:x	*Stop-WD-Timer* *Apply-Default-Connection-Parameters* *Disconnect-Indication(PrimaryConflict)*	NDM
	Recv x:x:cmd:~P	*Empty*	RECV
ERROR	*Recv* x:x:x:P	*Wait-Minimum-Turnaround-Delay* *Send* u:frmr:rsp:F *Start-WD-timer*	RECV
	Recv u:disc:cmd:P	*Stop-WD-Timer* *Send* u:ua:rsp:F *Release-Buffered-Data* *Apply-Default-Connection-Parameters* *Disconnect-Indication*	NDM

	Recv u:dm:rsp:P	*Stop-WD-Timer* *Release-Buffered-Data* *Apply-Default-Connection-Parameters* *Disconnect-Indication*	NDM
	Recv x:x:x:¬P	*Start-WD-timer (optional: see note 6)*	ERROR
RESET_ CHECK	*Reset-Response*	*Send* u:ua:rsp:F *Initialize-Connection-State* *Start-WD-timer* *release buffered data*	RECV
	Disconnect-Request	*Wait-Minimum-Turnaround-Delay* *Send* u:rd:rsp:F *Start-WD-timer*	SCLOSE
RESET	*Recv* u:snrm:cmd:P	*Initialize-Connection-State* *Wait-Minimum-Turnaround-Delay* *Send* u:ua:rsp:F *Reset-Confirm* *Start-WD-timer*	RECV
	Recv u:dm:x:P	*Stop-WD-timer* *Release-Buffered-Data* *Apply-Default-Connection-Parameters* *Disconnect-Indication*	NDM
	Recv x:x:x:x	*Empty*	RESET
	Recv x:x:cmd:P	*Wait-Minimum-Turnaround-Delay* *Send* u:rnrm:rsp:F *Start-WD-timer*	RESET
	WD-timer-expired	*Release-Buffered-Data* *Apply-Default-Connection-Parameters* *Disconnect-Indication*	NDM
BUSY	*Data-Request(data)* ∧ *Requests-Pending* ∧ window > 1 ∧ ¬remoteBusy	*Send-Data-With-F-Bit-Cleared*	BUSY
	Data-Request(data) ∧ ¬remoteBusy ∧ window = 1	*Send-Data-With-F-Bit-Cleared* *Send* s:rnr:Vr:F window := windowSize *Start-WD-timer*	BUSY_ WAIT
	Local-Busy-Cleared	*Wait-Minimum-Turnaround-Delay* *Send* s:rr:rsp:F *Start-WD-timer*	RECV
BUSY_ WAIT	*Recv* i:cmd:Ns:Nr:¬P ∨ *Recv* i:cmd:Ns:Nr:¬P *with-unexpected-Ns*	Update Nr Received *Start-WD-timer (optional: see note 6)*	BUSY_ WAIT
	(Recv i:cmd:Ns:Nr:P ∨ *Recv* i:cmd:Ns:Nr:P *with-unexpected-Ns)* ∧ *Pending-Requests* ∧ ¬ remoteBusy	Update Nr Received *Stop-WD-timer*	BUSY

(Recv i:cmd:Ns:Nr:P ∨ *Recv* i:cmd:Ns:Nr:P *with-unexpected-Ns)* ∧ *(No-Pending-Data-Requests* ∨ remoteBusy)	Update Nr Received *Wait-Minimum-Turnaround-Delay* *Send* s:rnr:rsp:Vr:F *Start-WD-timer*	BUSY_ WAIT
Recv u:ui:cmd:¬P	*Start-WD-timer (optional: see note 6)*	BUSY_ WAIT
Recv u:ui:cmd:P ∧ *Pending-Requests* ∧ ¬ remoteBusy	*Stop-WD-timer*	BUSY
Recv u:ui:cmd:P ∧ *(No-Pending-Data-Requests* ∨ remoteBusy)	*Wait-Minimum-Turnaround-Delay* *Send* s:rnr:rsp:Vr:F *Start-WD-timer*	BUSY_ WAIT
Recv u:xid:cmd:P	*Wait-Minimum-Turnaround-Delay* *Send* s:rnr:rsp:Vr:F *Start-WD-timer*	BUSY_ WAIT
Recv u:test:cmd:P	*Wait-Minimum-Turnaround-Delay* *send u:test:rsp:F* *Start-WD-timer*	BUSY_ WAIT
	Wait-Minimum-Turnaround-Delay *Send* s:rnr:rsp:Vr:F *Start-WD-timer*	BUSY_ WAIT
Recv s:rr:cmd:P ∧ *Pending-Requests* ∧ ¬ remoteBusy	*Stop-WD-timer* Update Nr Received remoteBusy := false	BUSY
Recv s:rr:cmd:P ∧ (*No-Pending-Data-Requests* ∨ remoteBusy)	Update Nr Received *Wait-Minimum-Turnaround-Delay* *Send* s:rnr:rsp:Vr:F *Start-WD-timer* remoteBusy := false	BUSY_ WAIT
Recv s:rnr:cmd:P ∧ ¬ *Pending-Busy-Cleared*	remoteBusy := true *Wait-Minimum-Turnaround-Delay* *Send* s:rnr:rsp:Vr:F *Start-WD-timer*	BUSY_ WAIT
Recv s:rnr:cmd:P Ù *Pending-Busy-Cleared*	remoteBusy := true	BUSY

Recv s:rej:cmd:F	Update Nr Received
	if (remoteBusy is false) then
	resend rejected frames
	else
	Wait-Minimum-Turnaround-Delay
	Send s:rnr:rsp:Vr:F
	Start-WD-timer
		BUSY_
		WAIT

	Recv s:srej:cmd:Nr:P	Update Nr Received
		if (remoteBusy is false) then
		resend rejected frame
		else
		Wait-Minimum-Turnaround-Delay
		Send s:rnr:rsp:Vr:F
		Start-WD-timer
			BUSY_ WAIT

	Recv u:disc:cmd:P	*Wait-Minimum-Turnaround-Delay* *Send* u:ua:rsp:F *Stop-WD-Timer* *Release-Buffered-Data* *Apply-Default-Connection-Parameters* *Disconnect-Indication*	NDM
	Recv u:snrm:cmd:P	*Stop-WD-timer* *Reset-Indication(remote)*	RESET_ CHECK
		Wait-Minimum-Turnaround-Delay *Send* u:rd:rsp:F *Start-WD-timer*	SCLOSE
	WD-timer-expired (see note 2)	*Status-Indication* *Start-WD-Timer*	BUSY_ WAIT
		Apply-Default-Connection-Parameters *Disconnect-Indication*	NDM
	Recv s:x:rsp:x ∨ *Recv* i:rsp:x:x:x	*Stop-WD-Timer* *Apply-Default-Connection-Parameters* *Disconnect-Indication(PrimaryConflict)*	NDM
	Recv x:x:cmd:~P	*Empty*	BUSY_ WAIT
	Recv Unknown-Frame	*Wait-Minimum-Turnaround-Delay* *Send* u:frmr:rsp:F *Start-WD-timer*	BUSY_ WAIT
SCLOSE	*Recv* u:disc:cmd:P	*Stop-WD-timer* *Send* u:ua:rsp:F *Apply-Default-Connection-Parameters* *Disconnect-Indication*	NDM
	Recv u:dm:rsp:F	*Stop-WD-timer* *Apply-Default-Connection-Parameters* *Disconnect-Indication*	NDM
		Empty	SCLOSE
	Recv s:x:rsp:x ∨ *Recv* i:rsp:x:x:x	*Stop-WD-Timer* *Apply-Default-Connection-Parameters* *Disconnect-Indication*	NDM
	Recv x:x:x:P	*Wait-Minimum-Turnaround-Delay* *Send* u:rd:rsp:F *Start-WD-timer*	SCLOSE
	Recv x:x:x:x	*Start-WD-timer* (see note 6)	SCLOSE
	WD-timer-expired	*Apply-Default-Connection-Parameters* *Disconnect-Indication*	NDM

6.12.4.2 **Notes**

1. Whenever a transition into XMIT state (from some other state) is made a *Wait-Minimum-Turnaround-Delay* action must be executed before the first frame is transmitted. Also note that a station may acknowledge received I frames with an I frame of its own (if it has data to be sent) - there is no need for a separate S frame in an I frame will be sent.

2. A count is kept of WD timer expirations and actions are taken based upon the negotiated link disconnect/threshold time (see the negotiation section). When the number of consecutive WD timer expirations equals the time negotiated for the disconnect warning threshold the *Status-Indication* action is taken to indicate to the service user that the data link connection has passed the threshold and a spontaneous disconnect is likely unless corrective action is taken. After the number of consecutive WD timer expirations equals the time negotiated for the link disconnect then the *Disconnect-Indication* action must be taken.

3. When an in-band SNRM connection reset occurs, the responsibility for all unacknowledged I frames assigned to the data link control reverts to a higher layer. Whether the content of the information fields of such unacked I frames is subsequently retransmitted is decided by the higher layer.

4. For a given window size “n”, a device must be able to receive n I-frames or UI-frames and one S-frame, for a total of n+1 frames, before requiring the link to turn around. This covers the case in which a device receives some I-frames (and therefore needs to acknowledge them), but sends a window-full of UI frames (recall that UI-frames cannot acknowledge I-frames), and therefore must send an S-frame as well with the acknowledgment.

5. It is permitted to ignore received frames which are too long; an FRMR response is not required.

6. Implementations should probably not start the WD-Timer in these cases - a secondary is not making progress unless it receives a frame with the poll bit set and can send data. The watchdog timer should only be restarted when the secondary sees a frame with the poll bit set. This will allow IrLMP and applications, which do not know or care whether their station is secondary or primary will see more consistent behavior from IrLAP.

7. Stations are allowed to treat I-Frames with invalid Ns as if they have unexpected Ns and therefore, are not required to reset the link.

6.12.4.3 State Definitions

XMIT. The secondary station has permission to transmit IrLAP response frames.

RECV. The secondary station does not have permission to transmit, it is expecting to receive frames from the primary station.

ERROR. The IrLAP layer has detected an error condition in a received frame that requires a FRMR response to be sent. The IrLAP layer is awaiting receipt of a frame with the P bit set so that it may send the appropriate FRMR response.

RESET_CHECK. The IrLAP layer is waiting for the service user to accept or refuse a remote reset request.

RESET. As a result of a service user request the local IrLAP layer has sent a RNRM command to the remote IrLAP peer layer to request a reset of the data link connection and is awaiting a reply.

BUSY. The secondary station is currently able to carry out all the functions that it is capable of when in the XMIT state. However, conditions at the local IrLAP layer make it likely that when the secondary gives transmit permission to the primary (by sending a frame with the F bit set) I frames received from the primary will have to be discarded.

BUSY_WAIT. The secondary station does not have permission to transmit any frames and is expecting to receive frames from the primary. Conditions at the local IrLAP layer make it likely that I frames received from the primary will have to be discarded. S and U frames will be received and processed as usual.

SCLOSE. The station is in the normal response mode playing the secondary role, and has transmitted an RD frame to the primary station to request to close the existing connection. It is currently awaiting a DISC command frame from the primary station.

6.12.4.4 Event Descriptions

Data-Request(data). The service user has requested that a data unit be sent over the connection by posting a IrLAP_DATA.request, or IrLAP_UNITDATA.request.

Pending-Data-Requests. There are additional data requests currently ready and awaiting service.

Pending-Requests. There are either additional data requests currently awaiting service or a local-busy-cleared event is pending.

RemoteBusy. This flag is set “true” when an RNR frame has received been from the remote connection component to indicate that I frames should not be sent, It is reset to “false” when an RR frame is received from the remote connection component or the connection is reset. *Data-Request* events are not recognized unless this flag is set to “false”.

Window. The number of I frames remaining in the current transmit “window”.

Disconnect-Request. The service user has requested that the data link connection be terminated.

Reset-Request. The service user has requested that the data link connection be reset.

Local-Busy-Detected. The local station has detected a busy condition and will not be able to accept I frames over the connection.

Recv i:cmd:Ns:Nr:ØP. An I frame has been received from the primary station, both the Ns and Nr fields are valid and the Ns value is the expected sequence number. The P bit was not set so additional frames will follow.

Recv i:cmd:Ns:Nr:P. An I frame has been received from the primary station, both the Ns and Nr fields are valid and the Ns value is the expected sequence number. The P bit was set indicating the secondary may now transmit frames.

No-Pending-Data-Requests. There are no data requests currently awaiting service.

Recv ... with-unexpected-Ns. An I frame has been received from the primary station The Ns field of the frame does not contain the expected sequence number (Vr) but it is within the window size. The Nr field is valid.

Recv ... with-unexpected-Nr. An I or S frame has been received from the primary station with the P bit set. The Nr field of the frame does not contain the expected sequence number (Vs) but it is within the window size.

Recv s:b:c:Nr:d. A supervisory frame has been received. Where *b* is the frame type, rr, rnr, rej, srej; *c* is command (cmd) or response (rsp). Nr is the value of the Nr field which is valid; *d* indicates P, poll bit set, ¬P, poll bit not set, F, final bit set, ¬F, final bit not set. When any of the *b,c,* or *d* fields is set to *x* this indicates the value of the field is "don't care",

Recv ... with-invalid-Ns. An I frame has been received from the primary station The Ns field of the frame is invalid. The Nr field is valid. Implementations are allowed to treat invalid-Ns as unexpected-Ns.

Recv ... with-invalid-Nr. An I or S frame has been received from the primary station the Nr field is invalid (i.e. the sequence number requests a frame that is not the next frame to send and is not an unacknowledged frame).

Recv u:b:c:d. An unsequenced frame addressed to this connection has been received. Where *b* is the frame type, e.g. disc; *c* is command (cmd) or response (rsp); *d* indicates P, poll bit set, ¬P, poll bit not set, F, final bit set, ¬F, final bit not set. When any of the *b,c,* or *d* fields is set to *x* this indicates the value of the field is "don't care",

Recv Unknown-Frame. An unknown or unsupported U, I, S or "other" frame has been received.

WD-timer-expired. The watchdog bit timer has expired.

Reset-Request. The service user has requested that the data link connection be reset.

Reset-Response. The service user has accepted a requested reset operation.

Local-Busy-Cleared. The local station busy condition has ended and it can again accept I frames over the connection. Note: Implementers are expected to avoid "silly window syndrome" i.e. a station that sends an RNR frame should wait until it has accumulated a non-trivial amount of buffer space before sending a RR frame.

Pending-Busy-Cleared. A local busy cleared condition has occurred and is waiting to be serviced.

6.12.4.5 Action Descriptions

Send-Data-With-F-Bit-Set. This action is carried out as the result of a *Data-Request* event. If this frame is the first frame it should be sent after waiting the minimum turn around time. If the event is for reliable data the following actions are carried out:

Store[Vs] := data; Ack[Vs] := false
Send i:rsp:Vr:Vs:F:*data*
Vs := Vs + 1 mod 8
window := windowSize
AckRequired := false
start-WD-timer

If the event is for unreliable data these actions are carried out:

*if (*AckRequired *is* true*)*
then *Send* u:ui:rsp: :¬F:*data*
Send s:rr:rsp:Vr:F
AckRequired := false
else *Send* u:ui:rsp:F:*data*

window := windowSize
start-WD-timer

Send-Data-With-F-Bit-Cleared. This action is carried out as the result of a *Data-Request* event. The first frame should be sent after waiting the minimum turn around time. If the event is for reliable data the following actions are carried out:

Store[Vs] := data; Ack[Vs] := false
Send i:rsp:Vr:Vs:¬F:*data*
Vs := Vs + 1 mod 8
AckRequired = false
window := window - 1

If the event is for unreliable data these actions are carried out:

Send u:ui:rsp:¬F:*data*
window := window - 1

Store[Vs] := data; Ack[Vs] := false. Save the data transmitted in the I frame with sequence number Vs ready for retransmission if requested, record that it has not been acknowledged yet.

Send i:rsp:Vr:Vs:a:data. Transmit an I frame with sequence number Vs, piggyback acknowledge I frames received up to Vr. The *a* field indicates F, final bit set, ¬F, final bit not set.

Vs := Vs + 1 mod 8. Increment Vs (modulo 8) to get the sequence number of the next I frame to send.

window := window-1. Decrement the number of I frames that may still be transmitted in the current transmit window.

window := windowSize. Set the number of I frames that may still be transmitted in the current transmit window (window) to the maximum allowed on this connection (windowSize). Note: windowSize is determined during the connection establishment negotiation.

Start-WD-timer. Start the watchdog bit timer from zero.

Send u: b:c:d. Send an unsequenced frame over the established connection. Where *b* is the frame type, e.g. disc; *c* is command (cmd) or response (rsp); *d* indicates P, poll bit set, ¬P, poll bit not set, F, final bit set, ¬F, final bit not set. When any of the *b*,*c*, or *d* fields is set to *x* this indicates the value of the field is “don’t care”,

retryCount := 0. Reset the number of retry attempts.

Release-Buffered-Data. Release buffered copies of unacknowledged I frames held in “Store”. Responsibility for these I frames reverts to the service user.

Data-Indication. Pass the information field of a received I frame to the service user.

Update Nr Received. If the Nr field of the received frame acknowledges receipt of one or more previously transmitted I frames, remove those frames from the “Store” buffer and mark them “true” in the “Ack” buffer.

Stop-WD-timer. Stop the watchdog bit timer.

Wait-Minimum-Turnaround-Delay. Whenever a transition into the XMIT state (from some receiving state) is made the station must wait for the minimum link turnaround time (established by negotiation during connection startup) before transmitting any frames.

Send s:b:Vr:d. Send a supervisory frame. Where *b* is the frame type, rr, rnr, rej, srej; *c* is command (cmd) or response (rsp). Vr is the sequence number of the next I frame expected by this layer; *d* indicates P, poll bit set, ¬P, poll bit not set, F, final bit set, ¬F, final bit not set. When any of the *b,c,* or *d* fields is set to *x* this indicates the value of the field is "don't care",

Resend Rejected Frame(s). If the Nr field of the received I or S frame "not" acknowledges one or more previously transmitted I frames, or one or more previously transmitted I frames is specifically rejected by receipt of a REJ or SREJ frame retransmit the rejected I frames. The first frame should be sent after waiting the minimum turn around time. If other data frames are pending it is legal to send additional frames to fit the window.

AckRequired When set true indicates that an I-Frame requires and acknowledgment so if only UI frames are transmitted then an S-Frame (RR) must also be transmitted to acknowledge the I-Frames.

RemoteBusy. When set to "true" the remoteBusy flag indicates that the remote IrLAP layer is currently unable to accept I frames due to a busy condition. When set "false" the remote IrLAP layer is able to accept I frames.

Reset-Indication. Inform the service user that either the remote station has initiated a reset of the data link connection, or that the local IrLAP layer has determined that the data link connection is in need of reinitialization.

Status-Indication. Inform the service user that the link has passed its "retry" threshold and a spontaneous disconnect is likely unless corrective action is taken.

retryCount := retryCount + 1. Increment the number of retry attempts.

Apply-Default-Connection-Parameters. Configure IrLAP layer to use the default connection and transmission parameters, e.g. return to default baud rate (9600bps).

Disconnect-Indication. Inform the service user that either a remote peer layer or the local layer has initiated disconnection of the data link connection.

Disconnect-Indication(PrimaryConflict). Inform the service user that the local layer has initiated disconnection of the data link connection due to detection of one or more stations behaving as primary stations.

Initialize-Connection-State. Initialize the connection state variables:

```
Vr := Vs := 0;
window := windowSize;
remoteBusy := false;
retryCount := 0;
```

Reset-Confirm. Inform the service user that the remote station has accepted the reset operation.

Empty. No actions.

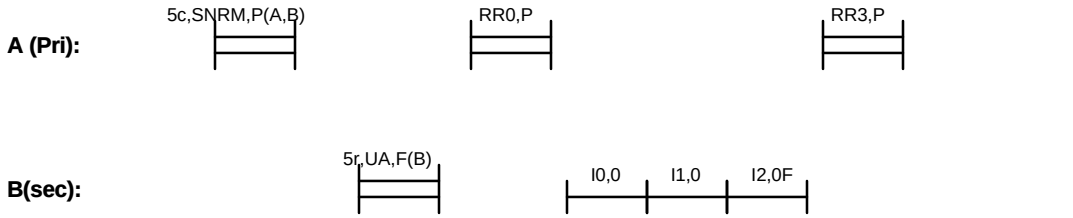
6.12.5 Information Exchange Without an IrLAP Connection

IrLAP allows unsequenced information frames to be exchanged by stations in NDM with some restrictions. Unconnected information exchange is requested by the service user through the IrLAP_UNITDATA.request and IrLAP_UNITDATA.indication primitives it can only be broadcast. The restrictions are as follows: If a station is (a) in NDM, (b) its mediaBusy flag is false and, (c) it has not

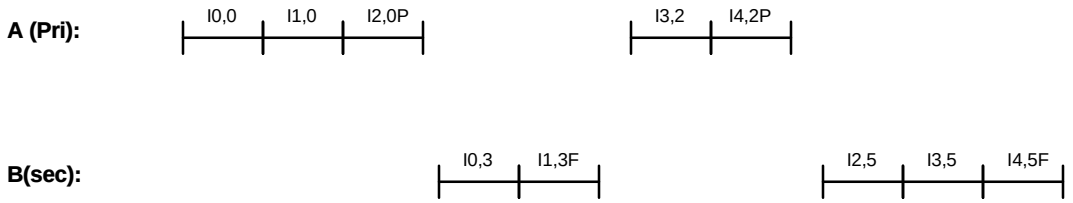
transmitted a UI frame in the last 500ms period, then the station is permitted to transmit a UI frame. The frame's A field must be set to X'FF", command with broadcast address.

6.12.6 Information Exchange Examples

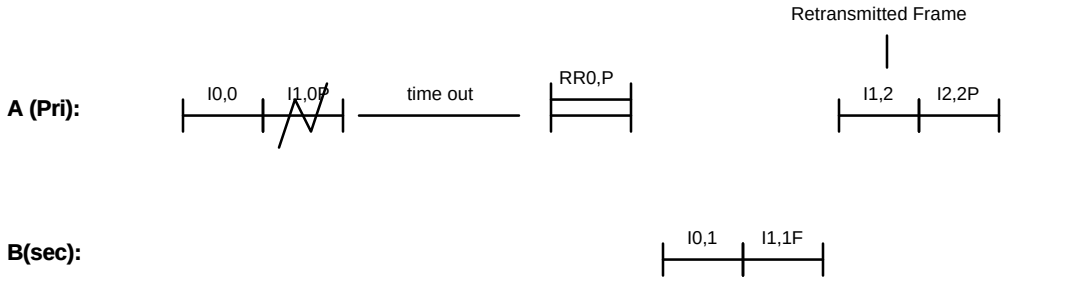
6.12.6.1 NRM Start-up Procedure and Secondary Only Information Transfer



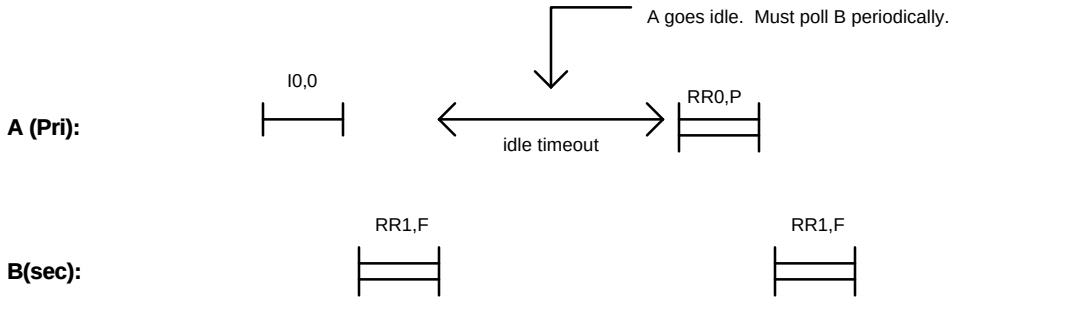
6.12.6.2 NRM Information Transfer by Primary and Secondary



6.12.6.3 NRM Primary Poll Frame Error



6.12.6.4 NRM Primary going Idle



6.13

Media Access Control Procedures

MAC Rules are the rules that govern access to the IR media. All communication on the IrDA serial infrared medium must follow these rules.

The following rules define the procedures that must be followed in order to gain access to the IR medium. Some of the rules use the word traffic. In these cases traffic is defined to be any IR pulses received including framing errors, random characters, invalid frames and good frames. Given that connections may be present at different baud rates it is necessary for stations to monitor all types of IR traffic.

1. All communication is done using IrLAP frames. Frames are divided into bytes (octets) of 8 bits.

2. Baud rates allowed on the link are limited to the following:

    - 2400 bps
    - 9600 bps
    - 19200 bps
    - 38400 bps
    - 57600 bps
    - 115200 bps
    - 576000 bps
    - 1152000 bps
    - 4000000 bps

3. All contention traffic will occur at 9600 bps. There is an optional capability for contention traffic at 2400 bps for devices that cannot perform at 9600 bps. Devices operating at 2400 bps will have more restrictive contention state rules. See 7.Appendix A (2400 bps Devices).

4. Connection traffic has priority over contention traffic. During a connection, the link must be turned around every 500 ms (1/2 second) or less. That means that a primary must poll a secondary within 500 ms and a secondary must return control (set Final bit) within 500 ms. If a primary is talking to two or more secondaries it is allowed to give each secondary 500 ms. It is possible for the primary to divide the time between the secondaries so all secondary traffic fits into 500 ms. This would have to be negotiated with the secondaries. It is possible to negotiate a window size and max frame size that would allow a device to transmit continuously for longer than 500 ms. The 500 ms turn around time rule has higher priority than any negotiated parameters. If a primary is currently receiving a frame when the 500ms period expires the primary may delay turning the line around until frame reception completes or is terminated.

5. When in the contention state (except Sniffing) a device first attempting to transmit (usually the XID discovery frame) must listen for a time period greater than 500 ms (560 - 600 ms recommended). If no traffic is detected then the device can transmit on the link. If any traffic is detected it is assumed to be connection traffic and the device cannot transmit. Section 0 6.13.3 Low Level Algorithm for Setting The MediaBusy Flag shows how this rule is related back to the state machines in section 6.0.

6. Devices performing Sniffing must follow the rules specified in the Sniffing subsection below.

7. Devices using time slots must follow the rules specified in the Time Slot subsection below.

8. A device in contention state may only repeat a procedure once to the same address (broadcast included), after which it must wait a time period greater than 500 ms. This allows address conflict resolution to different addresses to occur without the need to wait 500 ms, and will also allow a primary to connect to different machines for point-multipoint without having to wait. The three contention state procedures are discovery, address conflict resolution and connection establishment.

9. When sending frames the sender is required to keep the inter-character timing below 10 ms (regardless of baud rate). A frame with a single pair of characters that has a gap with a time greater than 10ms is considered an invalid frame. This should be verified in a certification test. However, receiving devices are not required to measure the inter-character gap; a receiving device may interpret this frame as valid. The burden lies on the sending device and certification.

6.13.1 Sniffing Rules

Devices performing sniffing (see the Node Discovery Section for more details on Sniffing) must give priority to connections. A Sniffing device following rule 4 can send out its frames as often as it wants. The following rules are given for devices that do not wish to follow rule 4,

1. A Sniffing device must listen for at least 50 ms before transmitting a frame.
2. The shortest time for which a Sniffing device must listen is 100 ms. It can divide the time into two pieces. The first piece occurs before transmitting it's XID frame and the second piece occurs after transmitting the XID frame. For example a Sniffing device could listen for 50 ms if it hears nothing it could transmit and then listen for 50 ms.
3. If a Sniffing device hears any traffic other than a frame directed toward it, it must assume a connection is in progress and follow media access rule 5 the next time it tries to sniff. If after following media access rule 5 it does not hear any traffic it can resume it's shortened listening cycle.
4. If a device chooses to listen for only 250 ms or less (down to 100 ms) it must sleep for at least 3 seconds between sniffs. If a device chooses to listen for greater than 250 ms up to 400 ms it must sleep for at least 2 1/2 seconds. If it listens for more than 400 ms it must sleep for at least 2 seconds. If a device uses media access rule 5 there is no requirement for sleeping.

6.13.2 Time Slot Rules

1. Successive Beginning of Slot frames (BOS) must be separated by at least 25 ms.

2. Devices responding to BOS frames must commence transmission of the first BOF (X'FF') of the 11 BOFs required in NDM within 10 ms and complete the response within 70 ms.

6.13.3 Low Level Algorithm for Setting The MediaBusy Flag

The variable, *mediaBusy*, is used throughout the state machines to indicate the state of the media and specify whether or not it is legal to transmit in the contention state. This use of this variable corresponds to rule 5 of the Mac rules above. The algorithm below describes how rule 5 is applied to set the *mediaBusy* flag correctly.

boolean *inFrame* **initially** *false* -- indicates if a frame is being received
boolean *mediaBusy* **initially** *false*

-- Note: mediaBusy timer timeout is > 500 ms (550 to 600 ms recommended).

Begin

case *infraredEvent* **is**

when receiver overrun or framing error
mediaBusy := *true*
start mediaBusy timer

when SOP character received

```
            if inFrame  and preceding character was not an SOP then
                mediaBusy := true
                start mediaBusy timer
            else
                inFrame := true

        when EOP character received
            if inFrame then
                -- A frame has been received
                inFrame := false
                if the CRC check fails or the frame is not addressed to this station then
                    mediaBusy := true
                    start mediaBusy timer
                endif
            else
                mediaBusy := true
                start mediaBusy timer

        when character received (not SOP or EOP)
            if not inFrame then
                mediaBusy := true
                start mediaBusy timer

        when mediaBusy timer expires
            mediaBusy := false

    end case

end.
```

6.13.4 High Level Rules for Setting MediaBusy Flag

The previous algorithm described how *mediaBusy* is set at a low level. Listed below are some rules for how mediaBusy is set between operations:

1. After a successful discovery/address conflict resolution procedure *mediaBusy* of the initiator is set to false. This allows a device performing discovery to immediately attempt to make a connection without having to wait 500+ms.

2. The mediaBusy flag of the responder to a successful discovery/address conflict resolution procedure can be set to false but it is highly recommended that the responder wait a small period of time (70 - 100 ms) before starting an operation to allow the initiator to perform an operation such as address conflict resolution or connection establishment.

3. If the responder's WD timer expires during a discovery/address conflict resolution procedure *mediaBusy* is set to true.

4. When a device transitions from a connection to the NDM state, *mediaBusy* is set to true.

6.13.5 Restrictions on State Machine Parameters

In order to comply with the media access rules some parameters defined in the state machines must be restricted to a certain range of values. These ranges are specified below:

P-Timer Timeout. The P timer timeout must never exceed 500ms.

F-Timer Timeout. The F timer timeout must never exceed 500ms.

Slot-Timer-Timeout. The discovery and address conflict resolution slot timer timeout must never exceed 85ms and must always be at least 25ms.

7. Appendix A (2400 bps Devices)

7.1 Optional operation to support 2400 bps-only stations

By definition, a station operating per the IrLAP protocol that is not in a connection is in the contention state. Typical contention state actions include the XID Discovery process to determine addresses of stations in the local communication space and the SNRM/UA negotiation process to agree on communication parameters for a subsequent connection.

The IrLAP protocol specifies that all contention state communication must be conducted at 9600 bps. This makes it a requirement that all stations must support 9600 bps to comply with the IrLAP protocol specification.

Some stations have the capabilities to support the IrLAP protocol layer and the IrDA physical layer in all other respects except that they can only communicate at 2400 bps. Without some method to switch the contention communication for "normal" stations to 2400 bps, the 2400 bps-only stations cannot set up connections with the multi-rate stations.

This Appendix describes an optional method to support 2400 bps-only stations within an IrLAP environment. Briefly, the contention processes need to be changed as follows:

1. A 2400 bps only station that wishes to initiate a discovery process or setup a connection conforms to the normal media access rules to send a frame. It carries out a normal IrLAP process with the following exceptions:

   a) Frames are sent at 2400 bps.
   b) Initial command frames begin with a minimum of five (5) start of frame delimiters (X'C0').

2. A 2400 bps-only station responds in the normal manner except that all communication is at 2400 bps.

3. A multi-rate station that supports this option and which is in it’s NDM state with its data rate set to 9600 bps will hear the initial X'C0' characters sent at 2400 bps as a sequence of characters X'77 77 FF'. When it recognizes this sequence, it changes its data rate to 2400 bps to attempt to receive valid frames being sent at 2400 bps. This transition must be done quickly before the last start of frame character is sent by the transmitting station. If it receives a valid frame that is expected in the NDM state, it responds to the discovery process in the normal manner except that it communicates at 2400 bps.

4. When the process is completed or when the multi-rate station times out, the multi-rate station returns to the NDM state at 9600 bps.

7.2 Discovery Process

The state diagram for a multi-rate station that supports communication with 2400 bps-only stations is not changed. The state chart is changed and is given below. The key changes are that the parameter dataRate is explicitly set to either 9600 bps or 2400 bps as appropriate and the query timer is set to values that are appropriate for either 9600 bps or 2400 bps operation. Changes from the state chart in section 0 are given in bold type.

7.2.1 State Chart

7.2.1.1 Multi-rate station with support for 2400 bps-only stations

Current State	Event	Action(s)	Next State	
NDM	*Discovery-Request(S)* ∧ mediaBusy = false	**dataRate := 9600** maxSlot := (S-1) slotCount := 0 *send Discovery-XID-Cmd*: maxSlot, slotCount *start-slot-timer* log := {∅}	QUERY	[2]
	Discovery-Request(S) ∧ mediaBusy = true	*Discovery-Indication(media-busy)* -- see note 1	NDM	[1]

	Recv Discovery-XID-Cmd:S,s
	-- see note 2
		slot := *Generate-Random-Time-Slot(S,s)*
		if slot = 0 then
		Send-Discovery-XID-Rsp:NA,discovery-info
		frameSent := true
		else
		frameSent := false
		if dataRate = 2400
		stop 2400-timer
		start-query-timer
			REPLY	[5]

	recv: 2400-probe **∧ dataRate=9600**	**dataRate := 2400** **query-timer := QT24** ***start 2400-timer***	**NDM**	**[1]**
	2400-timer expired	**dataRate := 9600** **query-timer := QT96**	**NDM**	**[1]**
QUERY	*slot-timer-expired* ∧ slotCount < maxSlot	slotCount := slotCount + 1 *send Discovery-XID-Cmd:* maxSlot, slotCount *start-slot-timer*	QUERY	[4]
	slot-timer-expired ∧ slotCount ≥ maxSlot	*send End-Discovery-XID-Cmd* *Discovery-Confirm(log)* -- see note 3	NDM	[3]
	Discovery-Abort-Condition	*stop-slot-timer* *send End-Discovery-XID-Cmd* *Discovery-Indication(aborted)*	NDM	[3]
	RecvDiscovery-XID-Rsp:sa,info	log := log ∪ { <*sa,info*> }	QUERY	[4]
	Response-Collision	log := log ∪ {<ϕ, ϕ>}	QUERY	[4]
		Empty	QUERY	[4]
	Recv x:x:x:x	*Empty*	QUERY	[4]
REPLY	*Recv Discovery-XID-Cmd:S,s* ∧ (s ≥ slot) ∧ ¬frameSent	*Send Discovery-XID-Rsp: NA,discovery-info* frameSent := true	REPLY	[7]

	Recv End-Discovery-XID-Cmd	*stop-query-timer* *Discovery-Indication(remote)* **if dataRate = 2400** **begin** **dataRate := 9600** **query-timer := QT96** **end**	NDM	[6]
	query-timer-expired	**if dataRate = 2400** **begin** **dataRate := 9600** **query-timer := QT96** **end**	NDM	[6]
	Recv x:x:x:x	*Empty*	REPLY	[6]

7.2.2 Additional Event Descriptions

Recv: 2400-probe: The 2400-probe is one or more repetitions of the character set X'77 77 FF'. This is the character sequence received by a station listening at 2400 bps when the start of frame delimiter X'C0' is sent at 2400 bps.

2400-timer-expired. The timer that times the interval that a station waits to receive an initial 2400 bps contention command frame after receiving a 2400-probe has expired.

7.2.3 Additional Parameters

dataRate: Keeps track of the data rate to be used for subsequent contention mode communication. When the station is initialized, dataRate is set to 9600.

QT96: The value set for the query-timer for normal contention at 9600 bps.

QT24: The value set for the query-timer for contention at 2400 bps. Typically this value should be approximately four times QT96.

7.2.4 Additional Action Descriptions

Start 2400-timer: Start the timer that times the duration that the station will stay in the condition with dataRate=2400. This timer is used to return to dataRate=9600 if valid commands are not received while in the 2400 bps contention mode.

Stop-2400-timer. When this timer expires, reset the system for 9600 bps contention operation.

7.3 Connect/Disconnect Process

7.3.1 Multi-rate station with support for 2400 bps-only stations

The connect/disconnect state diagram for a multi-rate station that supports communication with 2400 bps-only stations is not changed. The state chart is changed and is given below. Changes to the state chart from section 0 are given in bold type.

7.3.2 State Chart

Current State	Event	Action(s)	Next State	
NDM	*Connect-Request(da)* ∧ mediaBusy = false	**dataRate := 9600** *Generate-Random-ConnectionAdr(*ca*)* dest := *da* *send* u:snrm:cmd:P:ca:dest *start-F-timer* retryCount := 0	SETUP	[5]
		Disconnect-Indication	NDM	[1]
	Connect-Request(da) ∧ mediaBusy = true	*Disconnect-Indication* -- see note 1	NDM	[1]
	Recv u:snrm:cmd:P:*c*:*d*	dest := *d;* ca := *c* *Connect-Indication* **if dataRate = 2400** ***stop 2400-timer***	CONN	[3]
	recv x:x:cmd:P	*send* u:dm:rsp:x	NDM	[1]
	recv x:x:cmd: ¬P ∨ *recv* x:x:rsp:x	*Empty*	NDM	[1]
	recv: 2400-probe **∧ dataRate=9600**	**dataRate := 2400** **query-timer := QT24** ***start 2400-timer***	**NDM**	**[1]**
	2400-timer expired	**dataRate=9600** **query-timer := QT96**	**NDM**	**[1]**
CONN	*Connect-Response*	*Negotiate-Connection-Parameters* *send* u:ua:rsp:F **If dataRate=2400** **dataRate := 9600** *Apply-Connection-Parameters* *Initialize-Connection-State* *start-WD-timer* -- see note 2	NRM(S)	[11]
	Disconnect-Request	*send* u:dm:rsp:F **If dataRate=2400** **dataRate := 9600**	NDM	[4]
	recv x:x:x:x	*Empty*	CONN	[8]
SETUP	*F-timer-expired* ∧ retryCount < N3	*Perform-Random-Backoff* *send* u:snrm:cmd:P:ca:dest *start-F-timer* retryCount := retryCount + 1	SETUP	[12]
	F-timer-expired ∧ retryCount ≥ N3	*Disconnect-Indication*	NDM	[6]
	recv u:snrm:cmd:P: *ca:sa* ∧ (sa > *NA*)	*stop-F-timer* *Initialize-Connection-State* *Negotiate-Connection-Parameters* *send* u:ua:rsp:F *Apply-Connection-Parameters* *Connect-Confirm* *start-WD-timer* -- see note 2	NRM(S)	[9]
		Empty -- see note 3	SETUP	

	recv u:snrm:cmd:P: *sa:da* ∧ (sa < NA)	*Empty*	SETUP	[12]
	recv u:ua:rsp:F	*stop-F-timer* *Initialize-Connection-State* *Negotiate-Connection-Parameters* *Apply-Connection-Parameters* *Connect-Confirm* *send* s:rr:cmd:P *start-F-timer* -- see note 4	NRM(P)	[13]
	recv u:dm:rsp:x	*Disconnect-Indication*	NDM	[6]
	recv u:disc:cmd:x	*Disconnect-Indication*	NDM	[6]
	recv x:x:x:x		SETUP	[12]

7.4 2400 bps-Only Stations

Stations that are constrained to operate as 2400 bps-only have the same state diagrams as a conventional IrLAP station. The state charts are changed only slightly. Only the differences are discussed here.

The operation of a 2400 bps-only station differs from a conventional IrLAP compliant station in the follow ways:

1. All connection mode communication is at 2400 bps instead of 9600 bps.

2. The initial discovery XID and the SNRM command frames must begin with 5 start of frame delimiters X'C0' The start of frame delimiter characters are seen by multi-rate stations in NDM mode as the 2400-probe pattern. Such stations then switch to 2400 bps to listen for valid commands.

Timers like the slot timer and the query timer are set to values approximately 4 times longer than their counterparts for normal operation at 9600 bps.

8. Appendix B (Point-To-Multipoint)

[Point-To-Multipoint is an optional feature. It will be added later]

9. Appendix C (Exchange Primary/Secondary Roles)

Exchanging primary/secondary roles is an optional procedure that is used to exchange the roles of the primary and secondary. Exchanging roles only changes which device controls the link. All other parameters, such as those negotiated at link startup, remain unchanged. This procedure is allowed when a data link is operating in a point-to-point mode only (primary connected to one secondary). The main purpose for exchanging the primary/secondary roles is for a secondary to become a primary so it can provide certain services to the upper layer. These services like point-to-multipoint require primary capabilities. Thus, before a station can attempt to connect to more than one device is must obtain primary capabilities.

This appendix is divided into three main sections. The first section, Service Specifications, discusses the services needed to obtain primary capabilities. These are the services that might result in the exchange of primary/secondary roles. The second section, Frame Structure, discusses the new unnumbered frames required. The third section, Description of Procedures, gives the actual description of the procedures for both the primary and the secondary.

9.1 Service Specifications

The following services are provided for stations that want to obtain the capabilities of a primary.

IrLAP_PRIMARY.request()
IrLAP_PRIMARY.indication()
IrLAP_PRIMARY.response(deny)
IrLAP_PRIMARY.confirm(deny)

The request primitive is used to request that the IrLAP layer obtain primary capabilities. The indication primitive tells the upper layer that a request has been issues asking that the IrLAP layer give up its primary capabilities. The response primitive provides a deny flag which when set true specifies that the request to give up primary capabilities should be denied. The confirm primitive also contains the deny flag which when set true indicates that the request for primary capabilities was denied.

9.2 Frame Structure

The table below shows the new unnumbered frames required for exchanging primary/secondary roles.

7	6	5	4	3	2	1	0	
0	1	0	P	1	1	1	1	XCHG command
1	1	0	P	1	1	1	1	DXCHG command
1	1	0	F	1	1	1	1	RXCHG response

9.2.1 XCHG (Exchange Primary/Secondary Roles)

Sent by the primary to initiate the exchange of primary/secondary roles.

9.2.2 DXCHG (Deny Exchange Primary/Secondary Roles)

Sent by the primary to deny the exchange of primary/secondary roles. The primary sends this to the secondary after receiving a RXCHG frame if it wishes to deny the request to exchange.

9.2.3 RXCHG (Request Exchange Primary/Secondary Roles)

Sent by the secondary to request that the primary initiate the exchange of primary/secondary roles by sending an XCHG frame.

9.2.4 Other Frames used for Exchanging Primary/Secondary Roles

The original secondary sends an RR frame with the C/R bit set to 1 to indicate to the primary that it has assumed the primary role. Any frame sent by the original primary with the C/R bit set to 0 (response frame) indicates to the original secondary that the original primary is now a secondary.

9.3 Description of Procedures

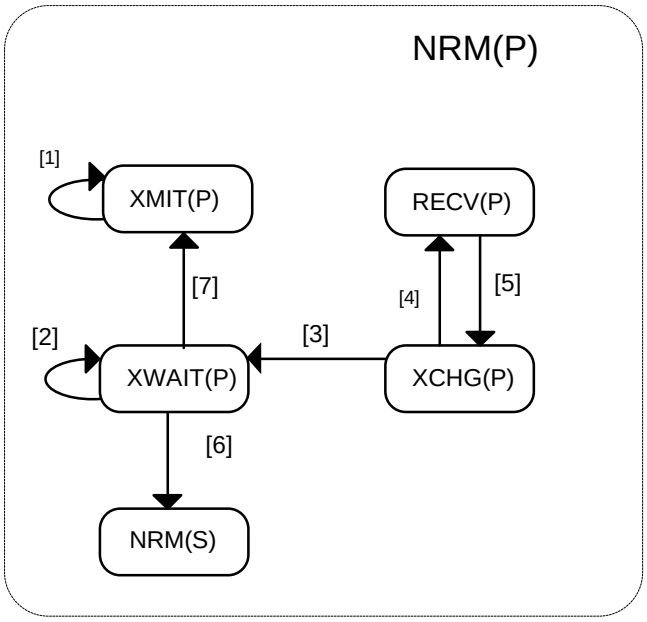
9.3.1 Procedure Overview

Only a secondary can request that the roles be exchanged. The secondary must ask the primary to initiate the procedure by sending an RXCHG frame. The primary can deny this request by sending an DXCHG frame. The primary initiates the exchange by sending an XCHG frame. The secondary can accept the exchange by sending an RR frame with the C/R bit set 1 showing that it is now a primary. After sending the XCHG frame the original primary does not know if it should become a secondary until it knows the original secondary has assumed the role. Therefore, it is possible that both machines may be primaries at the same time. Upon receiving the RR frame from the new primary the original primary switches to the secondary role and all future frames it sends have the C/R bit set to 0. When the original secondary sends the RR frame it waits until it receives a frame with the C/R bit set to 0 before it is assured the sole primary role.

9.3.2 Primary State Machine

The primary state machine for exchange of primary/secondary roles uses states from the primary information transfer and disconnect state machine. See section 0 6.12.3 Primary Role State Machine.

9.3.2.1 State Diagram



9.3.2.2 State Chart

Current State	Event	Action(s)	Next State	
XMIT(P)	*Primary-Request()*	*Primary_Confirm*	XMIT(P)	[1]
RECV(P)	*Recv* u:rxchg:rsp:F	*Primary-Indication* *start-P-timer* (note 1)	XCHG(P)	[5]
XCHG(P)	*Primary-Response(deny)*	*Send* u:dxchg:cmd:P *stop-P-timer* *start-F-timer*	RECV(P)	[4]
	Primary-Response()	*Send* u:xchg:cmd:P *stop-P-timer* *start-F-timer*	XWAIT(P)	[3]
	Disconnect-Request	*Send* u:disc:cmd:P *Release-Buffered-Data* *stop-P-timer* *start-F-timer* retryCount := 0	PCLOSE	
	P-Timer-Expired	*Send* s:rr:cmd:Vr:P *start-F-timer*	RECV(P) (note 2)	[4]
XWAIT(P)	*Recv* s:rr:cmd:Nr:P	*stop-F-timer* Update Nr Received *Send* s:rr:rsp:Vr:F *Switch-to-Secondary*	NRM(S)	[6]

Recv u:frmr:rsp:F	*Empty*	XMIT(P)	[7]
Recv u:rd:rsp:F	*Send* u:disc:cmd:P *Release-Buffered-Data* *start-F-timer* retryCount := 0	PCLOSE	
Recv u:disc:cmd:P	*Send* u:ua:rsp:F *Release-Buffered-Data* *Apply-Default-Connection-Parameters* *Disconnect-Indication*	NDM	
F-timer-expired ∧ retryCount < N2	*Perform-Random-Backoff* *Send* u:xchg:cmd:P *start-F-timer* retryCount := retryCount + 1	XWAIT(P)	[2]
F-timer-expired ∧ retryCount = N1	*Perform-Random-Backoff* *Status-Indication* *Send* u:xchg:cmd:P *start-F-timer* retryCount := retryCount + 1	XWAIT(P)	[2]
F-timer-expired ∧ retryCount ≥ N2	*Apply-Default-Connection-Parameters* *Disconnect-Indication*	NDM	
Recv x:x:rsp:F	*Send* u:xchg:cmd:P *start-F-timer* retryCount := 0	XWAIT(P)	[2]
Recv x:x:rsp: ¬F	*Empty*	XWAIT(P)	[2]
Recv x:x:cmd:x	*stop-F-timer* *Switch-to-Secondary* *Perform-Recv(S)-Action*	NRM(S)	[6]

9.3.2.3 Notes

1. The P timer is used here to provide time for the service user to respond to the indication. The length of the P timer should be the longest time allowed (max turn around time minus the time it takes to transmit a small frame like an RR, XCHG, or DXCHG.
2. If the P timer expires the state machine transitions back to the RECV state. It is assumed that the secondary will retransmit the RXCHG frame and another indication will be given to the service user. If the service user continues to ignore the indication then the secondary will give up.

9.3.2.4 State Definitions

XMIT(P). This is the primary XMIT state as specified in section 0 6.12.3 Primary Role State Machine. Only the events needed for primary/secondary exchange are shown here.

RECV(P). This the primary RECV state as specified in section 0 6.12.3 Primary Role State Machine. Only the events needed for primary/secondary exchange are shown here.

XCHG(P). The IrLAP layer has informed the service user of a request to exchange primary/secondary roles from a secondary and is awaiting the service user to indicate a *XCHG-Response(), XCHG-Response(deny),* or a *Disconnect-Request.*

XWAIT(P). The primary has sent an XCHG frame to the secondary and is waiting until the secondary switches to a primary and indicates the switch by sending an RR as a command frame.

NRM(S). This is the secondary NRM state which is governed by the state machine specified in section 0 6.12.4 Secondary Role State Machine.

9.3.2.5 Event Descriptions

Primary-Request. The service user has requested that the primary/secondary exchange roles.

***Recv* u:rxchg:rsp:F.** A frame from the secondary requesting that the primary initiate the exchange of primary/secondary roles has been received by the primary.

Primary-Response(deny). The service user has denied a request to initiate the exchange of primary/secondary roles.

Primary-Response. The service user has accepted the request to initiate the exchange of primary/secondary roles.

Disconnect-Request. The service user has requested that the link be disconnected.

P-timer-expired. The P-bit timer has expired indicating that it is time to give transmit permission to the secondary station.

F-timer-expired. The final bit timer has expired.

***retryCount* < *N2*.** The number of retried "send" attempts is less than that required to cause a spontaneous disconnect.

***retryCount* = *N1*.** The number of retried "send" attempts is less than that required to cause a spontaneous disconnect (N2), but has reached the threshold at which the service user should be warned of the problem via a *Status-Indication*.

***retryCount* ‡ *N2*.** The number of retried "send" attempts has reached or exceeded the maximum number allowed, a spontaneous disconnect will occur.

Recv a:b:c:d:e:f. A frame addressed to this station has been received. Where *a* is the frame format: unnumbered (u), supervisory (s) or information (i); *b* is the frame type e.g. disc, rr; *c* is command (cmd) or response (rsp); *d* indicates P, poll bit set, ¬P, poll bit not set, F, final bit set, ¬F, final bit not set; *e*, if present, is the (7 bit) connection address; *f*, if present, is the destination device address. When any of the *a,b,c,d,e,f* fields is set to *x* this indicates the value of the field is "don't care", e.g. *Recv* x:x:x:x indicates the event "receive any frame addressed to this station that has not been specifically enumerated for this state".

9.3.2.6 Action Descriptions

Primary_Confirm. Inform the service user that the request for primary capabilities was granted (confirmed).

Primary-Indication. Inform the service user that the secondary has made a request of the primary to initiate the exchange of primary/secondary roles.

Start-P-timer. Start the poll bit timer from zero.

Send u: b:c:d. Send an unsequenced frame over the established connection. Where *b* is the frame type, e.g. disc; *c* is command (cmd) or response (rsp); *d* indicates P, poll bit set, ¬P, poll bit not set, F, final bit set, ¬F, final bit not set. When any of the *b,c,* or *d* fields is set to *x* this indicates the value of the field is "don't care",

Start-F-timer. Start the final bit timer from zero.

Stop-P-timer. Stop the P bit timer.

Release-Buffered-Data. Release buffered copies of unacknowledged I frames held in "Store". Responsibility for these I frames reverts to the service user.

retryCount := 0. Reset the number of retry attempts.

Apply-Default-Connection-Parameters. Configure IrLAP layer to use the default connection and transmission parameters, e.g. return to default baud rate (9600bps).

Disconnect-Indication. Inform the service user that either a remote peer layer or the local layer has initiated disconnection of the data link connection.

Switch-to-Secondary. Perform necessary functions to switch over to the secondary NRM state machine maintaining current Vr and Vs counters, frame buffers and negotiated parameters. For information about the secondary NRM state machine see section, 0 6.12.4 Secondary Role State Machine.

Update Nr Received. If the Nr field of the received frame acknowledges receipt of one or more previously transmitted I frames, remove those frames from the "Store" buffer and mark them "true" in the "Ack" buffer.

Send s:b:c:Vr:d. Send a supervisory frame. Where *b* is the frame type, rr, rnr, rej, srej; *c* is command (cmd) or response (rsp). Vr is the sequence number of the next I frame expected by this layer; *d* indicates P, poll bit set, ¬P, poll bit not set, F, final bit set, ¬F, final bit not set. When any of the *b,c,* or *d* fields is set to *x* this indicates the value of the field is "don't care",

Perform-Random-Backoff. Wait a random number of time units, minimum duration half the time taken to transmit a XCHG frame, maximum duration 1.5 times the time taken to transmit a XCHG frame. At this point it is possible that both stations are primaries with F timers running so some randomness is needed.

retryCount := retryCount + 1. Increment the number of retry attempts.

Status-Indication. Inform the service user that the link has passed its "retry" threshold and a spontaneous disconnect is likely unless corrective action is taken.

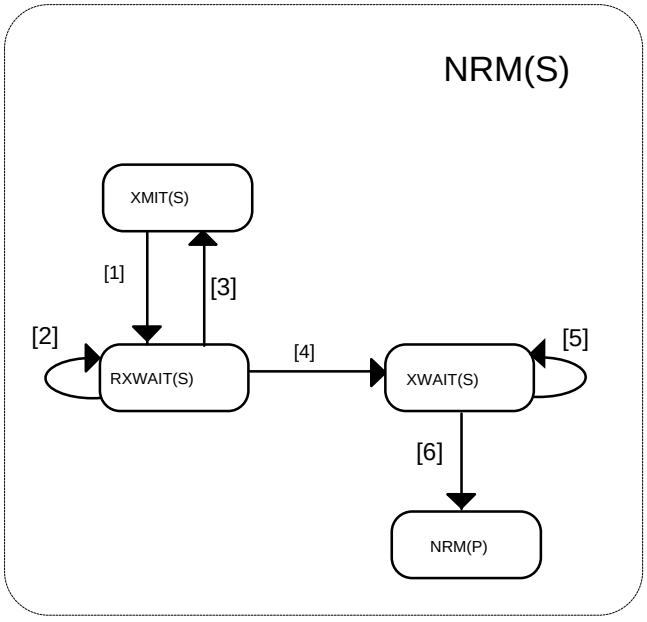
Stop-F-timer. Stop the final bit timer.

Perform-Recv(S)-Action. Perform the same actions for the given frame as the Secondary's RECV state, see section 0 6.12.4 Secondary Role State Machine. This includes going to the proper next state.

9.3.3 Secondary State Machine

The secondary state machine for exchange of primary/secondary roles uses states from the secondary information transfer and disconnect state machine. See section 0 6.12.4 Secondary Role State Machine.

9.3.3.1 State Diagram



9.3.3.2 State Chart

Current State	Event	Action(s)	Next State	
XMIT(S)	*Primary-Request()*	*Send* u:rxchg.rsp.F retryCount = 0	RXWAIT(S)	[1]
RXWAIT(S)	*Recv* u:xchg:cmd:P	*Send* s:rr:cmd:Vr:P *Primary-Confirm* *start-F-timer*	XWAIT(S)	[4]
	Recv u:dxchg:cmd:P	*Primary-Confirm(deny)*	XMIT(S)	[3]
	Recv u:disc:cmd:P	*Send* u:ua:rsp:F *Release-Buffered-Data* *Apply-Default-Connection-Parameters* *Disconnect-Indication*	NDM	

	Recv x:x:cmd:P ∧ retryCount < N4	*Send* u:rxchg:rsp:F retryCount := retryCount + 1	RXWAIT(P)	[2]
	Recv x:x:cmd:P ∧ retryCount ≥ N4	*Primary-Confirm(deny)* retryCount := 0	XMIT(S)	[3]
	Recv x:x:cmd: ¬P	*Empty*	RXWAIT(P)	[2]
XWAIT(S)	*Recv* u:xchg:cmd:P	*Send* s:rr:cmd:P:Vr *start-F-timer* *retryCount = 0*	XWAIT(S)	[5]
	Recv u:disc:cmd:P	*Send* u:ua:rsp:F *Release-Buffered-Data* *Apply-Default-Connection-Parameters* *Disconnect-Indication*	NDM	
	Recv u:rd:rsp:F	*Send* u:disc:cmd:P *Release-Buffered-Data* *start-F-timer* retryCount := 0	PCLOSE	
	F-timer-expired ∧ retryCount < N2	*Perform-Random-Backoff* *Send* s:rr:cmd:Vr:P *start-F-timer* retryCount := retryCount + 1	XWAIT(S)	[5]
	F-timer-expired ∧ retryCount = N1	*Perform-Random-Backoff* *Status-Indication* *Send* s:rr:cmd:Vr:P *start-F-timer* retryCount := retryCount + 1	XWAIT(S)	[5]
	F-timer-expired ∧ retryCount ≥ N2	*Apply-Default-Connection-Parameters* *Disconnect-Indication*	NDM	
	Recv x:x:rsp:x	*stop-F-timer* *Switch-to-Primary* retryCount := 0 *Perform-Recv(P)-Action*	NRM(P)	[6]

9.3.3.3 State Definitions

XMIT(S). This is the secondary XMIT state as specified in section 0 6.12.4 Secondary Role State Machine . Only the events needed for primary/secondary exchange are shown here.

RXWAIT(S). The secondary has sent an RXCHG frame to the primary and is waiting for an XCHG or DXCHG frame from the primary. It is possible that the primary might ignore the request to exchange because it does not implement the feature or the service user on the primary side does not understand the request (this is possible because primary/secondary exchange is an optional feature).

XWAIT(S). The secondary has received the XCHG frame from the primary and is waiting until the primary sends a command frame indicating that it has assumed a secondary role.

NRM(P). This is the primary NRM state which is governed by the state machine specified in section 0 6.12.3 Primary Role State Machine.

9.3.3.4 Event Descriptions

Primary-Request. The service user has requested that the primary/secondary exchange roles.

***Recv* u:xchg:cmd:P.** A frame from the primary initiating the exchange of primary/secondary roles has been received.

***Recv* u:dxchg:cmd:P.** A frame from the primary denying the request from the secondary to initiate the exchange of primary/secondary roles has been received.

Recv a:b:c:d:e:f. A frame addressed to this station has been received. Where *a* is the frame format: unnumbered (u), supervisory (s) or information (i); *b* is the frame type e.g. disc, rr; *c* is command (cmd) or response (rsp); *d* indicates P, poll bit set, ¬P, poll bit not set, F, final bit set, ¬F, final bit not set; *e*, if present, is the (7 bit) connection address; *f*, if present, is the destination device address. When any of the *a,b,c,d,e,f* fields is set to *x* this indicates the value of the field is "don't care", e.g. *Recv* x:x:x:x indicates the event "receive any frame addressed to this station that has not been specifically enumerated for this state".

retryCount* < *N4. The number of retried "send RXCHG" attempts is less than that required to cause the secondary to give up.

retryCount* ‡ *N4. The number of retried "send RXCHG" attempts has reached or exceeded the maximum number allowed (reasonable value is 3 - 5), the secondary will give up and tell the service user that the request was denied.

F-timer-expired. The final bit timer has expired.

retryCount* < *N2. The number of retried "send" attempts is less than that required to cause a spontaneous disconnect.

retryCount* = *N1. The number of retried "send" attempts is less than that required to cause a spontaneous disconnect (N2), but has reached the threshold at which the service user should be warned of the problem via a *Status-Indication*.

retryCount* ‡ *N2. The number of retried "send" attempts has reached or exceeded the maximum number allowed, a spontaneous disconnect will occur.

9.3.3.5 Action Descriptions

***Send u:rxchg.rsp.F*.** Send a frame to the primary requesting that the primary initiate an exchange of primary/secondary roles.

retryCount* := *0. Reset the number of retry attempts.

Send s:b:c:Vr:d. Send a supervisory frame. Where *b* is the frame type, rr, rnr, rej, srej; *c* is command (cmd) or response (rsp). Vr is the sequence number of the next I frame expected by this layer; *d* indicates P, poll bit set, ¬P, poll bit not set, F, final bit set, ¬F, final bit not set. When any of the *b,c,* or *d* fields is set to *x* this indicates the value of the field is "don't care",

Primary_Confirm. Tell the service user that the request for primary capabilities was granted (confirmed).

Start-F-timer. Start the final bit timer from zero.

Primary_Confirm(deny). Tell the service user that the request for primary capabilities was denied.

Send u: b:c:d. Send an unsequenced frame over the established connection. Where *b* is the frame type, e.g. disc; *c* is command (cmd) or response (rsp); *d* indicates P, poll bit set, ¬P, poll bit not set, F, final bit set, ¬F, final bit not set. When any of the *b*, *c*, or *d* fields is set to *x* this indicates the value of the field is "don't care",

Release-Buffered-Data. Release buffered copies of unacknowledged I frames held in "Store". Responsibility for these I frames reverts to the service user.

Apply-Default-Connection-Parameters. Configure IrLAP layer to use the default connection and transmission parameters, e.g. return to default baud rate (9600bps).

Disconnect-Indication. Inform the service user that either a remote peer layer or the local layer has initiated disconnection of the data link connection.

retryCount := retryCount + 1. Increment the number of retry attempts.

Perform-Random-Backoff. Wait a random number of time units, minimum duration half the time taken to transmit an RR frame, maximum duration 1.5 times the time taken to transmit an RR frame. At this point it is possible that both stations are primaries with F timers running so some randomness is needed.

Status-Indication. Inform the service user that the link has passed its "retry" threshold and a spontaneous disconnect is likely unless corrective action is taken.

Stop-F-timer. Stop the final bit timer.

Switch-to-Primary. Perform necessary functions to switch over to the primary NRM state machine maintaining current Vr and Vs counters, frame buffers and negotiated parameters. For information about the primary NRM state machine see section, 0 6.12.3Primary Role State Machine

Perform-Recv(P)-Action. Perform the same actions for the given frame as the primary's RECV state, see section 0 6.12.3 Primary Role State Machine. This includes going to the proper next state.

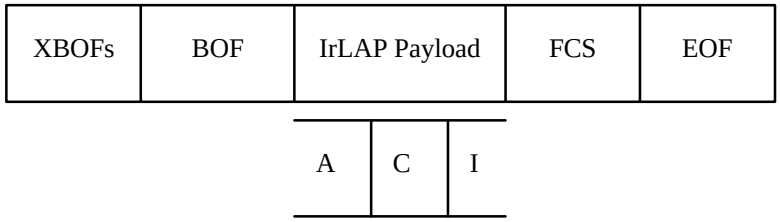
10. Appendix D (IrLAP Frame Wrappers)

Currently three different physical schemes exist for send IrLAP frames. Wrapper types for the three physical layer schemes are:

- Asynchronous (ASYNC) Framing (9600 bps - 115.2 kbps)
- Synchronous (SYNC) HDLC Framing (576 kbps and 1.152 Mbps)
- Synchronous 4 PPM Framing (4 Mbps)

10.1 ASYNC Wrapper (9600 bps - 115.2 kbps)

The ASYNC Wrapper is composed of beginning flag, ending flag and field check sequence fields as shown below.



XBOFs n occurrences of X'C0' or X'FF' (10 in NDM and negotiated in NRM)

BOF Beginning of Frame X'C0'

FCS 16 bit frame check sequence using CRC-CCITT algorithm on IrLAP Payload data

EOF End of Frame X'C1'

10.1.1 Flag

The beginning flag and the ending flag enclose the IrLAP frame. The beginning flag serves as a reference for the position of the A and C fields and initiates transmission error checking; the ending flag delimits the end of the FCS field and marks the end of the frame. The beginning flag is also referred to as Beginning of Frame (BOF) and the ending flag is referred to as End of Frame (EOF).

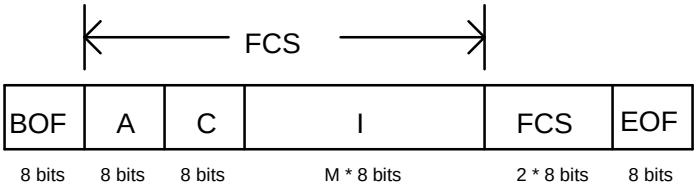
The BOF value is defined as X'C0'. The EOF value is defined as X'C1'. Multiple consecutive BOFs are allowed. The minimum number of BOFs sent with each packet can be negotiated. When multiple BOFs are sent, the value X'FF' is recommended (instead of X'C0') for all BOFs except the one immediately preceding the IrLAP payload - that one must still be X'C0'. The X'FF' value is easier for the receiver to synchronize itself on.

The ASYNC Wrapper defines a character transparency mode that transforms information bytes that would otherwise be interpreted as flags or other control characters into non-flag/control characters prior to transmission. This is described later.

10.1.2 Frame Check Sequence Field

Following the I field (or C field if no I field is present) is the frame check sequence field (FCS). The purpose of this field is to check the received frame for errors that may have been introduced during frame transmission. This field contains a 16 bit CRC-CCITT cyclic redundancy check. The CRC is computed from the A, C and I fields. See chapter 0 10.1.2.1.1 FCS Computation Method for the actual FCS algorithm. Important note: IrLAP uses the CRC-CCITT cyclic redundancy check (as shown in the appendix), **NOT** the popular XMODEM variation of CRC-CCITT.

Fields Involved in FCS computation



10.1.2.1 (Frame Check Sequence Algorithm)

This section contains an implementation of the CRC-CCITT FCS algorithm used for IrLAP. This code was copied from the **Network Working Group Request for Comments: 1171** (PPP protocol).

10.1.2.1.1 FCS Computation Method

The following code provides a table lookup computation for calculating the Frame Check Sequence as data arrives at the interface. The table is created by the code shown in the next section.

```
/*
 * u16 represents an unsigned 16-bit number.  Adjust the typedef for
 * your hardware.
 */
typedef unsigned short u16;


/*
* FCS lookup table as calculated by the table generator in section 2.
 */
static u16 fcstab[256] = {
   0x0000, 0x1189, 0x2312, 0x329b, 0x4624, 0x57ad, 0x6536, 0x74bf,
   0x8c48, 0x9dc1, 0xaf5a, 0xbed3, 0xca6c, 0xdbe5, 0xe97e, 0xf8f7,
   0x1081, 0x0108, 0x3393, 0x221a, 0x56a5, 0x472c, 0x75b7, 0x643e,
   0x9cc9, 0x8d40, 0xbfdb, 0xae52, 0xdaed, 0xcb64, 0xf9ff, 0xe876,
   0x2102, 0x308b, 0x0210, 0x1399, 0x6726, 0x76af, 0x4434, 0x55bd,
   0xad4a, 0xbcc3, 0x8e58, 0x9fd1, 0xeb6e, 0xfae7, 0xc87c, 0xd9f5,
   0x3183, 0x200a, 0x1291, 0x0318, 0x77a7, 0x662e, 0x54b5, 0x453c,
   0xbdcb, 0xac42, 0x9ed9, 0x8f50, 0xfbef, 0xea66, 0xd8fd, 0xc974,
   0x4204, 0x538d, 0x6116, 0x709f, 0x0420, 0x15a9, 0x2732, 0x36bb,
   0xce4c, 0xdfc5, 0xed5e, 0xfcd7, 0x8868, 0x99e1, 0xab7a, 0xbaf3,
   0x5285, 0x430c, 0x7197, 0x601e, 0x14a1, 0x0528, 0x37b3, 0x263a,
   0xdecd, 0xcf44, 0xfddf, 0xec56, 0x98e9, 0x8960, 0xbbfb, 0xaa72,
   0x6306, 0x728f, 0x4014, 0x519d, 0x2522, 0x34ab, 0x0630, 0x17b9,
   0xef4e, 0xfec7, 0xcc5c, 0xddd5, 0xa96a, 0xb8e3, 0x8a78, 0x9bf1,
   0x7387, 0x620e, 0x5095, 0x411c, 0x35a3, 0x242a, 0x16b1, 0x0738,
   0xffcf, 0xee46, 0xdcdd, 0xcd54, 0xb9eb, 0xa862, 0x9af9, 0x8b70,
   0x8408, 0x9581, 0xa71a, 0xb693, 0xc22c, 0xd3a5, 0xe13e, 0xf0b7,
```

```
    0x0840, 0x19c9, 0x2b52, 0x3adb, 0x4e64, 0x5fed, 0x6d76, 0x7cff,
    0x9489, 0x8500, 0xb79b, 0xa612, 0xd2ad, 0xc324, 0xf1bf, 0xe036,
    0x18c1, 0x0948, 0x3bd3, 0x2a5a, 0x5ee5, 0x4f6c, 0x7df7, 0x6c7e,
    0xa50a, 0xb483, 0x8618, 0x9791, 0xe32e, 0xf2a7, 0xc03c, 0xd1b5,
    0x2942, 0x38cb, 0x0a50, 0x1bd9, 0x6f66, 0x7eef, 0x4c74, 0x5dfd,
    0xb58b, 0xa402, 0x9699, 0x8710, 0xf3af, 0xe226, 0xd0bd, 0xc134,
    0x39c3, 0x284a, 0x1ad1, 0x0b58, 0x7fe7, 0x6e6e, 0x5cf5, 0x4d7c,
    0xc60c, 0xd785, 0xe51e, 0xf497, 0x8028, 0x91a1, 0xa33a, 0xb2b3,
    0x4a44, 0x5bcd, 0x6956, 0x78df, 0x0c60, 0x1de9, 0x2f72, 0x3efb,
    0xd68d, 0xc704, 0xf59f, 0xe416, 0x90a9, 0x8120, 0xb3bb, 0xa232,
    0x5ac5, 0x4b4c, 0x79d7, 0x685e, 0x1ce1, 0x0d68, 0x3ff3, 0x2e7a,
    0xe70e, 0xf687, 0xc41c, 0xd595, 0xa12a, 0xb0a3, 0x8238, 0x93b1,
    0x6b46, 0x7acf, 0x4854, 0x59dd, 0x2d62, 0x3ceb, 0x0e70, 0x1ff9,
    0xf78f, 0xe606, 0xd49d, 0xc514, 0xb1ab, 0xa022, 0x92b9, 0x8330,
    0x7bc7, 0x6a4e, 0x58d5, 0x495c, 0x3de3, 0x2c6a, 0x1ef1, 0x0f78
};


#define PPPINITFCS      0xffff  /* Initial FCS value */
#define PPPGOODFCS      0xf0b8  /* Good final FCS value */

/*
 * Calculate a new FCS given the current FCS and the new data.
 */
u16 pppfcs(fcs, cp, len)
    register u16 fcs;
    register unsigned char *cp;
    register int len;
{
    ASSERT(sizeof (u16) == 2);
    ASSERT(((u16) -1) > 0);
    while (len--)
        fcs = (fcs >> 8) ^ fcstab[(fcs ^ *cp++) & 0xff];

    return (fcs);
}
```

10.1.2.1.2 Fast FCS table generator

The following was copied from the **Network Working Group Request for Comments: 1171.**

```
The following code creates the lookup table used to calculate the
FCS.

/*
 * Generate a FCS table for the HDLC FCS.
 *
 * Drew D. Perkins at Carnegie Mellon University.
 *
 * Code liberally borrowed from Mohsen Banan and D. Hugh Redelmeier.
 */

/*
 * The HDLC polynomial: x**0 + x**5 + x**12 + x**16 (0x8408).
 */
#define P       0x8408

main()
```

```
{
    register unsigned int b, v;
    register int i;

    printf("typedef unsigned short u16;\n");
    printf("static u16 fcstab[256] = {");
    for (b = 0; ; ) {
        if (b % 8 == 0)
            printf("\n");

        v = b;
        for (i = 8; i--; )
            v = v & 1 ? (v >> 1) ^ P : v >> 1;

        printf("0x%04x", v & 0xFFFF);
        if (++b == 256)
            break;
        printf(",");
    }
    printf("\n};\n");
}
```

10.1.2.1.3 **FCS Usage**

Reference: Greg Morse, "Calculating CRCs by Bits and Bytes," BYTE (September 1986): pp. 115-124.

CRC generation:

1. All bits of a block are protected by the CRC.
2. The data is sent LSB first. The CRC is calculated on bits as they are sent.
3. The CRC is initialized to all ones. This allows detection of any missed or inserted zero bits at the beginning of a block. (Missed or inserted ones are still detected.)
4. The one's complement of the CRC is transmitted rather than the CRC itself. This allows detection of slippage-type errors.
5. The CRC is sent LSB first.
6. The polynomial used is $X^{16} + X^{12} + X^5 + 1$

To check incoming data block you have two options:

1. Calculate the CRC on all the protected bits and the CRC itself and then compare it to a known constant (0xF0B8).

OR

2. Calculate the CRC on all the protected bits only, omitting the CRC bits, and compare the calculated value to the received value.

```
/* This program shows how to use the CRC provided by PPP.
   The calculated CRC must bit inverted before being sent.
   This assumes the pppfcs() function, table, and constants
   are linked in elsewhere.
*/

#include <stdio.h>
#include “crc.h”

typedef short sint16;
typedef unsigned short uint16;
typedef unsigned char ubyte ;

#define FRM_SIZE  10
main()
{
    uint16 pppsendcrc, pppreccrc;
    ubyte frame[FRM_SIZE];
    int i;

    /* Fill the frame with sample data */
    for (i=0;i<FRM_SIZE-2;i++)
    {
        frame[i] = i;
    }

    pppsendcrc = pppfcs(PPPINITFCS,frame,FRM_SIZE-2);
    pppsendcrc = ~pppsendcrc;  /* Invert the crc */

    frame[FRM_SIZE-2] = pppsendcrc & 0xff; /* Put the least sig. value
in
                                              1st byte transmitted. */
    frame[FRM_SIZE-1] = (pppsendcrc>>8) & 0xff;

    /* At this point the frame would be sent. */

    /* Now let's simulate a receipt of the frame. */

    /* Run the CRC across the whole frame including the CRC */
    pppreccrc = pppfcs(PPPINITFCS, frame, FRM_SIZE);

    /* Compare against the constant */
    if (pppreccrc == PPPGOODFCS)
        printf("This equals PPPGOODFCS\n");
    else
        printf("This was not a valid CRC\n");
}
```

10.1.3 **ASYNC Transparency**

The contents of a frame is unrestricted, this can lead to problems, since an A, C, FCS or I-field byte that appears to be flag may occur. A Control Escape (CE) byte is defined as binary 01111101 (X'7D'). Prior to transmitting a frame a station examines each byte between the beginning and ending flags. For each byte it encounters with the same value as a flag or CE byte (X'C0', X'C1', or X'7D') it does the following.

1. Inserts a control escape (CE) byte preceding the byte
2. Complements bit 5 of the byte (i.e. exclusive OR's the byte with X'20')

The sending algorithm is shown below in a programmatic form.

```
Sending Algorithm
        Calculate FCS over Payload data
        For each byte in payload data and FCS
        {
                If byte is BOF, EOF, or CE
                {
                        Insert CE
                        Insert byte XOR X'20'
                }
        }
```

Prior to FCS computation, the receiving station also examines the entire frame contents between the flags. For each CE byte encountered it does the following:

1. Discards the CE byte
2. Complements bit 5 of the byte following CE

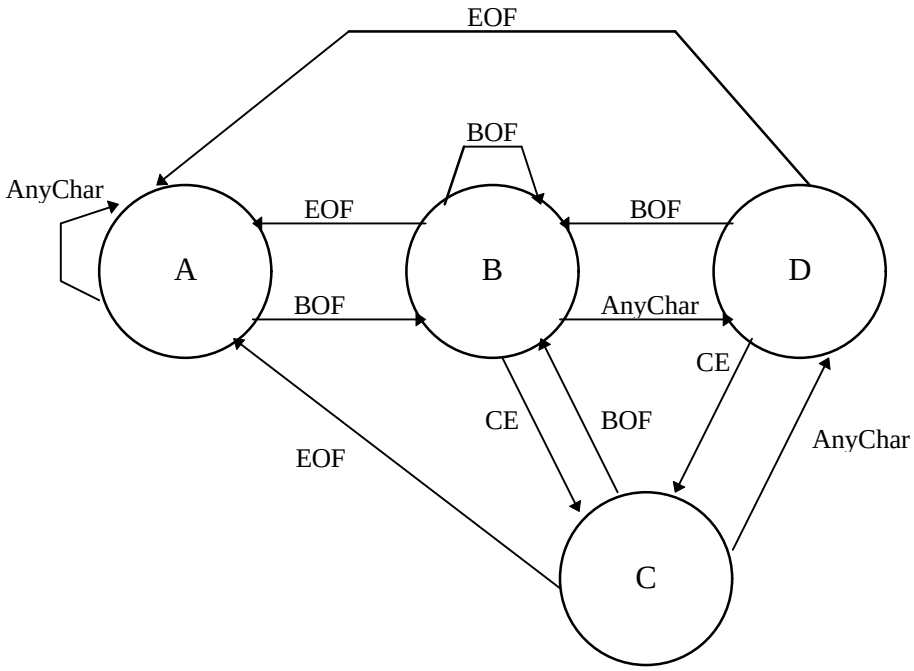
10.1.4 **Frame Abort**

The function of prematurely terminating a frame is call "abort". This function is reserved to the transmitter. The transmitting station aborts by sending a CE byte immediately followed by a flag sequence i.e. X'7DC1'. The abort pattern closes the frame without an FCS field or an ending flag.

Either a primary or a secondary station may abort. A secondary that sends an abort sequence must await permission from the primary before it may resume transmission, i.e. there is a implicit Final bit set in the sending of an abort sequence.

10.1.5 **ASYNC Transparency Receive Finite Automata**

Finite automata (State Machine for receiving frames)

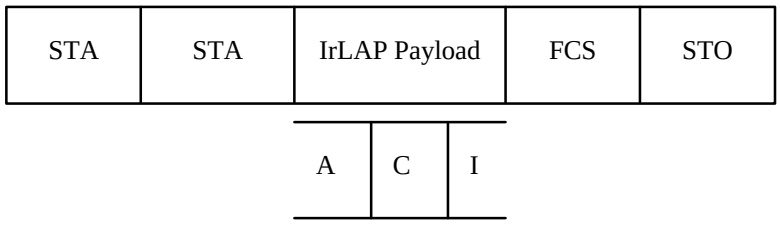


10.1.6 **Frame Transmission Order**

All bytes including connection addresses and control fields are transmitted low order bit first. Sequence numbers within control fields are also transmitted low order bit first (for example, the first bit of a sequence number that is transmitted will have the weight 2^0. The FCS will be transmitted least significant byte first. Every byte transmitted on the line in ASYNC mode consists of exactly one start bit, 8 data bits, and one stop bit.

10.2 SYNC Wrapper (576 kbps and 1.152 Mbps)

The SYNC Wrapper is composed of beginning flag, ending flag and field check sequence fields as shown below.



STA Begin Flag with value B'01111110'

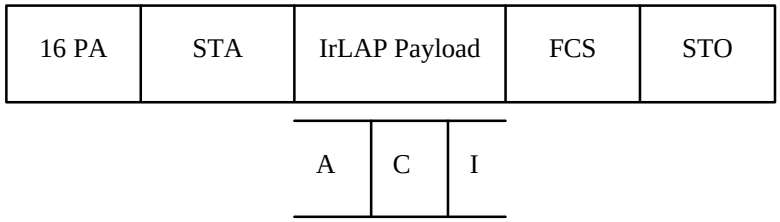
STO End Flag with value B'01111110'

FCS 16 bit Frame check sequence using CRC-CCITT performed on IrLAP Payload

Transparency is achieved with **bit stuffing**. When sending every time five 1's in a row is seen a 0 is inserted. When receiving every time five 1's in a row is seen the following 0 is removed.

10.3 4 PPM Wrapper (4Mbps)

The 4 PPM Wrapper is composed of a beginning sequence, ending flag and field check sequence fields as shown below.



PA Preamble of 4 chips equal to 1000 0000 1010 1000

STA Begin Flag 8 chips equal to 0000 1100 0000 1100 0110 0000 0110 0000

STO End Flag 8 chips equal to 0000 1100 0000 1100 0000 0110 0000 0110

FCS 32 bit Frame check sequence using IEEE CRC 32 performed on IrLAP Payload

Transparency is achieved via the 4 PPM encoding scheme.