FC-P/91-001R2.1
X3T9.3/90-019

FIBRE CHANNEL

PHYSICAL LAYER (FC-PH)

REV 2.1

working draft proposed
American National Standard
for Information Systems

May 25, 1991

Secretariat:

Computer & Business Equipment Manufacturers Association

ABSTRACT: This standard describes the point-to-point physical interface, transmission protocol, and signaling protocol of a high-performance serial link for support of the higher level protocols associated with HIPPI, IPI, SCSI and others.

NOTE:

This is an internal working document of X3T9.3, a Task Group of Accredited Standards Committee X3. As such, this is not a completed standard. The contents are actively being modified by the X3T9.3 Task Group. This document is made available for review and comment only. For current information on the status of this document contact the individuals shown below:

POINTS OF CONTACT:

Don Tolmie (X3T9.3 Chairman)
Los Alamos National Laboratory
C-5, MS-B255
Los Alamos, NM 87545
(505) 667-5502
E-Mail: det @ lanl.gov

I. Dal Allan (Fibre Channel Working Group Chairman)
ENDL
14426 Black Walnut Court
Saratoga, CA 95070
(408) 867-6630

Joe Mathis (Editor)
IBM Corporation, MS 4501
11400 Burnet Road
Austin, TX 78758
(512) 838-6247
E-Mail: MATHIS@AUSVM6.IINUS1.IBM.COM
Fax: (512) 838-7101

Contents

Tables

Figures

Foreword

The Fibre Channel provides a transport vehicle for the upper layer Intelligent Peripheral Interface (IPI) and Small Computer System Interface (SCSI) command sets, and the High-Performance Parallel Interface (HIPPI) data framing. The Fibre Channel is capable of replacing the SCSI, IPI and HIPPI physical interfaces with a protocol-efficient alternative that provides performance improvements in distance and/or speed.

IPI commands, SCSI commands, and HIPPI data framing operations may all be intermixed on the Fibre Channel. Proprietary and other command sets may also use and share the Fibre Channel, but such use is not defined.

The Fibre Channel is optimized for predictable transfers of large blocks of data such as used for file transfers between processors (super, mainframe, super-mini, etc.) storage systems (disk and tape), communications, and to output only devices such as laser printers and raster scan graphics terminals.

The Fibre Channel standard is organized in levels as follows:

- **FC-0** defines the physical portions of the Fibre Channel including the fibre, connectors, and optical parameters for a variety of industy supported data rates and physical media. A serial coax version may also be defined for limited distance applications.

FC-0 level provides the point-to-point physical portion of the Fibre Channel. Signaling rates from 132.813 MBaud, 265.625 Mbaud, 531.25MBaud, 1.0625Gbaud are specified. These signaling rates correspond to data transfer rates of 12.5 MBytes/s (100 Mbits/s), of 25 MBytes/s (200 Mbits/s), of 50 MBytes/s (400 Mbits/s), and of 100 MBytes/s (800 Mbits/s).

- **FC-1** defines the transmission protocol, which includes the serial encoding, decoding, and the error control.

- **FC-2** defines the signaling protocol which includes the frame structure and byte sequences.

- **FC-3** defines the common service interface to FC-4

- **FC-4** is the highest layer in the Fibre Channel standards set. It defines the channel protocol, or mapping, between the lower layer FC standards and the IPI and SCSI command sets, and the HIPPI data framing.

- **FC-F** describes the requirements placed on Fabrics which intend to support the Fibre Channel.

The Fibre Channel protocol is simple in order to minimize implementation cost and enhance throughput. The transmission medium is isolated from the control protocol so that implementation of point-to-point links, multi-drop bus, rings, crosspoint switches, or other special requirements may be made in a technology best suited to the environment of use.

American National Standard for Information Systems

Fibre Channel - Physical (FC-PH)

1 Scope

This standard provides the physical layer of a high performance serial link for support of the higher layer protocols associated with HIPPI, IPI, SCSI and others.

2 Normative references

The following standards contain provisions which, through reference in FC-PH, constitute provisions of this standard. At the time of publication, the editions indicated were valid. All standards are subject to revision, and parties to agreements based on this standard are encouraged to investigate the possibility of applying the most recent editions of the standards listed below. Members of IEC and ISO maintain registers of currently valid International Standards, and ANSI for American National Standards.

Fiber Distributed Data Interface Media Access Control - X3T9/84-100 X3T9.5/83-16 REV-10

All FOTPs are EIA/TIA-455-XXX and all OFSTPs are EIA/TIA-526-XXX.*

FOTP-30 - Frequency Domain Measurement of Multimode Optical Fiber Information Transmission Capacity

FOTP-45 - Microscopic Method for Measuring Fiber Geometry of Optical Waveguide Fibers

FOTP-48 - Measurement of Optical Fiber Cladding Diameter Using Laser-Based Instruments

FOTP-51 - Pulse Distortion Measurement of Multimode Glass Optical Fiber Information transmission Capacity

FOTP-54 - Mode Scrambler Requirements for Overfilled Launching Conditions to Multimode Fibers

FOTP-107 - Return Loss for Fiber Optic Components

FOTP-127 - Spectral Characteristics of Multimode Lasers

FOTP-168 - Chromatic Dispersion Measurement of Multimode Graded-Index and Single-Mode Optical Fibers by Spectral Group Delay Measurement in the time Domain.

FOTP-171 - Attenuation by Substitution Measurement for short length Multimode Graded-Index and Singlemode Optical Fiber Cable Assemblies.

---

* Fiber Optic Test Procedure (FOTP) and Fiber Optics Systems Test Procedure (OFSTP) standards are developed and published by the Electronics Industries Association under the EIA-RS-455 and the EIA-RS-526 series of standards. Copies may be obtained by writing to:

DIRECTOR OF TECHNICAL PROGRAMS
Information & Telecommunication Technologies
ELECTRONIC INDUSTRIES ASSOCIATION
2001 Eye Street, N.W.
Washington, D.C. 20006

FOTP-176 - Measurement Method of Optical Fiber Geometry by Automated Grey-Scale Analysis

FOTP-177 - Numerical Aperture Measurement of Graded-Index Optical Fibers

EIA/TIA-492AAAA - Detailed Specification for 62.5 μm Core Diameter/125 μm Cladding Diameter Class 1a Multimode, Graded Index Optical Waveguide Fibers.

EIA/TIA-492BAAA - Detail Specification for Class IVa Dispersion-Unshifted Single-Mode Optical Waveguide Fibers Used in Communications Systems

OFSTP-2 - Effective Transmitter Output Power Coupled Into Singlemode Fiber Optic Cable

OFSTP-3 - Fiber Optic Terminal Equipment Receiver Sensitivity and Maximum Receiver Input

OFSTP-14 - Optical Power Loss Measurement of Installed Multimode Fiber Cable Plant

3 Definitions and conventions

3.1 Definitions

For the purposes of this standard, the following definitions and figure 10 through figure 14 apply.

3.1.1 alias: an optional, alternate address identifier recognized by an N_Port in addition to its native address identifier.

3.1.2 attenuation: the Optical power loss expressed in units of dB.

3.1.3 average power: the optical power measured using an average reading power meter when the Fibre Channel is transmitting continuous valid symbols.

3.1.4 baud: the coded bit rate per second.

3.1.5 beginning running disparity: the Running Disparity present at a Transmitter when Encoding of the Special Code associated with an Ordered Set is initiated, or at a Receiver when Decoding of the Special Character associated with an Ordered Set is initiated.

3.1.6 broadcast: TBD

3.1.7 buffer: a logical construct which holds the Data Field contents of a single frame i.e. Data Field size ≤ a buffer.

3.1.8 byte: an eight-bit entity with its least significant bit denoted as bit 0 and most significant bit as bit 7.

3.1.9 cable plant: the cable plant consists of all the optical elements, for example fiber, connectors, splices, etc. between a transmitter and a receiver.

3.1.10 center wavelength (laser): the nominal value central operating wavelength. It is the wavelength, defined by a peak mode measurement (See FOTP-127) where the effective optical power resides.

3.1.11 center wavelength (LED): the average of the two wavelengths measured at the half amplitude points of the power spectrum.

3.1.12 character: any transmission character associated by the FC-1 transmission code with a FC-2 data byte or special code. Transmission characters are generated and interpreted only by FC-1.

3.1.13 circuit: a bidirectional path within the Fabric for the flow of electric signals.

3.1.14 code bit: the smallest time period used by the FC-0 for transmission on the media.

3.1.15 code violation: an error condition that occurs when a received Transmission Character cannot be decoded to a Valid Data Byte or Special Code using the validity checking rules specified by the Transmission Code.

3.1.16 comma: the seven bit sequence 0011111 or 1100000 when appearing in an encoded stream.

3.1.17 comma character: a Special Character containing a comma.

3.1.18 concatenation: a logical operation that "joins together" strings of data and is represented with a symbol "||". In FC-2, two or more fields are concatenated to provide a reference of uniqueness. (e.g., S_ID||X_ID)

3.1.19 Connection: see Dedicated Connection.

3.1.20 Connectionless service: communication between two N_Ports performed without a Dedicated Connection.

3.1.21 Connection Initiator: the source N_Port which initiates a Class 1 Connection with a destination N_Port through a Connection request and also receives a valid response from the destination N_Port to complete the Connection establishment.

3.1.22 Connection Recipient: the destination N_Port which receives a Class 1 Connection request from the Connection Initiator and accepts establishment of the Connection by transmitting a valid response.

3.1.23 Credit: the maximum number of receive buffers allocated to a transmitting N_Port or F_Port. It represents the maximum number of outstanding frames which can be transmitted by that N_Port without causing buffer overrun at the receiver.

3.1.24 current running disparity: the Running Disparity present at a Transmitter when Encoding of a Valid Data Byte or Special Code is initiated, or at a Receiver when Decoding of a Transmission Character is initiated.

3.1.25 data byte: a string of eight bits passed to and from the FC-2 layer as a unit. (See data character and transmission character.)

3.1.26 data character: any Transmission Character associated by the Transmission Code with a Valid Data Byte.

3.1.27 decoding: validity checking of received Transmission Characters and generation of Valid Data Bytes and Special Codes from those characters.

3.1.28 Dedicated Connection: a communicating route/circuit guaranteed and retained by the Fabric for two given N_Ports.

3.1.29 destination N_Port: the N_Port identified by the Destination_Identifier (D_ID) to which the frame is transmitted.

3.1.30 disconnection: the process of removing a Dedicated Connection between two N_Ports.

3.1.31 disparity: the difference between the number of ones and zeros in a Transmission Character.

3.1.32 dispersion TBD.

3.1.33 EIA: Electronic Industry Association

3.1.34 encoding: generation of valid Transmission Characters from Valid Data Bytes and Special Codes.

3.1.35 Exchange: the basic mechanism to transfer information consisting of one or more related non-concurrent Sequences which may flow in the same or opposite directions. An Exchange may span a set of related N_Ports with a common control structure. An Exchange may span across Class 1 disconnects. The Exchange is uniquely identified by an Originator Exchange_Identifier (OX_ID) and a Responder Exchange_Identifier (RX_ID).

| **3.1.36 Exchange Identifier (X_ID):** a generic reference to OX_ID and RX_ID.

3.1.37 Exchange_Status_Block: a logical construct which contains the state of an Exchange. An Originator N_Port has an Originator Exchange Status Block and the Responder N_Port has a Responder Exchange Status Block for each concurrently active Exchange.

3.1.38 extinction ratio: the ratio of the low, or off optical power level, (P_L)to the high, or on optical power level, (P_H) when the station is transmitting a stream of symbols. Extinction Ratio (%) = $(P_L/P_H)*100$

| **3.1.39 eye opening:** the portion of the bit time which is error free to a given bit error rate (BER $<10^{-12}$ for Fiber Channel).

3.1.40 F_Port: the link_Control_Facility within the Fabric which attaches to an N_Port through a link. An F_Port is not FC-2 addressable.

3.1.41 Fabric: an entity which can interconnect any two N_Ports attached to it and can be completely navigated by the D_ID contained in the FC-2 frame header.

3.1.42 Fabric Entry/Exit Point: the F_Port which directly attaches to an N_Port through a link.

3.1.43 fibre: the transmission media specified in FC-0, including glass, plastic, and copper.

3.1.44 fibre optic cable A jacketed optical fibre(s).

3.1.45 fibre optic test procedure: (FOTP) - Standards developed and published by the Electronic Industries Association (EIA) under the EIA-RS-455 series of standards.

3.1.46 frame: an indivisible unit of information demarcated by unique character strings called delimiters.

3.1.47 Frame Content: the information contained in a frame between its SOF and EOF delimiters, excluding the delimiters.

3.1.48 Hunt Group: a set of N_Ports with a common alias address identifier managed by a single common controlling entity.

3.1.49 Idle Sequence: a stream of Idle Words transmitted during periods of N_Port inactivity and as a spacing mechanism between frames.

3.1.50 Idle Word (Idle): an ordered set of four transmission characters as defined in FC-1 denoting a word boundary. The Idle Word is also referred to as an Idle.

3.1.51 ignored: a field that is interpreted by the receiver.

3.1.52 information transfer: information transferred which may be data, commands, or responses.

3.1.53 initialization: the period beginning with power on of an FC-1 level and continuing until the Transmitter and Receiver of that level become Operational.

3.1.54 Intermix: a service which interleaves Class 2 and Class 3 frames on an established Class 1 Connection.

3.1.55 InBand address: the address of the destination N_Port embedded within the transmitted frame itself, as part of the frame structure.

3.1.56 jitter: deviations from the ideal timing of an event which occur at high frequencies. For deviations at low frequencies and additional information see wander. Jitter is customarily subdivided into deterministic and random components.

3.1.57 jitter, deterministic (DJ): timing distortions caused by normal circuit effects in the transmission system. Deterministic jitter is often subdivided into duty cycle distortion (DCD) caused by propagation differences between the two transitions of a signal and data dependant jitter (DDJ) caused by the interaction of the limited bandwidth of the transmission system components and the symbol sequence.

3.1.58 jitter, random (RJ): jitter due to thermal noise which may be modeled as a Gaussian process. The peak-to-peak value of RJ is of a probabilistic nature and thus any specific value requires an associated probability.

3.1.59 level: a document artifice used to group related architectural functions. No specific implementation is intended between levels and actual implementations.

3.1.60 link: two unidirectional fibres transmitting in opposite directions.

3.1.61 Link_Control_Facility: a link hardware facility which attaches to the end of a link and manages transmission and reception of data. It is contained within each N_Port and F_Port and includes the link transmitter and receiver mechanism.

3.1.62 loopback: a mode of FC-1 operation in which information passed to the FC-1 Transmitter for transmission is shunted directly to the FC-1 Receiver, overriding any signal detected by the Receiver on its attached Fibre.

3.1.63 mandatory: a function which is supported by all compliant implementations.

3.1.64 mean launched power: the average power for a continuous valid symbol sequence coupled into a fibre at point S as shown in 8.

3.1.65 media interface connector: a fibre connector which connects the fibre media to the Fibre Channel attachment. The MIC consists of two halves. The MIC plug is the male half used to terminate an optical fibre signal transmission cable. The MIC receptacle is the female half which is associated with the Fibre Channel station.

3.1.66 MIC receptacle: the fixed or stationary female half of the MIC which is part of the transmitter and/or receiver in a Fibre Channel station.

3.1.67 MIC plug: the male half of the MIC which terminates an optical signal transmission cable.

3.1.68 multicast: TBD

3.1.69 Native address identifier: the address identifier required by each N_Port which is unique within the address domain of the Fabric.

3.1.70 Node: a collection of one or more N_Ports controlled by a level above FC-2.

3.1.71 N_Port: a hardware entity at the Node end of the link which includes a link_Control_Facility. An N_Port is assigned a system unique identifier called N_Port Identifier and is addressed by this Identifier. It may act as an Originator, a Responder, or both.

3.1.72 non-repeating ordered set: an Ordered Set which, when issued by FC-2 to FC-1 for transmission, is to be transmitted once.

3.1.73 Not Operational: a Receiver or Transmitter that is not capable of receiving or transmitting an encoded bit stream, respectively, based on the rules defined by this standard for error control. The FC-1 level is Not Operational during Initialization. A Receiver or Transmitter that becomes Not Operational after Initialization is complete requires external intervention (e.g., reset, repair) before it can become Operational. (The FC-1 level is Not Operational during Initialization.)

3.1.74 numerical aperture: the sine of the radiation or acceptance half angle of an optical fibre, multiplied by the refractive index of the material in contact with the exit or entrance face. See FOTP-177.

3.1.75 open fibre control (OFC): a safety interlock system that controls the optical power level on an open fibre cable.

3.1.76 operation: a group of one or more related Exchanges. An operation is a higher level construct bounded by an Operation_Identifier (O_ID).

3.1.77 operational: a Receiver or Transmitter that is capable of receiving or transmitting an encoded bit stream, respectively, based on the rules defined by this standard for error control. Those Receivers capable of accepting signals from Transmitters requiring laser safety procedures are not considered Operational after power on until a signal of a duration longer than that associated with laser safety procedures is present at the Fibre attached to the Receiver.

3.1.78 Operation Initiator: The source N_Port (or Node) which initiates an Operation with a destination N_Port (or Node) and assigns an Operation Identifier through the first Exchange of the Operation.

3.1.79 Operation Recipient: the destination N_Port (or Node) which receives the Operation initiation from the Operation Initiator and participates in the Operation so initiated.

3.1.80 optical fall time: the time interval for the falling edge of an optical pulse to transition from its 90% amplitude level to its 10% of amplitude level.

3.1.81 optical fibre: dielectric material that guides light; optical waveguide.

3.1.82 optical path power penalty: the additional loss budget required to account for degradations due to reflections and the combined effects of dispersion resulting from intersymbol interference, mode-partition noise, and laser chirp.

3.1.83 optical return loss (ORL): the ratio (expressed in units of dB) of optical power reflected by a component or an assembly to the optical power incident on a component port when that component or assembly is introduced into a link or system. (See FOTP-107).

3.1.84 optical rise time: the time interval for the rising edge of an optical pulse to transition from its 10% of amplitude levels to its 90% of amplitude level as specified.

3.1.85 optical reference plane: the plane that defines the optical boundary between the MIC plug and the MIC receptacle.

3.1.86 optional: features that are not required by the standard. However, if any optional feature defined by the standard is implemented, it will be implemented according to the standard.

3.1.87 ordered set: a Transmission Word containing a K28.5 Special Character in its first (leftmost) position and designated to have special meaning (e.g., frame delimiters and Idle Words).

3.1.88 Originator: the logical function in an N_Port responsible for initiating an Exchange. An Originator may be a master or slave (IPI), initiating controller or target controller (SCSI), or source (HIPPI).

3.1.89 Payload: contents of Data_Field of a frame, excluding Optional Header(s), if present. (see 34)

3.1.90 physical link: the full-duplex physical layer association between adjacent FC-1 entitles (in nodes or paths) in a Fibre Channel Path, i.e., a pair of Physical links.

3.1.91 PMD: physical media dependent as defined by ANSI X3T9.5 document.

3.1.92 power on: a Power on state has occurred when the power supply voltages are within their normal operating tolerances. At this point any circuits or optical devices may be assumed to be in a fully operational state.

3.1.93 receiver: the portion of a link Control Facility dedicated to receiving an encoded bit stream from a Fibre, converting this bit stream into Transmission Characters, and Decoding these characters using the rules specified by this standard.

3.1.94 receiver (RX): an optoelectronic circuit that converts an optical signal to an electrical retimed serial logic signal.

3.1.95 receiver sensitivity: defined as the minimum acceptable value of average received power at point R to achieve a 10^{-12} BER. It takes into account power penalties caused by use of a transmitter with worst-case values of extinction ratio, jitter, pulse rise and fall times, optical return loss at point S, receiver connector degradations and measurement tolerances. The receiver sensitivity does not include power penalties associated with dispersion, jitter, or reflections from the optical path; these effects are specified separately in the allocation of maximum optical path penalty. Sensitivity takes into account worst-case operating and end-of-life (EOL) conditions.

3.1.96 receiver overload : the maximum acceptable value of the received average power at point R for a 10^{-12} BER.

3.1.97 reflections: caused by refractive index discontinuities along the optical path.

3.1.98 repeating ordered set: an Ordered Set which, when issued by FC-2 to FC-1 for transmission, is to be repetitively transmitted until a subsequent transmission request is issued by FC-2.

3.1.99 return loss: see optical return loss.

3.1.100 relative intensity noise (RIN): laser noise where the noise is measured relative to the average optical power. Units are typically either 1/Hz or dB/Hz.

3.1.101 reflection induced intensity noise (RIIN): a form of RIN caused by optical feedback into a source.

3.1.102 reserved: a field which is filled with binary zero(s) by the Source and is not interpreted by the Destination. Each bit in the reserved field is denoted by "r".

3.1.103 Responder: the logical function in an N_Port responsible for supporting the Exchange initiated by the Originator in another N_Port. A Responder may be a master or slave (IPI), initiating controller or target controller (SCSI), or the destination (HIPPI).

3.1.104 running disparity: a binary parameter indicating the cumulative Disparity (positive or negative) of all previously issued Transmission Characters. Running Disparity is one of the inputs used to determine the proper Transmission Character to generate during Encoding and to check Transmission Character validity during Decoding.

3.1.105 Sequence: a set of one or more related Data frame(s), with a common Sequence_ID (SEQ_ID), transmitted unidirectionally from one N_Port to another N_Port with corresponding link_Control frame, if any, transmitted in response to each Data frame.

3.1.106 Sequence Initiator: the source N_Port which initiates the Sequence and transmits Data frame(s) to the destination N_Port.

3.1.107 Sequence Recipient: the destination N_Port which receives Data_Frames from the Sequence Initiator and, if applicable, transmits (link_Control) responses to the Sequence Initiator, unless suppressed.

3.1.108 Sequence Status Block: a logical construct and preferably a hardware facility which tracks the state of a Sequence. Both the Sequence Initiator and the Sequence Recipient have a Sequence Status Block for each concurrently active Sequence.

3.1.109 Source: the N_Port, identified by the Source_Identifier (S_ID), from which the frame is transmitted.

3.1.110 special character: any Transmission Character considered valid by the Transmission Code but not equated to a Valid Data Byte. Special Characters are provided by the Transmission Code for use in denoting special functions.

3.1.111 special code: a code which, when encoded using the rules specified by the Transmission Code, results in a Special Character. Special Codes are typically associated with control signals related to protocol management e.g., K28.5.

3.1.112 spectral width full width half maximum (FWHM): the absolute difference between the wavelengths at which the spectral radiant intensity is 50 percent of the maximum power.

3.1.113 spectral width lasers (RMS): the weighted root mean square (RMS) width of the Active Output Interface optical spectrum. (See FOTP-127.)

3.1.114 synchronization: receiver identification of a Transmission-Word boundary established by a Transmitter on a Fibre.

3.1.115 Transmission Character: any encoded character (valid or invalid) transmitted across a physical interface specified by FC-0. Valid Transmission Characters include Data and Special Characters and are specified by the Transmission Code.

3.1.116 Transmission Code: a means of Encoding data to enhance its Transmission Characteristics. The Transmission Code specified by this standard is byte-oriented, with (1) Valid Data Bytes and (2) Special Codes encoded by this Transmission Code converted to 10-bit Transmission Characters.

3.1.117 Transmission Word: a string of four Transmission Characters treated as a unit.

3.1.118 transmitter: the portion of a link Control Facility dedicated to converting Valid Data Bytes and Special Codes into Transmission Characters using the rules specified by this standard, converting these Transmission Characters into a bit stream, and transmitting this encoded bit stream onto a Fibre. In FC-0 **transmitter (Tx)** means an optoelectronic circuit that converts an electrical logic signal to an optical signal.

3.1.119 transceiver: A transmitter and receiver combined in one package.

3.1.120 unrecognized ordered set: a Transmission Word containing a K28.5 Special Character in its first (leftmost) position but not defined to have special meaning by the standard.

3.1.121 Upper Level Protocol (ULP): a higher level protocol user of FC-2.

3.1.122 valid data byte: a string of eight bits passed to and from the FC-2 level as a unit. Typically a Valid Data Byte contains either binary information or a character encoded according to the character representation of the using node.

3.1.123 Valid frame: a frame received with a valid SOF, a valid EOF, valid data characters, and proper CRC verification of the Frame Header and Data field.

3.1.124 vendor unique: those features that can be defined by a vendor in a specific implementation. Caution should be exercised in defining and using such features since they may or may not be standard between vendors.

3.1.125 wander: deviations from the ideal timing of an event which occur at low frequencies. Deviations which occur at high frequencies are termed jitter. The frequency division between jitter and wander is usually made based on the capabilities of the clock recovery. Wander is assumed to be tracked by the clock recovery and does not directly affect the timing allocations within a bit cell. Jitter is not tracked by the clock recovery and directly affects the timing allocations in a bit cell. For Fiber Channel the frequency division between jitter and wander corresponds to the bit rate divided by 2 500.

3.1.126 word: a sequence of four contiguous bytes. (See Transmission word.)

3.1.127 World_Wide_Name: a unique world wide address up to 60 bits wide, administered by a Network_Address_Authority such as CCITT or IEEE, preceded with 4 bit wide Network_Address_Authority identifier. (See 19.4 and 23.5.10)

3.2 Editorial conventions

In this standard, a number of conditions, mechanisms, sequences, parameters, events, English text, states, or similar terms are printed with the first letter of each word in uppercase and the rest lowercase (e.g., Exchange, Class). Any lowercase uses of these words have the normal technical English meanings.

Numbered items in this standard do not represent any priority. If prioritized, it will be specifically so indicated.

The American convention of numbering is used i.e., the thousands and higher multiples are separated by a comma and a period is used as the decimal point. This is equivalent to the ISO convention of a space and a comma.

American	ISO
0.6	0,6
1,000	1 000
1,323,462.9	1 323 462,9

In case of any conflict between figure, table, and text, the text takes precedence.

In all of the figures, tables, and text of this document, the most significant bit of a binary quantity is shown on the left side. Exception to this convention is indicated at the appropriate section.

4 Structure and concepts

The Fibre Channel (FC) is logically a point-to-point serial data channel, structured for high performance capability. Physically, the Fibre Channel can be an interconnection of multiple communication points called N_Ports, interconnected through a switching network, called a Fabric, or can be simply point-to-point. Fibre is a general term used to cover all physical media types supported by the FC Standard - such as glass, plastic and copper.

Fiber Channel is structured as a set of hierarchical functions as illustrated in figure 1. The Physical interface (FC-0) consists of transmission media, driver and receivers and their interfaces. The Physical interface specifies a variety of media, and associated drivers and receivers capable of operating at various speeds. The transmission code used is 8B/10B and specified in FC-1. The signaling protocol (FC-2) is performed through frames and specifies the rules and provides mechanisms needed to transfer block(s) of data from end-to-end. Device protocols constitute FC-4 which is the highest layer in the Fibre Channel structure. FC-3 provides services common to multiple device protocols (FC-4s) such as Striping and Multicast.

Fibre Channel Physical layer (FC-PH) consists of related functions FC-0, FC-1, and FC-2. Each of these functions is described as a level. The standard does not restrict implementations to specific interfaces between these levels.

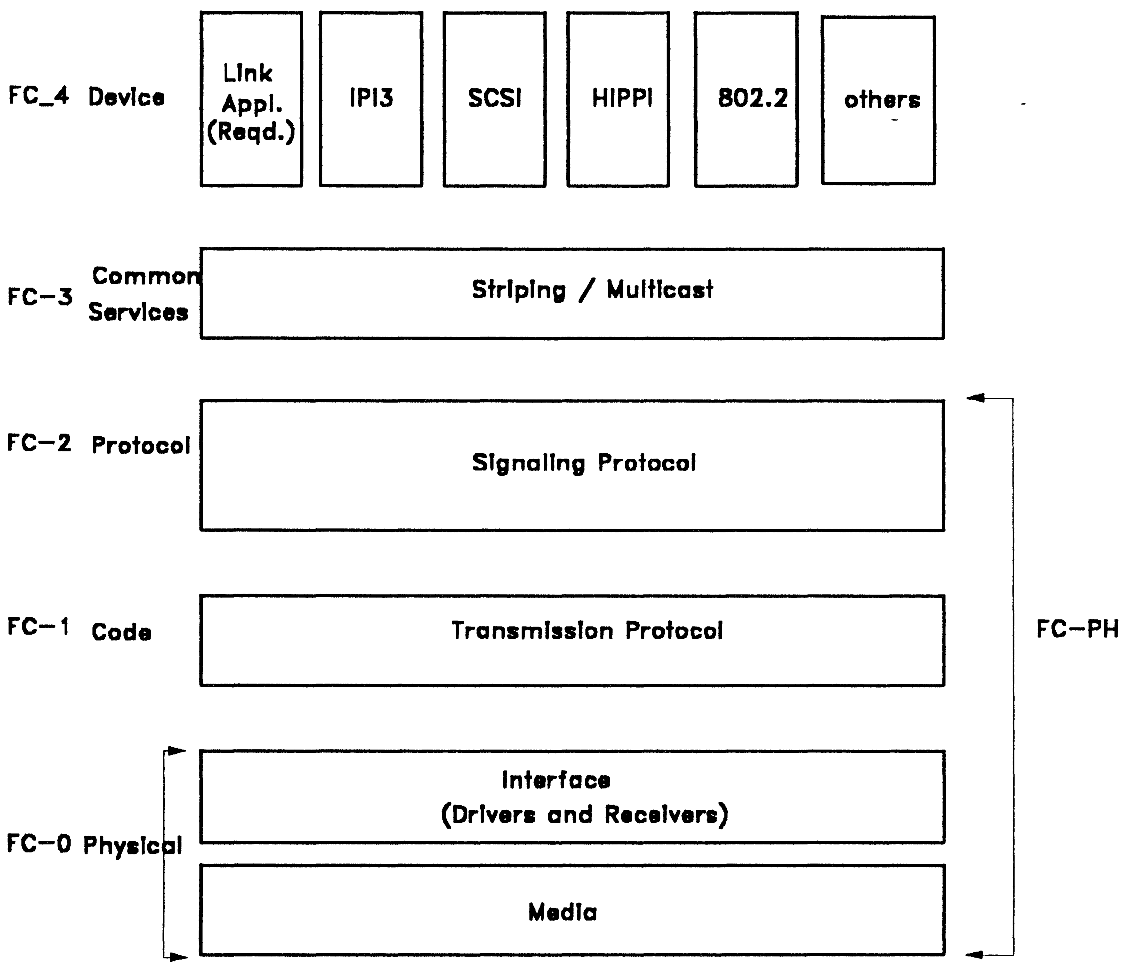


Figure 1. Fiber Channel structure

Fibre Channel provides a method for supporting any number of Upper Level Protocols (FC-4s). The Link Application is required, whereas other FC-4s are optional.

A Fibre Channel Node is functionally configured as illustrated in figure 2. A Node may support one or more N_Ports and one or more FC-4s. Each N_Port contains FC-0, FC-1 and FC-2 functions. FC-3 optionally provides the common services to multiple N_Ports and FC-4s.

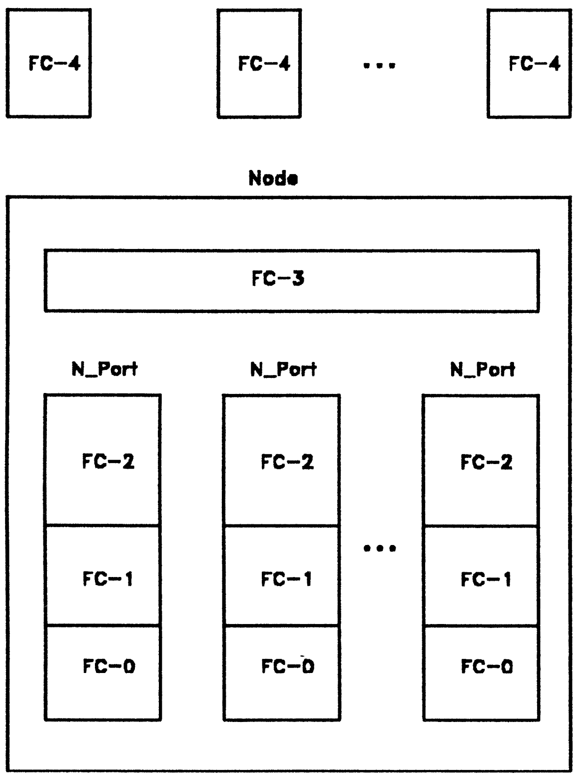


Figure 2. Fibre Channel functional configuration

4.1 FC-0 general description

The FC-0 level of the standard describes the physical link of the Fibre Channel. The FC-0 covers a variety of media and the associated drivers and receivers capable of operating at wide range of speeds. The FC-0 is designed for maximum flexibility and allows the use of a large number technologies to meet the widest range of system requirements.

Each fibre is attached to a transmitter of a port at one end and a receiver of another port at the other end (see figure 3). When a fabric is present in the configuration, a fibre may attach to an N_Port and an F_Port (see figure 4). Patch panels or portions of the active fabric

An end-to-end communicating route may be made up of physical links of different technologies. For example, it may have multimode fibre links attached to end ports but may have a single mode link in between as illustrated in figure Figure 5. In figure 6, a typical fibre channel building wire configuration is shown.

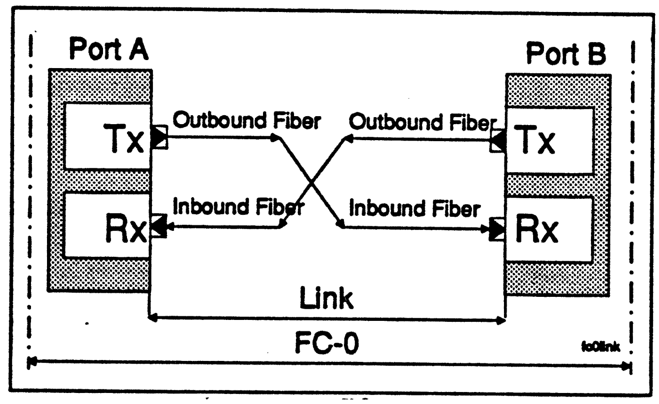


Figure 3. FC-0 Link

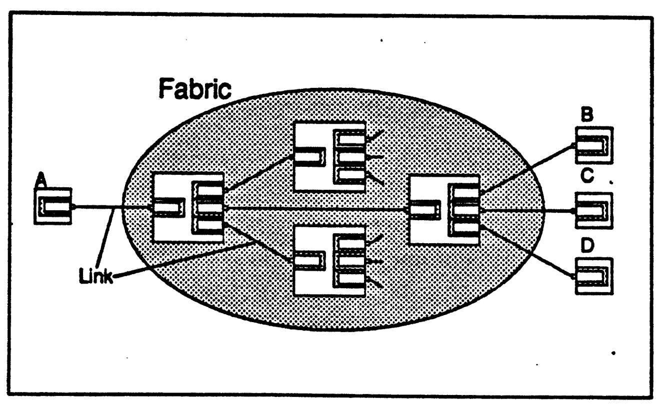


Figure 4. Fabric

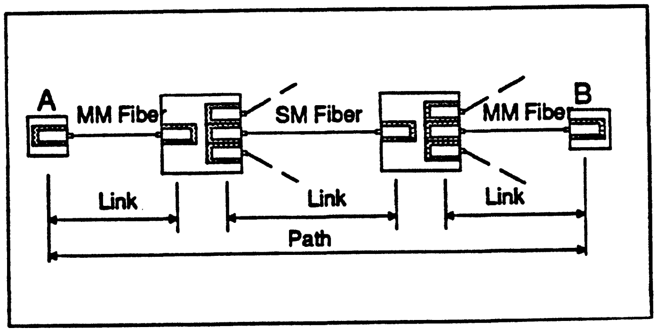


Figure 5. FC0 path

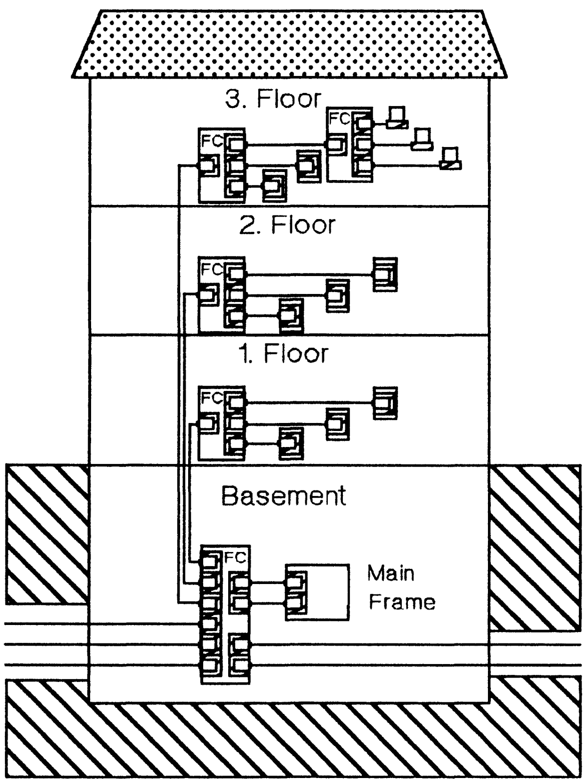


Figure 6. Fibre Channel building wiring

4.2 FC-0 interface overview

Figure 7 shows the cable plant topology.

Figure 8 and figure 9 show possible implementations of the transmitter and receiver, respectively.

Interfaces "a" through "e" are for reference only and are implementation dependent. Recommended interfaces are included in Appendix D.

Figure 18 shows the nomenclature used by FC-0 to reference various combinations of components.

FC-0 operates with a BER of less than 10^{-12}.

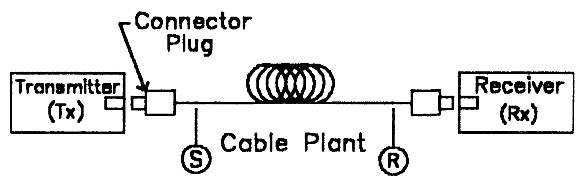


Figure 7. FC-0 interface

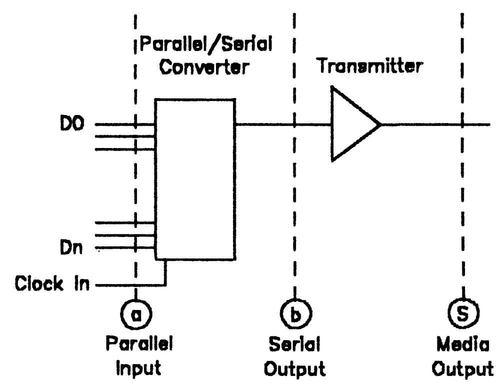


Figure 8. FC-0 transmitter interfaces

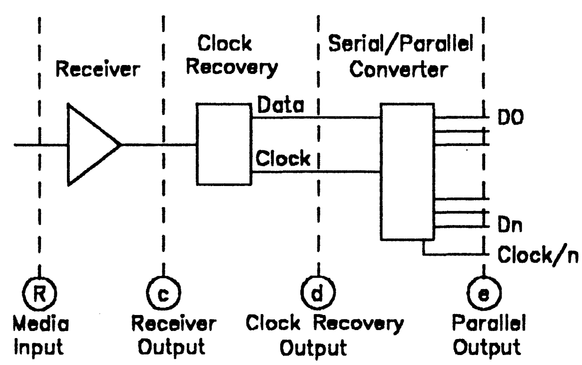


Figure 9. FC-0 receiver interfaces

4.3 FC-1 general description

The Fibre Channel ensures that the transmission characteristics of information to be transmitted over a Fibre are sufficiently robust through the use of a Transmission Code. This Transmission Code accepts unencoded data and converts it to a form which ensures that sufficient transitions are present on the encoded bit stream to allow clock recovery at the Receiver. Certain encodings specified by the Transmission Code have special characteristics which allow a Receiver to easily determine word alignment on the incoming bit stream.

Transmitter and Receiver behavior is specified via a set of states and their interrelationships. These states are divided into Operational and Not Operational classes. Error monitoring capabilities and special operational modes are also defined for Operational Receivers and Transmitters.

4.4 FC-2 general description

The Fibre Channel (FC) is logically a point-to-point serial data channel, structured for high performance capability. Physically, the Fibre Channel can be an interconnection of multiple communication points called N_Ports, interconnected through a switching network, called a Fabric, or can be simply point-to-point. Fibre is a general term used to cover all physical media types supported by the FC Standard - such as glass, plastic and copper.

The FC-2 level serves as the transport mechanism of the Fibre Channel. The transported data is transparent to FC-2 and visible to FC-3 and above.

The following concepts are defined and described:

- The physical components of the channel and their relationship as a Physical model.
- Modes of use of the channel in terms of half-duplex, duplex and dual-simplex communication models.
- Topologies based on the presence or absence of a Fabric.
- Classes of service provided by the Fabric and the N_Ports.
- General Fabric model
- Building Blocks and their hierarchy
- Sequence and Sequence Identifier
- Exchange and Exchange Identifier
- Exchange Status Block model
- Segmentation and Reassembly

A Request-Reply model is used in several clauses for illustration but other models are equally valid.

4.5 FC-2 Physical model

Figure 10 depicts the FC-2 Physical model and illustrates the FC-2 physical structure and components. The Fibre Channel (FC) physically consists a minimum of two Nodes, each with a minimum of one N_Port interconnected by a pair of Fibres - one outbound and the other inbound at each N_Port. This pair of unidirectional Fibres transmitting in opposite directions is referred to in this document as a Link. The Link is used by the interconnected N_Ports to perform data communication.

Physical equipment such as a processor, controller, or terminal can be interconnected to other physical equipment through these Links. Attached physical equipment supports one or more Nodes and each Node contains one or more N_Ports, with each N_Port containing a Transmitter and a Receiver.

The Physical model shown in figure 10 is inherently capable of simultaneous, symmetrical bidirectional flow. A Fabric may be present between the N_Ports and some Fabrics may not support this type of flow. From the perspective of a given N_Port, for instance A(1) or B(1), its Transmitter sends Data frames on the Outbound Fibre and its Receiver receives the responses on the Inbound Fibre. Data frames are throttled by flow control.

An N_Port logically performs only a point-to-point communication with another N_Port at any given time. This statement is true regardless of the presence of Fabric. Nevertheless, multiple N_Ports in a Node can simultaneously perform data transfers in parallel with single or multiple N_Ports contained in one or more Nodes. This structure provides the attached equipment flexible mechanisms to perform simultaneous data transfers in parallel to, or from single, or multiple equipment(s).

Each N_Port has a unique identifier used to identify the N_Port during communication.

4.5.1 Node and N_Port identifiers

Referring to figure 10, Node A consists of two N_Ports represented as A(1) and A(2) and Node B of two N_Ports represented as B(1) and B(2). Thus, single or multiple N_Ports contained within an Node are represented by a set of N_Port Identifiers. Each N_Port is represented as an element of the set, for example,

Node D as {D(1), D(2), ..., D(n)}.

4.5.2 Link_Control_Facility (LCF)

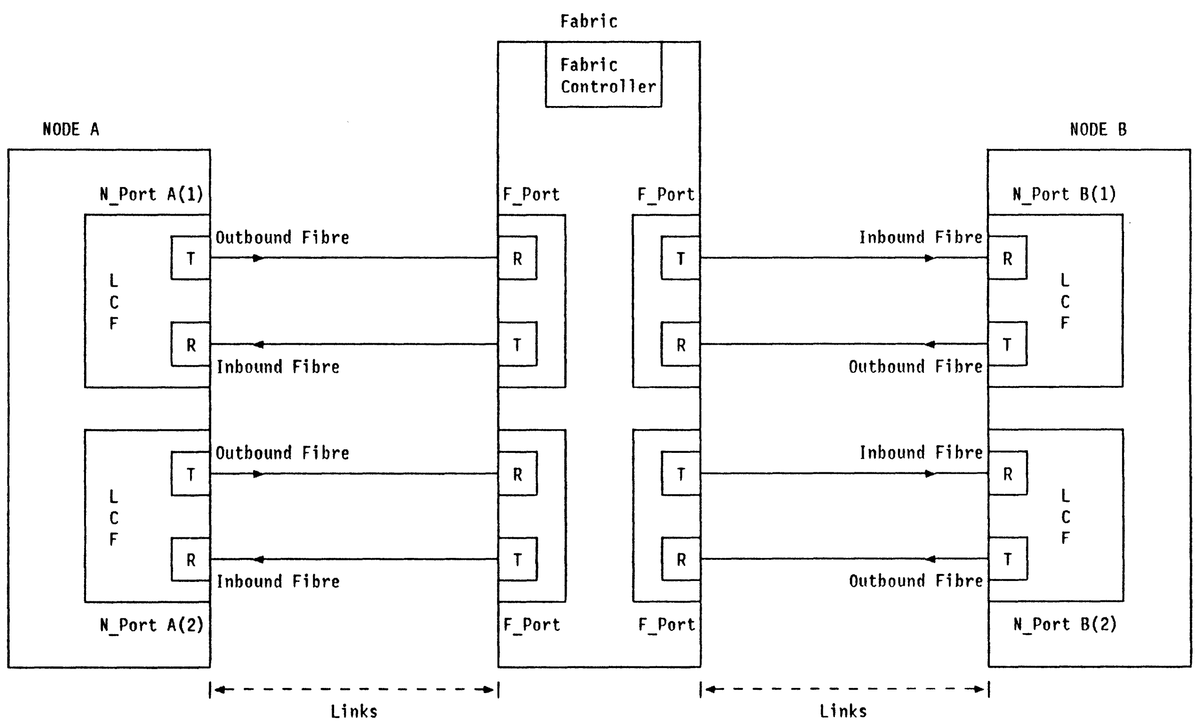
Each LCF shall perform the logical and physical control of the Link for each mode of use and provides the logical interface to the Originator and/or the Responder, as applicable to a communication model.

4.6 Communication models

An N_Port transmits Data frames at the request of ULP(s) within its Node, and transmits Link_Control responses to Data frames that it receives from other N_Port(s). An N_Port receives Link_Control responses to Data frames that it transmitted, and it receives Data frames from other N_Port(s).

An N_Port can operate in half-duplex, duplex, or dual-simplex modes as described below.

- Half-duplex operation is defined as an N_Port transferring Data frames in one direction only, with Link_Control frames flowing in the opposite direction.
- Duplex operation is defined as an N_Port simultaneously transmitting and receiving Data frames, with Link_Control frames flowing in both directions as well.
- Dual-simplex operation is defined as an N_Port both transmitting and receiving data, but not simultaneously. Data frames and Link_Control frames flow in both directions, but the flow is limited, possibly by a Fabric to a single direction at a time.



Legend:
T = Transmitter
R = Receiver
Fibre = Any medium type supported by Fibre Channel
LCF = Link_Control_Facility
A(1), A(2) = N_Port addresses within Node A
B(1), B(2) = N_Port addresses within Node B

Figure 10. FC-2 Physical model

4.7 Topology

Topologies defined, based on the capability and presence or the absence of Fabric between the N_Ports, are:

— Point-to-point Topology
— Single InBand Address Topology

Attributes of a Fabric may restrict operation to certain communication models. In figures 11 and 12, both the Inbound and Outbound Fibres are represented as a single line.

4.7.1 Point-to-point topology

The point-to-point topology, as shown in figure 11, in which communication between N_Ports occurs without the use of Fabric is defined as point-to-point. Here the communicating N_Ports are directly interconnected by a single dedicated link.

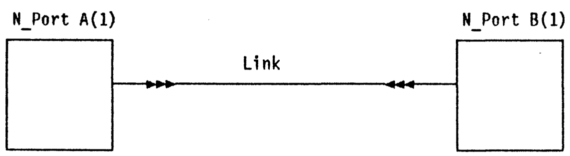


Figure 11. Point-to-point topology

4.7.2 Single InBand Address topology

This topology uses the Destination_Identifier (D_ID) embedded in the frame header to route the frame through a Fabric to the desired destination N_Port. Figure 12 illustrates multiple N_Ports interconnected by a Fabric.

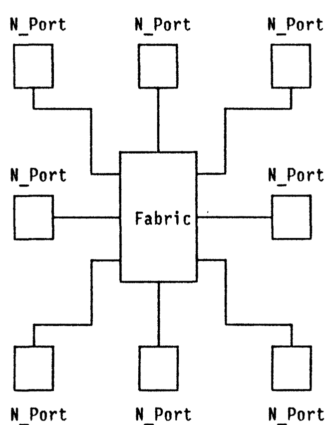


Figure 12. Single InBand Address topology

4.8 Classes of service

Three Classes of service are defined. These Classes of service are distinguished primarily by the methodology with which the communication circuit is allocated and retained between the communicating N_Ports and the level of delivery integrity required for an application. Certain Fabrics and/or N_Ports may not support all Classes of service.

4.8.1 Class 1 service - Dedicated Connection

Class 1 is a service, which once established, shall be retained and guaranteed by the Fabric (i.e. Dedicated Connection). This service guarantees full bandwidth available between two N_Ports across the established connection.

In Class 1, frames shall be delivered in the same order as they are transmitted by the Source N_Port.

4.8.2 Class 2 service - Multiplex

Class 2 is a connectionless service with the Fabric multiplexing frames at frame boundaries.

The transmitter shall transmit frames in a sequential order within a given Sequence. In Class 2, the Fabric may not guarantee the order of delivery and frames may be received out of order. ULPs using Class 3 are expected to provide the appropriate acknowledgements.

In Class 2, the Fabric guarantees notification of delivery or failure to deliver, in the absence of link errors. In case of link errors, notification is not guaranteed (since the S_ID may not be error free).

4.8.3 Class 3 service - Datagram

Class 3 is a connectionless service with the Fabric multiplexing frames at frame boundaries. Class 3 supports only open-ended delivery where the destination N_Port does not send any confirmation Link_Control frames on receipt of valid Data frame(s).

In Class 3, the Fabric is expected to make a best effort to deliver the frame to the intended destination and shall not issue busy or reject to the

Source if the Fabric is unable to deliver the frame.

4.9 Intermix

Intermix is an option of Class 1 service which allows interleaving of Class 2 and Class 3 frames during an established Class 1 Connection between two N_Ports. During a Class 1 Connection, an N_Port capable of Intermix may transmit and receive Class 2 and Class 3 frames interleaved with Class 1 frames. Support for the Intermix option of Class 1 service shall be indicated during Login. An N_Port shall be capable of receiving intermixed frames before it is allowed to transmit intermixed frames.

In a point-to-point topology, both interconnected N_Ports shall be required to support Intermix if Intermix is to be employed. In the presence of a Fabric, both the N_Port and the Fabric Port (F_Port) to which it is attached shall be required to support Intermix if Intermix is to be employed. Fabric support for Intermix requires that the full Class 1 bandwidth during a Dedicated Connection be maintained. Intermix permits the unused Class 1 bandwidth to be used for transmission of Class 2 and Class 3 frames.

4.10 General Fabric model

The primary function of the Fabric is to receive the frame(s) from a source N_Port and route the frame(s) to the destination N_Port whose address is specified in the frame(s). Each N_Port is physically attached through a Link to the Fabric. FC-2 specifies the protocol between the Fabric and the attached N_Ports. A Fabric is characterized by a homogeneous address space.

All Fabrics shall specify the classes of service they support. However, Fabrics are allowed to provide the Class(es) of service through equivalent mechanisms and/or functions. Refer to the Fabric requirements (FC-F) document for these equivalent functions provided by some Fabrics. See 23.6 for Fabric specific service parameters.

Figure 13 illustrates the general Fabric model. The model consists of the following major functional blocks and sub-blocks.

- Bidirectional Fabric Ports (F_Ports)
  - Receive buffer
- Connection Sub-Fabric
- Connectionless Sub-Fabric
  - Receive buffer queue management
- Fabric Controller

4.10.1 Fabric Ports (F_Ports)

The Fabric model contains two or more Fabric Ports (F_Ports). Each F_Port can be attached to an N_Port through a Link. Each F_Port is bidirectional and shall support one or more communication models.

The receiving F_Port responds to the sending N_Port according to the FC-2 protocol. The Fabric may or may not verify the validity of the frame as it passes through the Fabric (see 17.7). If the Fabric chooses to verify the validity and finds a frame to be invalid, it shall forward the frame to the destination if that frame could have been forwarded had the Fabric not performed the validity check (see 17.6.2).

An F_Port may or may not contain a receive buffer for the incoming frame(s). If present, the receive buffer size determines the maximum frame size that the Fabric can handle for Class 2 and Class 3 service and for the frame which establishes Class 1 service. (The receive buffer size is determined by the Login protocol.)

The Fabric directs the frame to the proper outgoing F_Port based on the N_Port identifier (D_ID) embedded in the frame. In some Fabrics, the frame may be routed to all F_Ports supported by the Fabric. Refer to the Fabric Requirements (FC-F) document for description of this broadcast function provided by some Fabrics. The address translation and the routing mechanisms within the Fabric shall be transparent to N_Ports and are not specified in this document.

4.10.2 Connection based Sub-Fabric

The Connection based Sub-Fabric provides Dedicated Connections between F_Ports and the N_Ports attached to these F_Ports. Such Dedicated Connections shall be bidirectional and support full data rate concurrently in both directions. A Dedicated Connection shall be retained until a removal request is received from one of the communicating N_Ports.

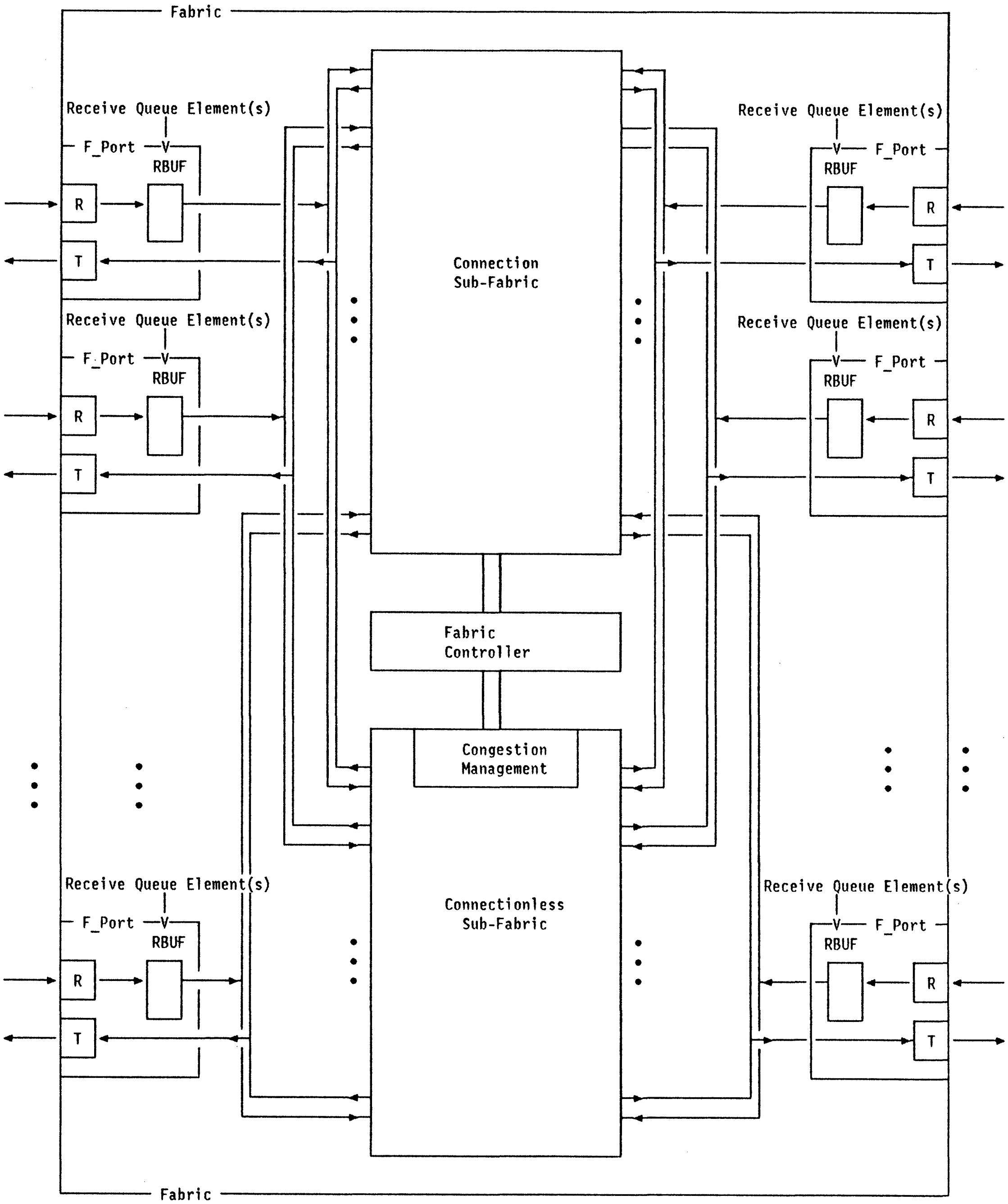
On receiving a request from an N_Port for Class 1 service, the Fabric shall establish a Dedicated Connection to the destination N_Port through the

Connection Sub-Fabric. The mechanisms used by the Fabric to establish the connection are transparent to FC-2.

The Sub-Fabric is not involved in flow control which is managed end-to-end by N_Ports.

If the Fabric is unable to establish a Class 1 service, it shall return a Busy or a Reject with a valid reason code. Once the Class 1 service is established, all frames between the communicating N_Ports shall be routed through the same circuit and all responses are issued by N_Ports.

If two N_Ports are in Class 1 Connection, a Class 1 connect request from another source shall be rejected, regardless of Intermix support. If Intermix is supported and the Fabric receives a



Legend:
F_Port - Bidirectional Fabric Port
RBUF - Receive Buffer
R - Receiver
T - Transmitter

Figure 13. General Fabric model

Class 2 or Class 3 frame from another N_Port, destined to one of the N_Ports connected in Class 1, the Fabric may

- allow delivery if the associated exit F_Port is not transmitting other frames, or
- disallow delivery if the F_Port is transmitting.

If the disallowed frame is Class 2, the transmitting N_Port shall be notified. The exit F_Port may be able to hold the frame for a period of time before discarding the frame or returning a busy. If it is Class 3, no notification shall be sent.

4.10.3 Connectionless Sub-Fabric

A Connectionless Sub-Fabric is characterized by the absence of Dedicated Connections. The Connectionless Sub-Fabric multiplexes frames at frame boundaries between an F_Port and any other F_Port and between the N_Ports attached to them.

The Fabric shall notify the transmitting N_Port with a reason code if it is unable to deliver a Class 2 frame. In the case of a Class 3 frame, the Fabric shall not notify the transmitting N_Port.

A given frame flows through the Connectionless Sub-Fabric for the duration of the routing. Once the routing is complete, the Connectionless Sub-Fabric has no memory of the routing or the destination.

When frames from multiple N_Ports are targeted for the same destination N_Port as in Class 2 and Class 3, congestion of frames may occur within the Fabric. Management of this congestion is part of the Connectionless Sub-Fabric and buffer-to-buffer flow control.

If an N_Port breaks flow control rules which would cause overflow, the Fabric may discard the overflow frame without notification and shall log the error.

4.10.4 Fabric Controller

A Fabric Controller controls and coordinates all Fabric functions. It is an FC-2 addressable N_Port with a well-known address.

4.11 Building Blocks

A set of building blocks, their interrelationships, behavior rules and usage hierarchy are defined in FC-2. These building blocks are:

- Frame
- Sequence
- Exchange
- Protocol

4.11.1 Frame

Frames are based on a common frame format. Frames are broadly categorized as Data_Frames and Link_Control frames. Data_Frames may be used as Link_Data frames or Device_Data frames. (see 20.2). Link_Control frames are further classified as Acknowledge (ACK) frames and Link_Response (Busy and Reject) frames. (see 20.3).

Selective retransmission of frames (i.e., retry partial Sequence) for error recovery is not supported in this standard. But busy conditions which are part of operating protocols need retransmission of busied frames up to its ability to retry.

4.11.2 Sequence

A Sequence is a set of one or more related Data_Frames transmitted unidirectionally from one N_Port to another N_Port with corresponding Link_Control frames, if applicable, transmitted in response. It may be used as a Request or a Reply Sequence. An N_Port which initiates a Sequence is referred to as the Sequence Initiator and the N_Port which responds to the initiation of a Sequence as the Sequence Recipient. Usage rules for these Sequences are specified in "24 Exchange, Sequence, and Sequence Count Management."

Error recovery is performed at the Sequence boundary. If a frame is not transmitted error free, and error policy requires error recovery, the Sequence to which the frame belongs shall be retransmitted. (See 29.)

4.11.2.1 Sequence_Identifier (SEQ_ID)

The Sequence Initiator (which transmits Data_Frames) shall assign the Sequence a Sequence_Identifier (SEQ_ID). The Sequence Recipient shall use the same SEQ_ID in its response. This N_Port pair shall have only one SEQ_ID active per Exchange_Identifier at any given time. (See 4.11.3.1.) In other words, an N_Port may initiate multiple Sequences for multiple Exchanges but only one SEQ_ID shall be active for a given Exchange_Identifier "irrespective of direction" at any given time.

4.11.2.2 Sequence Status Blocks

A Sequence Status Block is a logical construct representing the format of the status information. It is used to track the progress of a Sequence at an N_Port on a frame by frame basis. A Sequence Initiator Status Block and a Sequence Recipient Status Block are used by the respective N_Port to track the status of a given Sequence.

- When a Sequence Initiator starts a Sequence, the Sequence Initiator shall allocate a Sequence Status Block to be associated with the SEQ_ID it has assigned.
- The Sequence Recipient also shall allocate a Sequence Status Block at its end, associated with the SEQ_ID the Sequence Initiator has allocated.
- Both the Sequence Initiator and Sequence Recipient N_Ports track the status of this Sequence through the Sequence Initiator and the Sequence Recipient Status Blocks respectively.

The maximum number of concurrent Sequences (SEQ_IDs) the Sequence Initiator may initiate with a given destination is limited by the number of Recipient Sequence Status Blocks provided by the service parameter of that destination. This value is established during Login through the service parameters (See "23 Login and Service Parameters").

4.11.3 Exchange

Exchanges are composed of one or more non-concurrent Sequences. An Exchange is the basic mechanism used to transfer data between two N_Ports (or Nodes). An Exchange may be unidirectional, or bidirectional. A unidirectional Exchange results if Sequence(s) within the Exchange perform information transfer in a single direction. A bidirectional Exchange results if Sequence(s) within the Exchange perform information transfer in a both directions. An Exchange may span across one or more Class 1 Connections.

4.11.3.1 Exchange_Identifiers (OX_ID and RX_ID)

Exchange_Identifiers shall be used by the Originator and Responder to uniquely identify an Exchange.

- The Originator shall assign each new Exchange an Originator Exchange_Identifier (OX_ID) unique to the Originator and embed it in all frames of the Exchange.
- The Responder shall assign a Responder Exchange_Identifier (RX_ID) unique to the Responder in response and shall communicate it to the Originator before the end of first Sequence of the Exchange. The Responder embeds the RX_ID along with OX_ID in all subsequent frames of the Exchange.
- On receiving the RX_ID from the Responder, the Originator shall embed both the RX_ID, and OX_ID in all subsequent frames of the Exchange.

The Originator may initiate multiple concurrent Exchanges, but each shall use a unique OX_ID.

4.11.3.2 Exchange Status Blocks

An Exchange Status Block is a logical construct representing the format of the Exchange status information. It is used to track the progress of an Exchange on a Sequence by Sequence basis, An Originator Exchange Status Block and a Responder Exchange Status Block are respectively used by an Originator and a Responder to track the status of an Exchange.

- When an Originator initiates an Exchange, it shall assign an Originator Exchange Status Block associated with the OX_ID it has assigned to the Exchange (See 29.11.3).

— The Responder shall assign a Responder Exchange Status Block associated with the OX_ID the Originator has assigned to the Exchange. The Responder may reference it through the respective RX_ID the Responder optionally assigns or through the OX_ID (See 29.11.3).

— Both the Originator and the Responder shall track the status of the Exchange at their respective N_Ports via these Exchange Status Blocks.

4.11.4 Protocol

This standard provides and describes data transfer protocols and rules to be used by higher level protocols to accomplish a given level of function. This standard also provides Login and Logout control functions to manage the operating environment to perform data transfers.

— Fabric Login protocol: An N_Port shall establish a Class of service with the Fabric if any, by explicitly performing the Fabric Login protocol "or implicitly" through an equivalent method not defined in this standard.

— N_Port Login protocol: Before performing a Class 1 or Class 2 data transfer, the N_Port shall interchange its service parameters with the other N_Port through an N_Port Login protocol. (See 23.5 for the definition of service parameters).

— Data transfer protocol: The actual data transfer shall use Data transfer protocols. Examples are provided in Annex &dtp..

— N_Port Logout protocol: An N_Port may request removal of its service parameters from the other N_Port by performing an N_Port Logout protocol. This request may be used to free up resources at the other N_Port.

4.11.5 Usage hierarchy

These FC-2 building blocks are used in a hierarchical fashion, as illustrated in figure 14.

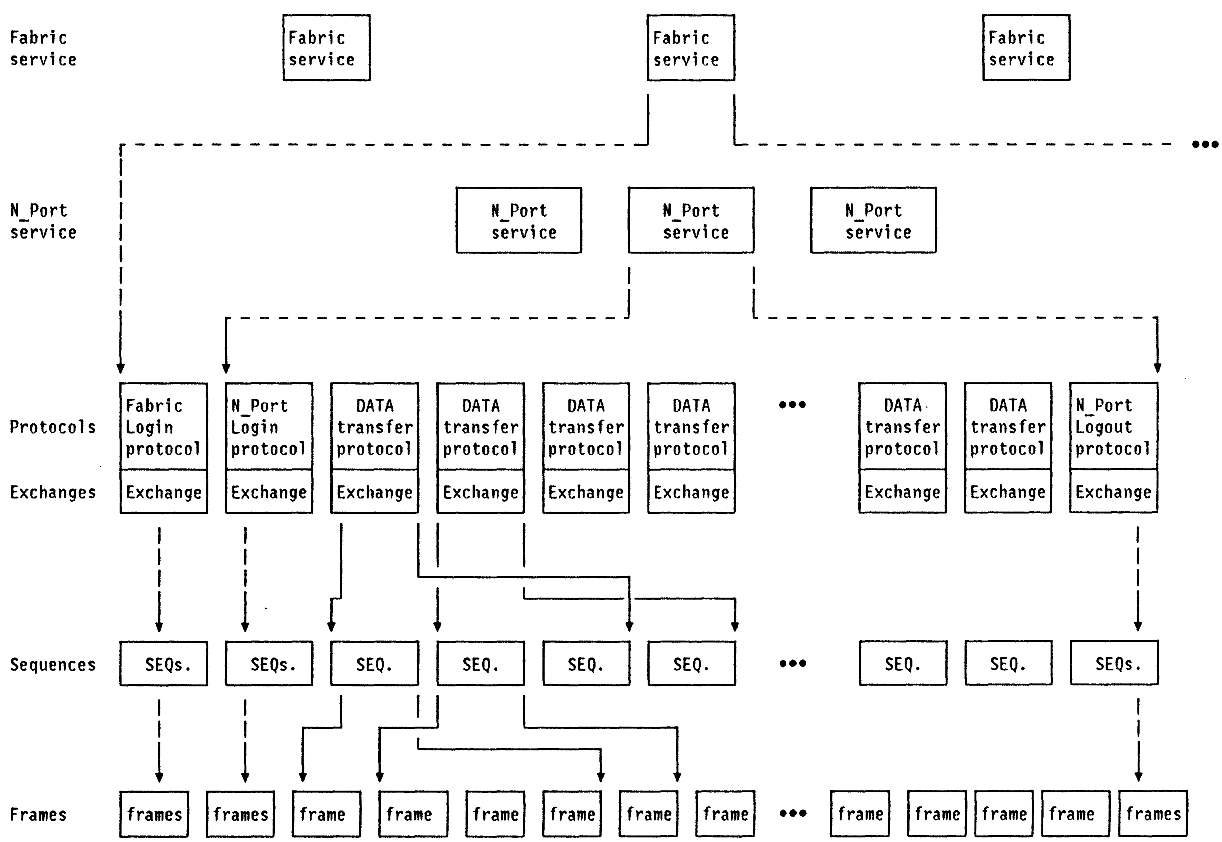


Figure 14. FC-2 building blocks hierarchy

4.12 Segmentation and Reassembly

Segmentation and Reassembly are the FC-2 functions provided to subdivide a block of data to be transferred into smaller "data segments", embed each "data segment" in an individual frame as the payload, transfer these frames over the channel, and reassemble the block of data at the receiving end.

In FC-2, the maximum payload size in each frame is governed by the receive buffer sizes of the Fabric and the destination N_Port. The Payload of a frame is allowed to be of equal or varied size within this maximum.

FC-2 Segmentation and Reassembly processes, along with ULP gather and scatter processes, are illustrated in figure 17.

4.12.1 Application data

Upper level protocols (ULPs) have the knowledge of the amount of application data to be transferred. The application data may be contiguous or scattered in memory. The ULP shall specify the Payload for the last frame to be transferred in its entirety to FC-2, before FC-2 starts the transfer of the last frame.

4.12.1.1 Application data Mapping

A block of data may be mapped to FC-2 as one or more Sequences. Multiple blocks of data may be mapped to FC-2 as a single Sequence. Mapping blocks of data to Sequences at the sending end and Sequences to blocks of data at the receiving end are outside the scope of this standard but are performed within the N_Ports. Examples of mapping blocks of data are illustrated in figures 15 and 16.

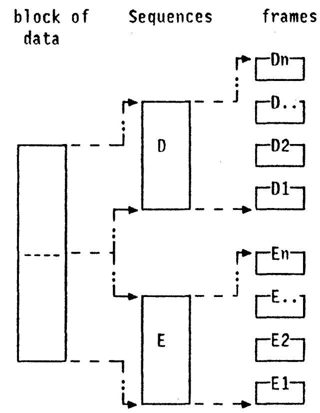


Figure 15. Mapping a block of data to multiple Sequences

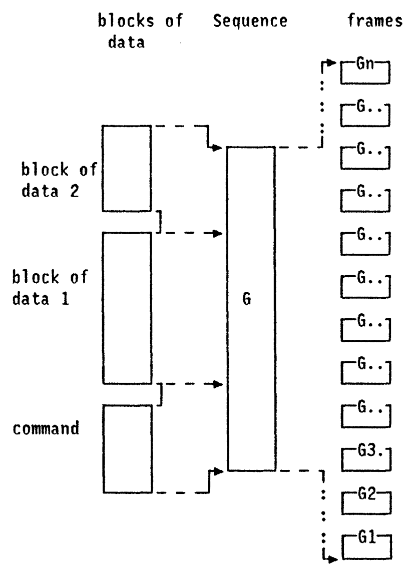


Figure 16. Mapping multiple blocks of data to a single Sequence

4.12.2 Sending end ULP

The sending end ULP shall completely specify each block of data, e.g., with starting address, ending address, size etc.

4.12.2.1 Offset

Offset is an FC_2 construct used at the sending ULP to indicate the displacement of the first data byte of each frame's payload with reference to the base address at the sending end for a block of data. The sending end ULP may provide FC-2 an initial Offset for each block of data. If a particular ULP does not provide an initial Offset, a value of all ones shall be used by FC-2. (The receiving end ULP shall be able to reassemble

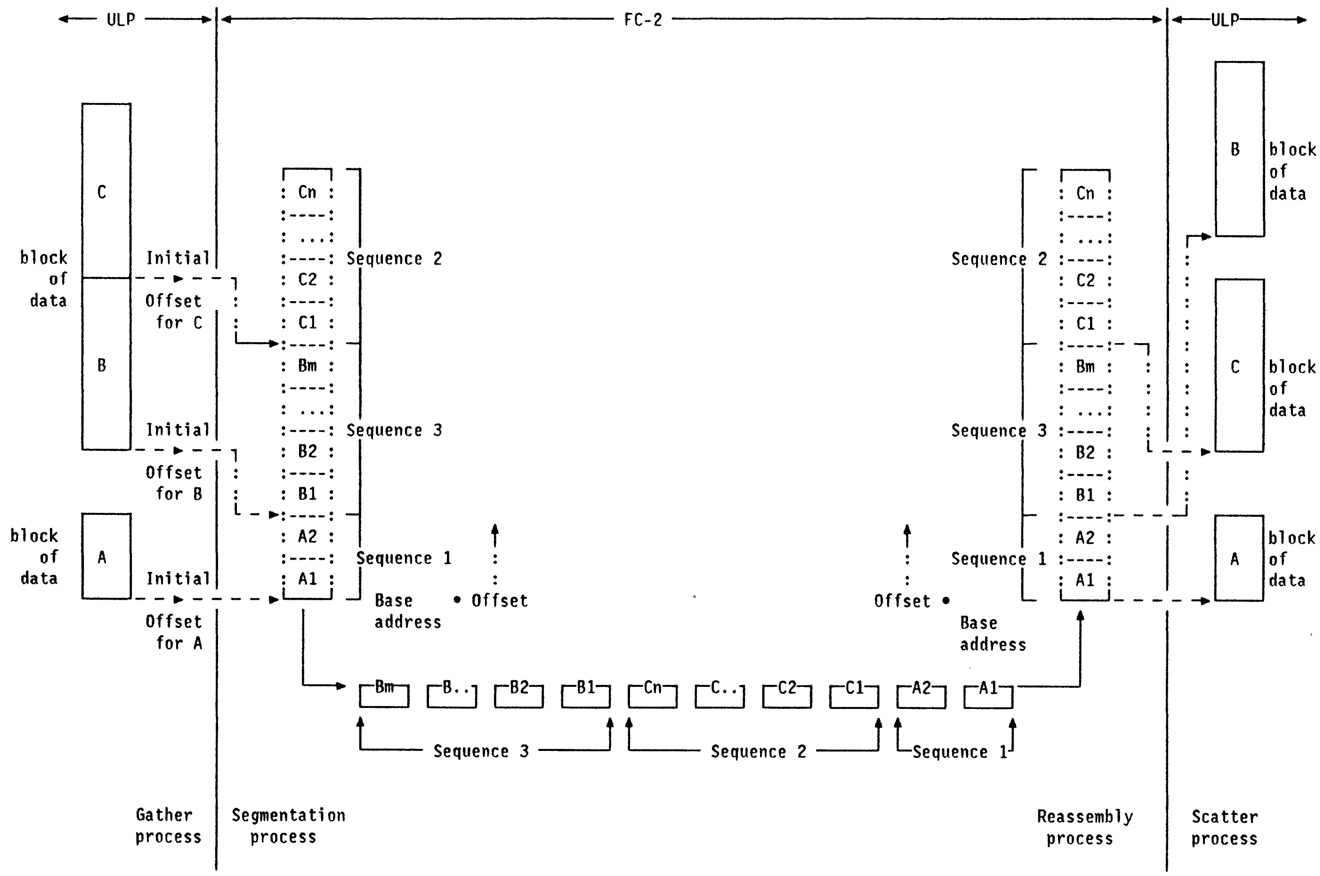


Figure 17. Segmentation and Reassembly

the application data without the aid of the offset value.)

4.12.2.2 Segmentation

"Segmentation" is the process by which FC-2 subdivides a block of data to be transferred into frames with payloads of equal or varied sizes and transmits them over the Link. During Segmentation, FC-2 determines the size of the payload for each frame in the Sequence. It embeds the data in the payload of the frame along with the Offset of the Payload "relative to the initial Offset of the block of data" and transfers each frame to the destination N_Port.

4.12.3 Receiving ULP

The receiving end ULP receives application data from the sending ULP as FC-2 Sequence(s).

4.12.3.1 Offset

The Offset within each frame is the displacement of the first data byte of the payload in that frame, with reference to the base address of the receiving ULP for the block of data.

4.12.3.2 Reassembly

Reassembly is the FC-2 process which reassembles a block of data by placing the Payload from each frame at the Offset specified in the frame. The receiving ULP may receive the data contiguously or scattered within its application buffer(s).

4.13 Error detection and recovery

In general, detected errors fall into two broad categories: frame errors and link-level errors. Frame errors result from missing or corrupted frames. Corrupted frames are discarded and the resulting error is detected at the Sequence level. At the Sequence level, an out of sequence frame is detected or the Sequence times out due to one or more missing Data frames or Acknowledgments. Once detected, the Sequence is aborted and the Sequence is retransmitted. Sequence errors may also cause Exchange errors which may also be aborted. Selective error recovery is performed on the failing Sequence or Exchange. Other properly performing Sequences are unaffected.

Link-level errors result from errors detected at a lower level of granularity than frames where the basic signal characteristics are in question. Link-level errors include such errors as loss of signal, loss of synchronization, and several link timeout errors which indicate no frame activity. When link-level errors are detected by the receiver, it is not possible to determine if the error was caused by receiver logic, the transmission medium, the transmitter logic, or the remotely attached Port. Recovery from link-level errors is accomplished by transmission and reception of Primitive Sequences. Recovery at the link-level disturbs normal frame flow and may introduce Sequence errors which must be recovered following recovery at the link-level.

Primitive Sequences are low-level Ordered Sets which provide word synchronization and a handshake mechanism in order to ensure that both the transmitter and receiver are able to transmit and receive "primitive" signals. Primitive Sequence protocols are specified for Link Failure, Link Initialization, Link Recovery, and Online-to-Offline transition.

4.13.1 Error recovery hierarchy

The recovery of errors may be described by the following hierarchy:

Abort Exchange
(Abort Exchange Protocol-frames)

Abort Sequence
(Abort Sequence Protocol-frames)

Link Reset
(Link Recovery Protocol-primitives)

Link Failure
(Link Failure Protocol-primitives)

Link Initialization
(Link Init Protocol-primitives)

American National Standard for Information Systems

FC-0 level - Physical interface and media

5 FC-0 functional characteristics

The FC-0 describes the lowest level physical link in the Fibre Channel system. It is designed for maximum flexibility and allows the use of a large number of technologies to meet the widest variety of system requirements.

5.1 General characteristics

FC-0 has the following general characteristics.

- Serial data streams are supported at data rates of 132,812 5 MBit/s, 265,625 MBit/s, 531,25 MBit/s and 1,062 5 GBit/s. All data rates have clock tolerances of $\pm$100 ppm.
- A system bit error rate (BER) $\leq 10^{-12}$ is supported.
- The interoperability specifications are at the external interface as shown at points S and R in figure 7.
- The interface to FC-1 occurs at the serial data interfaces. For the transmitter this is point b in figure 8. In the case of the receiver the interface is the retimed serial data indicated at point d in figure 9. Additional points are discussed and recommended in the annexes.

The physical links have the following characteristics:

- Physical point to point data links.
- Any transmitter is always connected to the same receiver, optical switches are not supported.
- An elastic buffer is required at all nodes in the switch fabric or repeaters. All signals must be retimed before retransmission to prevent jitter accumulation.
- All specifications are end-of-life worst case values over manufacturing, temperature, power supply and cable plant variations.

The interface between FC-0 and FC-1 is intentionally structured to be technology and implementation transparent. That is, the same set of commands and services may be used for all signal sources and communications media. The intent is to allow for the interface hardware to be interchangeable at the system level without regard to the technology of a particular implementation. As a result of this, all safety or other operational considerations which may be required for a specific communications technology are to be handled by the FC-0 level associated with that technology. An example of this would be the open fibre control system required for the short wave laser technologies as discussed in section 6.2.3.

5.2 FC-0 States

5.2.1 Transmitter States

The transmitter is controlled by the FC-1 level. Its function is to convert the serial data received from the FC-1 level into the proper signal types associated with the operating media.

- **Transmitter Not-Enabled State**: A not-enabled state is defined as the optical output off for optical communication media and a logical zero for coaxial media. This is the state of the transmitter at the completion of power on sequence unless the transmitter is specifically directed otherwise by the FC-1 level.
- **Transmitter Enabled State**: The transmitter shall be deemed to be in an enabled state when the transmitter is capable of operation within its specifications while sending valid data patterns.
- **Transition Between Not-Enabled and Enabled States**: The sequence of events required for the transition between the not-enabled and

enabled states will be media dependant, both as to the time period required and the optical or logical activity on the media interface.

- **Transmitter Failure State**: Some types of transmitters are capable of monitoring themselves for internal failures. Examples are laser transmitters where the monitor diode current is may be compared against a reference to determine proper operating point. Other transmitters, such as Light Emitting Diodes and coaxial transmitters do not typically have this capability. If the transmitter is capable of performing this monitoring function then a detection of a failure shall cause entry into the failure state.

5.2.2 Receiver States

The function of the receiver interface is to convert the incoming data from the form required by the communications media employed, retime the data, and present the data and an associated clock to the FC-1 level.

The receiver has no states.

5.3 Response to input data phase jumps

Some link implementations may have phase discontinuities in the incoming data. This may occur from causes such as the operation of a serial switch at the transmitter. In the event of such a phase discontinuity the recovery characteristics of the receiver shall be specified as follows:

Phase Jump
Uniform distribution between $\pm 180°$

Link Worst Case

Degree of Recovery
Within BER objective (10^{-12})

Probability of Recovery
95%

Recovery Time
2 500 baud intervals

Additional Wait Time Before Next Phase Jump
None

The FC-0 level shall require no intervention from higher levels to perform this recovery. If, at the end of the specified time, the higher levels determine that bit synchronization is not present these levels may assume a fault has occurred and take appropriate action.

5.4 Limitations on invalid code

The FC-0 is intended to operate with valid 8B/10B encoded data supplied by the FC-1. However, it is recognized that it may occasionally be desired to pass invalid data for testing or other purposes. This may be done without upseting the operation of the FC-0 within the following limitations:

1. No more than two (2) bytes of invalid data may occur within a group of thirty two (32) bytes.
2. There must be at least thirty two (32) byte of valid data between bytes of invalid data.
3. Ivalid data bytes are limited to a balance between 40% and 60%. The maximum run length is limited to twelve (12) consecutive ones or zeros.
4. The invalid data may occur only after bit synchronisation has been acheived.

5.5 Receiver initialization time

The time interval required by the receiver from the initial receipt of a valid input to the time that the receiver is synchronized to the bit stream and delivering valid retimed data within the BER objective of the system shall not exceed 1 millisecond. Should the retiming function be implemented in a manner that requires direction from a higher level to start the initialization process, the time interval shall start at the receipt of the initialization request.

5.6 Signal detect function

The FC-0 may optionally have a signal detect function. If implemented this function shall indicate when the receiver has determined that the signal on the input is above a predetermined level. In the case of optical media the activation level will lie in a range whose upper bound is the minimum specified sensitivity of the receiver and whose lower bound is the lower of either the point where the bit error rate will exceed 10^{-2} or 7 dB below the minimum receiver sensitivity.

While there is no defined hysteresis for this function there shall be a single transition for any monotonic increase or decrease in the optical power. The optical power at the transition points shall lie within the previously defined bounds.

For optical links that employ a link control function, such as annex Annex H, the signal detect is replaced by the link status function as provides the same service.

The reaction time to a change in the input optical power shall be less than 12 seconds.

5.7 FC-0 nomenclature

The nomenclature for the technology options are illustrated in figure 18.

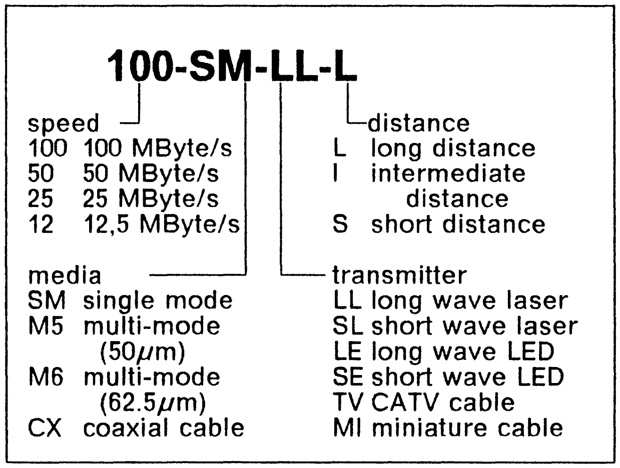


Figure 18. FC-0 nomenclature

5.8 FC-0 technology options

FC-0 provides for a large variety of technology options. Tables 1 and 2 list the presently supported signal interface types, both optical fibre and coaxial. For each listing the nomenclature is listed along with a reference to the section within this standard containing detailed information. There is also a brief description indicating the technology and the nominal communication distance.

Table 1. Fibre Media Signal Interface

100 MB/s 1,062 5 GBit/s				
100-SM-LL-L Section 6.1 SM 1 300nm 2m-10Km	**100-SM-LL-I** Section 6.1 SM 1 300nm 2m-2Km			
50 MB/s 531,25 MBit/s				
50-SM-LL-L Section 6.1 SM 1 300nm 2m-10Km	**50-M5-SL-I** Section 6.2 MM 780nm 2m-1Km			
25 MB/s 265,625 MBit/s				
25-SM-LL-L Section 6.1 SM 1 300nm 2m-10Km	**25-SM-LL-I** Section 6.1 SM 1 300nm 2m-2Km	**25-M6-LE-I** Section 6.3 MM(LED) 1 300nm 0-1Km	**25-M5-SL-I** Section 6.2 MM 780nm 2m-2Km	
12,5 MB/s 132,812 5 MBit/s				
12-M6-LE-I Section 6.4 MM(LED) 1 300nm 0-500m				

Table 2. Coaxial Media Signal Interface

100-CX-TV-S Section 7.2 0-50m	**50-CX-TV-S** Section 7.2 0-50m	**25-CX-TV-S** Section 7.2 0-50m	**12-CX-TV-S** Section 7.2 0-50m	

Tables 3 and 4 give information concerning the presently supported communications media types, both optical fibre and coaxial. Each signal type has an associated primary media type. However, in some cases the there is a listing for an alternate cable plant which may be used with degraded performance. For more information on alternate cable plants see Annex B and Annex E.

Table 3. Fibre Cable Plant.
(****** indicates alternate cable plant, see Annex B)

Single Mode				
100-SM-LL-L Section 6.1 SM 1 300nm 2m-10Km	**100-SM-LL-I** Section 6.1 SM 1 300nm 2m-2Km	**50-SM-LL-L** Section 6.1 SM 1 300nm 2m-10Km	**25-SM-LL-L** Section 6.1 SM 1 300nm 2m-10Km	**25-SM-LL-I** Section 6.1 SM 1 300nm 2m-2Km
Multi-mode (62.5μm)				
50-M6-SL-I Section 6.2 MM 780nm 2m-350m *********	**25-M6-SL-I** Section 6.2 MM 780nm 2m-700m *********	**25-M6-LE-I** Section 6.3 MM(LED) 1 300nm 0-1Km	**12-M6-LE-I** Section 6.4 MM(LED) 1 300nm 0-500m	
Multi-mode (50μm)				
50-M5-SL-I Section 6.2 MM 780nm 2m-1Km	**25-M5-SL-I** Section 6.2 MM 780nm 2m-2Km	**25-M5-LE-I** Section 6.3 MM(LED) 1 300nm *********	**12-M5-LE-I** Section 6.4 MM(LED) 1 300nm ********	

Table 4. Coaxial Cable Plant.
(****** indicates alternate cable plant, see Annex E)

CATV Coax				
100-CX-TV-S Section 7.2 0-50m	**50-CX-TV-S** Section 7.2 0-50m	**25-CX-TV-S** Section 7.2 0-50m	**12-CX-TV-S** Section 7.2 0-50m	
Miniature Coax				
100-CX-MI-S Section 7.3 0-10m *******	**50-CX-MI-S** Section 7.3 0-10m *******	**25-CX-MI-S** Section 7.3 0-10m *******	**12-CX-MI-S** Section 7.3 0-10m *******	

6 Fibre interface specification

This section defines the interfaces of the serial optical signal at the interconnect receptacles. Each conforming FC attachment shall be compatible with this serial optical interface to allow interoperability within an FC environment. The parameters specified in this section are based on a requirement that the bit error rate (BER) ≤ 10^{-12} be maintained under all conditions of this section, including the minimum active input interface power level. The corresponding cable plant specifications are described in section 8.

Table 5. FC-0 physical links for single mode classes

FC-0	Units	100-SM -LL-L	100-SM -LL-I	50-SM -LL-L	25-SM -LL-L	25-SM -LL-I
Section		6.1	6.1	6.1	6.1	6.1
Data Rate	MB/sec	100	100	50	25	25
Nominal Bit Rate	MBit/s	1 062,5	1 062,5	531,25	265,625	265,625
Tolerance	ppm	±100	±100	±100	±100	±100
Operating Range (typ)	m	2-10K	2-2K	2-10K	2-10K	2-2K
Fibre type	μm	9	9	9	9	9
Transmitter(S)						
Type		Laser	Laser	Laser	Laser	Laser
Spectral Center Wavelength	nm (min) nm (max)	1 285 1 330	1 270 1 355	1 270 1 355	1 270 1 355	1 270 1 355
RMS Spectral Width	nm (max)	3	6	3	6	30
Launched power, max **Launched power, min**	dBm (ave) dBm (ave)	-3 -9	-3 -12	-3 -9	-3 -9	-3 -12
Extinction Ratio	dB (min)	9	9	9	6	6
RIN 12 (max)	dB/Hz	-116	-116	-114	-112	-112
Eye Opening @ BER=10 -12	%	70	70	70	70	70
Receiver (R)						
Received power, min	dBm (ave)	-25	-20	-25	-25	-20
Received power, max.	dBm (ave)	-3	-3	-3	-3	-3
AC optical path penalty	dB (max)	2	2	2	2	2
Return loss of receiver	dB (min)	12	12	12	12	12

Table 6 (Page 1 of 2). FC-0 physical links for multi-mode classes

FC-0	Units	50-M5 -SL-I	25-M5 -SL-I	25-M6 -LE-I	12-M6 -LE-I	
Section		6.2	6.2	6.3	6.4	
Data Rate	MB/sec	50	25	25	12,5	
Nominal Bit Rate	MBit/s	531,25	265,625	265,625	132,813	
Tolerance	ppm	±100	±100	±100	±100	
Operating Range (typ)	m	2-1K	2-2K	0-1K	0-500	
Fibre type	μm	50	50	62,5	62,5	

Table 6 (Page 2 of 2). FC-0 physical links for multi-mode classes						
FC-0	**Units**	**50-M5 -SL-I**	**25-M5 -SL-I**	**25-M6 -LE-I**	**12-M6 -LE-I**	
Transmitter(S)						
Type		Laser	Laser	LED	LED	
Spectral Center Wavelength	nm (min) nm (max)	770 850	770 850	1 270 1 380	1 270 1 380	
Spectral Width RMS/FWHM	nm (max)	4	4	fig 22	200	
Launched power, max **Launched power, min**	dBm (ave) dBm (ave)	+1,3 -5	4,0 -5	-14 -20	-14 -23	
Extinction Ratio	dB (min)	6	6	9	10	
RIN$_{12}$ (max)	dB/Hz	-114	-112	NA	NA	
Eye Opening @ BER=10^{-12}	%	70	70	NA	NA	
Deterministic jitter	% (p-p)	NA	NA	15	19	
Random Jitter	% (p-p)	NA	NA	15	11	
Optical Rise/Fall Time	ns (max)	NA	NA	fig 22	4,5	
Receiver (R)						
Received power, min	dBm (ave)	-17	-17	-26	-29	
Received power, max	dBm (ave)	+1,3	0	-14	-14	
Return loss of receiver	dB (min)	12	12	NA	NA	
Discrete Conn return loss	dB (min)	26	26	NA	NA	
Determnistic jitter	% (p-p)	NA	NA	23	20	
Random Jitter	% (p-p)	NA	NA	15	11	
Optical Rise/Fall Time	ns (max)	NA	NA	3,0	5,0	

6.1 SM data links

Table 5 gives the link budgets for 2 and 10 km single mode optical fiber links running at 266, 531, and 1 062 MBit/s. The optical power coupled into the fiber is limited to a maximum value consistent with Class 1 laser safety operation per IEC 825, "Radiation safety of laser products, equipment classification, requirements and user's guide".

6.1.1 Optical output interface

Table 7. Eye diagram mask table			
Rates	**X1**	**X2**	**Y1**
266 MBit/s	0.15	0.35	0.20
531 MBit/s	0.15	0.35	0.20
1 062 MBit/s	0.15	0.35	0.20

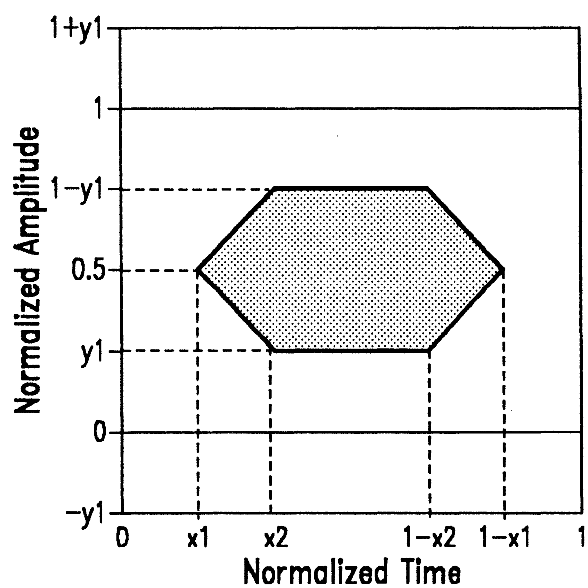


Figure 19. Eye diagram

The general laser transmitter pulse shape characteristics including rise time, fall time, pulse overshoot, pulse undershoot, and ringing, all of which should be controlled to prevent excessive degradation of the receiver sensitivity are specified in the form of a mask of the transmitter eye diagram at point S. For the purpose of a assesment of the transmit signal, it is important to consider not only the eye opening, but also the overshoot and undershoot limitations. The parameters specifying the mask of the transmitter eye diagram are shown in figure 19 and table 7. Annex A considers measurement setups for determining the eye diagram of the optical transmit signal.

6.1.2 Optical input interface

The receiver shall operate at a BER $\leq 10^{-12}$ when the input power falls in the range given when driven with a data stream output that fits the specified eye diagram mask.

Receiver characteristics specified in this document are receiver sensitivity, overload, receiver reflectance, and the allowable power penalty from the combined effects of dispersion, reflections, and jitter.

The minimum and maximum values of the average received power in dB give the input power range to maintain a BER $\leq 10^{-12}$. These values take into account power penalties caused by the use of a transmitter with worst-case, end of life, transmitter spectral, extinction ratio, and pulse shape characteristics specified for each application described,

The optical path power penalty (in dB) accounts for the total degradation along the optical path (from reflections, jitter, and the combined effects of dispersion resulting from intersymbol interference, mode-partition noise, and laser chirp).

The minimum acceptable value for receiver sensitivity shall equal the values specified in table 5. In addition, the receiver shall be designed to accommodate the dispersion power penalty.

The receiver overload shall equal or exceed the value given in table 5.

6.2 SW laser data links

The first two columns of table 6 give the link budget for short wavelength (SW) laser data links at 531 and 266 Mbit/s on multi-mode fibre.

6.2.1 Optical output interface

The Active Output interface shall exhibit the characteristics as shown in table 6.

Table 6 gives the link budgets, transmitter specifications, and receiver specifications for the short wavelength (SW) laser links operating at the 531,25 and 265,625 Mbit/s rates. The SW laser described for this link is the self-pulsating type, which provides a low noise transmitter source that essentially eliminates the problems of modal noise and laser reflection noise on the multi-mode fibre.

The self-pulsation frequency should be greater than three times the bit rate of the link to allow for efficient filtering of the self-pulsation noise. The self-pulsation frequency of most self-pulsating lasers will shift upward over a limited range with increased drive current. Hence, it is desirable to operate the SW laser for the 531 Mbit/s data link at a slightly higher power level than the 266 Mbit/s link.

The maximum and minimum of the allowed range of average transmitter power coupled into the fibre are **worst-case values** to account for manufacturing variances, drift due to temperature variations and aging effects, and operation within the specified minimum value of the extinction ratio.

The minimum value of the extinction ratio shall be the minimum acceptable value of the ratio (in dB) of the average optical energy in a logic one level to the average optical energy in a logic zero level. It shall be measured under fully modulated conditions in the presence of worst-case reflections.

The transmitter eye diagram requirements are identical to those presented in 6.1.1. The (normalized) mask of the transmitter eye diagram is shown in figure 19, and specified in table 7. The values given in this table are the same for both the 531 and 266 MBit/s SW laser transmitters of this section.

6.2.2 Optical input interface

The receiver shall operate at a BER of 10^{-12} over the link's lifetime and temperature range when the input power falls in the range given in table 6 and when driven with a data stream output that fits the specified mask of the eye diagram.

Receiver characteristics specified in this document are receiver sensitivity, overload, receiver reflectance, and the allowable power penalty from the combined effects of dispersion, reflections, and jitter.

The minimum sensitivity and maximum overload values of the average received power in dBm give the input power range over which a 10^{-12} BER is achieved. These values take into account power penalties caused by the use of a transmitter with worst-case transmitter spectral, extinction ratio, and pulse shape characteristics specified for each application described, and they include the effects of drifts due to temperature variation and aging.

The optical path power penalty (in dB) typically accounts for the total degradation along the optical path (from reflections, jitter, and the combined effects of dispersion resulting from intersymbol interference, mode-partition noise, and laser chirp). However, for multi-mode fibre data links (with both laser and LED sources), a common practice is to include or incorporate this optical path power penalty into the actual link budget specifications of both the power budget (producing amplitude degradation) and the timing budget (producing pulse spreading degradation). Therefore, the SW laser data links have no specified optical path power penalty (not included in table 6) because these link degradations of both amplitude and pulse spreading (primarily modal dispersion) are already "accounted for" in the power budget and time budget as specified between the transmitter and receiver in table 6.

The minimum acceptable value for receiver sensitivity shall equal the values specified in table 6. In addition, the receiver shall be designed to accommodate the maximum received power and the optical rise/fall time at the input to the receiver. The receiver overload shall equal or exceed the value given in table 6.

6.2.3 The Open Fibre Control Safety System (OFC)

An overview of an open fibre link detection and laser control system referred to as the Open Fibre Control (OFC) system is presented in this section. This laser control system can be used as a safety interlock for point-to-point optical fibre links that use semiconductor laser diodes as the optical source. The main reason for implementing this safety interlock system is that the optical power levels required to obtain the desired level of system performance would exceed the Class 1 limits defined by one or more national or international laser safety standards in the event that the optical fibre link between two optical ports is disconnected (i.e., an opened connector or cut fibre). Meeting the requirements for Class 1 classification is very important for an optical interconnect system in a computer environment due to the potential for customer exposure to laser radiation. Simply certifying the laser system at a higher classification level is impractical unless access can be restricted to only trained service personnel.

The following description of the OFC system applies to both the 531 Mbit/s and 266 Mbit/s SW laser data links. For purposes of this standard, the OFC timings have been chosen to be the same for both data rates. The timings are, however, based upon the higher allowed optical power levels of the 531 Mbit/s link.

6.2.3.1 Operation of the OFC system

During normal operation the optical link is a closed system and therefore clearly Class 1 regardless of the power in the fibre. It is only when the link is opened that a person can be exposed to laser radiation; hence by implementing a safety interlock that can detect when the optical link has been disrupted and shut down the laser or reduce the optical power level, one can design a Class 1 system. A block diagram showing a basic optical link with OFC circuitry is shown in Figure 20 and will be used in the following description of the operation.

The sequence of events which follow a disconnection in the optical data link are as follows:

1. Suppose data link A to B (1) is disconnected (for example, an opened connector or cut fibre).

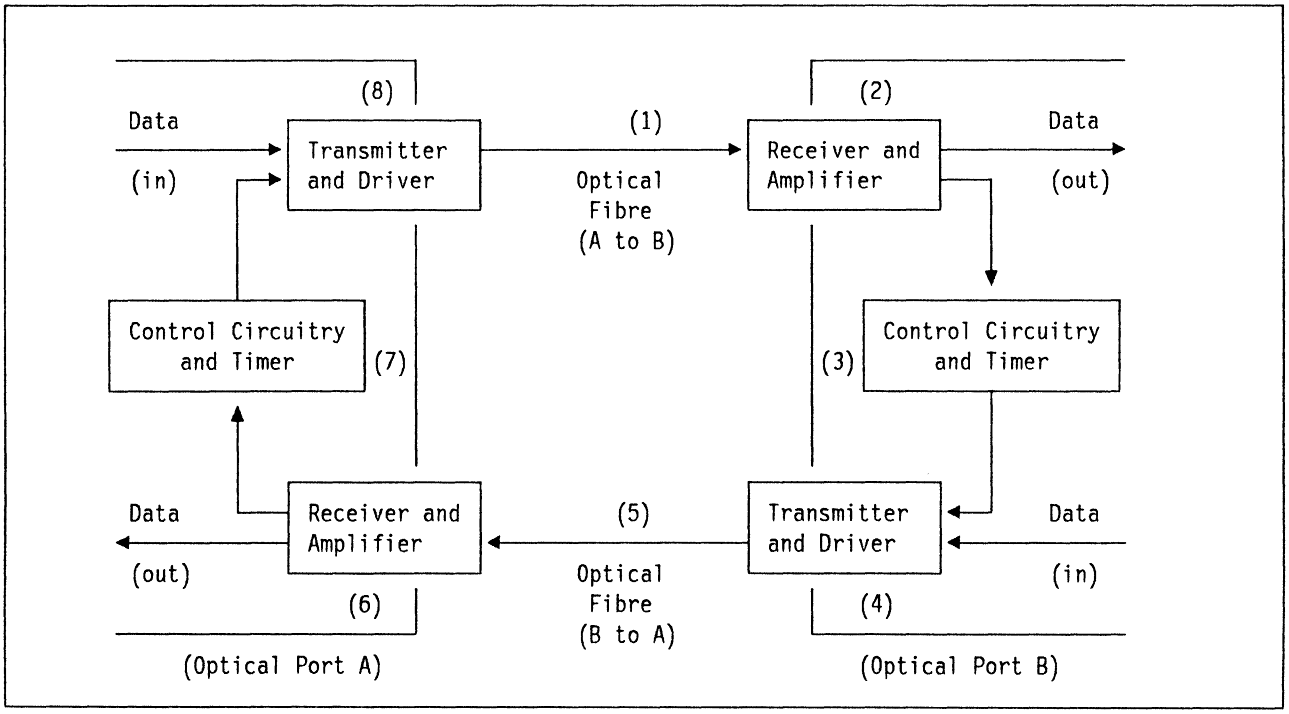


Figure 20. Block diagram of the OFC safety system

2. The receiver on optical port B (2) is no longer receiving an optical signal and triggers a loss-of-light flag to be raised in the control circuitry (3) on port B.

3. The control circuitry (3) responds by forcing off the port B transmitter (4) and starting an internal T second timer.

4. Since the port B transmitter is now off, no optical signal is received by the port A receiver (6). This triggers a loss-of-light flag to be raised in the control circuitry (7) on port A.

5. The control circuitry (7) responds by forcing off the port A transmitter (8) and starting an internal T second timer. Thus a safe condition with respect to the opened end of the link (1) has been created since both transmitters are now off.

6. When the internal T second timers expire, the control circuitry on each port will turn on the port's transmitter for t microseconds to check the link status (i.e. check for a closed link).

7. If the link is now a closed loop (i.e. data link A to B (1) has been reconnected), then a reconnect handshake will take place between the two ports and the transmitters will return to normal operation. If the link is still open, no optical signal will be received by at least one of the two receivers, the reconnect handshake will fail and the transmitters will once again be forced off for T seconds before the link check is repeated.

8. Note that to allow for synchronization of the control circuitry on the two ports during the link status check and reconnect handshake, either the expiring of the port's internal timer or the receiving of an optical signal from the other port causes an attempt to reconnect and the port timer to be reset.

9. If both data links A to B (1) and B to A (5) are disconnected at the same time, both ports A and B independently turn off their transmitters since a loss-of-light signal is generated at each receiver. Normal operation can not return until both data links are reconnected and the proper reconnect handshake has taken place between the ports.

Turning the transmitters on briefly after a predetermined time period T, allows the system to return to a normal mode of operation after an accidental or purposeful disconnection/reconnection of one or more of the connectors. Hence, the timed link checking and reconnection make the OFC safety interlock system automatic and nearly transparent to the user.

6.2.4 Link reconnection and OFC time periods

During system start-up or link reconnection, a handshaking takes place between the two optical ports. This handshaking occurs after a closed link condition is detected and insures that both optical ports contain functioning safety control circuitry and are capable of shutting down in the event of a break in the link. If either port in the link (for example port B) does not respond to the handshaking, then the control circuitry at the other end of the link (port A) will keep its transmitter inactive (i.e. either no emission or brief pulses every T seconds) and thereby maintain a safe link. Hence, the electronic safety control functions as a safety interlock which can not be defeated by attaching an optical port without OFC safety control circuitry.

The reconnection procedure is implemented with an on-off-on algorithm and four-state state machine as follows:

1. The OFC sytem is in the Active state during normal operation. In this state, the receiver is continuously monitored for a loss-of-light (LOL) condition. If a LOL condition is detected (LOL high), the OFC system transfers control to the Disconnect state.

2. The Disconnect state is used to maintain a safe (Class 1) link and to repetitively check the link for a closed link condition. Every T seconds the transmitter is activated for D1=t microseconds and the receiver is monitored for the LOL condition to disappear (LOL low). To exit from this state and proceed to the Stop state, light must be both sent and received during a t microsecond check period. This can occur in two ways:

   A. The T second timer expires, the transmitter is activated for D1=t microseconds and during this t microsecond period, a response is received from the other end of the link (i.e. LOL goes low). The transceiver is then considered "master" of the reconnection attempt. (NOTE: Due to timing variations, it is possible for both ends of the link to be "master" at the same time; this will not interfere with proper functioning of the OFC system.)

   B. A light signal is received from the other end of the link sometime during the T second wait period (i.e. LOL goes low). The T second timer is reset and the transmitter activated for D1=t microseconds in order to send a response. In this situation, the transceiver is considered "slave" of the reconnection attempt.

3. The Stop state is the "off" portion of the handshake algorithm. In the Stop state, the transmitter is disabled for D2 microseconds and the link monitored for a LOL condition (LOL high). The OFC system will remain in the Stop state until a LOL condition exists, this could be for an indefinite period of time. The system will exit from the Stop state in one of two ways:

   A. If the LOL condition (LOL high) is detected within the D2 microsecond time period, then the OFC system will transfer control to the Reconnect state.

   B. If the LOL condition (LOL high) is detected after the D2 microsecond time period has expired, then control is passed back to the Disconnect state.

4. The Reconnect state verifies that a closed link exists by requiring once again that light be sent and received during a D3=t microsecond time period. The behavior of this state is different depending upon whether-or-not the transceiver is "master" or "slave" of the reconnection attempt.

   A. If the transceiver is "master" of the reconnection attempt, then it once again initiates the send/receive sequence by activating its transmitter for D3=t microseconds and monitoring its receiver for a response signal (LOL low). If it receives a response within the t microsecond time period, it keeps the transmitter activated and transfers control to the Active state; otherwise, it disables the transmitter and transfers control back to the Disconnect state.

   B. If the transceiver is "slave" of the reconnection attempt, then it monitors its receiver for D3=t microseconds for a light present signal (LOL low) but does not activate its transmitter. If it receives a light present signal within the t microsecond time period, it activates its transmitter in order to send a response and transfers control to the Active state; otherwise, it keeps its transmitter disabled and transfers control back to the Disconnect state.

The timing periods used during each of the states of the state machine depend upon the time lags in the control circuitry of the transceivers and the optical link length. One must wait a minimum of one worst case round trip time for a response. For more detailed information on a recommended implementation of the OFC system and state machine, refer to Annex H.

6.2.4.1 OFC time periods

The OFC system makes use of a repetitive pulsing technique (t microseconds on every T seconds) during the time that a link is open (i.e. in the Disconnect state) instead of continuous operation in order to reduce the maximum possible viewing exposure to laser radiation and allow for classification as a Class 1 laser product. The maximum power level per pulse is a function of the wavelength, pulse duration (t), and pulse repetition frequency (PRF = 1/T) which determines the number of pulses (N) that occur during the time base used for classification purposes. Note that the use of the word "pulse" above refers to the "on time" during which the laser is powered on and being modulated with a valid full rate data pattern. The **worst case** maximum transmitter receptacle power allowed for the SW laser system is

Transmitter receptacle (max) = 2,3 dBm (1,7 mW).

To function correctly, each SW optical data link port must contain a transceiver that has implemented the OFC system with compatible OFC interface timings. The timing values referred to above that are consistent with the assumed maximum transmitter receptacle power and current (1990) IEC laser safety restrictions for a Class 1 system are as follows:

1. Pulse repetition time, T, (Disconnect state)
   - maximum = 11,9 seconds
   - minimum = 11,7 seconds
2. Pulse duration, D1=D3=t, (Disconnect and Reconnect states)
   - maximum = 410 microseconds
   - minimum = 400 microseconds
3. D2 stop time (Stop state)
   - maximum = 1 260 microseconds
   - minimum = 1 250 microseconds

These time periods when used according to the OFC interface specification described in this section, should result in a laser product which conforms to current (1990) emission requirements for Class 1 classification world wide. Note, however, that classification of a laser product must always be verified with measurements and calculations and not assumed.

6.2.5 Safety redundancy

The safety standards require that the maximum permissible exposure level for Class 1 operation can not be exceeded under any condition of operation, including single fault conditions. One way to handle this requirement is to introduce redundancy into the OFC system. The OFC system described above is implemented using totally redundant paths. Each of the paths can independently turn off the laser and only when they agree to turn on the laser will the laser receive any power. Hence, a double fault is required in the control system before an unsafe situation can occur.

The bloc diagram of the OFC system shown in figure 21 shows two control paths that must be satisfied before the laser can be activated. Each path contains a loss-of-light detector, digital filter, state machine, timers (counters) and laser driver control line. The two loss-of-light detectors monitor the receiver's signal. At least one of the detectors must be AC coupled and therefore respond only to a modulated signal.

The two loss-of-light detectors each feed a digital filter, the output of each filter is 'OR/EQUAL'd to form an internal Loss-of-Light (LOL) signal. The two filter outputs must be 'EQUAL' during the Disconnect, Stop and Reconnect states to activate the laser and bring the link up to a normal operating mode. But once the system is in the Active state, the 'OR' is implemented so that only one light detector is required to bring the link down and deactivate the laser.

The digital filters integrate the incoming signals to improve their reliability. The filters sample at a faster rate when acquiring a light present signal and at a slower rate when dropping a light present signal (i.e. setting LOL high), while still maintaining the correct handshake timings.

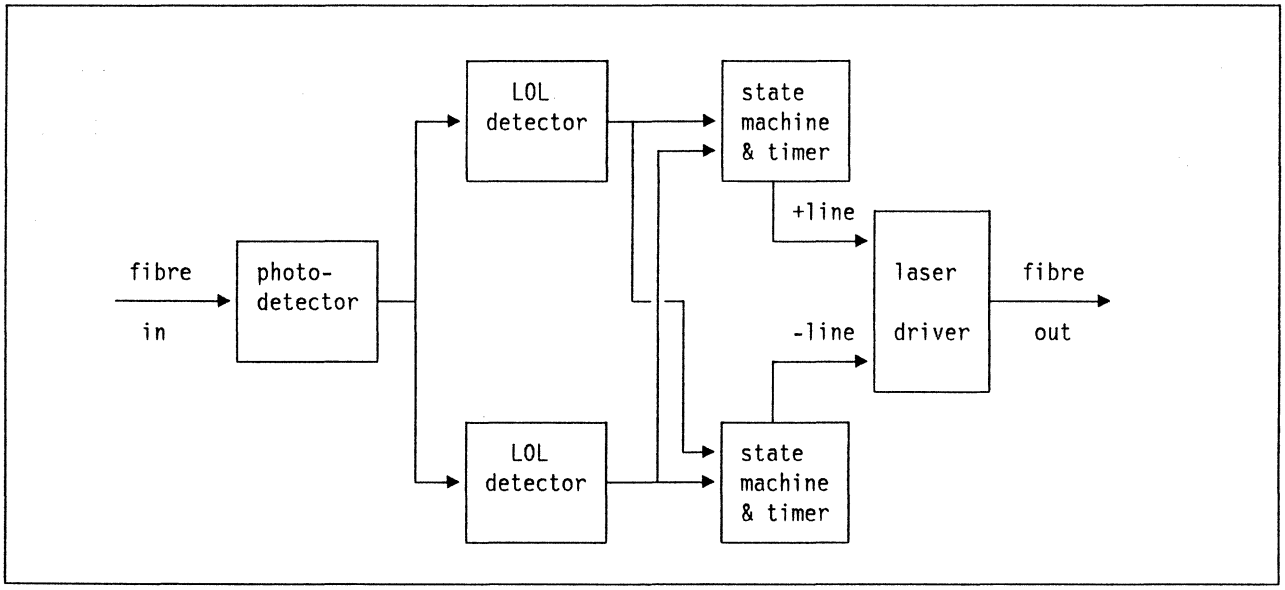


Figure 21. Block diagram of the OFC safety redundancy

The internal LOL signals are used to synchronize the counters and state machines. The state machines control the handshake algorithm implemented in the OFC system. Each state machine controls a 'Laser Off' output that connects to separate laser drive control circuits. The counters control the pulse repetition, pulse duration and stop times. The counters also provide the low frequency sampling clock to the digital filters.

6.3 25 MB/s MM 1 km LED data link

6.3.1 Optical output interface

The Active Output interface shall exhibit the characteristics as shown in table 6.

6.3.2 Spectral width

The spectral width is a function of source center wavelength and source rise and fall times. These specifications in conjunction with the fibre's chromatic dispersion and modal bandwidth parameters given in section 7 result in an optical rise time of less than 3 ns exiting a 1 kilometer fibre cable. Figure 22 shows curves for source rise and fall times ranging from 2,0 to 2,5 ns.

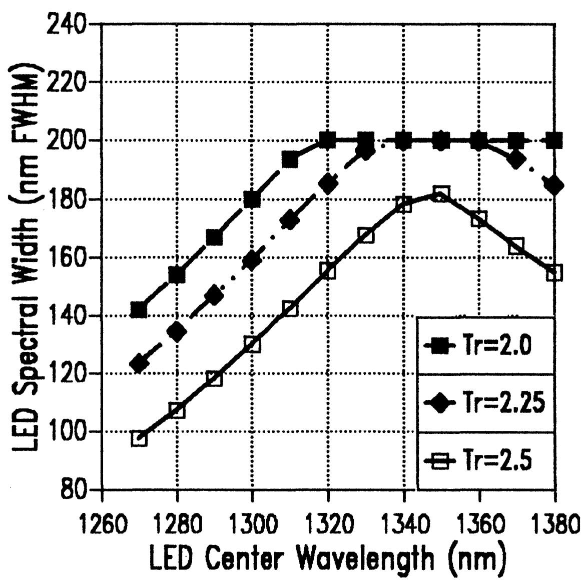


Figure 22. LED spectral width

6.3.3 Optical input interface

The Active Input Interface shall operate when provided with a signal corresponding to table 6.

6.4 12,5 MB/s 1 300nm MM LED 500m data link

6.4.1 Optical output interface

The Active Output interface shall exhibit the characteristics as shown in table 6.

6.4.2 Optical input interface

The Active Input Interface shall operate when provided with a signal corresponding to table 6.

7 Coaxial cable interface specification

This clause defines the interfaces of the serial electrical signal at the interconnect receptacles. Each conforming FC attachment shall be compatible with this serial electrical interface to allow interoperability within an FC environment. The parameters specified in this clause are based on a requirement that the bit error rate (BER) ≤ 10^{-12} be maintained under all conditions of section 7, including the minimum active input interface power level. The corresponding cable plant specifications are described in section 9.

Table 8. FC-0 coaxial links

FC-0	Units	100-CX-TV-S 100-CX-MI-S	50-CX-TV-S 50-CX-MI-S	25-CX-TV-S 25-CX-MI-S	12-CX-TV-S 12-CX-MI-S
Data Rate	MB/sec	100	50	25	12,5
Nominal Bit Rate	MBit/s	1062,5	531,25	265,625	132,813
Tolerance	ppm	100	100	100	100
Operating Range (typ)	m	10m/50m	10m/50m	10m/50m	10m/50m
Cable type	Ω	75	75	75	75
Transmitter(S)					
Type		ECL	ECL	ECL	ECL
Output Voltage					
- maximum	mV (p-p)	580	580	580	580
- minimum	mV (p-p)	235	235	235	235
S_{11} (75Ω)	dB	-30	-30	-30	-30
Duty Cycle Distortion (p-p)	ns	0,05	0,05	0,1	0,2
Random Jitter (p-p)	ns	0,1	0,1	0,2	0,4
Data Dependent Jitter (p-p)	ns	0,1	0,1	0,2	0,4
Rise/Fall Time (20-80%)	ns	0,3	0,3	0,6	0,8
Mask of Eye Diagram	ns	0,6	1,2	3,0	6,0
Receiver (R)					
Min Sensitivity	mV	50	50	50	50
Min Overload	mV	1000	1000	1000	1000
Max AC path penalty	dB	9	10	11	12
AC frequency	MHz	531	265	133	62,4
S_{11} (75Ω)	dB	-20	-20	-20	-20
Min Discrete Conn. Return Loss	dB	30	30	30	30

7.1 Coaxial ECL data links

The coaxial ECL data link definitions apply to two styles of coaxial cable, the 75 Ω CATV style coax and miniature coaxial cable. The electrical characteristics of these links are defined in table 8. The 75 Ω CATV style coax and miniature coaxial cables are interoperable, i.e., electrically compatible with minor impact on link length capability when intermixed. The interoperability implies that the transmitter and receiver level specifications defined in table 8 are preserved with the trade-off being distance capability in an intermixed system. Other electrically compatible, interoperable coaxial cables may be used to achieve goals of longer distance, higher frequency, or lower cost as desired in the system implementation provided that they are connector compatible.

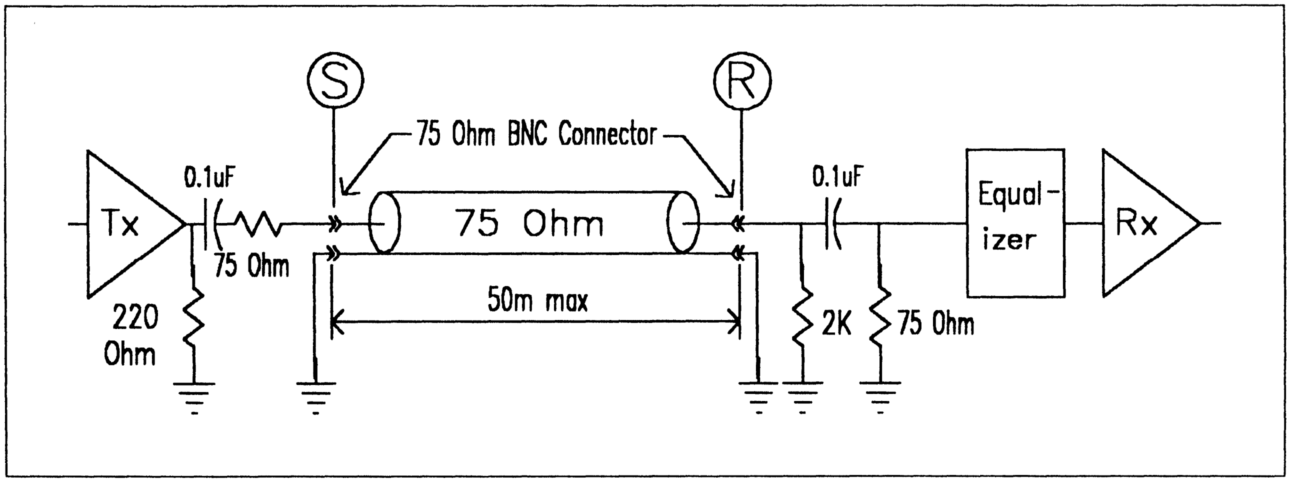


Figure 23. FC-0 with CATV style coaxial cable

7.2 75 Ω CATV style coaxial data link

This coaxial ECL data link definition applies to 75 Ω CATV style cable with industry standard 75 Ω BNC type connectors as shown in figure 23. The electrical performance of these 75Ω BNC connectors is compatible with CATV style connectors specified by EIA-403-A. The mechanical compatibility is defined by MIL-C 39012 for this bayonet lock coupling connector which intermates with comparable BNC UG/U connectors. Primary use of these links is for interconnection of cabinets.

7.2.1 ECL compatible driver characteristics

The output driver is assumed to have Emitter Coupled Logic (ECL) output levels. Due to the AC coupled drive configuration (figure 23), this driver may be either negative voltage referenced ECL or positive voltage referenced ECL (PECL). The output driver shall have output levels, measured at the input to the coax (point S), as shown in table 8, when terminated as shown in figure 23.

7.2.2 ECL compatible receiver characteristics

The normalized mask of the transmitter eye diagram is given in figure 24. The transmitter output, as specified in table 9, shall be measured using 50 m of cable (e.g. type RG-6/U from table 64 of annex Annex E) in the configuration shown in figure 23. The series terminating resistor (in figure 23) will cause a loss of 6 dB at the receiver and enough gain must be provided to minimize edge recovery losses.

The equalizer network corrects for timing loss variations due to the differences in propagation delay time between higher frequency components as well as attenuation loss of the transmitted signal. A proper equalizer design will have fixed values and need no adjustment relative to coaxial line length or levels.

A more detailed discussion of the equalizer is found in section 7.4.

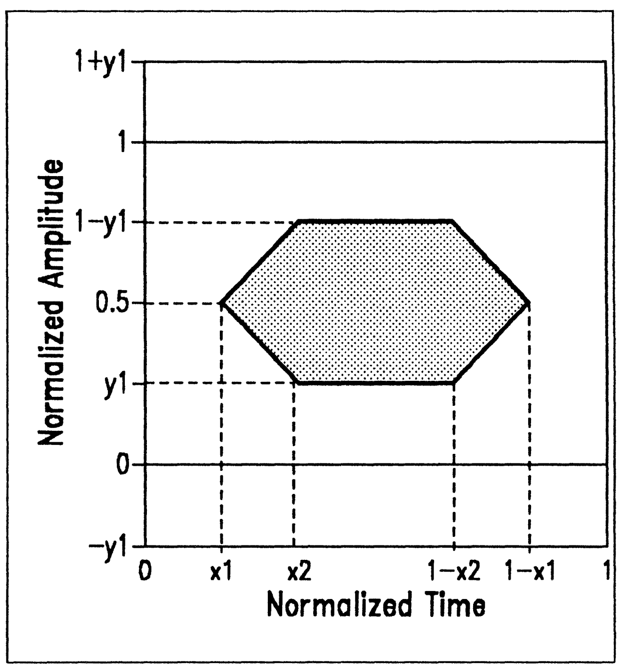


Figure 24. Eye Diagram

Table 9. CATV eye diagram mask				
Rates	X1	X2	Y1	Cable length
132 MBit/s	0,15	0,35	0,20	50m
266 MBit/s	0,15	0,35	0,20	50m
531 MBit/s	0,25	0,40	0,20	50m
1 062 MBit/s	0,25	0,40	0,20	50m

7.2.3 Coaxial cable characteristics

The CATV style coaxial cables defined for this interface have characteristics defined in table 10.

Table 10. Char of 75 Ω, CATV coax		
Category	Impedance	Attenuation (nom.)
Electrical	75±5 Ω	12dB/50m @531 MHz

7.3 75 Ω miniature coaxial data link

This ECL data link is a 75 Ω miniature coax with a optional shielded connector (figure 25) that interfaces with industry standard headers with 0,64 mm (0,025 in) square posts on 2.54 mm (0,100 in) spacing. Primary use of these links is within a cabinet.

Due to cost reasons, these connectors are not entirely shielded and leakage of RFI will occur. A shielded cabinet and leakage control techniques such as ferrite beads will be required to successfully meet EMC standards even with double shielded miniature coax (refer to annex E.4).

7.3.1 ECL compatible driver characteristics

The output driver is assumed to have Emitter Coupled Logic (ECL) output levels. Due to the AC coupled drive configuration (figure 25), this driver may be either negative voltage referenced ECL or positive voltage referenced ECL (PECL) levels. The output driver shall have output levels, measured at the input to the coax (point S), as shown in table 8, when terminated as shown in figure 25.

7.3.1.1 ECL compatible receiver characteristics

The normalized mask of the transmitter eye diagram is given in figure 24. The transmitter output, as specified in table 11, shall be measured using 10 m of cable (e.g. type RG-179 B/U from table 66 of annex Annex E) in the configuration shown in figure 25. The series terminating resistor (in figure 25) will cause a loss of 6 dB at the receiver and enough gain must be provided to minimize edge recovery losses.

Table 11. miniature coax eye diagram mask				
Rates	X1	X2	Y1	Cable length
132 MBit/s	0,15	0,35	0,20	10m
266 MBit/s	0,15	0,35	0,20	10m
531 Mbit/s	0,30	0,45	0,25	10m
1 062 MBit/s	0,30	0,45	0,25	10m

The equalizer for the miniature 75Ω coax is a design option for lower data rates and, in most

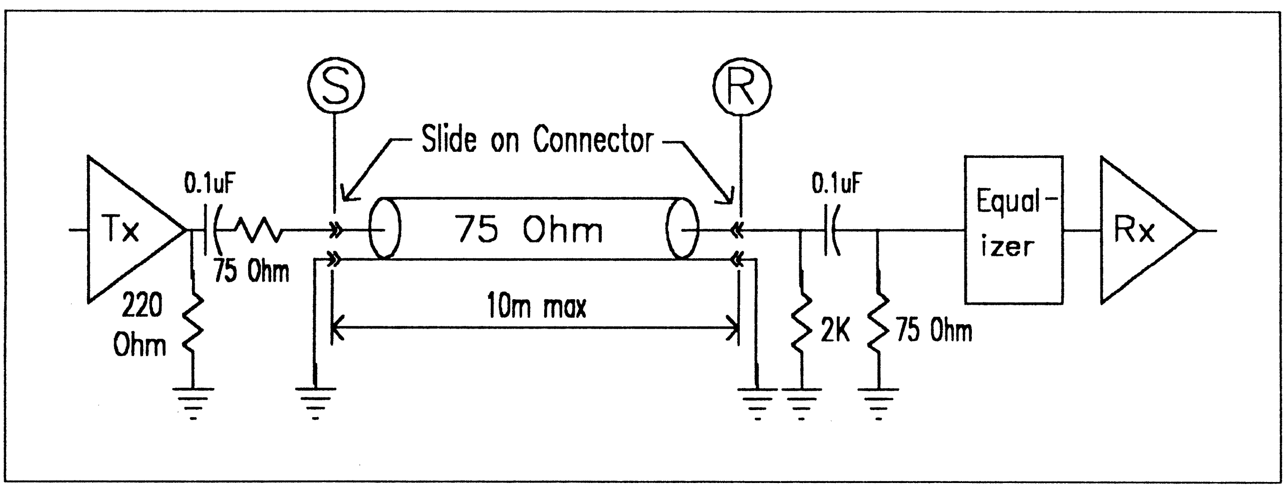


Figure 25. FC-0 with miniature coaxial cable

cases, is probably not required for adequate error rate. As in the case of the CATV style coaxial data link described in section 7.2, the purpose of this equalizer is to compensate for propagation time variations between high frequency and lower frequency components of the transmitted signal. A proper equalizer design will have fixed values and need no adjustment relative to line length or levels.

7.3.2 Miniature coaxial cable characteristics

The miniature style coaxial cables defined for this interface have characteristics defined in table 12.

It should be noted that the attenuation of the miniature coax at 531 MHz is less than the CATV coax characteristics specified in table 10. The miniature coax, by virtue of its smaller physical size, is more lossy to high frequency components of the signal than the CATV style cable. Therefore, the length and attenuation characteristics have been deliberately compromised to result in nearly the same signal levels and characteristics at the receiver that would be seen with CATV style cable.

Table 12. Char of 75 Ω, miniature coax

Category	Impedance	Attenuation (nom.)
Electrical	75±5 Ω	10dB/10m @531 MHz

7.4 Equalizer network

The equalizer network shown by block diagram in figures 23 and 25 corrects for timing loss due to the differences in propagation delay time between higher frequency components of the transmitted signal and amplitude loss at higher data rates. The primary physical mechanism of the loss is the skin effect characteristics of the coaxial cable. Except for these losses, the coaxial cable is extremely phase linear and very predictable for attenuation loss.

The equalizer network needed for coaxial data link compensation should be designed to peak up the 1F (highest nominal data rate, e.g. 531 MHz for 100 MB/sec transfers) and to accomplish a linear phase delay of the lower frequencies to adjust the phase. An example of an equalizer for a 25 MB/sec (132.813 MHz on the coax) is shown in figure 26 and 13. This equalizer attenuates the low frequency (≈34 MHz) by approximately 1 dB while also delaying the lower frequency components as much as 0,12 ns. Attempting to compensate for the 3rd and 5th order harmonics of the digital signal into the coaxial cable is usually not profitable in that it results in a fairly elaborate and costly equalizer.

For lower data rates, one may make the economic trade-off of elimination of the equalizer at modest risk to error rate performance. At higher data rates, > 50 MB/sec, an equalizer is judged to be almost mandatory unless the cable length is short.

Table 13. Equalizer design example

Frequency	Attenuation	Delay
MHz	dB	ns
40	-0,9625	0,113
50	-0,6845	0,191
60	-0,5205	0,196
70	-0,4067	0,179
80	-0,3255	0,156
90	-0,2660	0,135
100	-0,2213	0,116
110	-0,1869	0,100
120	-0,1600	0,087
130	-0,1385	0,076
140	-0,1211	0,066

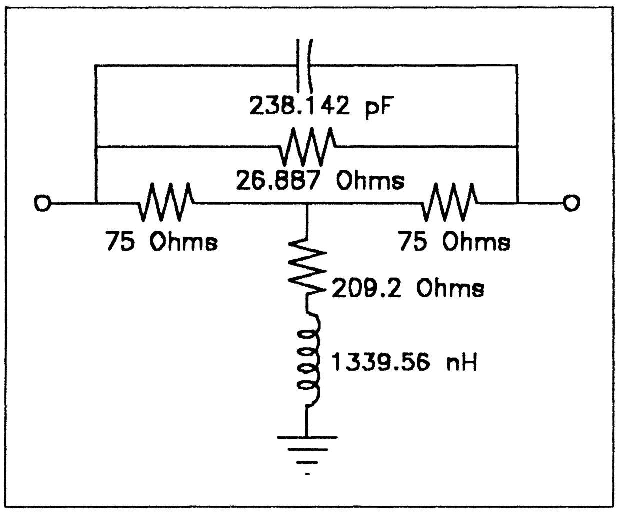


Figure 26. Equalizer design example

8 Fibre cable plant specification

The term "cable plant" encompasses all the fibre optic components between any two communicating ports and the associated connector plug at each end of the link.

8.1 SM cable plant specification

This sub-section specifies a single mode cable plant which is capable of supporting communication at distances of 2 and 10 Km at data rates of 265,625 MBit/s, 531,25 MBit/s and 1,062 5 GBit/s and provides for system upgrade. The cable plant is generally insensitive to data rate and therefore any installed portions of the cable plant may be used at any data rate. (See table 14.)

Table 14. Single mode cable plant

FC-0		100-SM-LL-L	100-SM-LL-I	50-SM-LL-L	25-SM-LL-L	25-SM-LL-I
Section		6.1	6.1	6.1	6.1	6.1
Operating Range	m	2-10K	2-2K	2-10K	2-10K	2-2K
Cable Plant Dispersion	ps/nm	35	12	60	60	12
Dispersion Related Penalty	dB	1	1	1	1	1
Reflection Related Penalty	dB	1	1	1	1	1
Loss Budget	dB	14	6	14	14	6

8.1.1 Fibre type

The fibre shall conform to EIA/TIA 492BAAA, "Detail Specification for Class IVa Dispersion Unshifted Single Mode Optical Waveguide Fibers Used In Communications Systems".

8.1.2 Cable plant loss budget

The loss budget for single mode fibre shall be no greater than specified in table 14. These limits were arrived at by taking the difference between the minimum transmitter output power and the receiver sensitivity and subtracting 2 dB to account for the dispersion and reflection link penalties. See annex D.3 for cable plant examples.

The cable plant loss shall be verified by the methods of FOTP-171, Method B3 or C3.

There are no requirements for fixed attenuators in the single mode classes. All receivers and transmitters, of a given data rate, may intercommunicate without operability limit as long as the distance requirement of the shorter class is not exceeded.

8.1.3 Optical return loss

The cable plant optical return loss, with the receiver connected shall be greater than or equal to 12 dB. This is required to keep the reflection penalty under control. The receiver shall have a return loss greater than or equal to one glass air interface, hence equivalent to the termination of the cable plant.

Mid-link connectors and splices are required to have a return loss greater than 30dB in order to control interferometric noise.

8.2 MM 62,5 μm Cable plant specification

The 62,5 μm Cable plant is the preferred cable plant for the 1 300nm LED based components. The short wave length 780nm CD laser can also be used on the 62,5 μm cable plant with some limited distance restrictions. See Annex B and Annex D.

Table 15. 62,5 μm multi-mode cable plant			
FC-0		**25-M6-LE-I**	**12-M6-LE-I**
Section		6.3	6.4
Data Rate	MB/sec	25	12
Operating Range	m	0 1K	0-500
Loss Budget	dB	6	6

8.2.1 Fibre types

The fibre shall conform to EIA/TIA - 492AAAA, "Detail Specification for 62,5 μm Core Diameter/125 μm Clading Diameter Class 1a Multimode, Graded Index Optical Waveguide Fibres".

Table 16. 62,5 μm fibre type		
Nominal Core Diameter EIA-455-58	**Cladding Diameter EIA-455-45 & EIA-455-176 or EIA-455-48**	**Nominal Numerical Aperture EIA-455-177**
62,5 μm	125 μm	0,275

8.2.2 Bandwidth

The following normalized bandwidth values are based on a nominal source wavelength of 850 and 1 300 nm. (See table 17.)

Table 17. 62,5 μm multi-mode bandwidth		
Wavelength	**Modal band-width (-3dB optical min)**	**Test per**
850 nm	160MHz*km	EIA-455-30 or EIA-455-51 with EIA-455-54
1 300nm	500MHz*km	EIA-455-30 or EIA-455-51 with EIA-455-54

NOTE - Some users may wish to install higher modal bandwidth fibre to facilitate later future use of the cable plant for higher bandwidth applications. For shorter distances, a lower bandwidth fibre may be substituted provided the performance requirements are met.

8.2.3 Cable plant loss budget

The loss budget for the multi-mode fibre cable plant shall be no greater than specified in table 15. These limits were arrived at by taking the difference between the minimum transmitter output power and the receiver sensitivity. This includes the losses of the fiber and other components in the link such as splices and connectors. The connectors at the ends of the links are included in the transmitter and receiver specifications and not in the cable plant limit.

In some cases the modal dispersion limit may be reached in an installaton before the installation loss limit of table 15. See annex D.4 for examples.

Conformace to the loss budget requirements shall be verified by means of OFSTP-14.

8.2.4 Fibre chromatic dispersion parameters

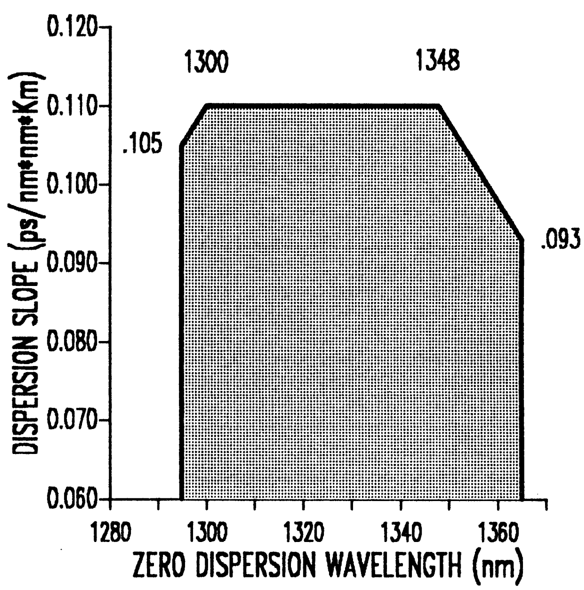


Figure 27. Fiber chromatic dispersion parameters

Figure 27 illustrates the required zero dispersion wavelength and the dispersion slope values for multimode fibre.

Sufficient optical bandwidth is provided when the zero dispersion wavelength and the dispersion slope measured at the zero dispersion wavelength (as defined by EIA-455-168) fall within the highlighted area.

These modal bandwidth and chromatic dispersion requirements in conjunction with the specifications on the center wavelength, spectral width, and risetimes of the transmitter in the active output interface specification in section 6 assures a 3 ns. exit response time over the respective distance.

8.3 MM 50 μm cable plant specification

The 50 μm cable plant is the preferred cable plant for the 780nm short wave length CD laser. The long wave length 1 300nm LED based components may also be used on the 50 μm cable plant with some distance restrictions. See annexes Annex B and Annex D.

Table 18. 50 μm multi-mode cable plant

FC-0		50-M5-SL-I	25-M5-SL-I
Section		6.2	6.2
Data Rate	MB/sec	50	25
Operating Range	m	2 1K	2- 2K
Loss Budget	dB	12	12

8.3.1 Fibre type

The 50μm fibre shall meet the requirements in tables 18, 19, and 20.

Table 19. 50 μm fibre type

Nominal Core Diameter EIA-455-58	Cladding Diameter EIA-455-45 & EIA-455-176 or EIA-455-48	Nominal Numerical Aperture EIA-455-177
50 μm	125 μm	0,20

8.3.2 Bandwidth

The following normalized bandwidth values are based on a nominal source wavelength of 850 and 1 300 nm. (See table 20.)

Table 20. 50 μm multi-mode bandwidth

Wavelength	Modal bandwidth (-3dB optical min)	Test per
850 nm	500MHz*km	EIA-455-30 or EIA-455-51 with EIA-455-54
1 300nm	500MHz*km	EIA-455-30 or EIA-455-51 with EIA-455-54

Note: The bandwidth shown in the table will meet the performance requirements for 780 nm laser operation.

8.3.3 Cable plant loss budget

The loss budget for the 50μm multi-mode fibre cable plant shall be no greater than specified in table 18. These limits were arrived at by taking the difference between the minimum transmitter output power and the receiver sensitivity. This includes the losses of the fiber and other components in the link such as splices and connectors. The connectors at the ends of the links are included in the transmitter and receiver specifications and not in the cable plant limit.

In some cases the modal dispersion limit may be reached in an installaton before the installation loss limit of table 18. See annex D.4 for examples.

The loss of the cable plant shall be verified by the methods of OFSTP-14.

8.3.4 Fibre chromatic dispersion parameters

The effects of chromatic dispersion on system bandwidth are illustrated in figure 27.

Sufficient optical bandwidth is provided when the optical transmitter center wavelength and spectral width are as specified in table 6.

These modal bandwidth and chromatic dispersion requirements in conjunction with the specifications on the center wavelength, spectral width and risetimes of the transmitter in the active output interface specification in section 6 assures a 3 ns exit response time over the respective distance.

8.4 Connectors and splices

Connectors and splices of any nature are allowed inside the cable plant. The number and quality of connections affect the loss of the cable plant and represent a design trade-off outside the scope of the standard.

9 Coaxial cable plant specification

This section defines the link requirements for an FC coaxial cable plant. The term "cable plant" encompasses all the components between any two communicating ports and the associated connector plug at each end of the link.

Performance to this standard shall be met by meeting the data link characteristics defined in table 8 of section 7, the coaxial cable interface specification. A specific goal of the coaxial cable plant is to allow interoperability with restricted lengths of miniature coax cable and CATV style coax. For example, as long as the miniature coax cable is less than 2 m in length for both cabinets, the full 50 m capability of the CATV style coax should be achievable for cabinet interconnections.

Especially at data rates 50 MB/sec or greater, particular attention must be given to the transition board that changes signals from miniature coax to the BNC coaxial connector for the CATV style coax. No more than four miniature coaxial connectors should be present from the ECL transmitter to the receiver of the data link.

Careful attention to details as noted above will allow the system designer to construct a low-cost and yet high performance system capable of reliable error rate performance, The coaxial data link will meet the requirements of many shorter distance Fiber Channel applications.

American National Standard for Information Systems

FC-1 level - Transmission protocol

10 8B/10B Transmission Code

Information to be transmitted over a Fibre shall be encoded eight bits at a time into a 10-bit Transmission Character and then sent serially by bit. Information received over a Fibre shall be collected ten bits at a time, and those Transmission Characters that are used for data, called Data Characters, are decoded into the correct eight-bit codes. The 10-bit Transmission Code supports all 256 eight-bit combinations. Some of the remaining Transmission Characters, referred to as Special Characters, are used for functions which are to be distinguishable from the contents of a frame.

NOTE - The primary rationale for use of a Transmission Code is to improve the transmission characteristics of information to be transferred across a Fibre. The Encodings defined by the Transmission Code ensure that sufficient transitions are present in the serial bit stream to make clock recovery possible at the Receiver. Such Encoding also greatly increases the likelihood of detecting any single or multiple bit errors that may occur during transmission and reception of information. In addition, some of the Special Characters of the Transmission Code contain a distinct and easily recognizable bit pattern (a Comma) which assists a Receiver in achieving word alignment on the incoming bit stream.

10.1 Notation conventions

FC-1 uses letter notation for the bits of an eight-bit byte. Such notation differs from the bit notation specified by the remainder of this standard (see 3.2). The following text describes the translation process between these notations and provides a translation example. It also describes the conventions used to name valid Transmission Characters.

The bit notation of A,B,C,D,E,F,G,H is used for an eight-bit byte. The bits A,B,C,D,E,F,G,H are translated to bits a,b,c,d,e,i,f,g,h,j of 10-bit Transmission Characters. There is a correspondence between bit A and bit a, B and b, C and c, D and d, E and e, F and f, G and g, and H and h. Bits i and j are derived, respectively, from (A,B,C,D,E) and (F,G,H).

The bit labeled A corresponds to bit 0 in the numbering scheme of the FC-2 specification, B corresponds to bit 1, as shown in figure 28.

```
FC-2 bit
designation—▸   7 6 5 4 3 2 1 0

8B/10B bit
designation—▸   H G F E D C B A
```

Figure 28. Bit designations

Figure 29 shows the conversion from an FC-2 Valid Data Byte to a Transmission Character (using 8B/10B Transmission-Code notation).

```
FC-2 byte notation:

   Hex 45
   Bits: 7654 3210
         ---- ----
         0100 0101

Converted to 8B/10B notation
(note carefully that the order of
bits is reversed):

  Bits: ABCDE FGH
        ----- ---
        10100 010

Translated to a 10-bit Transmission Character:

  Bits: abcdei fghj
        ------ ----
        101001 0101
```

Figure 29. Conversion example

Each valid Transmission Character has been given a name using the following convention: zxx.y, where z is used to indicate whether the Transmission Character is a Data Character (z = D) or a Special Character (z = K). When z = D, xx is the decimal value of the binary number composed of the bits E, D, C, B, and A in that order, and y is the decimal value of the binary number composed of the bits H, G, and F in that order. When z = K, xx and y are derived by comparing the encoded bit patterns of the Special Character to those patterns derived from encoded Valid Data Bytes and selecting the names of the patterns most similar to the encoded bit patterns of the Special Character.

Under the above conventions, the Transmission Character in figure 29 is referred to by the name D5.2. The Special Character K29.7 is so named because the first six bits (abcdei) of this character make up a bit pattern similar to that resulting from the Encoding of the unencoded 11101 pattern (29), and because the second four bits (fghj) of this character make up a bit pattern similar to that resulting from the Encoding of the unencoded 111 pattern (7).

NOTE - This definition of the 10-bit Transmission Code is based on (and is in basic agreement with) the following references. Both of these references describe the same 10-bit transmission code.

A. X. Widmer and P. A. Franaszek. "A DC-Balanced, Partitioned-Block, 8B/10B Transmission Code," *IBM Journal of Research and Development, 27*, No. 5: 440-451 (September, 1983).

U.S. Patent 4,486,739. Peter A. Franaszek and Albert X. Widmer. *Byte Oriented DC Balanced (0,4) 8B/10B Partitioned Block Transmission Code* (December 4, 1984).

10.2 Encoding and Decoding

The following information describes how the tables shall be used for both generating valid Transmission Characters (Encoding) and checking the validity of received Transmission Characters (Decoding). It also specifies the ordering rules to be followed when transmitting the bits within a character and the characters within the higher-level constructs specified by the document (i.e., Ordered Sets and frames).

10.2.1 Transmission order

Within the definition of the 8B/10B Transmission Code, the bit positions of the Transmission Characters are labeled a,b,c,d,e,i,f,g,h, and j. Bit "a" shall be transmitted first, followed by bits "b," "c," "d," "e," "i," "f," "g," "h,," and "j," in that order. (Note that bit "i" shall be transmitted between bit "e" and bit "f," rather than in the order that would be indicated by the letters of the alphabet.)

Characters within Ordered Sets (as specified by 10.3) shall be transmitted sequentially beginning with the Special Character used to distinguish the Ordered Set (e.g., K28.5) and proceeding character by character from left to right within the definition of the Ordered Set until all characters of the Ordered Set are transmitted.

The contents of a frame (as specified by FC-2 in 17) shall be transmitted sequentially beginning with the Ordered Set used to denote the start of frame (the SOF delimiter) and proceeding character by character from left to right within the definition of the frame until the Ordered Set used to denote the end of frame (the EOF delimiter) is transmitted.

10.2.2 Valid and invalid Transmission Characters

Tables 21 and 22 define the valid Data Characters and valid Special Characters (K characters), respectively. The tables are used for both generating valid Transmission Characters (Encoding) and checking the validity of received Transmission Characters (Decoding). In the tables, each Valid-Data-Byte or Special-Code entry has two columns that represent two (not necessarily different) Transmission Characters. The two columns correspond to the current value of the Running Disparity ("Current RD –" or "Current RD +"). Running Disparity is a binary parameter with either the value negative (–) or the value positive (+).

After powering on, the Transmitter shall assume the negative value for its initial Running Disparity. Upon transmission of any Transmission Character, the Transmitter shall calculate a new value for its Running Disparity based on the contents of the transmitted character.

After powering on, the Receiver may assume either the positive or negative value for its initial Running Disparity. Upon reception of any Transmission Character, the Receiver shall determine whether the Transmission Character is valid or invalid according to the following rules and tables and shall calculate a new value for its Running Disparity based on the contents of the received character.

The following rules for Running Disparity shall be used to calculate the new Running Disparity value for Transmission Characters that have been transmitted (Transmitter's Running Disparity) and that have been received (Receiver's Running Disparity).

Running Disparity for a Transmission Character shall be calculated on the basis of sub-blocks, where the first six bits (abcdei) form one sub-block (six-bit sub-block) and the second four bits (fghj) form the other sub-block (four-bit sub-block). Running Disparity at the beginning of the six-bit sub-block is the Running Disparity at the end of the last Transmission Character. Running Disparity at the beginning of the four-bit sub-block is the Running Disparity at the end of the six-bit sub-block. Running Disparity at the end of the Transmission Character is the Running Disparity at the end of the four-bit sub-block.

Running Disparity for the sub-blocks shall be calculated as follows:

1. Running Disparity at the end of any sub-block is positive if the sub-block contains more ones than zeros. It is also positive at the end of the six-bit sub-block if the six-bit sub-block is 000111, and it is positive at the end of the four-bit sub-block if the four-bit sub-block is 0011.
2. Running Disparity at the end of any sub-block is negative if the sub-block contains more zeros than ones. It is also negative at the end of the six-bit sub-block if the six-bit sub-block is 111000, and it is negative at the end of the four-bit sub-block if the four-bit sub-block is 1100.
3. Otherwise, Running Disparity at the end of the sub-block is the same as at the beginning of the sub-block.

10.2.2.1 Use of the tables for generating Transmission Characters

The appropriate entry in the table shall be found for the Valid Data Byte or Special Code for which a Transmission Character is to be generated (encoded). The current value of the Transmitter's Running Disparity shall be used to select the Transmission Character from its corresponding column. For each Transmission Character transmitted, a new value of the Running Disparity shall be calculated. This new value shall be used as the Transmitter's Current Running Disparity for the next Valid Data Byte or Special Code to be encoded and transmitted.

10.2.2.2 Use of the tables for checking the validity of received Transmission Characters

The column corresponding to the current value of the Receiver's Running Disparity shall be searched for the received Transmission Character. If the received Transmission Character is found in the proper column, then the Transmission Character shall be considered valid and the associated data byte or Special Code determined (decoded). If the received Transmission Character is not found in that column, then the Transmission Character shall be considered invalid and a Code Violation detected. Independent of the Transmission Character's validity, the received Transmission Character shall be used to calculate a new value of Running Disparity. This new value shall be used as the

Receiver's Current Running Disparity for the next received Transmission Character.

NOTE - Detection of a Code Violation does not necessarily indicate that the Transmission Character in which the Code Violation was detected is in error. Code Violations may result from a prior error which altered the Running Disparity of the bit stream but which did not result in a detectable error at the Transmission Character in which the error occurred. The example shown in figure 30 exhibits this behavior:

	RD	Character	RD	Character	RD	Character	RD
Transmitted character stream	-	D21.1	-	D10.2	-	D23.5	+
Transmitted bit stream	-	101010 1001	-	010101 0101	-	111010 1010	+
Bit stream after error	-	101010 1011	+	010101 0101	+	111010 1010	+
Decoded character stream	-	D21.0	+	D10.2	+	Code Violation	+

Figure 30. Delayed Code Violation example

Table 21 (Page 1 of 3). Valid Data Characters							
Data Byte Name	Bits HGF EDCBA	Current RD - abcdei fghj	Current RD + abcdei fghj	Data Byte Name	Bits HGF EDCBA	Current RD - abcdei fghj	Current RD + abcdei fghj
D0.0	000 00000	100111 0100	011000 1011	D16.1	001 10000	011011 1001	100100 1001
D1.0	000 00001	011101 0100	100010 1011	D17.1	001 10001	100011 1001	100011 1001
D2.0	000 00010	101101 0100	010010 1011	D18.1	001 10010	010011 1001	010011 1001
D3.0	000 00011	110001 1011	110001 0100	D19.1	001 10011	110010 1001	110010 1001
D4.0	000 00100	110101 0100	001010 1011	D20.1	001 10100	001011 1001	001011 1001
D5.0	000 00101	101001 1011	101001 0100	D21.1	001 10101	101010 1001	101010 1001
D6.0	000 00110	011001 1011	011001 0100	D22.1	001 10110	011010 1001	011010 1001
D7.0	000 00111	111000 1011	000111 0100	D23.1	001 10111	111010 1001	000101 1001
D8.0	000 01000	111001 0100	000110 1011	D24.1	001 11000	110011 1001	001100 1001
D9.0	000 01001	100101 1011	100101 0100	D25.1	001 11001	100110 1001	100110 1001
D10.0	000 01010	010101 1011	010101 0100	D26.1	001 11010	010110 1001	010110 1001
D11.0	000 01011	110100 1011	110100 0100	D27.1	001 11011	110110 1001	001001 1001
D12.0	000 01100	001101 1011	001101 0100	D28.1	001 11100	001110 1001	001110 1001
D13.0	000 01101	101100 1011	101100 0100	D29.1	001 11101	101110 1001	010001 1001
D14.0	000 01110	011100 1011	011100 0100	D30.1	001 11110	011110 1001	100001 1001
D15.0	000 01111	010111 0100	101000 1011	D31.1	001 11111	101011 1001	010100 1001
D16.0	000 10000	011011 0100	100100 1011	D0.2	010 00000	100111 0101	011000 0101
D17.0	000 10001	100011 1011	100011 0100	D1.2	010 00001	011101 0101	100010 0101
D18.0	000 10010	010011 1011	010011 0100	D2.2	010 00010	101101 0101	010010 0101
D19.0	000 10011	110010 1011	110010 0100	D3.2	010 00011	110001 0101	110001 0101
D20.0	000 10100	001011 1011	001011 0100	D4.2	010 00100	110101 0101	001010 0101
D21.0	000 10101	101010 1011	101010 0100	D5.2	010 00101	101001 0101	101001 0101
D22.0	000 10110	011010 1011	011010 0100	D6.2	010 00110	011001 0101	011001 0101
D23.0	000 10111	111010 0100	000101 1011	D7.2	010 00111	111000 0101	000111 0101
D24.0	000 11000	110011 0100	001100 1011	D8.2	010 01000	111001 0101	000110 0101
D25.0	000 11001	100110 1011	100110 0100	D9.2	010 01001	100101 0101	100101 0101
D26.0	000 11010	010110 1011	010110 0100	D10.2	010 01010	010101 0101	010101 0101
D27.0	000 11011	110110 0100	001001 1011	D11.2	010 01011	110100 0101	110100 0101
D28.0	000 11100	001110 1011	001110 0100	D12.2	010 01100	001101 0101	001101 0101
D29.0	000 11101	101110 0100	010001 1011	D13.2	010 01101	101100 0101	101100 0101
D30.0	000 11110	011110 0100	100001 1011	D14.2	010 01110	011100 0101	011100 0101
D31.0	000 11111	101011 0100	010100 1011	D15.2	010 01111	010111 0101	101000 0101
D0.1	001 00000	100111 1001	011000 1001	D16.2	010 10000	011011 0101	100100 0101
D1.1	001 00001	011101 1001	100010 1001	D17.2	010 10001	100011 0101	100011 0101
D2.1	001 00010	101101 1001	010010 1001	D18.2	010 10010	010011 0101	010011 0101
D3.1	001 00011	110001 1001	110001 1001	D19.2	010 10011	110010 0101	110010 0101
D4.1	001 00100	110101 1001	001010 1001	D20.2	010 10100	001011 0101	001011 0101
D5.1	001 00101	101001 1001	101001 1001	D21.2	010 10101	101010 0101	101010 0101
D6.1	001 00110	011001 1001	011001 1001	D22.2	010 10110	011010 0101	011010 0101
D7.1	001 00111	111000 1001	000111 1001	D23.2	010 10111	111010 0101	000101 0101
D8.1	001 01000	111001 1001	000110 1001	D24.2	010 11000	110011 0101	001100 0101
D9.1	001 01001	100101 1001	100101 1001	D25.2	010 11001	100110 0101	100110 0101
D10.1	001 01010	010101 1001	010101 1001	D26.2	010 11010	010110 0101	010110 0101
D11.1	001 01011	110100 1001	110100 1001	D27.2	010 11011	110110 0101	001001 0101
D12.1	001 01100	001101 1001	001101 1001	D28.2	010 11100	001110 0101	001110 0101
D13.1	001 01101	101100 1001	101100 1001	D29.2	010 11101	101110 0101	010001 0101
D14.1	001 01110	011100 1001	011100 1001	D30.2	010 11110	011110 0101	100001 0101
D15.1	001 01111	010111 1001	101000 1001	D31.2	010 11111	101011 0101	010100 0101

Table 21 (Page 2 of 3). Valid Data Characters

Data Byte Name	Bits HGF EDCBA	Current RD - abcdei fghj	Current RD + abcdei fghj	Data Byte Name	Bits HGF EDCBA	Current RD - abcdei fghj	Current RD + abcdei fghj
D0.3	011 00000	100111 0011	011000 1100	D16.4	100 10000	011011 0010	100100 1101
D1.3	011 00001	011101 0011	100010 1100	D17.4	100 10001	100011 1101	100011 0010
D2.3	011 00010	101101 0011	010010 1100	D18.4	100 10010	010011 1101	010011 0010
D3.3	011 00011	110001 1100	110001 0011	D19.4	100 10011	110010 1101	110010 0010
D4.3	011 00100	110101 0011	001010 1100	D20.4	100 10100	001011 1101	001011 0010
D5.3	011 00101	101001 1100	101001 0011	D21.4	100 10101	101010 1101	101010 0010
D6.3	011 00110	011001 1100	011001 0011	D22.4	100 10110	011010 1101	011010 0010
D7.3	011 00111	111000 1100	000111 0011	D23.4	100 10111	111010 0010	000101 1101
D8.3	011 01000	111001 0011	000110 1100	D24.4	100 11000	110011 0010	001100 1101
D9.3	011 01001	100101 1100	100101 0011	D25.4	100 11001	100110 1101	100110 0010
D10.3	011 01010	010101 1100	010101 0011	D26.4	100 11010	010110 1101	010110 0010
D11.3	011 01011	110100 1100	110100 0011	D27.4	100 11011	110110 0010	001001 1101
D12.3	011 01100	001101 1100	001101 0011	D28.4	100 11100	001110 1101	001110 0010
D13.3	011 01101	101100 1100	101100 0011	D29.4	100 11101	101110 0010	010001 1101
D14.3	011 01110	011100 1100	011100 0011	D30.4	100 11110	011110 0010	100001 1101
D15.3	011 01111	010111 0011	101000 1100	D31.4	100 11111	101011 0010	010100 1101
D16.3	011 10000	011011 0011	100100 1100	D0.5	101 00000	100111 1010	011000 1010
D17.3	011 10001	100011 1100	100011 0011	D1.5	101 00001	011101 1010	100010 1010
D18.3	011 10010	010011 1100	010011 0011	D2.5	101 00010	101101 1010	010010 1010
D19.3	011 10011	110010 1100	110010 0011	D3.5	101 00011	110001 1010	110001 1010
D20.3	011 10100	001011 1100	001011 0011	D4.5	101 00100	110101 1010	001010 1010
D21.3	011 10101	101010 1100	101010 0011	D5.5	101 00101	101001 1010	101001 1010
D22.3	011 10110	011010 1100	011010 0011	D6.5	101 00110	011001 1010	011001 1010
D23.3	011 10111	111010 0011	000101 1100	D7.5	101 00111	111000 1010	000111 1010
D24.3	011 11000	110011 0011	001100 1100	D8.5	101 01000	111001 1010	000110 1010
D25.3	011 11001	100110 1100	100110 0011	D9.5	101 01001	100101 1010	100101 1010
D26.3	011 11010	010110 1100	010110 0011	D10.5	101 01010	010101 1010	010101 1010
D27.3	011 11011	110110 0011	001001 1100	D11.5	101 01011	110100 1010	110100 1010
D28.3	011 11100	001110 1100	001110 0011	D12.5	101 01100	001101 1010	001101 1010
D29.3	011 11101	101110 0011	010001 1100	D13.5	101 01101	101100 1010	101100 1010
D30.3	011 11110	011110 0011	100001 1100	D14.5	101 01110	011100 1010	011100 1010
D31.3	011 11111	101011 0011	010100 1100	D15.5	101 01111	010111 1010	101000 1010
D0.4	100 00000	100111 0010	011000 1101	D16.5	101 10000	011011 1010	100100 1010
D1.4	100 00001	011101 0010	100010 1101	D17.5	101 10001	100011 1010	100011 1010
D2.4	100 00010	101101 0010	010010 1101	D18.5	101 10010	010011 1010	010011 1010
D3.4	100 00011	110001 1101	110001 0010	D19.5	101 10011	110010 1010	110010 1010
D4.4	100 00100	110101 0010	001010 1101	D20.5	101 10100	001011 1010	001011 1010
D5.4	100 00101	101001 1101	101001 0010	D21.5	101 10101	101010 1010	101010 1010
D6.4	100 00110	011001 1101	011001 0010	D22.5	101 10110	011010 1010	011010 1010
D7.4	100 00111	111000 1101	000111 0010	D23.5	101 10111	111010 1010	000101 1010
D8.4	100 01000	111001 0010	000110 1101	D24.5	101 11000	110011 1010	001100 1010
D9.4	100 01001	100101 1101	100101 0010	D25.5	101 11001	100110 1010	100110 1010
D10.4	100 01010	010101 1101	010101 0010	D26.5	101 11010	010110 1010	010110 1010
D11.4	100 01011	110100 1101	110100 0010	D27.5	101 11011	110110 1010	001001 1010
D12.4	100 01100	001101 1101	001101 0010	D28.5	101 11100	001110 1010	001110 1010
D13.4	100 01101	101100 1101	101100 0010	D29.5	101 11101	101110 1010	010001 1010
D14.4	100 01110	011100 1101	011100 0010	D30.5	101 11110	011110 1010	100001 1010
D15.4	100 01111	010111 0010	101000 1101	D31.5	101 11111	101011 1010	010100 1010

Table 21 (Page 3 of 3). Valid Data Characters							
Data Byte Name	Bits HGF EDCBA	Current RD - abcdei fghj	Current RD + abcdei fghj	Data Byte Name	Bits HGF EDCBA	Current RD - abcdei fghj	Current RD + abcdei fghj
D0.6	110 00000	100111 0110	011000 0110	D0.7	111 00000	100111 0001	011000 1110
D1.6	110 00001	011101 0110	100010 0110	D1.7	111 00001	011101 0001	100010 1110
D2.6	110 00010	101101 0110	010010 0110	D2.7	111 00010	101101 0001	010010 1110
D3.6	110 00011	110001 0110	110001 0110	D3.7	111 00011	110001 1110	110001 0001
D4.6	110 00100	110101 0110	001010 0110	D4.7	111 00100	110101 0001	001010 1110
D5.6	110 00101	101001 0110	101001 0110	D5.7	111 00101	101001 1110	101001 0001
D6.6	110 00110	011001 0110	011001 0110	D6.7	111 00110	011001 1110	011001 0001
D7.6	110 00111	111000 0110	000111 0110	D7.7	111 00111	111000 1110	000111 0001
D8.6	110 01000	111001 0110	000110 0110	D8.7	111 01000	111001 0001	000110 1110
D9.6	110 01001	100101 0110	100101 0110	D9.7	111 01001	100101 1110	100101 0001
D10.6	110 01010	010101 0110	010101 0110	D10.7	111 01010	010101 1110	010101 0001
D11.6	110 01011	110100 0110	110100 0110	D11.7	111 01011	110100 1110	110100 1000
D12.6	110 01100	001101 0110	001101 0110	D12.7	111 01100	001101 1110	001101 0001
D13.6	110 01101	101100 0110	101100 0110	D13.7	111 01101	101100 1110	101100 1000
D14.6	110 01110	011100 0110	011100 0110	D14.7	111 01110	011100 1110	011100 1000
D15.6	110 01111	010111 0110	101000 0110	D15.7	111 01111	010111 0001	101000 1110
D16.6	110 10000	011011 0110	100100 0110	D16.7	111 10000	011011 0001	100100 1110
D17.6	110 10001	100011 0110	100011 0110	D17.7	111 10001	100011 0111	100011 0001
D18.6	110 10010	010011 0110	010011 0110	D18.7	111 10010	010011 0111	010011 0001
D19.6	110 10011	110010 0110	110010 0110	D19.7	111 10011	110010 1110	110010 0001
D20.6	110 10100	001011 0110	001011 0110	D20.7	111 10100	001011 0111	001011 0001
D21.6	110 10101	101010 0110	101010 0110	D21.7	111 10101	101010 1110	101010 0001
D22.6	110 10110	011010 0110	011010 0110	D22.7	111 10110	011010 1110	011010 0001
D23.6	110 10111	111010 0110	000101 0110	D23.7	111 10111	111010 0001	000101 1110
D24.6	110 11000	110011 0110	001100 0110	D24.7	111 11000	110011 0001	001100 1110
D25.6	110 11001	100110 0110	100110 0110	D25.7	111 11001	100110 1110	100110 0001
D26.6	110 11010	010110 0110	010110 0110	D26.7	111 11010	010110 1110	010110 0001
D27.6	110 11011	110110 0110	001001 0110	D27.7	111 11011	110110 0001	001001 1110
D28.6	110 11100	001110 0110	001110 0110	D28.7	111 11100	001110 1110	001110 0001
D29.6	110 11101	101110 0110	010001 0110	D29.7	111 11101	101110 0001	010001 1110
D30.6	110 11110	011110 0110	100001 0110	D30.7	111 11110	011110 0001	100001 1110
D31.6	110 11111	101011 0110	010100 0110	D31.7	111 11111	101011 0001	010100 1110

Table 22. Valid Special Characters			
Special Code Name	Current RD - abcdei fghj	Current RD + abcdei fghj	Notes
K28.0	001111 0100	110000 1011	Reserved
K28.1	001111 1001	110000 0110	Reserved
K28.2	001111 0101	110000 1010	Reserved
K28.3	001111 0011	110000 1100	Reserved
K28.4	001111 0010	110000 1101	Reserved
K28.5	001111 1010	110000 0101	
K28.6	001111 0110	110000 1001	Reserved
K28.7	001111 1000	110000 0111	Reserved
K23.7	111010 1000	000101 0111	Reserved
K27.7	110110 1000	001001 0111	Reserved
K29.7	101110 1000	010001 0111	Reserved
K30.7	011110 1000	100001 0111	Reserved

Explanation:

Reserved - valid Transmission Characters which are not defined for use by this standard

The K28.7 Special Character shall not be followed by any of the following Special or Data Characters: K28.x, D3.x, D11.x, D12.x, D19.x, D20.x, or D28.x, where x is a value in the range 0 to 7, inclusive.

NOTE - The above restriction simplifies word synchronization hardware.

10.3 Ordered Sets

Tables 23, 24, and 25 specify the Ordered Sets (composed of Special and Data Characters) which are defined for use by this standard. The following Ordered Set types are defined by FC-2 in 16:

— Delimiter Functions
— Primitive Signals
— Primitive Sequences

Table 23. Delimiter Functions

Delimiter Function	Beginning RD	Ordered Set
SOF Connect Class 1 (SOFc1)	Negative	K28.5 D21.5 D23.0 D23.0
SOF Initiate Class 1 (SOFi1)	Negative	K28.5 D21.5 D23.2 D23.2
SOF Normal Class 1 (SOFn1)	Negative	K28.5 D21.5 D23.1 D23.1
SOF Initiate Class 2 (SOFi2)	Negative	K28.5 D21.5 D21.2 D21.2
SOF Normal Class 2 (SOFn2)	Negative	K28.5 D21.5 D21.1 D21.1
SOF Initiate Class 3 (SOFi3)	Negative	K28.5 D21.5 D22.2 D22.2
SOF Normal Class 3 (SOFn3)	Negative	K28.5 D21.5 D22.1 D22.1
SOF Fabric (SOFf)	Negative	K28.5 D21.5 D24.2 D24.2
EOF Terminate (EOFt)	Negative Positive	K28.5 D21.4 D21.3 D21.3 K28.5 D21.5 D21.3 D21.3
EOF Disconnect-Terminate (EOFdt)	Negative Positive	K28.5 D21.4 D21.4 D21.4 K28.5 D21.5 D21.4 D21.4
EOF Abort (EOFa)	Negative Positive	K28.5 D21.4 D21.7 D21.7 K28.5 D21.5 D21.7 D21.7
EOF Normal (EOFn)	Negative Positive	K28.5 D21.4 D21.6 D21.6 K28.5 D21.5 D21.6 D21.6
EOF Disconnect-Terminate-Invalid (EOFdti)	Negative Positive	K28.5 D10.4 D21.4 D21.4 K28.5 D10.5 D21.4 D21.4
EOF Normal-Invalid (EOFni)	Negative Positive	K28.5 D10.4 D21.6 D21.6 K28.5 D10.5 D21.6 D21.6

Explanation:

SOF - Start-of-frame delimiter
EOF - End-of-frame delimiter

Table 24. Primitive Signals

Primitive Signal	Beginning RD	Ordered Set
Idle Word	Negative	K28.5 D21.4 D21.5 D21.5
Receiver_Ready (R_RDY)	Negative	K28.5 D21.4 D10.2 D10.2

Table 25. Primitive Sequences		
Primitive Sequence	**Beginning RD**	**Ordered Set**
Offline (OLS)	Negative	K28.5 D21.1 D10.4 D21.2
Not_Operational (NOS)	Negative	K28.5 D21.2 D31.5 D5.2
Link_Reset (LR)	Negative	K28.5 D9.2 D31.5 D9.2
Link_Reset_Response (LRR)	Negative	K28.5 D21.1 D31.5 D9.2

NOTES

1. Each EOF-delimiter Ordered Set is defined such that negative Current Running Disparity will result after processing of the final (right-most) character of the Ordered Set. This, in combination with the Running Disparity Initialization rules specified in 10.2.2, ensures that the first Ordered Set following an EOF delimiter or Transmitter power on will always be transmitted with negative Beginning Running Disparity. The Ordered Sets defined for the primitive signals and primitive sequences preserve this negative Disparity, ensuring that the Ordered Sets associated with SOF delimiters, primitive signals, and primitive sequences will also always be transmitted with negative Beginning Running Disparity. As a result, primitive signal, primitive sequence, and SOF delimiter Ordered Sets are defined for the negative Beginning Running Disparity case only. (The primary benefit of such a definition is that it allows idle words to be removed and added from an encoded bit stream one word at a time without altering the Beginning Running Disparity associated with the Transmission Word subsequent to the removed idle word.)

2. The K28.5 Special Character is chosen as the first character of all Ordered Sets for the following reasons:

   - Bits abcdeif make up a Comma; this is a singular bit pattern which in the absence of transmission errors cannot appear in any other location of a Transmission Character and cannot be generated across the boundaries of any two adjacent Transmission Characters. The Comma can be used to easily find and verify character and word boundaries of the received bit stream.

   - The encoded character presents a high number of transitions, simplifying Receiver acquisition of bit synchronization.

3. The second character of the Ordered Sets used to represent EOF delimiters differentiates between normal and invalid frames. It also ensures that the Running Disparity resulting after processing of an EOF Ordered Set is negative independent of the value of Beginning Running Disparity. In all other Ordered Sets, it serves only as a placeholder which provides a high number of bit transitions.

4. The third and fourth characters of the delimiter functions, Receiver_Ready, and the idle word are repeated to ensure that an error affecting a single character will not result in the recognition of an Ordered Set other than the one transmitted. The third and fourth characters of the other Ordered Sets defined by the standard have been selected to provide distinct patterns which provide a large number of transitions and good spectral characteristics.

5. The categories of Delimiter, Primitive Signal, and Primitive Sequence have meaning to FC-2 and are defined by FC-2. FC-1 recognizes the character combinations defined in this section as Ordered Sets only.

6. More Ordered Sets may be defined as the standard progresses.

11 State description

11.1 Receiver state description

Whenever a signal (as defined by FC-0 in 5.6) is present on a Fibre, the Receiver attached to that Fibre shall attempt to achieve synchronization on both bit and Transmission-Word boundaries of the received encoded bit stream. Bit synchronization is defined by FC-0 in 5.5. Transmission-Word synchronization is defined in 11.1.2.2. Synchronization failures on either bit or Transmission-Word boundaries are not separately identifiable and cause loss-of-synchronization errors. Such failures may ultimately cause a Receiver to become Not Operational.

11.1.1 Receiver states

The Receiver state diagram is shown in figure 32.

11.1.1.1 Operational states

Synchronization to Transmission-Word boundaries, hereafter referred to as *Synchronization*, is achieved when the Receiver has identified the Transmission-Word boundary as the one that is established by the input signal from the Transmitter at the other end of the Fibre. The procedure used to achieve this condition is described in 11.1.2.2. When this condition is achieved, the Receiver shall enter the *Synchronization-Acquired* state. A Receiver in the Synchronization-Acquired state shall provide to the FC-2 level information that has been received from its attached Fibre and decoded.

When the Transmission-Word boundary identified by the Receiver no longer matches the boundary established by the Transmitter to which it is connected, the Receiver shall enter the *Loss-Of-Synchronization* state. Such a Receiver shall remain Operational but shall cease to provide received and decoded information to the FC-2 level. When the Receiver is in the Loss-Of-Synchronization state, the procedure described in 11.1.2 shall be used to regain Synchronization.

When the Receiver is in the Synchronization-Acquired state and receives a request to enter the Offline state from FC-2, it shall enter the *Offline* state. (See 27 for a discussion of Primitive Sequences and their relationship to the Offline state.) Such a Receiver shall remain Operational and shall provide to the FC-2 level information that has been received from its attached Fibre and decoded. It shall also continue to process received Transmission Words under the rules specified by "Invalid Transmission Word rules." See 11.1.4 for additional details.

When the Receiver is Operational, it is always in one of the three states described above. The determination of an Operational Receiver's state is based on the conditions described in 11.1.2 and 11.1.3 and upon the presence or absence of the FC-2 requests associated with entry and exit of the Offline state.

11.1.1.2 Not Operational states

If an Operational Receiver remains in the Loss-Of-Synchronization state for longer than an extended period of time as defined by FC-0 in 5.6, an FC-1-level-Receiver-failure condition shall be recognized and the Receiver shall become Not Operational. If a loss-of-signal condition is not detected by the FC-0 Receiver at the time of failure recognition, this condition shall be reported to FC-2 as a loss-of-Synchronization failure condition and the Receiver shall enter the *Loss-Of-Synchronization-Failure* state. If a loss-of-signal condition is detected by the FC-0 Receiver at the time of failure recognition, this condition shall be reported to FC-2 as a loss-of-signal failure condition and the Receiver shall enter the *Loss-Of-Signal-Failure* state. (Signal detection is discussed by FC-0 in 5.6.)

A Receiver shall become Not Operational and shall enter the *Reset* state when a reset condition is imposed upon it, either internally or externally.

11.1.2 Entry into Synchronization-Acquired state

A Receiver shall enter into the Synchronization-Acquired state when it has achieved both bit and Transmission-Word synchronization. It shall also enter the Synchronization-Acquired state upon receipt of a request from FC-2 to exit the Offline state while currently in that state as the result of a prior FC-2 request. (See 27 for a discussion of Primitive Sequences and their relationship to the Offline state.)

11.1.2.1 Bit synchronization

An Operational Receiver that is in the Loss-Of-Synchronization state shall first acquire bit synchronization before attempting to acquire Transmission-Word synchronization. Bit synchronization is defined by FC-0 in 5.5. After achieving bit synchronization, the Receiver shall remain in the Loss-Of-Synchronization state until it achieves Transmission-Word synchronization.

11.1.2.2 Transmission-Word synchronization

When an Operational Receiver that is in the Loss-Of-Synchronization state and that has acquired bit synchronization recognizes three Ordered Sets, as specified in tables 23, 24, and 25, without an intervening invalid Transmission Word, as specified by "Invalid Transmission Word rules," the Receiver shall enter the Synchronization-Acquired state. The third recognized Ordered Set shall be considered valid information and shall be decoded and provided to the FC-2 level. Ordered Set recognition shall include not only recognition of the individual characters that make up an Ordered Set but also proper alignment of those characters (i.e., the Special Character used to designate an Ordered Set shall be aligned in the leading character position of the received Transmission Word). The method used by the Receiver to align and recognize Ordered Sets is not defined by this standard.

NOTES

1. The Comma contained within the K28.5 Special Character is a singular bit pattern which in the absence of transmission errors cannot appear in any other location of a Transmission Character and cannot be generated across the boundaries of any two adjacent Transmission Characters. This bit pattern is sufficient to identify the word alignment of the received bit stream.

2. The recommended electrical interface allows the word alignment function to be enabled in either of two modes:

   - Continuously-enabled word alignment (continuous-mode alignment)

   - Explicitly-enabled word alignment (explicit-mode alignment)

   Continuous-mode alignment allows the FC-1 Receiver to reestablish word alignment at any point in the incoming bit stream while the Receiver is Operational. Such realignment is likely (but not guaranteed) to result in Code Violations and subsequent loss of Synchronization. Under certain conditions, it may be possible to realign an incoming bit stream without loss of Synchronization. If such a realignment occurs within a received frame, detection of the resulting error condition is dependent upon FC-2 function (e.g., invalid CRC, missing EOF Delimiter).

   Explicit-mode alignment allows the FC-1 Receiver to reestablish word alignment only while in the Loss-Of-Synchronization state or when Synchronization has been lost in the Offline state. Once Synchronization has been acquired, the word alignment function of the Receiver is disabled.

11.1.3 Entry into Loss-Of-Synchronization state

The four conditions which shall cause an Operational Receiver to enter the Loss-Of-Synchronization state are:

- Completion of the Loss-Of-Synchronization procedure
- Transition to power on
- Exit from Receiver reset condition
- Detection of loss of signal

NOTE - While in the Loss-Of-Synchronization state, the FC-1 Receiver may request that the FC-0 Receiver attempt bit resynchronization. In some instances, this may allow the FC-1 Receiver to regain Synchronization and enter the Synchronization-Acquired state when it otherwise would be forced to enter a Not Operational state. However, initiation of bit resynchronization at the FC-0 Receiver may also delay the resynchronization process by forcing the FC-0 Receiver to reestablish a clock reference when such reestablishment is otherwise unnecessary. See 5.5 in FC-0 for a detailed discussion of bit synchronization.

11.1.3.1 Loss-of-Synchronization procedure

The loss-of-Synchronization procedure defines the method by which the Receiver changes from the Synchronization-Acquired state to the Loss-Of-Synchronization state. The procedure tests each received Transmission Word according to the rules defined in "Invalid Transmission Word rules" to determine the validity of the Transmission Word.

Starting in the Synchronization-Acquired state, the Receiver shall check each received Transmission Word according to the rules of "Invalid Transmission Word rules" to determine if the word is valid. The Receiver shall continue to check the Transmission Words and shall remain in the Synchronization-Acquired state until the loss-of-Synchronization procedure, as described by "Loss-of-Synchronization-procedure states," is completed.

Invalid Transmission Word rules

An invalid Transmission Word shall be recognized by the Receiver when one of the following conditions is detected:

- A Code Violation on a character basis, as specified by tables 21 and 22, is detected within a Transmission Word. This is referred to as a Code Violation condition.

- Any valid Special Character is detected in the second, third, or fourth character position of a Transmission Word. This is referred to as an invalid Special Code alignment condition.

- Any valid Special Character other than K28.5 is detected in the first (leftmost) character position of a Transmission Word. This is referred to as a reserved Special Code Violation condition.

Loss-of-Synchronization-procedure states

The following five detection states are defined as part of the loss-of-Synchronization procedure:

1. No invalid Transmission Word has been detected (the No-Invalid-Transmission-Word detection state).
2. The first invalid Transmission Word is detected (the First-Invalid-Transmission-Word detection state).
3. The second invalid Transmission Word is detected (the Second-Invalid-Transmission-Word detection state).
4. The third invalid Transmission Word is detected (the Third-Invalid-Transmission-Word detection state).
5. The fourth invalid Transmission Word is detected (the Fourth-Invalid-Transmission-Word detection state).

A Receiver in the Synchronization-Acquired state may be in any of the first four detection states listed above. A Receiver in the fifth detection state listed above (the Fourth-Invalid-Transmission-Word detection state) shall enter the Loss-Of-Synchronization state.

When the procedure is in detection state 1, checking for an invalid Transmission Word shall be performed. After each invalid Transmission Word is detected, one of the detection states 2 through 5 shall be entered. When the procedure is in detection state 2, 3, or 4, checking for additional invalid Transmission Words shall be performed in groups of two consecutive Transmission Words. If two consecutive valid Transmission Words are received, the count of previously detected invalid Transmission Words shall be reduced by one, and the previous detection state shall be entered. The loss-of-Synchronization procedure is completed when detection state 5 is entered.

The No-Invalid-Transmission-Word detection state shall be entered on a transition to the Synchronization-Acquired state due to acquisition of Synchronization, or after the First-Invalid-Transmission-Word detection state is reset.

The First-Invalid-Transmission-Word detection state shall be entered after the first invalid Transmission Word is detected or after the previous detection state (detection of the second invalid Transmission Word) is reset. Subsequent Transmission Words received shall be checked to determine whether an additional invalid Transmission Word is detected within the next consecutive two or fewer Transmission Words received. If an additional invalid Transmission Word is detected within the next consecutive two or fewer Transmission Words received, the Second-Invalid-Transmission-Word detection state shall be entered. If two consecutive valid Transmission Words are received, the No-

Invalid-Transmission-Word detection state shall be entered.

The Second-Invalid-Transmission-Word detection state shall be entered after the second invalid Transmission Word is detected or after the previous detection state (detection of the third invalid Transmission Word) is reset. Subsequent Transmission Words received shall be checked to determine whether an additional invalid Transmission Word is detected within the next consecutive two or fewer Transmission Words received. If an additional invalid Transmission Word is detected within the next consecutive two or fewer Transmission Words received, the Third-Invalid-Transmission-Word detection state shall be entered. If two consecutive valid Transmission Words are received, the First-Invalid-Transmission-Word detection state shall be entered.

The Third-Invalid-Transmission-Word detection state shall be entered after the third invalid Transmission Word is detected. Subsequent Transmission Words received shall be checked to determine whether an additional invalid Transmission Word is detected within the next consecutive two or fewer Transmission Words received. If an additional invalid Transmission Word is detected within the next consecutive two or fewer Transmission Words received, the Fourth-Invalid-Transmission-Word detection state shall be entered and the Receiver shall immediately enter the Loss-Of-Synchronization state. If two consecutive valid Transmission Words are received, the Second-Invalid-Transmission-Word detection state shall be entered.

The Fourth-Invalid-Transmission-Word detection state shall be entered after the fourth invalid Transmission Word is detected. Upon entering this detection state, the Receiver shall immediately enter the Loss-Of-Synchronization state. Subsequent Transmission Words received shall not be checked as the loss-of-Synchronization procedure is complete. The Receiver shall remain in the Loss-Of-Synchronization state until one of the following occurs:

- The Receiver regains Synchronization
- The Receiver is reset
- The Receiver recognizes an FC-1-level-Receiver-failure condition, becomes Not Operational, and enters one of the two failure states (Loss-Of-Synchronization-Failure or Loss-Of-Signal-Failure)

The following figure graphically portrays the Loss-of-Synchronization procedure. States 1 through 5 are keyed to the detection states described by the ordered list at the beginning of this section. State 1 is the initial detection state entered by a Receiver upon acquisition of Synchronization. States 1 through 4 are detection states which are possible only when the Receiver has achieved Synchronization. Entry into State 5 results in loss of Receiver Synchronization.

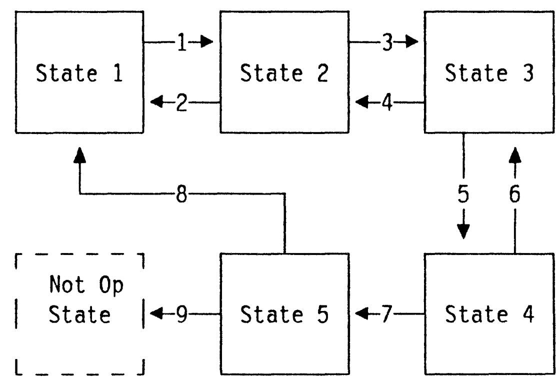


Figure 31. Loss-of-Synchronization procedure state diagram

State Transition Conditions:

1. The first invalid Transmission Word is detected
2. An additional invalid Transmission Word is not detected in the next two consecutive Transmission Words
3. An additional invalid Transmission Word is detected in the next two consecutive Transmission Words
4. An additional invalid Transmission Word is not detected in the next two consecutive Transmission Words
5. An additional invalid Transmission Word is detected in the next two consecutive Transmission Words
6. An additional invalid Transmission Word is not detected in the next two consecutive Transmission Words
7. An additional invalid Transmission Word is detected in the next two consecutive Transmission Words

8. The Receiver regains Synchronization
9. The Receiver is Reset or recognizes an FC-1-level-Receiver-failure condition

NOTE - The rationale for the loss-of-Synchronization procedure is to reduce the likelihood that a single error will result in a loss of Synchronization. For example, a single two-bit error positioned to overlap two Transmission Words could result in the detection of three invalid Transmission Words; the two Transmission Words directly affected by the error and a subsequent Transmission Word which was affected by a disparity change resulting from the error. The procedure described above would maintain Synchronization in such a case.

11.1.3.2 Transition to power on

When the Receiver is Operational after being powered on, the Receiver shall enter the Loss-Of-Synchronization state.

11.1.3.3 Exit From Receiver reset condition

When the Receiver is Operational after exiting from a Receiver reset condition imposed upon it, either externally or internally, and was not previously in the Offline state, the Receiver shall enter the Loss-Of-Synchronization state.

NOTE - The conditions required for a Receiver in the Reset state to exit that state are not defined by this standard. Such conditions may be based on explicit indications. They may also be time-dependent in nature.

11.1.3.4 Detection of loss of signal

When a loss-of-signal condition is recognized by an Operational Receiver, the Loss-Of-Synchronization state shall be entered (if the Receiver is not presently in that state). The Receiver shall remain in this state until one of the following conditions occur:

- The loss-of-signal condition is corrected and Synchronization is regained
- The Receiver is reset
- The Receiver recognizes an FC-1-level-Receiver-failure condition, becomes Not Operational, and enters the Loss-Of-Signal-Failure state

11.1.4 Entry into Offline state

When a request to enter the Offline state is received from FC-2 by an Operational FC-1 Receiver, the Offline state shall be entered. The Receiver shall remain in the Offline state until it receives a request from FC-2 to exit the Offline state. (See 27 for a discussion of Primitive Sequences and their relationship to the Offline state.) This request may take the form of a request to return to the Synchronization-Acquired state or a reset request. A Receiver in the Offline state that is reset shall return to the Offline state upon completion of the reset.

While in the Offline state, the FC-1 Receiver shall remain Operational and shall provide received and decoded information to the FC-2 level according to the rules specified by "Invalid Transmission Word rules." The Receiver shall also continue to process received Transmission Words under the rules of the Loss-of-Synchronization procedure specified in "Invalid Transmission Word rules." Consequently, while in the Offline state, the Receiver may lose Synchronization. In addition, it may also detect a loss of signal condition. Such conditions shall not cause the Receiver to leave the Offline state. Instead, a Receiver in the Offline state which encounters such conditions shall attempt to reacquire Synchronization according to the rules described in 11.1.2 when such conditions occur.

11.1.5 Entry into Not Operational states

The two conditions which shall cause an Operational Receiver to enter one of the Not Operational states are:

- Recognition of FC-1-Level-Receiver-Failure condition
- Imposition of Receiver reset condition

11.1.5.1 Recognition of FC-1-level-Receiver-failure condition

A Receiver shall enter into the Loss-Of-Synchronization-Failure or Loss-Of-Signal-Failure state based upon the conditions present at the time of FC-1-level-Receiver-failure condition recognition. Once one of these failure states is entered, the Receiver shall become Not Operational and shall remain in a failure state until it is subsequently made Operational through external intervention. Transitions between the Loss-Of-Synchronization-Failure and Loss-Of-

Signal-Failure states may occur as the result of changes to the output of the FC-0 signal detection function.

Signal detection is an optional function of FC-0. When a failure condition is recognized by an FC-1 Receiver and the signal detection function is not provided by FC-0, the Loss-Of-Synchronization-Failure state shall be entered. (Signal detection is discussed by FC-0 in 5.6.)

11.1.5.2 Imposition of Receiver reset condition

When a Receiver reset condition is imposed on a Receiver, either internally or externally, the Receiver shall enter the Reset state. Once the Reset state is entered, the Receiver shall become Not Operational (if not already Not Operational as a result of previously entering one of the failure states) and shall remain in the Reset state until it is subsequently made Operational by exiting the Receiver reset condition.

NOTES

1. A typical use of Receiver reset is to force a Receiver in the Loss-Of-Synchronization or Loss-Of-Synchronization-Failure state to attempt reacquisition of bit synchronization. (Note that entry into either of these states does not necessarily indicate loss of bit synchronization.) As such, an FC-1 Receiver reset may result in a request for bit resynchronization to the FC-0 Receiver.
2. The conditions required for a Receiver in the Reset state to exit that state are not defined by this standard. Such conditions may be based on explicit indications. They may also be time-dependent in nature.

11.2 Receiver state diagram

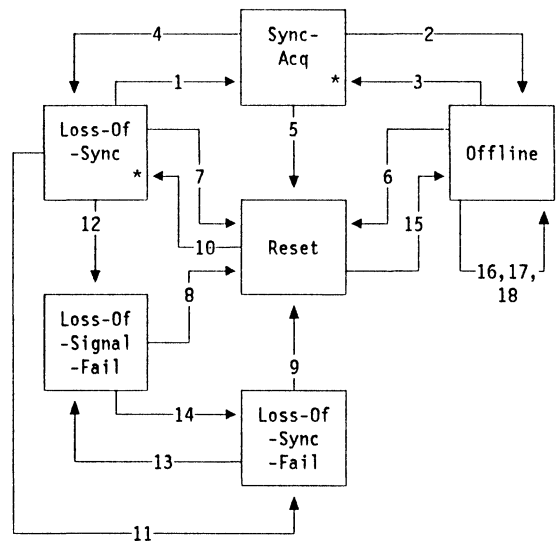


Figure 32. FC-1 Receiver state diagram

State Transition Conditions:

1. Acquisition of Synchronization (see 11.1.2)
2. Receipt of Receiver Offline request from FC-2 (see 11.1.4)
3. Receipt of Receiver request to return to Synchronization-Acquired state from FC-2 (see 11.1.2)
4. Completion of loss-of-Synchronization procedure (see 11.1.3.1)
5. Receipt of Receiver reset request from FC-2 (see 11.1.5.2)
6. Receipt of Receiver reset request from FC-2 (see 11.1.5.2)
7. Receipt of Receiver reset request from FC-2 (see 11.1.5.2)
8. Receipt of Receiver reset request from FC-2 (see 11.1.5.2)
9. Receipt of Receiver reset request from FC-2 (see 11.1.5.2)
10. Exiting of Receiver reset condition -- Receiver previously not in Offline state (see 11.1.3.3)
11. Elapsed time since entry into Loss_Of_Synchronization state = an extended period of time as defined by FC-0 and signal currently present on attached Fibre (see 11.1.1.2 and 11.1.5.1)
12. Elapsed time since entry into Loss_Of_Synchronization state = an extended period of time as defined by FC-0

and signal currently not present on attached Fibre (see 11.1.1.2 and 11.1.5.1)

13. Signal currently not present on attached Fibre (see 11.1.5.1)
14. Signal currently present on attached Fibre (see 11.1.5.1)
15. Exiting of Receiver reset condition -- Receiver previously in Offline state (see 11.1.4)
16. Completion of loss-of-Synchronization procedure (see 11.1.4)
17. Detection of a loss of signal condition (see 11.1.4)
18. Acquisition of Synchronization (see 11.1.4)

NOTES

1. The Loss_Of_Synchronization state is the default state upon completion of Receiver Initialization.
2. The data path width of a variable-path-width Receiver shall be established before that Receiver may enter the Synchronization_Acquired state.
3. The Receiver may be in either Loopback or normal mode when in states denoted by * (see 13 for a discussion of Loopback mode).
4. The Loss-Of-Signal-Failure state will never be entered by a Receiver if FC-0 does not provide a signal detection function.

11.3 Transmitter state description

While Operational and not disabled, a Transmitter attached to a Fibre shall constantly attempt to transmit an encoded bit stream onto that Fibre. Some Transmitters are capable of monitoring this transmitted signal and verifying its validity. Such Transmitters may become Not Operational as a result of this monitoring. Transmitters shall also become Not Operational if requests for information transfer not consistent with the transmission rules established by FC-1 are received. A Transmitter may also be placed in a disabled state under certain internal or external conditions.

The Transmitter state diagram is shown in figure 33.

11.3.1 Transmitter states

11.3.1.1 Operational states

A Transmitter actively attempting to transmit an encoded bit stream onto its attached Fibre is defined to be in the *Working* state.

Under certain conditions, it may be necessary or desirable to cease the transmission of signals by a Transmitter onto its attached Fibre. A Transmitter that has ceased transmission is in the *Not-Enabled* state. A Transmitter in the Not-Enabled state shall remain Operational.

NOTE - A Transmitter in the Not-Enabled state is free to transmit those signals associated with laser safety procedures. See 15 for additional information related to laser safety.

11.3.1.2 Not Operational states

A Transmitter shall become Not Operational and shall enter the *Failure* state under certain conditions. One determination of the Transmitter's state is based on signal monitoring rules not defined by this standard. The Transmitter's state is also based on monitoring of requests for information transfer to ensure consistency with the transmission rules established by FC-1 (see 11.3.4.1).

An FC-1-level-Transmitter-failure condition shall be recognized and reported to FC-2 when the Transmitter becomes Not Operational and enters the Failure state.

11.3.2 Entry into Working state

An Operational Transmitter shall enter the Working state when its Transmitter becomes enabled. A Transmitter shall become enabled under either of the following conditions:

- Processing of an FC-1 request to enter the enabled state when the Receiver was not placed in the Not-Enabled state as a result of laser safety procedures

- Determination by the FC-0 laser safety procedures that a laser safety condition (i.e., a condition which requires that the Transmitter cease transmission) no longer exists

11.3.3 Entry into Not-Enabled state

Entry into the Not-Enabled state may result from error conditions detected externally by the Link Control Facility, or it may result from internally generated signals. Specific conditions under which the Transmitter shall enter the Not-Enabled state are as follows:

- Completion of Transmitter Initialization

- Processing of an FC-1 request to enter the Not-Enabled state

- Determination by the FC-0 laser safety procedures that a laser safety condition exists

The Transmitter shall remain in the Not-Enabled state until the conditions causing it to cease transmission are removed.

11.3.4 Entry into Failure state

A Transmitter shall enter into the Failure state under the following conditions:

- Detection of a request for information transfer that is not consistent with the transmission rules established by FC-1 while in the Working or Not-Enabled state

- Recognition (via signal monitoring) of an FC-1-level-Transmitter-failure condition while in the Working state

Once the Failure state is entered, the Transmitter shall become Not Operational and shall remain in the Failure state until it is subsequently made Operational through external intervention.

11.3.4.1 Transmission rules

Requests for information transfer shall conform to the following rules in order for an FC-1 Transmitter to consider them valid requests:

- Requests for Ordered Set transmission shall be aligned with the word alignment established by the Transmitter upon entry into the Working state.

- Requests for Valid Data Byte transmission shall be aligned with the byte alignment established by the Transmitter upon entry into the Working state. Each byte of a four-byte word is aligned based on the established word alignment.

The following requests for information transfer are inconsistent with the transmission rules established by FC-1 and shall cause a Transmitter to enter the Failure state:

- Transmission requests improperly aligned with transmission (word and byte) boundaries

- Transmission requests not consistent with the currently active transmission boundary

  - Requests for Ordered Set transfers on a non-word boundary

  - Requests for data transfers overlapping an Ordered Set transfer currently in progress

- Lack of transmission requests (i.e., additional bytes required to complete a Transmission Word) when a Transmission Word has been partially transmitted as a result of prior transmission requests

11.3.4.2 Transmitter signal monitoring

Each Transmitter capable of monitoring its transmitted signal shall define a method by which this monitoring is to take place. It shall also define the conditions necessary to cause the Transmitter to recognize an FC-1-level-Transmitter-failure condition and change from the Working state to the Failure state.

NOTE - Monitoring of average light levels is a typical method of Transmitter monitoring. Typically, an error condition is detected when the transmitted light level of a Transmitter falls outside of the range of allowed values specified by FC-0 for its Transmitter class. (See 5.2 in FC-0 for a discussion of signal monitoring.)

11.4 Transmitter state diagram

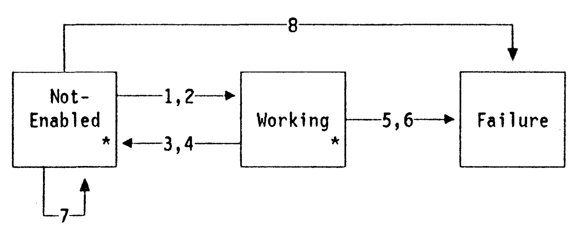


Figure 33. FC-1 Transmitter state diagram

State Transition Conditions:

1. Receipt of Transmitter enable request from FC-2 when Transmitter was not previously disabled as a result of a laser safety condition (see 11.3.2)
2. Determination by FC-0 that a laser safety condition no longer exists (see 11.3.2)
3. Receipt of Transmitter disable request from FC-2 (see 11.3.3)
4. Determination by FC-0 that a laser safety condition exists (see 11.3.3)
5. Detection by FC-0 of a Transmitter failure condition (see 11.3.4)
6. Detection of a request for information transfer inconsistent with the transmission rules established by FC-1 (see 11.3.4)
7. Receipt of Transmitter enable request from FC-2 when Transmitter was previously disabled as a result of a laser safety condition (see 11.3.3)
8. Detection of a request for information transfer inconsistent with the transmission rules established by FC-1 (see 11.3.4)

NOTES

1. The Not_Enabled state is the default state upon completion of Transmitter Initialization.
2. The data path width of a variable-path-width Transmitter shall be established before that Transmitter may enter the Working state.
3. The Transmitter may be in either Loopback or normal mode when in states denoted by * (see 13 for a discussion of Loopback mode).

12 Clock reference

A clock reference is provided by FC-1 to ensure that a gross frequency reference is available for the FC-0 Receiver to lock onto as a precursor to attempting bit synchronization based on the incoming bit stream. The method used to provide this reference is not defined by this standard.

NOTE - See C.4 for the recommended usage of this clock reference.

13 Loopback mode

Loopback mode is provided by FC-1 as a diagnostic function to FC-2. When in Loopback mode, transmission requests passed to the FC-1 Transmitter shall be shunted directly to the FC-1 Receiver, overriding any signal detected by the Receiver on its attached Fibre. A Link Control Facility must be explicitly placed in Loopback mode (i.e., Loopback mode is not the normal mode of operation of a Link Control Facility). The method of implementing Loopback mode is not defined by this standard.

NOTE - Loopback mode may be implemented either in the parallel or the serial circuitry of a Link Control Facility. As such, it may be implemented entirely within FC-1 or may be implemented by the combination of FC-1 and FC-0.

13.1 Receiver considerations

A Receiver in the Loss-Of-Synchronization or Synchronization-Acquired state may be placed in Loopback mode. A Receiver in any other state shall not be placed in Loopback mode.

Entry into or exit from Loopback mode may result in a temporary loss of Synchronization. Under such conditions, decoded information shall not be presented by the FC-1 Receiver to FC-2 until Synchronization has been reestablished.

13.2 Transmitter considerations

A Transmitter in the Working or Not-Enabled state may be placed in Loopback mode. A Transmitter in any other state shall not be placed in Loopback mode. The external behavior of a Transmitter (i.e., the activity of a Transmitter with respect to its attached Fibre) simultaneously in the Working state and in Loopback mode is unpredictable.

14 Diagnostic mode

Diagnostic functions may be provided at the FC-1 level by certain implementations. Compliance with the standard shall not be affected by the provision or exclusion of such functions by an implementation, and any functions that are provided by an implementation are not defined by the standard.

NOTE - A typical diagnostic function is the ability to transmit invalid Transmission Characters within an otherwise valid bit stream. Certain combinations of invalid Transmission Characters may damage certain FC-0 components (e.g., a bit stream of continuous '1's may overheat an LED Transmitter, resulting in a reduced lifetime for that Transmitter). Other invalid bit streams may cause a Receiver to lose word and/or bit synchronization. See 5.4 in FC-0 for a more detailed discussion of Receiver and Transmitter behavior under various diagnostic conditions.

15 Transmitter safety

Certain FC-0 Transmitter classes require special procedures to ensure that Class 1 laser safety standards are met. These procedures have the following effect on the FC-1 definition:

1. A special Transmitter state (the Not-Enabled state) is defined.
2. When the FC-0 laser safety procedures determine that a laser safety condition (i.e., a condition which requires that the Transmitter cease transmission) exists, the Transmitter shall enter the Not-Enabled state.
3. When the FC-0 laser safety procedures determine that a laser safety condition no longer exists, the Transmitter shall exit the Not-Enabled state.
4. FC-1 requests are incapable of forcing a Transmitter which enters the Not-Enabled state as a result of FC-0 laser safety procedures to exit that state.

For a detailed description of the laser safety procedures defined for the Fiber Channel Standard, see 6.2.3 in FC-0.

American National Standard for Information Systems

FC-2 level - Signaling protocol

16 Ordered Sets

16.1 Introduction

An Ordered Set is a four-byte combination of data and special transmission characters which is designated by the standard to have special meaning. Ordered Sets provide the ability to obtain bit and word synchronization which also establishes word boundary alignment. See FC-1 for additional information on Ordered Sets and rules for synchronization. The following types of Ordered Sets are defined:

- Start_of_Frame delimiters
- End_of_Frame delimiters
- Primitive Signals
  - Idle
  - Receiver_Ready (R_RDY)
- Primitive Sequences
  - Not_Operational (NOS)
  - Offline (OLS)
  - Link_Reset (LR)
  - Link_Reset_Response (LRR)

16.2 Frame delimiters

A frame delimiter is an Ordered Set that immediately precedes or follows the contents of a frame. Separate and distinct delimiters identify the start of a frame and the end of a frame. Frame delimiters shall be recognized when a single Ordered Set is detected.

16.2.1 Start_of_Frame (SOF)

The Start_of_Frame delimiter is an Ordered Set that immediately precedes the Frame Content. There are multiple **SOF** delimiters defined for Fabric and N_Port Sequence control. The **SOF** delimiter shall be transmitted on a word boundary.

16.2.2 End_of_Frame (EOF)

The End_of_Frame (**EOF**) delimiter is an Ordered Set that immediately follows the Frame_Trailer and shall be transmitted on a word boundary. The **EOF** delimiter designates the end of the Frame Content and is followed by Idle words.

16.3 Primitive Signals

A Primitive Signal is an Ordered Set designated by the standard to have special meaning. Primitive Signals are recognized when a single Ordered Set is detected. (See FC-1 for details on ordered sets and the bit encodings for Primitive Signals).

16.3.1 Idle Word (Idles)

An Idle Word is a Primitive Signal transmitted on the link to indicate an Operational Port facility ready for frame transmission and reception. Idle Words shall be transmitted on the link to obtain and maintain bit and word synchronization as well as word boundary alignment.

Idle Words shall be transmitted on the link during those periods of time when frames, R_RDY, or other Primitive Sequences are not being transmitted. A word is composed of four transmission characters during transmission. A word within the Frame Content, prior to transmission and after reception, is composed of four valid data bytes. (See 17.1 regarding frame transmission.)

16.3.2 Receiver_Ready (R_RDY)

The R_RDY Primitive Signal shall indicate that a single Class 1 connect-request (**SOFc1**), Class 2, or Class 3 frame was received and that the interface buffer which received the frame is available for further frame reception. Validity of the frame content is not required. Transmission of the R_RDY Primitive Signal shall be preceded and followed by a minimum of two Idle Words. (Since R_RDY is not passed through the Fabric, there is no requirement for additional Idles for clock skew management.) (See 17.1.) See 20.3.2.1 for a discussion regarding the use of R_RDY in flow control in conjunction with ACK frames.

16.4 Primitive Sequences

A Primitive Sequence is an Ordered Set that is transmitted repeatedly and continuously. Primitive Sequences are transmitted to indicate specific conditions within a Port (N_Port or F_Port) or conditions encountered by the Receiver logic of a Port.

Primitive Sequences are transmitted continuously while the condition exists. When a Primitive Sequence is received and recognized, a corresponding Primitive Sequence or Idles are transmitted in response. The following Primitive Sequence Protocols are described in clause 27:

— Link Initialization (see 27.3)
— Online to Offline (see 27.4)
— Link Failure (see 27.5)
— Link Recovery (see 27.6)

During a Class 1 Dedicated Connection, Primitive Sequences transmitted from an N_Port to the entry F_Port of a Fabric shall remove an existing Class 1 Dedicated Connection. The Fabric entry F_Port shall respond appropriately to the Primitive Sequence received and shall notify the exit F_Port which shall transmit the Link Reset Primitive to the other Connected N_Port (ie., initiate Link Recovery).

If a Class 1 Dedicated Connection does not exist, a Primitive Sequence transmitted by an N_Port shall be received and recognized by the F_Port, but not transmitted through the Fabric.

16.4.1 Primitive Sequence Recognition

Recognition of a Primitive Sequence requires detection of three consecutive Ordered Sets of the same Ordered Set without any intervening code violation errors.

16.4.2 Not_Operational (NOS)

The Not_Operational Primitive Sequence is transmitted to indicate that the Port transmitting this Sequence has detected a Link Failure condition.

Link failure conditions are defined as:

— Loss of Synchronization for more than the timeout period while not in the Offline State, as defined in FC-1,
— Loss of Signal while not in the Offline State, as defined in FC-1, or
— Timeout during the Link Recovery Protocol (see 27.6).

NOS shall be transmitted continuously as a Primitive Sequence. The Port transmitting NOS shall transmit the Sequence as long as the Link Failure condition exists, and until the Offline Sequence (OLS) is received.

Receiving NOS

A Port recognizing NOS shall transmit the Offline Sequence (OLS) in response.

16.4.3 Offline (OLS)

The Offline Primitive Sequence is transmitted to indicate that the Port transmitting this Sequence is:

— receiving and recognizing NOS,
— entering the Offline State, or
— exiting the Offline State.

Entry into Offline State

A Port enters the Offline Transmit State to notify an attached Port that it is preparing to:

— perform power-on initialization,
— reinitialize,
— perform diagnostic procedures, or
— power-down.

To enter the Offline State, a Port performs the Online to Offline Protocol as specified in 27.4. While the Port is in the Offline State, it may perform diagnostic procedures, turn off its transmitter, and transmit any signal (excluding Link Reset Response) without errors being detected by the other attached Port. While in the Offline State, a Port may also power-down.

Exiting Offline State

A Port exits the Offline State when initialization or diagnostics are complete by performing the Link Initialization Protocol specified in 27.3.

Receiving OLS

A Port receiving and recognizing OLS shall enter the Offline Receive State and shall:

1. transmit the Link Reset (LR) Primitive Sequence in response, and
2. shall ignore Receiver errors such as Loss of Synchronization or Loss of Signal.

A Port in Offline Receive State exits this State when the Link Reset Response (LRR) Primitive Sequence is received and recognized.

16.4.4 Link Reset (LR)

The Link Reset Primitive Sequence is transmitted by a Port after a Link Timeout error has occurred, or the OLS Sequence is received and recognized. LR shall be transmitted continuously as a Primitive Sequence.

In Class 1, a Link Timeout error is detected during a Dedicated Connection if Sequence timeouts have been detected for all active Sequences.

In Class 1, 2, or 3, a Link Timeout error is detected if one or more R_RDY Primitive Signals is not received within the timeout period after the buffer-to-buffer Credit_CNT has reached zero.

Receiving LR

Link Reset Response (LRR) is transmitted after LR is received and recognized.

Dedicated Connection

In Class 1, an existing Dedicated Connection shall be removed and the end-to-end Credit associated with the Connected N_Port shall be reset to the Login value. An N_Port supporting Class 1 Service may also transmit the LR Primitive Sequence when it is unable to determine its Connection status.

All Class 1 frame Sequences which are active in an N_Port when LR is received and recognized, or transmitted shall be abnormally terminated. When the Link Reset Primitive is being transmitted by an N_Port, all Class 1 frames received are discarded. See 28.8 for more discussion on Connection Recovery.

R_RDY

In Class 1, 2 and 3, the buffer-to-buffer Credit_CNT within the N_Port is reset to its Login value and an F_Port processes or discards any Class 1 connect-request, Class 2, or Class 3 frame(s) currently held in the receive buffer associated with the outbound fibre of the attached N_Port. end-to-end Credit is not affected.

16.4.5 Link Reset Response (LRR)

The Link Reset Response Primitive Sequence shall be transmitted by a Port to indicate that it is receiving and recognizes the LR Primitive Sequence. LRR shall be transmitted continuously as a Primitive Sequence.

Receiving LRR

After a Port receives and recognizes the LRR Sequence, it shall transmit the Idles.

17 Frame formats

All FC-2 frames follow the general frame format as shown in figure 34. An FC-2 frame is composed of a Start_of_Frame delimiter, Frame Content, and an End_of_Frame delimiter. The Frame Content is composed of a Frame_Header, Data Field, and CRC.

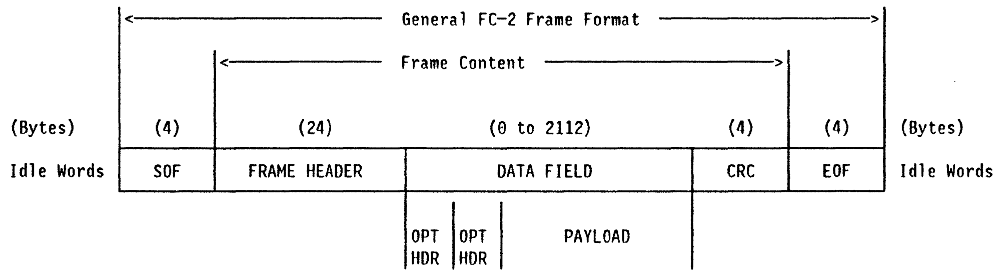


Figure 34. FC-2 general frame format

17.1 Frame transmission

Frame transmission shall be performed by inserting a frame immediately following a sequence of Idle Words. Idles are then transmitted immediately upon completion of frame transfer.

At the transmitter there shall be a minimum of six Primitive Signals (Idle Words and R_RDY) between frames. A minimum of two Idles shall be guaranteed to precede the start (**SOF**) of each frame received by a destination N_Port. The Fabric may insert or remove Idles as long as the destination receives at least two Idles preceding each frame.

NOTE - See Annex N for Fabric requirements.

17.2 Start_of_Frame (SOF)

The Start_of_Frame delimiter is an Ordered Set that immediately precedes the Frame Content. There are multiple **SOF** delimiters defined for Fabric and N_Port Sequence control. The **SOF** delimiter shall be transmitted on a word boundary. The bit encodings for the **SOF** delimiters are defined in FC-1.

17.2.1 Start_of_Frame Connect Class 1 (SOFc1)

SOFc1 shall be used to request a Class 1 Dedicated Connection. The frame Data Field size is limited to the maximum size specified by the destination N_Port or by the Fabric, if present, whichever size is smaller. The Data Field size is determined during the Login procedure. This delimiter also identifies the start of the first Sequence (ie. implicit **SOFi1**).

17.2.2 Start_of_Frame Initiate (SOFix)

A Sequence is initiated and identified by using the **SOFix** delimiter in the first frame. **SOFix** is used to indicate **SOFi1**, **SOFi2**, or **SOFi3**. The following three delimiters identify the start of a Sequence based on Class of Service.

17.2.2.1 Start_of_Frame Initiate Class 1 (SOFi1)

The first Sequence is initiated with **SOFc1**. After Class 1 Service is established, all subsequent Sequences within a specific Dedicated Connection are initiated with **SOFi1**.

17.2.2.2 Start_of_Frame Initiate Class 2 (SOFi2)

The **SOFi2** shall be used on the first frame to initiate a Sequence for Class 2 Service.

17.2.2.3 Start_of_Frame Initiate Class 3 (SOFi3)

The **SOFi3** shall be used on the first frame to initiate a Sequence for Class 3 Service.

17.2.3 Start_of_Frame Normal (SOFnx)

The following three delimiters identify the start of all frames other than the first frame of a Sequence based on Class of Service. **SOFnx** is used to indicate **SOFn1**, **SOFn2**, or **SOFn3**.

17.2.3.1 Start_of_Frame Normal Class 1 (SOFn1)

The **SOFn1** shall be used for all frames except the first frame of a Sequence for Class 1 Service.

17.2.3.2 Start_of_Frame Normal Class 2 (SOFn2)

The **SOFn2** shall be used for all frames except the first frame of a Sequence for Class 2 Service.

17.2.3.3 Start_of_Frame Normal Class 3 (SOFn3)

The **SOFn3** shall be used for all frames except the first frame of a Sequence for Class 3 Service.

17.2.4 Start_of_Frame Fabric (SOFf)

The **SOFf** shall be used by the Fabric for Fabric_Frames. If a Fabric_Frame is received by an N_Port, the frame shall be discarded. See 17.9 for description of Fabric_Frame.

17.3 Frame_Header

The Frame_Header shall be the first field of the Frame Content and immediately follow the **SOF** for all frames. The Frame_Header shall be transmitted on a word boundary. The Frame_Header is used to control link operations, control device protocol transfers, and detect sequence errors in the link transport facility. The Frame_Header is described in detail in clause 18.

17.4 Data Field

The Data Field shall follow the Frame_Header. Two Frame_Types are defined based on the value of the DL Bits (Bits 31-30) in the R_CTL field of the Frame_Header. Data Fields are aligned on word boundaries.

A discussion of FT_0 (Link_Control) and FT_1 (Data) frames is found in clause 20. The two Frame_Types are:

1. FT_0 (Data Field = 0 bytes)

   When the DL Bits are set to 1 1, an FT_0 frame (Link_Control) is specified which has a Data Field length of zero.

2. FT_1 (Data Field = 0 to 2112 bytes)

   When the DL Bits are set to any values other than 1 1, an FT_1 frame (Data frame) is specified whose Data Field size is a multiple of four bytes and ranges in size from 0 to 2112 bytes. If any of the last three bytes are not meaningful, they shall contain fill bytes as specified in the F_CTL field in table 28.

   FT_1 frames may contain optional headers, as defined by the DF_CTL field as well as the Payload. Optional Data Field headers are described in clause 19.

17.5 CRC

The Cyclic Redundancy Check (CRC) is a four byte field that is used to verify the data integrity of the Frame_Header and Data Field. **SOF** and **EOF** delimiters are not included in the CRC verification. The CRC field shall be calculated on the Frame_Header and Data Field prior to encoding for transmission or after decoding upon reception. The CRC field shall be aligned on a word boundary.

The CRC uses the following 32-bit polynomial (from the Fiber Distributed Data Interface Media Access Control document) and is described in Annex L.

$$X^{32} + X^{26} + X^{23} + X^{22} + X^{16} + X^{12} + X^{11} + X^{10} + X^{8} + X^{7} + X^{5} + X^{4} + X^{2} + X + 1$$

17.6 End_of_Frame (EOF)

The End_of_Frame (**EOF**) delimiter is an Ordered Set that immediately follows the CRC. The **EOF** delimiter designates the end of the Frame Content and shall be followed by Idle words. There are three categories of **EOF** delimiters. One category of delimiter indicates that the frame contents are valid. The second category indicates that the frame contents are invalid. The third category indicates the frame content is corrupted and the frame was truncated during transmission. The bit encoding for the **EOF** delimiters is defined in FC-1.

17.6.1 Valid frame content

Three types of End_of_Frame delimiter are defined to indicate that the contents of the frame are valid.

17.6.1.1 End_of_Frame Terminate (EOFt)

The **EOFt** shall end the last frame of a Sequence and shall indicate that the Sequence associated with this SEQ_ID is complete. It is required to properly close a Sequence without error.

17.6.1.2 End_of_Frame Disconnect_Terminate (EOFdt)

The **EOFdt** shall remove a Class 1 Dedicated Connection through a Fabric, if present. The **EOFdt** shall also identify the last ACK of a Sequence and indicate that all Sequences associated with this S_ID are terminated. Open Sequences, other than the SEQ_ID specified in the frame containing **EOFdt**, shall be abnormally terminated and may require Sequence recovery (FC-4 protocol dependent). The **EOFdt** is only used for removing for Class 1 Service and is not applicable to Class 2 and Class 3 Service.

17.6.1.3 End_of_Frame Normal (EOFn)

The **EOFn** shall identify the end of frame when one of the other **EOF** delimiters indicating valid frame contents is not required.

17.6.2 Invalid frame content

There are two **EOF** delimiters which indicate that the frame contents are invalid. If a frame is received at an intermediate Fabric F_Port or an intermediate N_Port and an error is detected within the frame, the frame is passed to the destination N_Port with a modified **EOF** to indicate that an error was previously detected. Errors such as code violation or CRC errors are examples of detectable frame errors. The following **EOF** delimiters indicate that the frame contents are invalid.

17.6.2.1 End_of_Frame Disconnect_Terminate_Invalid (EOFdti)

The **EOFdti** shall replace an **EOFdt** delimiter on a frame with invalid frame content. The **EOFdti** shall remove a Class 1 Dedicated Connection through a Fabric, if present. It shall also indicate that all Sequences associated with the Connected N_Port are abnormally terminated and may require Sequence recovery (FC-4 protocol dependent).

The frame containing the **EOFdti** is processed in the following manner:

1. no response frame is transmitted,
2. the frame content may or may not be used at the receiver's discretion, and
3. the receiver knows that the Dedicated Connection has been removed.

17.6.2.2 End_of_Frame Invalid (EOFni)

The **EOFni** shall replace an **EOFn**, or **EOFt** indicating that the frame content is invalid.

The frame containing the **EOFni** is processed in the following manner:

1. no response frame is transmitted, and
2. the frame content may or may not be used at the receiver's discretion.

17.6.3 Aborted frame content

17.6.3.1 End_of_Frame Abort (EOFa)

The **EOFa** shall terminate a partial frame due to a malfunction in a link facility during transmission. The frame shall end on a word boundary and shall be discarded by the receiver without reply. The transmitter shall retransmit the aborted frame with the same Sequence count (SEQ_CNT) up to its ability to retry, which may be zero.

An invalid **EOF** (**EOFdti** or **EOFni**) delimiter may be changed to an **EOFa** delimiter under the conditions specified for **EOFa**.

EOFa delimiters shall not be changed to an invalid **EOF** delimiter under any conditions.

17.7 Frame validity

An FC-2 frame shall be considered "valid" if it contains a valid **SOF**, a valid **EOF**, valid data characters, and proper CRC verification of the Frame Header and Data Field. If a received frame is invalid, it may be discarded and no response frame or acknowledgment frame shall be transmitted. However, the R_RDY Primitive is transmitted in response to an appropriate frame regardless of Frame_Content validity. The detected error shall be accumulated in a Link Error Status Block. Frame reception is discussed in 29.10.

17.8 Frame field order

The Frame Content is transmitted sequentially following the **SOF** delimiter and proceeds character by character from left to right within the definition of the frame until the **EOF** delimiter is transmitted. Each field within the Frame Content is transmitted character by character from left to right.

17.9 Fabric_Frame

A Fabric_Frame is a special frame format which is different than an FC-2 frame and is used by the Fabric for intrafabric communication. A Fabric_Frame is composed of a **SOFf** delimiter, Frame Content, and an **EOFn** delimiter. If a Fabric_Frame is received by an N_Port, it shall be discarded and ignored.

17.9.1 Fabric_Frame Content

The Fabric_Frame Content is not defined within the FC-2 document.

18 Frame_Header

18.1 Introduction

The Frame_Header shall be subdivided into fields as shown in figure 35.

Word	Bits 3322 2222 1098 7654	2222 1111 3210 9876	1111 11 5432 1098	7654 3210
0	R_CTL	D_ID		
1	rrrr rrrr	S_ID		
2	TYPE	F_CTL		
3	SEQ_ID	DF_CTL	SEQ_CNT	
4	OX_ID		RX_ID	
5	Parameters			

Figure 35. Frame_Header

The Frame_Header shall be the first field of the Frame Content and immediately follow the **SOF** delimiter for all frames. The Frame_Header shall be transmitted on a word boundary. The Frame_Header is used to control link operations, control device protocol transfers, and detect sequence errors in the link transport facility.

18.1.1 Frame identification

Identification fields such as OX_ID, RX_ID, and SEQ_ID are provided for frame tracking, error detection, and frame processing. Frames are tracked through uniqueness of those fields from the time of assignment over the life of the field constructs.

An Exchange Originator tracks frames on the basis of OX_ID||SEQ_ID||SEQ_CNT. An Exchange Responder tracks frames on the basis of RX_ID||SEQ_ID||SEQ_CNT.

A given N_Port Originator may choose to provide frame tracking outside of FC-2. This is indicated by setting the OX_ID to hexadecimal 'FFFF' with the New X_ID F_CTL bit = 1. This implies that the Exchange Originator shall only have one Exchange active with a given destination N_Port. If an N_Port chooses a mechanism outside of FC-2, it is still responsible for providing proper SEQ_ID and SEQ_CNT values. In addition, it shall return the RX_ID assigned by the Responder.

A given N_Port Responder may choose to provide frame tracking outside of FC-2. This is indicated by setting the RX_ID to hexadecimal 'FFFF' with the New X_ID F_CTL bit = 1. If an N_Port chooses a mechanism outside of FC-2, it is still responsible for providing proper SEQ_ID and SEQ_CNT values. In addition, it shall return the OX_ID assigned by the Originator.

The following sections describe the contents of the Frame_Header fields:

18.2 R_CTL field

Routing Control (Word 0, Bits 31-24) is a one byte field that contains control Bits to assist in frame routing, data routing, or addressing.

- **Bits 31-30 - DL Bits**

  0 0 - Device_Data frame
  0 1 - Reserved
  1 0 - Link_Data frame
  1 1 - Link_Control frame

DL bits differentiate device and link frames for routing purposes within an N_Port or F_Port.

When the DL Bits (Word 0, Bits 31-30) are set to zeros, it indicates a Device_Data frame.

The DL Bits (Word 0, Bits 31-30) indicate a Link frame when Bit 31 is set to a one. Bit 30 further distinguishes between Link_Data and Link_Control frames. See clause 20 for a discussion of Data and Link_Control frames. See clause 21 for a discussion on Link Application (Link_Data) frames.

- **Bits 29-24 - Data Category**

  Category bits are available to each FC-4 to indicate data type or data control information relating to the data within the Payload portion of the Data Field. This information may assist the receiver of the Data frame in buffer management or routing based on data type. Category Bits shall specify a Data category for the Data Field of a single frame. Consecutive frames within a Sequence may contain different Categories of Data.

18.3 Address identifiers

Each N_Port shall have a single native address identifier which is unique within the address domain of a Fabric. In addition, an N_Port may optionally have one or more alias addresses which may be shared across multiple N_Ports within a node or common controlling entity. Alias addressing may be used to implement a Hunt Group.

An N_Port address identifier of binary zeros indicates that the N_Port is unidentified. Address identifiers in the range of hexadecimal 'FFFFF0' to 'FFFFFE' are well-known and reserved for Fabric use. See table 26. The Address identifier of hexadecimal 'FFFFFF' is reserved. See clause 23 for more information on address identifier assignment.

Table 26. Fabric Address Identifiers

Address Value	Description
FFFFF0 to FFFFF9	Reserved
FFFFFA	Management Server
FFFFFB	Time Server
FFFFFC	Name Server
FFFFFD	Fabric Controller
FFFFFE	Fabric F_Port

18.3.1 Destination_ID (D_ID)

The Destination Identification (D_ID) is a three byte field (Word 0, Bits 23-0) that contains the address identifier of an N_Port within the destination node.

18.3.2 Source_ID (S_ID)

The Source Identification (S_ID) is a three byte field (Word 1, Bits 23-0) that contains the address identifier of an N_Port within the source node.

18.4 Data structure type (TYPE)

The data structure type (TYPE) is a one byte field (Word 2, Bits 31-24) that identifies the Upper Level Protocol of the frame content for data frames.

When the DL bits in R_CTL indicate a Link_Control frame, the TYPE field is decoded as a Link_Control command as specified in 20.3.1.

When the DL bits in R_CTL indicate Link_Data or Device_Data, the TYPE codes are decoded as shown in table 27.

Table 27. TYPE codes

Encoded Value Wd 2, bits 31-24	Description
0000 0000	Link Application
0000 0001	Reserved
0000 0010	Memory Port
0000 0011	Memory Initialization
0000 0100	802.2
0000 0101	Reserved
0000 0110	IPI-3 Master
0000 0111	IPI-3 Slave
0000 1000	SCSI Initiator
0000 1001	SCSI Target
0000 1010	HIPPI-FP
0000 1011	Reserved
0000 1100	Channel Command Word
0000 1101 to 1100 1111	Reserved
1101 0000 to 1111 1111	Vendor Specific

18.5 Frame Control (F_CTL)

The Frame Control (F_CTL) field (Word 2, Bits 23-0) is a three byte field that contains control information relating to the frame content. The format of the F_CTL field is defined in table 28.

Table 28 (Page 1 of 2). F_CTL field			
Control Field	**Word 2, Bits**	**Description**	**Reference**
Exchange/Sequence Control	23-13		
Exchange context	23	0 = Originator of Exchange 1 = Responder of Exchange	see 24.3.1
Sequence context	22	0 = Sequence Initiator 1 = Sequence Recipient	see 24.5.1
First_Sequence	21	0 = intermediate or last Sequence of Exchange 1 = first Sequence of Exchange	see 24.3.1
Last_Sequence	20	0 = first or intermediate Sequence of Exchange 1 = last Sequence of Exchange	see 24.9
End_Sequence	19	0 = first or intermediate Data frame of Sequence 1 = last Data frame of Sequence	see 24.7
End_Connection	18	0 = Connection active 1 = End of Connection Pending (Class 1)	see 28.6.2
Connection_Resource	17	0 = Connection Resource Required 1 = Connection Resource Not Required (Class 1)	see 28.6.1
Sequence Initiative	16	0 = hold Sequence Initiative 1 = transfer Sequence Initiative	see 24.7
new X_ID assigned	15	0 = X_ID assignment retained 1 = new X_ID assigned	see 24.4
Invalidate X_ID	14	0 = X_ID assignment retained 1 = invalidate X_ID	see 24.4
Reserved	13-6		
Abort Sequence Condition	5-4	ACK frame - Sequence Recipient 0 0 Continue Sequence 0 1 Abort Sequence, retransmit 1 0 Stop Sequence, read Status 1 1 Sequence timeout Data frame - Sequence Initiator 0 0 Defer to Recipient 0 1 Discard policy required 1 0 Reserved 1 1 Reserved	see 24.7
Reserved	3		
Exchange reassembly	2	0 = Exchange reassembly not active 1 = Exchange reassembly active	see 26.3.2
Fill Data Bytes	1-0	End of Data field - bytes of fill 0 0 0 Bytes of fill 0 1 1 Byte of fill 1 0 2 Bytes of fill 1 1 3 Bytes of fill	

Bit 23 - Exchange Context

An Exchange is started by the Originator facility within an N_Port. The destination N_Port of the Exchange is known as the Responder. Each frame for this Exchange indicates whether the S_ID is associated with the Originator or Responder.

Bit 22 - Sequence Context

A Sequence is started by a Sequence Initiator facility within an N_Port. The destination N_Port of the Sequence is known as the Sequence Recipient. Each frame of the Sequence indicates whether the S_ID is associated with the Sequence Initiator or the Sequence Recipient.

NOTE: Ownership is required for proper handling of Link_Control frames received in response to Data frame transmission. When a Busy frame is received, it may be in response to a Data frame (Sequence Initiator) or to an ACK frame (Sequence Recipient). This Bit simplifies the necessary constructs to distinguish between the two cases.

Bit 21 - First_Sequence

This Bit is set to one on all Data frames in the First Sequence of an Exchange. It is set to zero for all other Sequences within an Exchange.

Bit 20 - Last_Sequence

This Bit is set to one on all Data frames in the Last Sequence of an Exchange. It is set to zero for all other Sequences within an Exchange.

Bit 19 - End_Sequence

This Bit is set to one on the last Data frame of a Sequence. This indication is used for Sequence termination by the two N_Ports involved in the Sequence. This Bit is set to zero for the first or intermediate frames within a Sequence.

Bit 18 - End_Connection

This Bit is set to one to indicate that the N_Port transmitting E_C is beginning the disconnect procedure. The N_Port transmitting E_C set to one is completing its last Sequence of the Connection and is requesting the receiving N_Port to transmit a frame terminated by **EOFdt** if the receiving N_Port has completed all active Sequences. If the receiving N_Port is not able to transmit **EOFdt**, E_C set to one requests that the N_Port complete all active Sequences and not initiate any new Sequences during the current Connection.

See 28.6.2 for a discussion on removing Class 1 Dedicated Connections (E_C bit).

Bit 17 Connection_Resource (C_R bit)

The C_R bit is an information bit set by the Sequence Initiator on the first and the last frame of a Sequence. When the C_R bit = 0, it indicates that Dedicated Connection resources are required by the Initiator in the Recipient N_Port. When the C_R bit = 1, it indicates that Dedicated Connection resources other than the current Sequence are no longer required and the Initiator is preparing to begin the disconnect procedure. The Sequence Initiator may reset the C_R bit to 0 after it has transmitted C_R = 1 based on the current conditions in the N_Port. The setting of C_R is not binding and either N_Port may initiate the disconnect procedure by setting E_C = 1 regardless of the C_R setting.

Bit 16 - Sequence Initiative

The Originator of an Exchange initiates the first Sequence as the Sequence Initiator. If the Sequence Initiative Bit is set to zero, the Sequence Initiator holds the initiative to continue transmitting the current Sequence in addition to the initiative to start new Sequences for this Exchange. The Sequence Recipient gains the initiative to transmit a new Sequence for this Exchange after the Sequence Initiative has been transferred to the Recipient. This is accomplished by setting the Sequence Initiative to one in the last Data frame of a Sequence (End_Sequence = 1).

Bit 15 - New X_ID assigned

This bit is set to one on a Data frame transmitted by the Sequence Initiator to indicate that its X_ID is new. New_X_ID is indicated at the beginning of an Exchange or following an X_ID reassignment transition. The bit continues to be set to one for each frame of the Sequence or until the ACK is received when X_ID transition interlock is active.

This bit is set to one on an ACK frame transmitted by the Sequence Recipient to indicate that its X_ID is new. New_X_ID is indicated at the beginning of an Exchange or following an X_ID reassignment transition. The bit continues to be set to one for each frame of the Sequence or the ACK is transmitted when X_ID transition interlock is active.

If X_ID interlock is active, Bit 15 is only used on the frame(s) in which the new X_ID is assigned. The new X_ID may be any value including the previous value specified. If Bit 15 is set to zero, it indicates that the current X_ID assignment is valid.

Bit 14 - Invalidate X_ID

Invalidate X_ID may be indicated by the Sequence Initiator in the last Data frame of a Sequence (End_Sequence = 1) if the Initiator has indicated X_ID reassignment is required during Login. When the Sequence Initiator indicates that the current X_ID is being invalidated or unassigned, the Sequence Recipient may also unassign its X_ID by setting Bit 14 to 1 in the ACK frame corresponding to the last Data frame of the Sequence. X_ID invalidation shall occur at the end of a Sequence but does not require transfer of Sequence Initiative at the same time.

X_ID invalidation by an N_Port requires use of an Association Header as described in 24.4.

Bits 5-4 Abort Sequence Condition

Bits 5-4 are set to a value other than zeros by the Sequence Initiator on the first frame of an Exchange to indicate that the Originator is requiring a specific error policy on this Exchange.

```
0 0   Defer to Recipient
0 1   Discard policy required
1 0   reserved
1 1   reserved
```

Bits 5-4 are set to a value other than zeros by the Sequence Recipient in an ACK or Link_Response frame to indicate to the Sequence Initiator that a malfunction or error has been detected by the Sequence Recipient.

```
0 0   Continue Sequence
0 1   Abort Sequence and retransmit
1 0   Stop Sequence and read status
1 1   Sequence timeout by Recipient
```

An aborted Sequence is a Sequence which has been terminated in a manner other than the normal termination process.

A setting of 0 1 indicates a request by the Sequence Recipient to the Sequence Initiator to terminate this Sequence using the Abort Sequence Protocol and then retransmit the Sequence.

A setting of 1 0 indicates a request by the Sequence Recipient to the Sequence Initiator to stop this Sequence using the Abort Sequence Protocol and then read FC-4 status. This allows for an early termination by the Sequence Recipient. Some of the data received may have been processed. See 21.4.4 for a description of the Abort Sequence protocol.

A setting of 1 1 indicates that the Sequence Recipient has aborted the Sequence due to Sequence timeout (See 24.6.1). The Sequence status is saved by the Sequence Recipient in the Exchange Status Block associated with the aborted SEQ_ID.

Bit 2 - Exchange reassembly

The Sequence Initiator shall set bit 2 = 0 to indicate that the Payload in this Data frame is associated with an Exchange between a single pair of N_Ports. Therefore, reassembly is confined to a single destination N_Port.

The Sequence Initiator shall set bit 2 = 1 to indicate that the Payload in this Data frame is associated with an Exchange being managed by a single controlling entity using multiple N_Ports at either the source, destination, or both. Therefore, segmentation and reassembly may be distributed across multiple N_Ports. (See 26.3.2.)

Bits 1-0 - Fill Data Bytes

The Bits associated with the Fill Data Bytes notifies the Data frame receiver that one or more of the last three bytes of the Data Field do not contain data specified by the transmitter and shall be ignored. The fill character is not specified by the Standard but shall be a valid data character.

18.6 Sequence_ID (SEQ_ID)

The SEQ_ID is a one byte field (Word 3, Bits 31-24) that is assigned by the Sequence Initiator and shall be unique for a specific D_ID and S_ID at the time it is assigned. Both the Sequence Initiator and the Sequence Recipient track the status of the Sequence. The Sequence Initiator tracks the Sequence on the basis of OX_ID and SEQ_ID. The Sequence Recipient tracks the

Sequence on the basis of RX_ID and SEQ_ID. (See 18.9.)

The combination of Initiator and Recipient Sequence Status Blocks identified by a single SEQ_ID describe the complete status of that Data Sequence for a given Exchange. See clause 29 for a description of the Sequence Status Block.

18.7 DF_CTL

The Data_Field Control (DF_CTL) is a one byte field (Word 3, Bits 23-16) that specifies the presence of optional headers at the beginning of the Data_Field. Control Bit usage is shown in table 29.

Table 29. DF_CTL Bit definition

Word 3, Bit(s)	Optional Header
23	Reserved for expansion
22	0 - No Expiration_Security Header 1 - Expiration_Security Header
21	0 - No Network_Header 1 - Network_Header
20	0 - No Association_Header 1 - Association_Header
19	0 - No Operation_Header 1 - Operation_Header
18	Reserved
17-16	0 0 - No Device_Header 0 1 - 16 Byte Device_Header 1 0 - 32 Byte Device_Header 1 1 - 64 Byte Device_Header

The Optional Headers are positioned in the Data Field in the sequence specified with the Bit 23 header as the first header in the Data Field, Bit 22 header as the second header in the Data Field, and so forth, in a left to right manner corresponding to Bits 23, 22, 21, and so forth.

If either Bit 17 or 16 is set to one, then a Device Header is present. The size of the Device Header is specified by the encoded value of Bits 17 and 16 as shown.

If an Optional Header is not present, no space in the Data Field is reserved. Therefore, for example, if Bits 23 and 22 are zero and Bit 21 is one, the first data byte of the Data Field contains the first byte of the Network_Header.

See clause 19 for a discussion on Optional Headers.

18.8 Sequence Count (SEQ_CNT)

The sequence count (SEQ_CNT) is a two byte field (Word 3, Bits 15-0) that shall indicate the sequential order of frame transmission within a single Data frame Sequence. The first frame of a Sequence shall contain a sequence count of binary zero. If back to back Sequences for the same Exchange occur, it is the responsibility of the Sequence Initiator to use a unique SEQ_ID since frame uniqueness is based on X_ID||SEQ_ID||SEQ_CNT. The sequence count shall be incremented by one on all subsequent Data frames within this Sequence. ACK and Link_Response frames shall be identified by the same SEQ_ID and SEQ_CNT as the frame to which it is responding. Frames are tracked on a SEQ_ID||SEQ_CNT basis associated with the OX_ID for the Originator and the RX_ID for the Responder.

The sequence count shall wrap to zero after reaching a value of 65535. Sequences of Data frames and sequence count values are discussed in clause 24.

18.9 Originator Exchange_ID (OX_ID)

The Originator Exchange_ID is a two byte field (Word 4, Bits 31-16) that identifies the Exchange_ID assigned by the Originator of the Exchange. The Originator of the Exchange shall assign a unique value for OX_ID, if used, or a value of hexadecimal 'FFFF' if not used. An OX_ID of hexadecimal 'FFFF' indicates that uniqueness is not being enforced by the Originator by the OX_ID mechanism.

An Originator Exchange Status Block associated with the OX_ID is used to track the progress of a series of Sequences which composes an Exchange. See clause 24 for a discussion of Sequences and Exchanges.

NOTE - If hexadecimal 'FFFF' is used as the OX_ID, then the Originator uses an alternate tracking mechanism. If the OX_ID is unique, it may be used as an index into a control block structure which may be used in conjunction with other constructs to track frames.

18.10 Responder Exchange_ID (RX_ID)

The Responder Exchange_ID is a two byte field (Word 4, Bits 15-0) assigned by the Responder which provides a unique, locally meaningful identifier at the Responder for an Exchange established by an Originator and identified by an OX_ID.

A Responder Exchange Status Block associated with the RX_ID is used to track the progress of a series of Sequences which composes an Exchange. See clause 24 for a discussion of Sequences and Exchanges.

18.11 Parameters

The Parameter field (Word 5, Bits 31-0) has two meanings based on frame type. For Link_Control frames, the Parameter field is used to carry data specific to the Link_Control frame.

For Data frames, the Parameter field specifies Relative Offset, a four-byte field that specifies the relative displacement of the first byte of the Payload of the frame from the base address. Relative Offset is specified in terms of bytes.

The offset value may be relative to one or more Sequences within an Exchange. The number of bytes transmitted in a single Sequence shall not exceed the maximum value of the Relative Offset (Parameter) field ($2^{32} - 1$). A value of hexadecimal 'FFFFFFFF' indicates that this field shall be ignored. Performance may be improved if data is aligned on natural boundaries.

See clause 26 for a discussion concerning Relative Offset. See clause 20 for a discussion concerning use of the Parameter field in Link_Control frames.

19 Optional headers

19.1 Introduction

Five optional headers defined within the Data Field of a frame are:

1. Extended Frame_Header
2. Expiration_Security_Header
3. Network_Header
4. Association_Header
5. Operation_Header

The presence of optional headers is defined by control Bits in the DF_CTL field of the Frame_Header. The sequential order of the optional headers, Device_Header and Payload and their sizes are indicated in figure 36.

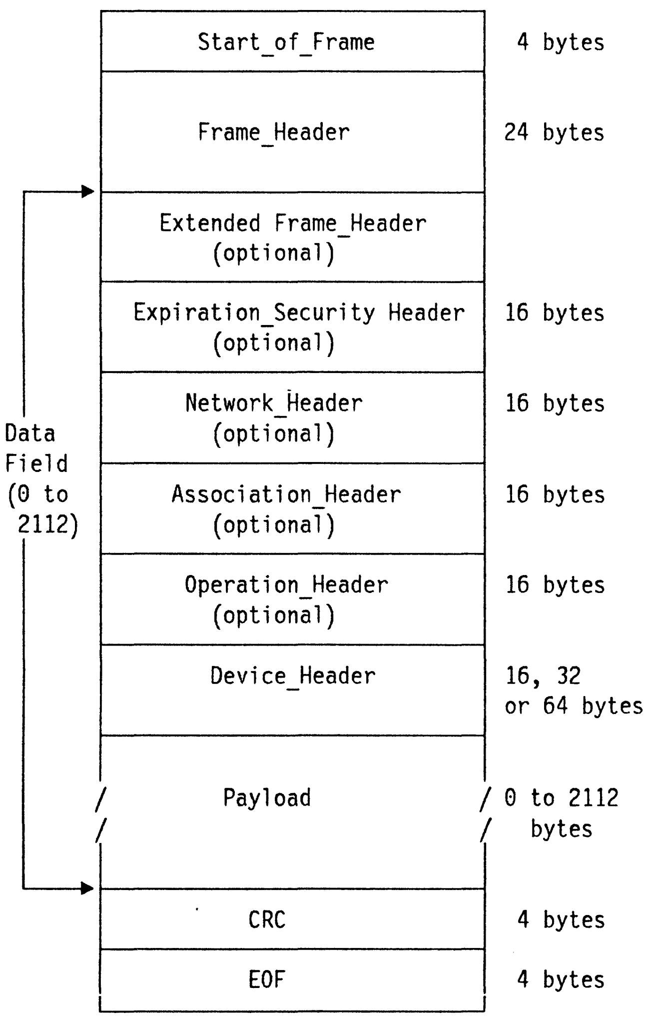


Figure 36. Optional headers order

If present, an Extended Frame_Header shall be the first optional header to follow the Frame_Header. If present, an Expiration_Security_Header shall be the next 16 bytes of the Data Field following the Extended Frame_Header. If present, a Network_Header shall be the next 16 bytes of the Data Field. If present, a Operation_Header shall be the next 16 bytes of the Data Field. If present, a Device_Header shall follow the optional headers. If none of the optional headers is present, no space in the Data Field shall be reserved.

The Network Header may be used by a bridge or a gateway node which interfaces to an external Network. The Network_Header, if present, shall be 16 bytes in size.

The Operation_Header is used for correlating multiple exchanges within a given Operation. This header may also be used for striping of data across multiple N_Ports within a single node. The Operation_Header, if present, shall be 16 bytes in size.

The Device_Header, if present, shall be 16, 32, or 64 bytes in size. The contents of the Device Header are entirely under the control of a level above FC-2 based on the TYPE field.

19.2 Extended Frame_Header

The Extended Frame_Header shall provide the capability to extend the Frame_Header functions for future needs. Its presence shall be indicated by Bit 23 in the DF_CTL field, located in the Frame_Header, being set to one. The Extended Frame_Header is reserved for future expansion.

19.3 Expiration_Security_Header

The Expiration_Security_Header, as shown in figure 37, is an optional header within the Data_Field content. Its presence shall be indicated by Bit 22 in the DF_CTL field, located in the Frame_Header, being set to one. The Expiration_Security_Header shall be 16 bytes in size.

Word	Bits		
0	63 Expiration Time (Most significant word) 32		
1	31 Expiration Time (Least significant word) 0		
2	31 24 S_Type	Reserved	15 S_Length 0
3	31 Security 0		

Figure 37. Expiration_Security_Header

19.3.1 Expiration Time

An Expiration Timer in a system shall be used to compute the Expiration Time to be included in this field. The Expiration Timer shall be a 64-bit fixed-point number based, in seconds relative to 0000 UT on 1 January 1900. The integer part of the number shall be the most significant 32 bits, and the fractional part shall be the least significant 32 bits.

On start up, a value of zeros shall be used to indicate an invalid or undefined time. When an N_Port begins communication within a system, it may obtain the Expiration Timer value from the Fabric or other specified N_Port. The Expiration Time shall be determined by adding an expiration time value to the current system Expiration Timer value. If a frame is received after the Expiration Time has been exceeded, the frame shall be discarded by the N_Port. The frame may be discarded by the Fabric.

In a network of heterogeneous Fabrics, the start date of 1 January 1900 shall be used. For local use, the fractional part of the timer shall be optional. Multiple Fabrics in a system shall synchronize their Expiration Timers to an accuracy of $\pm$ 2 secs.

19.3.2 Security type (S_Type)

The field indicates the type of Security supported.

19.3.3 Security length (S_Length)

The field indicates the length in bytes of the security information.

19.4 Network_Header

The Network_Header, as shown if figure 38, is an optional header within the Data_Field content. Its presence shall be indicated by Bit 21 in the DF_CTL field, located in the Frame_Header, being set to one. The Network_Header shall be 16 bytes in size.

Word	Bits	
0	63 56 D_NAA	55 Network_Destination_ID (high order bits) 32
1	31 Network_Destination_ID (low order bits) 0	
2	63 56 S_NAA	55 Network_Source_ID (high order bits) 32
3	31 Network_Source_ID (low order bits) 0	

Figure 38. Network_Header

The Network_Header provides information to an N_Port for traversing a heterogeneous network.

19.4.1 D_NAA or S_NAA

Destination Network_Address_Authority (D_NAA) or Source Network_Address_Authority (S_NAA) field indicates the authority responsible for the administration of the network address (destination or source) used. The following are Network_Address_Authority (NAA) indicators:

Table 30. NAA identifiers

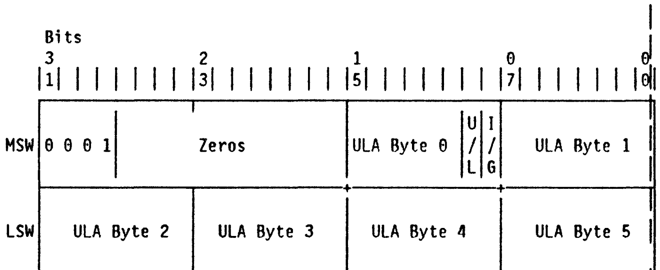
Bits 31 30 29 28	NAA
0 0 0 0	Ignored
0 0 0 1	IEEE
0 0 1 0	CCITT
0 0 1 1	Reserved
...	...
1 1 1 1	Reserved

19.4.2 Network_Destination_ID or Network_Source_ID

The Network_Destination_ID or Network-_Source_ID shall be a 60 bit field indicating the network address being used. These network addressed are also referred to as World_Wide_Names.

19.4.2.1 IEEE 48-bit address

When D_NAA (or S_NAA) is IEEE, Network_Destination_ID (or Network_Source_ID) field will contain a 48-bit IEEE Standard 802.1A Universal LAN MAC Address (ULA). The ULA shall be represented as an ordered string of six bytes numbered from 0 to 5. The least significant bit of byte 0 shall be the Individual/Group Address (I/G) bit. The next least significant bit shall be the Universally or Locally Administered Address (U/L) bit. This IEEE Standard 802.1A further specifies that the bytes be transferred in the order 0 to 5. Figure 39 shows how the bytes of an ULA shall be mapped to two words on the Network Header.



MSW - Most significant word
LSW - Least significant word

Figure 39. IEEE 48-bit address format

19.4.2.2 CCITT 60-bit address

When D_NAA (or S_NAA) is CCITT, Network-_Destination_ID (or Network_Source_ID) field will contain a 60-bit CCITT standard address.

19.5 Association_Header

The Association_Header is an optional header within the Data_Field content. Its presence shall be indicated by Bit 20 in the DF_CTL field, located in the Frame_Header, being set to one. The Association_Header shall be 32 bytes in size.

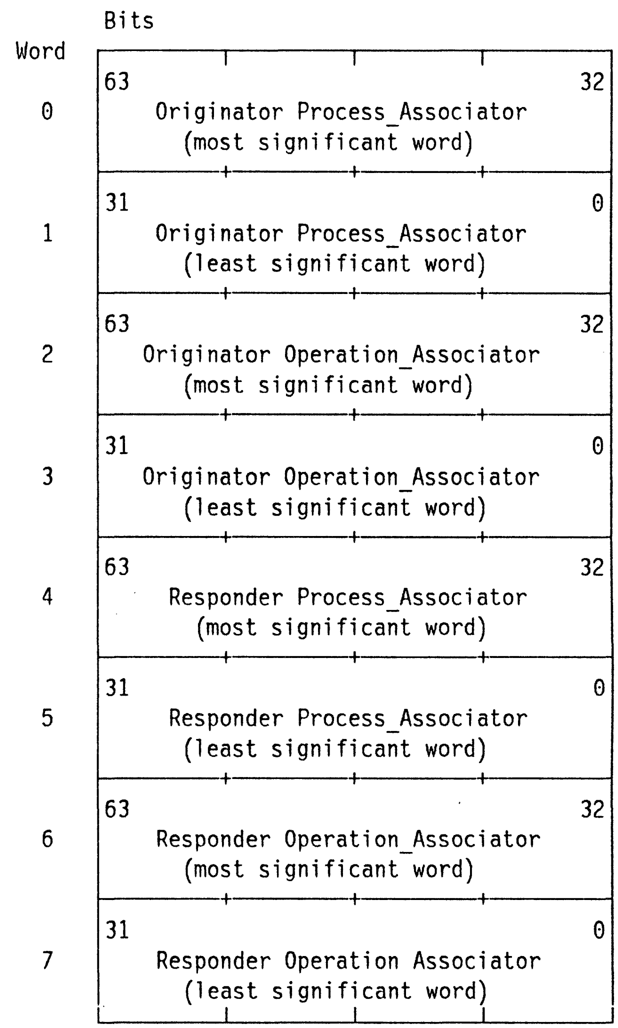


Figure 40. Association_Header

The Association_Header shall be used to track an Exchange when an X_ID is reassigned during the Exchange (see 24.4). The Association_Header shall be subdivided into four fields as illustrated in figure 40. Contents of Association_Header are meaningful to the Node which generates it. It may not be meaningful to the other Node receiving it.

19.5.1 Process_Associators

Process_Associators (Originator and Responder) have following characteristics:

- The Process_Associators shall be known by each end prior to the initiation of an Operation between the two processes.
- The Process_Associators shall be remembered for the duration of the process life.
- The Process_Associators may not change between Operations.

19.5.2 Operation_Associators

Operation_Associators (Originator and Responder) have following characteristics:

- The Operation_Associators shall be remembered for the duration of the Operation.
- The Operation_Associators may not change within an Operation.

19.6 Operation_Header

The Operation_Header is an optional header within the Data_Field content. Its presence shall be indicated by Bit 19 in the DF_CTL field, located in the Frame_Header, being set to one. The Operation_Header shall be 16 bytes in size.

Word	Bits	
0	63 – 32: Operation Identifier (O_ID)	
1	31 – 0: Operation Identifier (O_ID)	
2	31 – 16: Operation Control (O_CTL)	15 – 0: Reserved
3	31 – 0: Reserved	

Figure 41. Operation_Header

An Operation is a group of one or more related Exchanges. The Operation_Header contains information which shall correlate one or more Exchanges within a single Operation specified by the TYPE field specified (ie., FC-4).

The Operation_Header shall be subdivided into four fields as illustrated in figure 41. The Operation_Header format shall be common to all Upper Level Protocols using the Operation_Header.

19.6.1 Operation_Identifier

The Operation_Identifier (O_ID) shall be an eight-byte field specified by the system which owns or initiates the first exchange of this operation. The O_ID shall correlate one or more exchanges within a single operation. The O_ID is interpreted by the owner FC-4.

19.6.2 Operation_Control

Operation_Control (O_CTL) shall be a two-byte field containing general control information common to FC-4s. The format shall be as specified in figure 42.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Figure 42. O_CTL field

- **Bit 15 - Operation Context**
  - 0 = specifies that this frame is transmitted by the N_Port (node) which initiated this Operation.
  - 1 = specifies that this frame is transmitted by the N_Port (node) which is the Operation Recipient.

- **Bit 14 - first OpHdr in Operation**
  Bit 14 = 1 indicates that this is the first Operation_Header in an operation. This Bit shall be set to 0 for all other frames.

- **Bit 13 - last OpHdr in Operation**
  Bit 13 = 1 indicates that this is the last Operation_Header in this operation. This Bit shall be set to 0 for all other frames.

- **Bits 12-0 - Reserved**

19.7 Device_Header

A Device_Header shall be a 16, 32, or 64 byte field. The content of the this Header shall be specified by FC-4 based on the TYPE field specified.

20 Data and Link_Control frames

20.1 Introduction

There are two categories of frames and a Primitive Signal:

— Data (Link_Data, Device_Data) frames, and

— Link_Control (ACK, Link_Response) frames.

— R_RDY Primitive Signal.

Data frames convey information from a source N_Port to destination N_Port. Data frames may or may not contain Payload in the Data Field following the Frame_Header. A variety of Protocols using Data frame Sequences may be employed. An example of a Request-Reply protocol for Link Applications is discussed in clause 21.

Link_Control frames are individual Link-Level frames which indicate successful or unsuccessful frame delivery and participate in end-to-end flow control. Successful delivery is indicated by ACK frames, while unsuccessful delivery is indicated by Link_Response frames.

The R_RDY Primitive Signal is used for buffer to buffer flow control which is discussed in clause 25.

20.1.1 R_RDY response

In Class 1, a connect-request (**SOFc1**) frame is responded to by transmitting the R_RDY Primitive Signal. In Class 2 and Class 3, all frames received (ie., both Data frames and Link_Control frames) are responded to by transmitting the R_RDY Primitive Signal.

20.1.2 Data frame responses

Link_Control response frames are ACK frames (ACK_1 and ACK_N) and Link_Response frames (P_BSY, P_RJT, F_BSY, and F_RJT). Link_Control responses to Data frames are class dependent.

20.1.2.1 ACK frames

Successful Data frame delivery

- Class 1
  1. ACK_1, or
  2. ACK_N
- Class 2
  1. ACK_1, or
  2. ACK_N
- Class 3
  1. No response

20.1.2.2 Link_Response frames

Unsuccessful Data frame delivery

- Class 1
  1. F_BSY (Fabric Busy), or
  2. P_BSY (N_Port Busy), or
  3. F_RJT (Fabric Reject), or
  4. P_RJT (N_Port Reject).
- Class 2
  1. F_BSY (Fabric Busy), or
  2. P_BSY (N_Port Busy), or
  3. F_RJT (Fabric Reject), or
  4. P_RJT (N_Port Reject).
- Class 3
  1. No response

20.2 Data frames

Data frames are used to transfer information from a source N_Port to a destination N_Port. Device_Data frames are identified by the DL Bits 31-30 in the R_CTL field being set to 0 0 and the TYPE field identifying a specific Upper Level Protocol.

Data frames may be used to transfer information such as data, control information, header information, and status as specified by upper levels. In Class 1 and 2, each Data frame successfully transmitted shall be acknowledged by an ACK frame to indicate successful delivery to the destination N_Port. An indication of unsuccessful delivery shall be returned to the transmitter by a Link_Response frame in Class 1 and 2.

Data frames may be streamed, i.e., multiple, unidirectional frames may be transmitted before a response frame is received. The number of outstanding, unacknowledged Data frames allowed is specified by a Service Parameter during the Login procedure (Credit). See clause 23 for a discussion on Login and Service Parameters and clause 25 for a discussion on flow control.

Multiple, unidirectional Data frames are called a Sequence. See clause 24 for a discussion on Sequences and Exchanges.

Delimiters: **SOFc1**, **SOFix**, **SOFnx**, **EOFn**, **EOFt**, **EOFdt**.
Format: FT_1
Addressing: The S_ID field designates the source N_Port (Sequence Initiator) transmitting the Data frame. The D_ID field designates the destination N_Port (Sequence Recipient) of the Data frame.
Data Field: The Data_Field for Data frames is variable in size but a multiple of four bytes. The Data Field may contain optional Headers whose presence is indicated by the DF_CTL field in the Frame_Header. (See clause 19.)

In order to accommodate message content within the Data field that is not a multiple of four bytes, fill characters shall be appended to the end of the Data Field. The number of fill characters is specified by F_CTL Bits 1-0. (See 18.5.) The fill character is not specified by the standard.
Responses:

R_RDY Primitive (**SOFc1**, **SOFx2**, **SOFx3**)
ACK
- ACK_1, or
- ACK_N

Link_Response
- F_RJT, P_RJT
- F_BSY, P_BSY

20.2.1 Transfer Mechanisms

Several mechanisms are available for a Sequence which allow an Upper Level Protocol to convey information to a destination N_Port. Those mechanisms include:

- Data category within R_CTL field
- options within F_CTL field
- Device Header

Data Category:

The Data Category is included in R_CTL to assist the receiver of a Data frame in directing the Data Field content to the appropriate buffer pool.

F_CTL field options:

Within the F_CTL field, Exchange and Sequence CTL bits are used to manage the initiative to transmit Data frames, manage Sequences, and manage Exchanges.

Device Header:

Provisions have been made for a Device_Header to precede the actual Data contained in the Data Field of a Data frame. The size of the Device_Header is encoded in the DF_CTL field of the Frame_Header. This provides an additional means to specify unique protocol-dependent information to an Upper Level.

20.3 Link_Control

Link_Control frames are used by the N_Port Link Facility functions to control frame transfer at the link level.

Link Control frames are the only Frame_Type_0 frames. In FT_0 frames, the Parameter field is used to carry up to four bytes of information. FT_0 frames are identified by the DL Bits 31-30 in the R_CTL field being set to 1 1 for all Link_Control frames.

Link_Control frames provide:

- indication of successful delivery,
- indication of unsuccessful delivery,
- flow control and buffer management feedback.

An N_Port shall be required to provide sufficient link control facilities such that Link_Control frames received in response to Data frames transmitted do not result in P_BSY response frames.

20.3.1 Link_Control command codes

When DL Bits 31-30 in the R_CTL field are set to 1 1, the TYPE field of the Frame_Header is encoded as shown in table 31.

Table 31. Link Control codes		
Encoded Value Word 2, bits 31-24	Description	Abbr.
0000 0000	Acknowledge_1	ACK_1
0000 0001	Acknowledge_N	ACK_N
0000 0010	N_Port Busy	P_BSY
0000 0011	Fabric Busy	F_BSY
0000 0100	N_Port Reject	P_RJT
0000 0101	Fabric Reject	F_RJT

20.3.2 Link_Continue function

ACK frames shall be sent in response to valid Data frames frames (see 17.7) to acknowledge end to end frame transfers. The R_RDY Primitive shall be sent in response to a single frame (see 17.7) to acknowledge a buffer to buffer frame transfer based on **SOF** delimiter. The following list specifies flow control elements:

- R_RDY - buffer to buffer flow control for frames between F_Ports and N_Ports if a Fabric is present, or between N_Ports without a Fabric present. The R_RDY Primitive is transmitted on receipt of any Class 2 or 3 frame as well as a Class 1 connect-request frame.
- ACK_1 - end to end flow control for a single Data frame transfer between N_Ports with or without a Fabric present. The ACK_1 frame is transmitted on receipt of Class 1 or 2 Data frames.
- ACK_N - end to end flow control for one or more consecutive Data frame transfers between N_Ports with or without a Fabric present. The ACK_N frame is transmitted on receipt of Class 1 or 2 Data frames.

Either ACK_1 or ACK_N may be used for acknowledgment of Data frames between N_Ports for a given active Sequence, but both forms shall not be used within the same Sequence.

20.3.2.1 Receiver Ready (R_RDY)

The R_RDY Primitive shall indicate that a single frame was received and that the interface buffer which received the frame is available for further frame reception. The R_RDY Primitive provides buffer to buffer flow control for a single frame.

R_RDY is transmitted between an N_Port and an F_Port with a Fabric present, or between two N_Ports without a Fabric present for all Class 2 and Class 3 frames and connect-request frames in Class 1. The R_RDY Primitive is not forwarded to any higher levels within an N_Port or passed beyond the entry or exit point of the Fabric.

See 16.3.2 for specification of the number of Idles before and after the R_RDY Primitive during transmission.
Responses: There is no response to an R_RDY Primitive.

20.3.2.2 Acknowledge_1 (ACK_1)

The ACK_1 frame shall indicate that a single valid Data frame was received by the destination N_Port for the corresponding X_ID||SEQ_ID||SEQ_CNT and that the interface buffer which received the frame is available for further frame reception. ACK_1 frames are used in Class 1 and 2. The ACK_1 frame provides end to end flow control for a single frame between two N_Ports.

ACK_1 is not forwarded to any higher levels within an N_Port.
Delimiters: SOFnx, EOFn, EOFt, EOFdt
Format: FT_0
Addressing: The D_ID field designates the source of the Data frame (Sequence Initiator) being replied to by ACK_1, while the S_ID field designates the source of the ACK_1 frame (Sequence Recipient).
F_CTL: The F_CTL field is returned with Sequence and Exchange Context Bits inverted in the ACK_1 frame.
SEQ_ID: the SEQ_ID matches the SEQ_ID of the frame being replied to by ACK_1.
SEQ_CNT: The sequence count (SEQ_CNT) shall be equal to the sequence count of the Data frame being replied to by the ACK_1.
Parameter field: The field shall contain a binary one.

Responses:

R_RDY Primitive (**SOFx2**)
Link_Response
— F_RJT, P_RJT
— F_BSY

20.3.2.3 Acknowledge_N (ACK_N)

The ACK_N frame shall indicate that one or more consecutive Data frames for the corresponding X_ID||SEQ_ID were received by the destination N_Port. Buffers are available for reception of additional Data frames as noted in the credit acknowledged. The ACK_N frame provides end to end flow control for one or more Data frames between two N_Ports. Support for the ACK_N frame is optional and is specified in the Service Parameters during Login.

A specific Data frame is allowed to be acknowledged once and only once. ACK_N is only used for Class 1 and Class 2 Data frames. (See 23.5)

ACK_N is not forwarded to any higher levels within an N_Port.
Delimiters: SOFnx, EOFt, EOFdt
Format: FT_0
Addressing: The D_ID field designates the source of the Data frame (Sequence Initiator) being replied to by ACK_N, while the S_ID field designates the source of the ACK_N frame (Sequence Recipient).
F_CTL: The F_CTL field is returned with the Sequence and Exchange Context Bits inverted in the ACK_N frame. Parameters. (See clause 23.)
SEQ_ID: The SEQ_ID matches the SEQ_ID of the frames being acknowledged.
SEQ_CNT: The SEQ_CNT specifies the SEQ_CNT of the highest Data frame being acknowledged. (See Annex M, "ACK_N Usage (Informative)" for examples of use.)
Parameter field: Bits 31 - 16 are reserved. Bits 15 - 0 contain the number of consecutive Data frames up to and including the frame identified by sequence count (SEQ_CNT) of the ACK_N.

ACK_N parameter

	First Byte	Second Byte	Third Byte	Fourth Byte
Bits	31 24	23 18 17 16	15 8	7 0
	rrrr rrrr	rrrr rrrr	cccc cccc	cccc cccc
			Credit Acknowledged (m)	

Figure 43. ACK_N parameter

Responses:

R_RDY Primitive (**SOFx2**)
Link_Response
— F_RJT, P_RJT
— F_BSY

NOTE - For example, if the SEQ_CNT is 7 and the value of the Parameter field is 3, frames 5, 6, and 7 are being acknowledged. (See Annex M, "ACK_N Usage (Informative)" for examples of use.)

20.3.3 Link_Response

Link_Response frames shall be sent by either the destination N_Port or a Fabric F_Port in reply to frames for Class 1 and 2. Link_Response frames shall only be sent in reply to valid frames (see 17.7).

A Link_Response indicates that the frame identified by the SEQ_ID and SEQ_CNT was not delivered to or processed by the destination N_Port. When a Link_Response frame (either Reject or Busy) is generated by an F_Port or a facility internal to the Fabric, the frame is routed to the destination N_Port of the Link_Response frame. Link_Response frames may be:

— Busy - indicates a Busy condition was encountered in the Fabric or destination N_Port.

— Reject - indicates that delivery of the frame is being denied.

20.3.3.1 Reject (P_RJT, F_RJT)

The Reject Link_Response shall indicate that delivery of a frame is being denied. A four-byte reject action and reason code is contained in the Parameter field. Rejects are transmitted for a variety of conditions. For certain conditions retry is possible, whereas other conditions may require specific intervention. This standard does not define the specific intervention required.

Reject Link_Response frames are transmitted for Class 1 and 2 frames. In Class 3, the frame in question shall be discarded. A Reject frame (F_RJT, P_RJT) shall not be transmitted in response to another Reject frame (either F_RJT or P_RJT). The Reject frame received in error shall be discarded.

P_RJT: N_Port Reject indicates that the Reject was issued by the destination N_Port.

F_RJT: Fabric Reject indicates that the Reject was issued by an F_Port within a Fabric.

Delimiters: SOFnx, EOFt, EOFdt
Format: FT_0
Addressing: The D_ID field designates the source of the frame being rejected while the S_ID field designates the destination of the frame being rejected.
F_CTL: The F_CTL field is returned with the Sequence and Exchange Context bits inverted in the F_RJT or P_RJT frame.
SEQ_ID: The SEQ_ID matches the SEQ_ID of the frame being rejected.
SEQ_CNT: The SEQ_CNT shall be equal to the SEQ_CNT of the frame being rejected.
Parameter field: The four bytes of this field indicate the action code and reason for rejecting the request. (See figure 44 and tables 32 and 33.)

```
Reject parameter

            First     Second     Third    Fourth
            Byte      Byte       Byte     Byte
         +---------+---------+---------+---------+
  Bits   |3       2|2     111|1        |         |
         |1       4|3     876|5       8|7       0|
         +---------+---------+---------+---------+
         |AAAA AAAA|CCCC CCCC|rrrr rrrr|VVVV VVVV|
         +---------+---------+---------+---------+
          |_______| |_______|           |_______|
              |         |                   |
            Action    Reason          Vendor Unique
```

Figure 44. Reject code format

Table 32. Action Codes

Encoded Value Wd 5, bits 31-24	Description
0000 0001 (1)	Sequence terminated - retry after correction
0000 0010 (2)	Sequence terminated - non-retryable error
0000 0011 (3)	Abort Sequence requested - retry after correction
0000 0100 (4)	Abort Sequence requested - non-retryable error
Other codes	Reserved

Action 1

Action 1 indicates that the Recipient or the Fabric has abnormally terminated the Sequence (**EOFt** or **EOFdt**). For a Class 1 connect-request the reject frame is ended with **EOFdt**. The Recipient shall only terminate the Sequence on a Reject in response to a connect-request (**SOFc1**) or in response to an interlocked Data frame associated with X_ID assignment or reassignment. The Sequence is retryable if the corrective action indicated by the reason code is performed:

— Class not supported
— Invalid D_ID
— Invalid S_ID
— N_Port not available-temporary
— N_Port not available-permanent
— Login required

Applicability:

— by Fabric when D_ID = Fabric
— by Fabric when D_ID = N_Port
— by N_Port when D_ID = N_Port

Action 2

Action 2 indicates that the Recipient or the Fabric has abnormally terminated the Sequence (**EOFt** or **EOFdt**). For a Class 1 connect-request the reject frame is ended with **EOFdt**. The Recipient shall only terminate the Sequence on a Reject in response to a connect-request (**SOFc1**) or in response to an interlocked Data frame associated with X_ID assignment or reassignment. The Sequence is non-retryable and further recovery such as Abort

Exchange may be required. The following reason codes are non-retryable:

- Delimiter usage error
- Type not supported
- Invalid Link_Control
- Invalid R_CTL
- Invalid F_CTL
- Invalid OX_ID
- Invalid RX_ID
- Invalid SEQ_ID
- Invalid DF_CTL
- Invalid SEQ_CNT
- Invalid Parameter field
- Exchange error
- Protocol error
- Data Field too large
- Unexpected ACK
- Unexpected Link_Response

Applicability:

- by Fabric when D_ID = Fabric
- by N_Port when D_ID = N_Port

Action 3

Action 3 indicates that the Recipient is requesting the source of the rejected frame to abort the Sequence. The Sequence is retryable if the corrective action indicated by the reason code is performed:

- Class not supported
- Invalid D_ID
- Invalid S_ID
- Login required

Applicability:

- by Fabric when D_ID = Fabric
- by Fabric when D_ID = N_Port
- by N_Port when D_ID = N_Port

Action 4

Action 4 indicates that the Recipient is requesting the source of the rejected frame to abort the Sequence or Exchange. The Sequence is non-retryable for the following reason codes:

- Delimiter usage error
- N_Port not available, temporary
- N_Port not available, permanent
- Type not supported
- Invalid Link_Control
- Invalid R_CTL
- Invalid F_CTL
- Invalid OX_ID
- Invalid RX_ID
- Invalid SEQ_ID
- Invalid DF_CTL
- Invalid SEQ_CNT
- Invalid Parameter field
- Exchange error
- Protocol error
- Data Field too large
- Unexpected ACK
- Unexpected Link_Response

Applicability:

- by Fabric when D_ID = Fabric
- by Fabric when D_ID = N_Port
- by N_Port when D_ID = N_Port

Table 33 (Page 1 of 2). Reason Codes

Encoded Value Wd 5, bits 23-16	Description
0000 0001	Class not supported
0000 0011	Delimiter usage error
0000 0101	Invalid D_ID
0000 0111	Invalid S_ID
0000 1001	N_Port not available, temporary
0000 1011	N_Port not available, permanent
0000 1101	TYPE not supported
0000 1111	Invalid Link_Control
0001 0001	Invalid R_CTL field
0001 0011	Invalid F_CTL field
0001 0101	Invalid OX_ID
0001 0111	Invalid RX_ID
0001 1001	Invalid SEQ_ID
0001 1011	Invalid DF_CTL
0001 1101	Invalid SEQ_CNT
0001 1111	Invalid Parameter field
0010 0001	Exchange Error
0010 0011	Protocol Error
0010 0101	Data Field too large
0010 0111	Unexpected ACK
0010 1001	Unexpected Link_Response
0010 1011	Login Required
0010 1101 to 1111 1110	Reserved

Table 33 (Page 2 of 2). Reason Codes	
Encoded Value Wd 5, bits 23-16	**Description**
1111 1111	Vendor Unique Error (See Bits 7-0)

Reject reason codes are arranged in descending priority as the encoded value increases. Therefore, if Class not supported is the reject reason, no additonal checking of the frame is required and no other errors are reported. The first error encountered according to the priority specified is the error reported and further error checking is not required.

Class not supported

The Class of Service specified by the **SOF** delimiter of the frame being rejected is not supported.

Delimiter usage error

The **SOF** or **EOF** is not appropriate for the current conditions. For example, a frame started by **SOFc1** is received while a Class 1 Dedicated Connection already exists with the same N_Port.

Invalid D_ID

F_RJT - the Fabric is unable to locate the destination N_Port address.

P_RJT - the N_Port which received this frame does not recognize the D_ID as its own Identifier.

Invalid S_ID

F_RJT - the S_ID does not match the N_Port Identifier assigned by the Fabric.

P_RJT - the destination N_Port does not recognize the S_ID as valid.

N_Port not available, temporary

F_RJT - The N_Port specified by the D_ID is a valid destination address but the N_Port is not functionally available. The N_Port is online and may be performing a Link Failure or Recovery Protocol, for example.

N_Port not available, permanent

F_RJT - The N_Port specified by the D_ID is a valid destination address but the N_Port is not functionally available. The N_Port is Offline or Powered Down.

TYPE not supported

The TYPE field of the frame being rejected is not supported by the N_Port replying with the Reject frame.

Invalid Link_Control

The command specified in the TYPE field in the frame being rejected is invalid or not supported as a Link_Control frame.

Invalid R_CTL field

The R_CTL field is invalid or inconsistent with the other Frame Header fields or conditions present.

Invalid F_CTL field

The F_CTL field is invalid or inconsistent with the other Frame_Header fields or conditions present.

Invalid OX_ID

The OX_ID specified is inconsistent with the other Frame_Header fields or conditions present.

Invalid RX_ID

The RX_ID specified is inconsistent with the other Frame_Header fields or conditions present.

Invalid SEQ_ID

The SEQ_ID specified is inconsistent with the other Frame_Header fields or conditions present.

Invalid DF_CTL

The DF_CTL field is invalid.

Invalid SEQ_CNT

The SEQ_CNT specified is inconsistent with the other Frame_Header fields or conditions present. A SEQ_CNT reject is not used to indicate out of order or missing Data frames (see F_CTL Abort Sequence). An example of use is a Data frame initiating a Sequence with a SEQ_CNT other than zero.

Invalid Parameter field

The Parameter field is incorrectly specified or invalid.

Exchange Error

An error has been detected in the identified Exchange (OX_ID). This could indicate

Data frame transmission without Sequence Initiative or other logical errors in handling an Exchange.

Protocol Error

This indicates that an error has been detected which violates the rules of FC-2 signaling protocol which are not specified by other error codes.

Unexpected ACK

An ACK_1 or ACK_N was received from an unexpected S_ID.

Unexpected Link_Response

A P_BSY was received from an unexpected S_ID.

Data Field too large

F_RJT or P_RJT - the frame being rejected exceeded the maximum Receive Data Field size specified in the Service Parameters.

Login Required

F_RJT or P_RJT - an Exchange is being initiated before the interchange of Service Parameters (ie. Login) has been performed. F_RJT may be issued by the Fabric in order to notify an N_Port that a re-Login is required due to changes within the Fabric.

Vendor Unique Error

The Vendor Unique Error bits shall be used by specific Vendors to specify additional reason codes.

Responses:

R_RDY Primitive (**SOFx2**, **SOFx3**)
Link_Response
— F_BSY

20.3.3.2 Busy (P_BSY, F_BSY)

The Busy Link_Response shall indicate that a Fabric, if present, or the Responder N_Port is temporarily occupied with other link activity and is not able to receive the frame. A reason code is identified in the Parameter field. For Class 1 Service, Busy shall only be transmitted in response to the **SOFc1** frame and shall be ended with **EOFdt**. For Class 2 Service, any Data frame may receive a Busy response. A Busy response is not used with Class 3 Service.

A Busy frame (F_BSY, P_BSY) shall not be transmitted in response to another Busy frame (either F_BSY or P_BSY).

When a Busy frame is received in response to a frame transmission, the transmitter of the frame shall retransmit the busied frame up to its ability to retry. Therefore, a Port shall save sufficient information for frames with an **SOFc1**, or **SOFx2** delimiter for retransmission until an ACK or RJT is received.

F_BSY

Fabric Busy indicates that the Busy was issued by the Fabric.

P_BSY

N_Port Busy indicates that the Busy was issued by the destination N_Port. P_BSY is not an acceptable response to a Link_Control frame. An N_Port shall be required to process Link_Control response frames up to its ability to transmit unacknowledged Data frames.

Delimiters: SOFnx, EOFt, EOFdt
Format: FT_0
Addressing: The D_ID field designates the source of the frame encountering the busy condition while the S_ID field designates the destination of the frame encountering the busy condition.
F_CTL: The F_CTL field is returned as received with the Sequence and Exchange Context bits inverted in the F_BSY or P_BSY frame.
SEQ_ID: The SEQ_ID matches the SEQ_ID of the frame being busied.
SEQ_CNT: The SEQ_CNT shall be equal to the SEQ_CNT of the frame encountering the busy condition.
Parameter field The four bytes of this field indicate the reason for the busy response as defined in figure 45.

```
Busy parameter

          First     Second      Third     Fourth
          Byte       Byte       Byte       Byte
        ┌─────────┬─────────┬─────────┬─────────┐
Bits    │3       2│2     111│1        │         │
        │1       4│3     876│5       8│7       0│
        ├─────────┼─────────┼─────────┼─────────┤
        │AAAA AAAA│CCCC CCCC│rrrr rrrr│VVVV VVVV│
        └─────────┴─────────┴─────────┴─────────┘
         └───┬───┘ └───┬───┘           └───┬───┘
             │         │                   │
          Action     Reason           Vendor Unique
```

Figure 45. Busy code format

Table 34. Busy action codes

Encoded Value Wd 5, bits 31-24	Description
0000 0001 (1)	Sequence terminated - retry later
0000 0010 (2)	Sequence active - retry later
Others	Reserved

Action 1

Action 1 indicates that the Recipient has busied the Sequence ($\mathbf{EOF_t}$ or $\mathbf{EOF_{dt}}$). For a Class 1 connect-request the busy frame is ended with $\mathbf{EOF_{dt}}$. The Recipient shall only terminate the Sequence on a Busy in response to a connect-request ($\mathbf{SOF_{c1}}$) or in response to an interlocked Data frame associated with X_ID assignment or reassignment ($\mathbf{SOF_{i1}}$, or $\mathbf{SOF_{i2}}$). The frame and Sequence are retryable at a later time.

Applicability:

- by Fabric when D_ID = Fabric
- by Fabric when D_ID = N_Port
- by N_Port when D_ID = N_Port

Action 2

Action 2 indicates that the Recipient has busied a Class 2 frame and that the Sequence has not been terminated. The frame is retryable at a later time.

Applicability:

- by Fabric when D_ID = Fabric
- by Fabric when D_ID = N_Port
- by N_Port when D_ID = N_Port

Table 35. Busy reason codes

Encoded Value Wd 5, bits 23-16	Description
0000 0001	Fabric busy
0000 0011	Physical N_Port busy
0000 0101	Logical N_Port busy
0000 0110 to 1111 1110	Reserved
1111 1111	Vendor Unique Busy (See Bits 7-0)

Busy reason codes are defined as follows:

Fabric busy

F_BSY - the Fabric is blocked from delivering a frame to the destination N_Port specified in the frame being rejected.

Physical N_Port busy

F_BSY - the Fabric is unable to deliver the frame being transmitted due to a Class 1 Dedicated Connection active at the destination N_Port.

P_BSY - the destination N_Port link facilities are currently busy and the N_Port is unable to process the frame received.

Logical N_Port Busy

P_BSY - the destination N_Port is unable to complete the Data frame request at the present time due to the logical state of the N_Port facilities. Logical N_Port Busy may also be indicated by an Application Reject (A_RJT) Link_Data reply frame (see 21.5.2).

Vendor Unique Error

The Vendor Unique Error bits shall be used by specific Vendors to specify additional reason codes.

Responses:

R_RDY Primitive (**SOF_{x2}**, **SOF_{x3}**)
Link_Response
— F_RJT, P_RJT

20.4 Data Protocol Summary

Table 36 summarizes the Data transfer by Data frames and Sequences for Class 1,2, and 3. The "PROTOCOL" columns represent an example Request-Reply protocol with frames interchanged by two N_Ports in order to accomplish an Upper Level Protocol request and its associated reply. Other Protocols are also possible within the definition of FC-2.

The "FRAME RESPONSE" columns represent possible Link_Control frame responses for successful ("expected"), or unsuccessful ("alternate") delivery to the N_Port plus R_RDY.

An FC-4 Data protocol may be defined by specifying the request Data Sequence and the reply Data Sequence such as the Link Application protocol. Other protocol models may be derived using Data frame Sequences. The protocol shown represents an example of how a protocol relates to frame flow.

Table 36. Data Protocol Summary

PROTOCOL		FRAME RESPONSE (LINK_CONTROL)	
Request	**Reply**	**Expected**	**Alternate**
Data frame Sequence	**Data frame Sequence**	**Expected ACK**	**Alternate Link_Response**
Class 1			
Data frame	Data reply frame Sequence or No reply Sequence	R_RDY Primitive to **SOF_{c1}**, ACK_1 or ACK_N	F_BSY / PBSY or F_RJT / P_RJT
Class 2			
Data frame	Data reply frame Sequence or No reply Sequence	R_RDY primitive plus ACK_1 or ACK_N	R_RDY primitive plus F_BSY / P_BSY or F_RJT / P_RJT
Class 3			
Data frame	Data reply frame Sequence or No reply Sequence	R_RDY primitive	None

21 Link Applications

21.1 Introduction

Link_Data application frames represent a mandatory FC-4 for each N_Port. These protocols shall operate according to R_RDY Primitive, ACK, and Link_Response rules specified in clause 20.

A Sequence of Link_Data frames is logically viewed as a request or a reply to a previous request. A request Sequence solicits a destination N_Port to perform a function or service. The request may include command, control, data, or status information. A reply Sequence may be transmitted in answer to a request Sequence and may contain command, control, data, or status information.

Link_Data Sequences provide "Link Application" or N_Port level functions required to support an N_Port. Combinations of Request and Reply Sequences constitute a Link Application Upper Level Protocol.

21.2 Routing control

Since Link Application requests and replies may deal with low-level FC-2 functions, an N_Port or F_Port may use the Link_Data identification in the DL bits in R_CTL in order to route the frame. All Link application frames set TYPE = Link Application.

21.3 Link_Application command codes

When TYPE indicates Link Application, the first word of the Payload (LA_Command code) of the Request or Reply Sequence shall be encoded as shown in table 37 with bits 23-0 reserved.

Table 37. LA_Command codes

Encoded Value (Bits 31-24)	Description	Abbr.
0000 0000	No Operation	NOP
0000 0001	Link_Application Reject	LA_RJT
0000 0010	Accept	ACC
0000 0011	Login	LOGI
0000 0100	Logout	LOGO
0000 0101	Abort Exchange	ABTX
0000 0110	Abort Sequence	ABTS
0000 0111	Remove Connection	RMC
0000 1000	Read Connection Status	RCS
0000 1001	Read Exchange Status Block	RES
0000 1010	Read Sequence Status Block	RSS
0000 1011	Establish Streaming	ESTS
0000 1100	Estimate Credit	ESTC
0000 1101	Advise Credit	ADVC

21.4 Link_Data Requests

A Sequence Initiator transmits a Link_Data Sequence in order to solicit the destination N_Port to perform a Link-Level function or service. A Link_Data Protocol is composed of a Link_Data request Sequence and a Link_Data reply Sequence. The last Data frame of a Link_Data request Sequence transfers the Sequence Initiative to the Recipient in order to allow the reply to be transmitted. (See clause 24.) The following Link_Data Protocols are defined:

- Login
- Logout
- Abort Exchange
- Abort Sequence
- Read Connection Status
- Read Exchange Status Block
- Read Sequence Status Block
- Establish Streaming

— Advise Credit

The following Link_Data requests have no reply Sequence:

— No Operation
— Remove Connection
— Estimate Credit

This section specifies Link_Data request Sequences. In the Link commands described in the following sections, emphasis is placed on the function performed by the Link_Data Request and its associated Link_Data Reply Sequences. Frame_Header fields used in normal Exchange and Sequence management are not discussed in detail. See clause 24 for more information regarding use of F_CTL Bits for Exchange and Sequence control as well as OX_ID, RX_ID, SEQ_ID, and Sequence Count.

Payload
Link_Data requests and Link_Data reply frames use an FT_1 frame type. The Payload contains a multiple of four bytes of data based on the individual Link_Data Request or Reply. If required, more than one frame may be used to form a request or reply Sequence.

21.4.1 Login (LOGI)

The Login Link_Data frame shall transfer Service Parameters from the initiating N_Port to the N_Port associated with the Destination Identifier using single InBand addressing. The LOGI frame provides the means by which an N_Port may request "Login" with a Fabric or another N_Port prior to other Data frame transfers (see 23.1).

The interchange of Service Parameters, which includes World_Wide_Names, establishes the operating environment between the two N_Ports. Three Classes of Service are available depending on the support provided by the Fabric, if present. (See 23.5 for a definition of Service Parameters).

In order to Login with the Fabric and determine the Fabric operating characteristics, an N_Port specifies the Destination Identifier as the well-known Fabric F_Port Identifier (i.e. hexadecimal 'FFFFFE').

In order to direct the Login Link_Data frame to the Fabric Server, an N_Port specifies the appropriate well-known Address Identifier (see table 26 in 18.3).
Protocol:

Login request Sequence
Accept reply Sequence

Delimiters: SOFc1, SOFix, EOFn.
Format: FT_1
Addressing: The S_ID field designates the source N_Port requesting Login. If unknown, as in Fabric Login, binary zeros are used. The D_ID field designates the destination N_Port of the Login.
Payload: The Payload shall be 68 bytes in size. The first word shall contain the LA_Command code. The next 48 bytes shall contain the Service Parameters for all Classes of Service. The next eight bytes shall contain the World_wide_Name of the Originator N_Port initiating the LOGI Data frame. Service Parameters are defined in 23.5. The remaining bytes of this field are reserved.
Reply Link_Data Sequence:

— Application Reject (LA_RJT)
  signifies rejection of the LOGI command (See 21.5.2)
— Accept (ACC)
  signifies successful completion of the LOGI command
— Accept Payload
  contains the LA_Command code followed by the Service Parameters and World_wide_Name of the Responder N_Port or the Fabric in the same format as the LOGI Data frame. (See 23.5)

21.4.2 Logout (LOGO)

The Logout Link_Data frame shall request invalidation of the Service Parameters and World_wide_Name which have been saved by an N_Port, freeing those resources. This provides a means by which an N_Port may request "Logout" or remove Service between two N_Ports. (See 23.4)
Protocol:

Logout request Sequence
Accept reply Sequence

Delimiters: SOFc1, SOFix, EOFn.
Format: FT_1
Addressing: The S_ID field designates the source N_Port requesting Logout. The D_ID field designates the destination N_Port of the Logout request.

Payload: The Payload shall contain the LA_Command code.

Reply Link_Data Sequence:

— Application Reject (LA_RJT)
  signifies rejection of the LOGO command (See 21.5.2)

— Accept (ACC)
  signifies that Service has been removed.

— Accept Payload:
  contains the LA_Command code.

21.4.3 Abort Exchange (ABTX)

The Abort Exchange Link_Data frame shall be used to request abnormal termination of an Exchange in progress. The Payload shall contain the OX_ID and assigned RX_ID for the Exchange being aborted, as well as the S_ID of the N_Port which initiated the Exchange. Resources associated with the OX_ID in the Originator, and with the RX_ID in the Responder, shall be released. Either the Originator or Responder are permitted to Abort an Exchange.

Transmission of an ABTX frame is allowed while the identified Exchange is active. The Responder shall insure that the RX_ID being terminated is currently associated with the OX_ID specified in the ABTX request. If the Sequence Initiator has indicated that X_ID reassignment is required during Login, the Sequence Initiator shall include the Association Header in the ABTX frame.

Both the OX_ID and RX_ID are available by the respective N_Ports for reuse after the Exchange has been successfully aborted. Any active Sequences associated with the Exchange are also abnormally terminated by each N_Port The ACC reply confirms that all Sequences and the Exchange have been abnormally terminated. The Abort Sequence Protocol is not explicitly required.

The addressing is fully specified by the D_ID and S_ID fields using InBand addressing.

Protocol:

Abort Exchange request Sequence
Accept reply Sequence

Delimiters: **SOF_{nx}**, **EOF_{n}**.

Format: FT_1

Addressing: The D_ID field designates the destination N_Port of the Exchange being aborted while the S_ID field designates the source N_Port which is requesting that the Exchange be aborted.

X_ID: A separate and distinct Exchange is required other than the Exchange being aborted in order to properly track status.

SEQ_ID, SEQ_CNT: The SEQ_ID and the SEQ_CNT shall be appropriate for an active Sequence.

Payload: The first word shall contain the LA_Command code. The first byte of the second word shall be reserved. The second, third, and fourth bytes of the second word shall contain the S_ID of the N_Port which originated the Exchange. The third word contains the Exchange_IDs. The first two bytes of the third word shall contain the OX_ID and the third and fourth bytes of the second word shall contain the RX_ID for the Exchange being aborted. If the Sequence Recipient has indicated that X_ID reassignment is required during Login, the Sequence Initiator shall include the Association Header in the Payload of the ABTX request associated with the Exchange being aborted immediately following the first three words.

Reply Link_Data Sequence:

— Application Reject (LA_RJT)
  signifies rejection of the ABTX command (See 21.5.2)

— Accept (ACC)
  signifies that the destination N_Port has terminated the Exchange.

— Accept Payload
  shall contain the LA_Command code.

21.4.4 Abort Sequence (ABTS)

The Abort Sequence Link_Data frame shall be used to request abnormal termination of the Sequence in progress which is identified by the SEQ_ID of this frame. The RX_ID and OX_ID specified shall be associated with the Sequence (SEQ_ID) being terminated. Resources associated with the SEQ_ID shall be released.

Transmission of the ABTS frame shall only be allowed before the identified Sequence (SEQ_ID) is terminated. The Sequence Initiator shall transfer the Sequence Initiative to the Recipient by the appropriate F_CTL bit setting.

The Sequence Recipient may request that an active Sequence in progress be aborted by setting the Abort Sequence Condition bits to values of 0 1 or 1 0 on an ACK frame.

After receiving the ABTS frame, the Recipient shall insure that the SEQ_ID being terminated is currently associated with the SEQ_ID specified in the ABTS request. The Recipient of the ABTS frame shall withhold transmission of the Accept frame until all frames of the Sequence have been accounted for or timed out. Data frames for the Sequence may be processed, or discarded based on the policy of the FC-4 receiving the Data. One Sequence may be aborted by the Recipient. See 29.7.1.

The SEQ_ID shall be retired by the Sequence Initiator until the Accept reply Sequence is received. For purposes of further Data transmission the Exchange using this Sequence is suspended. The SEQ_ID shall be available for reuse by the Recipient after transmission of the Accept. The Recipient shall update the status of the Sequence in the Exchange Status Block.

The addressing is fully specified by the D_ID and S_ID fields using InBand addressing.
Protocol:

Abort Sequence request Sequence
Accept reply Sequence

Delimiters: **SOFnx**, **EOFn**.
Format: FT_1
Addressing: The D_ID field designates the destination N_Port (Sequence Recipient) of the Sequence being aborted while the S_ID field designates the source N_Port (Sequence Initiator) which is requesting that the Sequence be aborted.
X_ID: Both the RX_ID and OX_ID shall correspond to the Sequence being aborted.
SEQ_ID, SEQ_CNT: The SEQ_ID shall be appropriate for the active Sequence. The SEQ_CNT shall be set to its normal value for the Sequence in progress, indicating the highest SEQ_CNT transmitted for this SEQ_ID.
Payload: The Payload shall contain the LA_Command code.

Reply Link_Data Sequence:

- Application Reject (LA_RJT)
  signifies rejection of the ABTS command (See 21.5.2)
- Accept (ACC)
  signifies that the destination N_Port has terminated the Sequence.
- Accept Payload
  The first word is the LA_Command code. Following the command code, the aborted Sequence is identified by two words of data. The first byte of the two words shall specify the SEQ_ID being aborted. The second, third, and fourth bytes of the two words shall contain the S_ID of the N_Port which originated the Exchange. The fifth and sixth bytes of the two words shall contain the OX_ID and the seventh and eighth bytes of the two words shall contain the RX_ID for the Sequence being aborted.

21.4.5 No Operation (NOP)

The No Operation Link_Data frame shall be used with any **SOF** and **EOF** delimiters appropriate to the Class in which it is being used. The NOP frame is discarded by the recipient. However, the F_CTL field and the **SOF** and **EOF** delimiters are examined and the appropriate action taken by both the N_Port and Fabric, if present.
Protocol:

No Operation request

Delimiters: **SOFc1**, **SOFix**, **SOFnx**, **EOFn**, **EOFt**, **EOFdt**.
Format: FT_1
Addressing: The D_ID field designates the destination of the frame while the S_ID field designates the source of the frame.
Payload: The first word of the Payload shall contain the LA_Command code. The remaining Payload size and content shall be unspecified and may contain any appropriate number of words.
Reply Link_Data Sequence:

- none or another NOP

21.4.6 Remove Connection (RMC)

The Remove Connection Link_Data frame shall be used with the **EOF**$_{dt}$ to remove a Class 1 Dedicated Connection. The RMC frame shall be discarded by the Recipient, however, the F_CTL field and the **EOF**$_{dt}$ shall be examined and processed appropriately by both the N_Port and Fabric, if present.

Protocol:

Remove Connection request

Delimiters: SOF$_{c1}$, **SOF**$_{i1}$, **SOF**$_{n1}$, **EOF**$_{dt}$
Format: FT_1
Addressing: The D_ID field designates the Recipient of the Sequence while the S_ID field designates the Initiator of the Sequence.
Payload: The Payload contains the LA_Command code.
Reply Link_Data Sequence:

— none

21.4.7 Read Connection Status (RCS)

The RCS Link_Data frame requests the Fabric Controller to return the current Dedicated Connection status for the N_Port specified in four bytes of the Payload of the RCS frame. The RCS request provides the means by which an N_Port may interrogate the Fabric for its current Connection status or the Connection status of other N_Ports within the Fabric.

In order to direct the RCS Link_Data request frame to the Fabric, an N_Port specifies the Destination Identifier as a well-known Fabric F_Port address (i.e. hexadecimal 'FFFFFE').

Protocol:

Read Connection Status request Sequence
Accept (ACC) reply Sequence

Delimiters: SOF$_{c1}$, **SOF**$_{ix}$, **EOF**$_{n}$.
Format: FT_1
Addressing: The S_ID field designates the source N_Port requesting Connection status. The D_ID field is the Fabric F_Port, hexadecimal 'FFFFFE'.

Payload: The first word of the Payload shall contain the LA_Command code. The second word shall contain the N_Port address identifier for which Connection status is being requested. The format of the second word of the Payload is specified in figure 46.

RCS Payload Definition – Second Word

	First Byte	Second Byte	Third Byte	Fourth Byte
Bits	31 24	23 16	15 8	7 0
	rrrr rrrr	AAAA AAAA	AAAA AAAA	AAAA AAAA

Reserved (First Byte); Address Identifier (Second through Fourth Byte)

Figure 46. Connection address identifier

Bits 31 - 24 Reserved

Bits 23 - 0

Bits 23 through 0 specify the address identifier of the N_Port whose Connection status is being requested. If the address is the same as the address of the source N_Port, then the status returned indicates whether the N_Port transmitting the RCS Link_Data request frame is currently Connected and to which N_Port.

Reply Link_Data Sequence:

— Application Reject (LA_RJT)
signifies rejection of the RCS command (See 21.5.2)
— Accept (ACC)
signifies that the Fabric has completed the request.
— ACC Payload
The first word of the Payload shall contain the LA_Command code. The second word shall contain the Connection status and the address identifier to which the requested N_Port is Connected, if that N_Port is in a Dedicated Connection. The format of the Payload is specified in figure 47.

ACC Payload Definition - second word

	First Byte	Second Byte	Third Byte	Fourth Byte
Bits	3 2 1 4	2 111 3 876	1 5 8	 7 0
	SSSS rrrr	AAAA AAAA	AAAA AAAA	AAAA AAAA
	Status	Address Identifier		

Figure 47. Connection status information

Connection Status Codes:

Bit 31 - Connect-request delivered

If bit 31 is zero, the specified N_Port is either not Connected, or is involved in an Established Connection based on the setting of bit 29. If bit 31 is one, a connect-request has been delivered to the specified N_Port, but the N_Port has not yet responded with a proper response frame and a Dedicated Connection does not yet exist.

Bit 30 - Connect-request queued

If bit 30 is zero, no connect-request is queued for the specified N_Port. If bit 30 is one, a connect-request is queued, but has not been delivered to the specified N_Port.

Bit 29 - Connection Established

If bit 29 is zero, the specified N_Port in the RCS request is not in a Dedicated Connection. If bit 29 is one, the specified N_Port is involved in a Dedicated Connection. When bit 29 is one, the address identifier in bits 23-0 identifies the N_Port involved in the Dedicated Connection.

Bit 28 - Intermix

If bit 28 is zero, the N_Port specified in the RCS frame is not supporting intermix. If bit 28 is one, the N_Port specified in the request is actively supporting intermix for Class 2 or Class 3 when a Dedicated Connection exists. See 23.5. for more discussion of intermix.

Bits 23 - 0

Bits 23 through 0 specify the address identifier of the N_Port involved in a Dedicated Connection with the N_Port specified by the RCS Link_Data request Sequence frame.

21.4.8 Read Exchange Status Block (RES)

The RES Link_Data frame requests an N_Port to return the Exchange Status Block for the RX_ID or OX_ID originated by the S_ID specified in the Payload of this frame. This provides the N_Port transmitting the request with information regarding the current status of the Exchange specified.

Protocol:

Read Connection Status request Sequence
Accept (ACC) reply Sequence.

Delimiters: SOF_{c1}, SOF_{ix}, EOF_{n}.

Format: FT_1

Addressing: The S_ID field designates the source N_Port requesting the Exchange Status Block. The D_ID field designates the destination N_Port to which the request is being made.

Payload: The first word of the Payload shall contain the LA_Command code. The first byte of the second word of the Payload shall be reserved. The second, third, and fourth bytes of the second word shall contain the S_ID of the N_Port which originated the Exchange. The first and second bytes of the third word of the Payload shall contain the OX_ID. The third and fourth bytes of the third word shall contain the RX_ID. The OX_ID and RX_ID specified are associated with the Exchange Status Block being requested. If the Sequence Recipient has indicated that X_ID reassignment is required during Login, the Sequence Initiator shall include the Association Header in the Payload of the RES request immediately following the first twelve bytes.

RES Payload Definition - after LA_Command

	First Byte	Second Byte	Third Byte	Fourth Byte
Bits	3 2 1 4	2 111 3 876	1 5 8	 7 0
2nd Wd	rrrr rrrr	Originating S_ID		
3rd Wd	OX_ID		RX_ID	

Figure 48. RES Payload

Reply Link_Data Sequence:

- Application Reject (LA_RJT)
  signifies rejection of the RES command (See 21.5.2)
- Accept
  signifies that the N_Port has transmitted the requested data.
- Accept Payload:
  The first word of the Payload shall contain the LA_Command code. The remainder of the Payload shall contain the information representing the Exchange Status Block for the RX_ID or OX_ID specified in the RES request Sequence. The format of the Exchange Status Block is specified in 29.11.3.

21.4.9 Read Sequence Status Block (RSS)

The RSS Link_Data frame requests an N_Port to return the Sequence Status Block for the SEQ_ID specified in the Payload of this frame. The Payload also specifies the S_ID of the Sequence Initiator as well as the associated OX_ID and RX_ID. This provides the N_Port transmitting the request with information regarding the current status of the Sequence it identified.

Protocol:

Read Sequence Status request Sequence
Accept (ACC) reply Sequence

Delimiters: SOF$_{c1}$, SOF$_{ix}$, EOF$_{n}$.
Format: FT_1
Addressing: The S_ID field designates the source N_Port requesting the Sequence Status Block. The D_ID field designates the destination N_Port to which the request is being made.

Payload: The first word of the Payload shall contain the LA_Command code. The first byte of the second word of the Payload shall contain the SEQ_ID of the Status Block being requested. The second, third, and fourth bytes of the second word contain the S_ID of the N_Port which initiated the Sequence. The first and second bytes of the third word of the Payload shall contain the OX_ID. The third and fourth bytes of the third word shall contain the RX_ID. The OX_ID and RX_ID specified are associated with the Sequence Status Block being requested.

RSS Payload Definition - after LA_Command

	First Byte	Second Byte	Third Byte	Fourth Byte
Bits	3 2 1 4	2 111 3 876	1 5 8	 7 0
2nd Wd	SEQ_ID	Originating S_ID		
3rd Wd	OX_ID		RX_ID	

Figure 49. RES Payload

Reply Link_Data Sequence:

- Application Reject (LA_RJT)
  signifies rejection of the RSS command (See 21.5.2)
- Accept
  signifies that the N_Port has transmitted the requested data.
- Accept Payload:
  The first word of the Payload shall contain the LA_Command code. The remainder of the Payload shall contain the information representing the Exchange Status Block for the RX_ID or OX_ID specified in the RSS request Sequence. The format of the Sequence Status Block is specified in 29.11.4.

21.4.10 Establish Streaming (ESTS)

The ESTS Link_Data requests a temporary allocation of Credit known as Streaming Credit large enough to perform continuous streaming of Data frames. (See 23.7.1.1 for the usage of this frame).

Protocol:

Establish Streaming request Sequence
Accept reply Sequence

Delimiters: SOFc1, SOFix, EOFn.
Format: FT_0
Addressing: The S_ID field designates the source N_Port requesting Streaming. The D_ID field designates the destination N_Port addressed.
Payload: The Payload shall contain the LA_Command code.
Reply Link_Data Sequence:

- Application Reject (LA_RJT) signifies rejection of the RSS command (See 21.5.2)
- Accept (ACC) signifies successful completion of the ESTS function.
- Accept Payload shall contain Payload in the same format as the ACC response to LOGI Link_Data frame for Service Parameters. The Payload shall contain Streaming Credit (L) allocated in Credit field of the appropriate Class of the Service Parameters. The other Service Parameter fields shall be ignored by the receiver.

21.4.11 Estimate Credit (ESTC)

The ESTC Link_Data request shall be used to estimate the minimum Credit required to achieve the maximum bandwidth for a given distance between an N_Port pair. The ESTC Link_Data request shall have the maximum frame size as determined by Login with the destination N_Port.

The ESTC Link_Data frame shall be transmitted as a streamed Sequence. The destination N_Port shall respond with an ACK_1 for each ESTS Link_Data frame received. (See 23.7.1.2 for the usage of this frame.)

Protocol:

Estimate Credit request Sequence
No reply Sequence

Delimiters: SOFix, EOFn.
Format: FT_1
Addressing: The S_ID field designates the source N_Port requesting the Credit estimate. The D_ID field designates the destination N_Port specified in the Establish Streaming frame.
Data Field: The first word of the Data Field shall contain the LA_Command code. The remainder of the Data Field size shall be the maximum as determined by Login. The content of the Data Field after LA_Command shall be valid data characters.
Reply Link_Data Sequence:

- None.

21.4.12 Advise Credit (ADVC)

The ADVC Link_Data request shall be used to advise the destination N_Port of the estimated Credit which the source N_Port requests to be allocated. (See 23.7.1.3 for the usage of this frame) The ADVC request may also be used independently from the Estimate Credit procedure.

Protocol:

Revise Credit request Sequence
Accept reply Sequence

Delimiters: SOFix, EOFn.
Format: FT_1
Addressing: The S_ID field designates the source N_Port requesting Credit revision. The D_ID field designates the destination N_Port.
Payload: The first word of the Payload shall contain the LA_Command code. The data which follows is in the same format LOGI Link_Data frame for Service Parameters. The Payload shall contain estimated Credit (M+1) in Credit field of the appropriate Class of the Service Parameters. The other Service Parameter fields shall be ignored by the receiver.
Reply Link_Data Sequence:

- Application Reject (LA_RJT) signifies rejection of the ADVC command (See 21.5.2)
- Accept (ACC) signifies successful completion of the ADVC function and modifies the actual Credit value.
- Accept Payload shall contain data in the same format as the ACC response to a LOGI Link_Data frame for Service Parameters. The Payload shall contain revised Credit allocated in Credit field of the appropriate Class of the Service Parameters. The other Service Parameter fields shall be ignored by the receiver. This revised Credit value is determined by the desti-

nation N_Port based on its buffering scheme, buffer management, buffer availability, and N_Port processing time. (see 23.7.1.3 for the determination of this value)

21.5 Link_Data Reply Sequences

A Link_Data reply Sequence signifies that the Link_Data request Sequence is completed. The reply Sequence may contain data in the Payload following the LA_Command code word. The format and meaning of the Payload is specified in the request Link_Data definition. Link_Data frames set DL Bits 31-30 in R_CTL to 1 0.

21.5.1 Accept (ACC)

The Accept Link_Data reply Sequence notifies the transmitter of a Link_Data request that a previous Link_Data Sequence has been completed. The first word of the Payload contains the LA_Command code. The remainder of the Payload is unique to the request Link_Data Sequence being replied to.

Protocol

Accept is the reply Sequence to Login, Logout, Abort Exchange, Abort Sequence, Read Connection Status, Read Exchange Status Block, and Read Sequence Status Block, Establish Streaming, and Advise Credit request Sequences.

Delimiters: SOFnx, EOFt, EOFdt
Format: FT_1
Addressing: The D_ID field designates the source of the Link_Data frame being accepted while the S_ID field designates the destination of the request Data frame Sequence being accepted.
Payload: The Payload content following the LA_Command code is defined within individual Link_Data requests.
Reply Link_Data Sequence:

— none

21.5.2 Link Application Reject (LA_RJT)

The Link Application Reject (LA_RJT) notifies the transmitter of a Link_Data request that a previous Link_Data Sequence has been rejected. A four-byte reason code is contained in the Data_Field. Application Rejects may be transmitted for a variety of conditions which may be unique to a specific Link_Data request.

For example, if the Service Parameters specified in a Login frame were logically inconsistent or in error, a P_RJT frame would not be transmitted in response, but rather an Application Reject from the Upper Level.

Protocol

LA_RJT may be a reply Sequence to any Link Application request.

Delimiters: SOFnx, EOFt, EOFdt
Format: FT_1
Addressing: The D_ID field designates the source of the Link_Data request being rejected while the S_ID field designates the destination of the request Data frame Sequence being rejected.
Payload: The first word of the Payload shall contain the LA_Command code. The next four bytes of this field indicate the reason for rejecting the request. (See figure 50 and tables 38 and 39.)

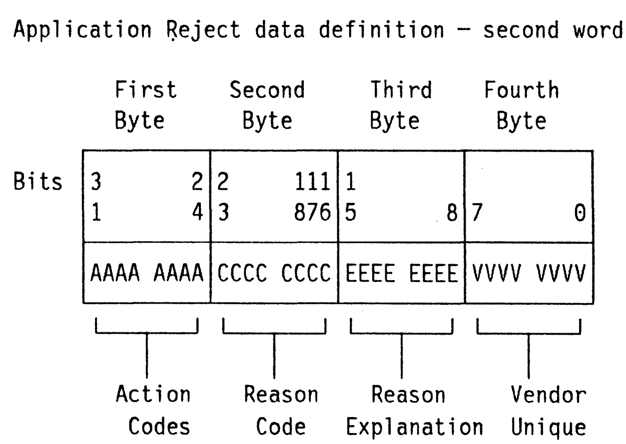


Figure 50. LA_RJT format

Table 38. Action Codes	
Encoded Value bits 31-24	**Description**
0000 0001 (1)	Abort Sequence requested - retry after correction
0000 0010 (2)	Abort Sequence requested - non-retryable error
Other codes	Reserved

Action 1

Action 1 indicates that the Recipient is requesting the source of the rejected frame to abort the Sequence. The Sequence is retryable if the corrective action indicated by the reason code is performed:

— Logical error
— Logical busy
— Protocol error
— Vendor unique error

Action 2

Action 2 indicates that the Recipient is requesting the source of the rejected frame to abort the Sequence or Exchange. The Sequence is non-retryable for the following reason codes:

— Invalid LA_Command code
— Vendor unique error

Table 39. Link Application reason codes	
Encoded Value (Bits 23-16)	**Description**
0000 0001	Invalid LA_Command code
0000 0011	Logical error
0000 0101	Logical busy
0000 0111	Protocol error
Others	Reserved
1111 1111	Vendor Unique Error (See Bits 7-0)

Reject reason codes are arranged in descending priority as the encoded value increases. Therefore, if invalid LA_Command code is the reject reason code, no additonal checking of the frame is required and no other errors are reported. The first error encountered according to the priority specified is the error reported and further error checking is not required.

Bits 15-8 specify an additional reason explanation which may be unique for a specific request.

• Bits 23-16 Description

Invalid LA_Command code

The LA_Command code in the Sequence being rejected is invalid or not supported for the Link Application TYPE.

Logical error

The request identified by the LA_Command code and Payload content is logically inconsistent for the conditions present.

Logical busy

The FC-4 identified by the Type field is logically busy and unable to process the request at this time.

Protocol Error

This indicates that an error has been detected which violates the rules of FC-4 protocol which are not specified by other error codes.

Vendor Unique Error

The Vendor Unique Error bits shall be used by specific Vendors to specify additional reason codes.

• Bits 15-8 Reason explanation

The reason explanation table shown (table 40) is based on an LA_RJT to a Login request. The explanations listed help identify the specific field in the Service Parameters which is in error.

Table 40 (Page 1 of 2). Application Reject Explanation	
Encoded Value (Bits 15-8)	**Description**
0000 0001	Service Parm error - Options
0000 0011	Service Parm error - Transmitter Ctl
0000 0101	Service Parm error - Receiver Ctl
0000 0111	Service Parm error - Rec Data field size

Table 40 (Page 2 of 2). Application Reject Explanation

Encoded Value (Bits 15-8)	Description
0000 1001	Service Parm error - Concurrent Seq
0000 1011	Service Parm error - Credit
0000 1111 to 1111 1110	Reserved

Reply Link_Data Sequence:

— none

21.5.3 No Operation (NOP)

The No Operation Link_Data frame shall be used with any **SOF** and **EOF** delimiters appropriate to the Class in which it is being used. The NOP frame shall be discarded by the Recipient, however, the F_CTL field and the **SOF** and **EOF** delimiters are examined and the appropriate action taken by both the N_Port and Fabric, if present.

Delimiters: **SOFc1**, **SOFix**, **SOFnx**, **EOFn**, **EOFt**, **EOFdt**.

Format: FT_1

Addressing: The D_ID field designates the destination of the frame while the S_ID field designates the source of the frame.

Payload: The first word of the Payload shall contain the LA_Command code. The remaining Payload size and content shall be unspecified and may contain any appropriate number of words.

Reply Link_Data Sequence:

— none, or
— NOP

21.6 Link Application Summary

Table 41 summarizes the Link Application request and reply Sequences. The Payload content and size in bytes are specified.

Table 41. Link Application Payload

REQUEST SEQUENCE Request	Payload	REPLY SEQUENCE Reply	Payload
Login (LOGI)	LA_Command code, Service Parameters (Class 1, 2, 3) World_Wide_Name, (68 bytes)	Accept (ACC) or A_RJT	LA_Command code, Service Parameters (Class 1, 2, 3) World_Wide_Name, (68 bytes)
Logout (LOGO)	LA_Command code (4 bytes)	Accept (ACC) or A_RJT	LA_Command code (4 bytes)
Abort Exchange (ABTX)	LA_Command code, S_ID, OX_ID , RX_ID (12 bytes)	Accept (ACC) or A_RJT	LA_Command code (4 bytes)
Abort Sequence (ABTS)	LA_Command code (4 bytes)	Accept (ACC) or A_RJT	LA_Command code (4 bytes)
No Operation (NOP)	LA_Command code (4 bytes)	No Operation, or None	LA_Command code (4 bytes)
Remove Connection (RMC)	LA_Command code (4 bytes)	None	LA_Command code (4 bytes)
Read Connection Status (RCS)	LA_Command code, Address_Identifier (8 bytes)	Accept (ACC) or A_RJT	LA_Command code, Status, Address_Identifier (8 bytes)
Read Exchange Status Block (RES)	LA_Command code, Originating S_ID, OX_ID , RX_ID (12 bytes)	Accept (ACC) or A_RJT	LA_Command code, Exchange SB Format (N bytes)
Read Sequence Status Block (RSS)	LA_Command code, SEQ_ID, S_ID, RX_ID, OX_ID (12 bytes)	Accept (ACC) or A_RJT	LA_Command code, Sequence SB format (N bytes)
Establish Streaming (ESTS)	LA_Command code (4 bytes)	Accept (ACC) or A_RJT	LA_Command code, Service Parameters (52 bytes)
Estimate Credit (ESTC)	LA_Command code, Data (maximum size)	None	LA_Command code (4 bytes)
Advise Credit (ADVC)	LA_Command code, Service Parameters (52 bytes)	Accept (ACC) or A_RJT	LA_Command code, Service Parameters (52 bytes)

22 Classes of service

Three Classes of service applicable to a Fabric and an N_Port are specified in FC-2. These Classes of service are distinguished primarily by the methodology with which the communication circuit is allocated and retained between the communicating N_Ports and the level of delivery integrity required for an application. A given Fabric or N_Port may support one or more Classes of service. These Classes of service are:

1. Class 1 - Dedicated Connection
2. Class 2 - Multiplex
3. Class 3 - Datagram

Each Class of service may be supported with any of the Communication Models.

22.1 Class 1 -- Dedicated Connection

Class 1 is a service which provides Dedicated Connections. A Class 1 Connection is requested by an N_Port with another N_Port. Once a Connection is established, it is retained and guaranteed by the Fabric.

22.1.1 Class 1 function

A Class 1 service is requested by an N_Port with another N_Port via transmission of a frame containing a **SOFc1**. The Fabric, if present, allocates a circuit between the requesting N_Port and the destination N_Port. The destination N_Port transmits a frame indicating its acceptance to the requesting N_Port. The Fabric retains the allocated circuit between these two N_Ports, until one of these N_Ports requests the service to be removed.

Even if a Fabric is not present, the requesting N_Port and the destination N_Port follow the same protocol.

Class 1 Delimiters as specified in 22.1.3 are used to establish and remove the service and to initiate and terminate one or more Sequences within the service.

22.1.2 Class 1 rules

The rules specified in this section apply to "exclusive" Class 1 Connections. (See 22.4 for additional rules which apply to Intermix). To provide a Class 1 Connection, the transmitting and receiving N_Ports, and the Fabric shall obey the following rules:

1. An N_Port requesting Class 1 service is required to have logged in with the Fabric and the N_Port(s) with which it intends to communicate, either implicitly or explicitly. To login explicitly, the requesting N_Port shall use Fabric and N_Port Login protocols (See "23 Login and Service Parameters"). **(Login)**
2. The Fabric is responsible for establishing a Connection at the request of an N_Port and retaining it until one of the communicating N_Ports explicitly requests the Connection to be removed. To establish or remove the service, the requesting N_Port shall use the Class 1 Delimiters as specified in 22.1.3. (**Dedicated Connection** through Connection Sub-Fabric)
3. The N_Port requesting the Connection shall not transmit additional frames for that Sequence within the Connection, after the frame that requests the Connection, until it receives confirmation through ACK from the destination N_Port. Once the Connection is confirmed, only frames for that Connection shall be sent until the Connection is removed. **(Dedicated Connection confirmation)**
4. A destination N_Port shall provide a confirmation to the Source through ACK, for each valid frame received, if mutually agreed to during N_Port Login. (See 22.1.5) **(end-to-end confirmation)**
5. The transmitter shall increment SEQ_CNT field of each successive frame transmitted within a Sequence. The Fabric shall guarantee delivery of the frames at the receiver in the same order of transmission within the Sequence. (See 24.11.4) **(sequential delivery)**
6. An N_Port is allowed to perform multiple Exchanges concurrently on an established Connection. The N_Port initiating the Exchanges shall assign unique OX_ID to each Exchange. (See 24.3). **(concurrent multiple Exchanges)**

7. The Fabric does not exercise any flow control. Communicating N_Ports are responsible for the flow control. ACK frames are used to perform the flow control. (See 22.1.5) **(end-to-end flow control)**
8. The Fabric may reject a request for Class 1 service or issue a busy with a valid reason code. Otherwise, once the service is established, the Fabric shall not interfere with the Connection. (See 20.3.3.1) **(Fabric reject or busy)**
9. The destination N_Port specified in the Connection-request frame may respond with a busy or a reject with a valid reason code to this frame only. Once the service is established, the destination N_Port shall not issue busy but may issue a reject. (See 20.3.3.1) **(N_Port busy or reject)**
10. The source of any frame transmission shall use only the address of the N_Port with which it has established the Connection. **(established address)**
11. The Credit established during the Login protocol by interchanging service Parameters shall be honored. (See 25.3) Class 2 and Class 3 share the Credit for connectionless service. (See 25.3) **(Credit)**
12. The Fabric shall guarantee full bandwidth availability to the connected N_Ports. **(bandwidth)**
13. Frames are tracked on an (OX_ID or RX_ID)||SEQ_ID||SEQ_CNT basis. **(tracking)**
14. An N_Port or F_Port shall be able to recognize SOF delimiters of all Classes of service, whether or not all Classes are supported by the N_Port and provide appropriate responses for all Classes. During a Class 1 Connection, unless Intermix is supported, a Class 1 N_Port shall issue a P_RJT for Class 2 frames and discard Class 3 frames. **(invalid Class)**

22.1.3 Class 1 delimiters

A Dedicated Connection is requested by transmitting a frame using an **SOFc1** delimiter. **SOFc1** initiates the first Sequence; subsequent Sequences are initiated with an **SOFi1**. Frames within a Sequence are started by **SOFn1**.

Each Sequence is terminated using an **EOFt**. An **EOFdt** contained in a frame terminates the Sequence in which the frame is sent and it also serves to remove the Dedicated Connection. (Other Sequences in progress are left in an indeterminate state.) All frames other than the last frame within a Sequence are terminated by **EOFn**.

22.1.4 Class 1 frame size

The size of a frame using the **SOFc1** delimiter is limited by the smaller value of the maximum Data Field size supported for frames with **SOFc1** by the Fabric and the destination N_Port. Subsequent frames, after a Dedicated Connection is established, are limited by the maximum Data Field size supported by the destination N_Port.

22.1.5 Class 1 flow control

ACK frames are used to perform Class 1 end-to-end flow control. ACK frames are started with **SOFn1**. If ACK is used to terminate a Sequence, it will end with **EOFt**. If used to terminate the service, it will end with **EOFdt**. Otherwise it will end with **EOFn**.

All Class 1 frames except one with **SOFc1** delimiter shall follow the end-to-end flow control rules. (See 25.4.2). The Class 1 frame with **SOFc1** delimiter shall follow the buffer-to-buffer flow control rules. (See 25.5.4).

22.2 Class 2 -- Multiplex

This operating environment provides Connectionless service with guaranteed notification of non-delivery between two N_Ports. This service allows one N_Port to transmit consecutive frames to multiple destinations without establishing a Dedicated Connection with any specific N_Port. Conversely, this service also allows one N_Port to receive consecutive frames from one or more N_Ports without having established Dedicated Connections with any of them.

22.2.1 Class 2 function

A Class 2 service is requested by an N_Port on a frame by frame basis. The Fabric, if present, routes the frame to the destination N_Port. If the N_Port transmits consecutive frames to multiple destinations, the Fabric demultiplexes them to the requested destinations.

Class 2 Delimiters are used to indicate the requested service and to initiate and terminate one or more Sequences as described in 22.2.3.

Since Class 2 is Connectionless, the question of service removal does not arise.

22.2.2 Class 2 rules

To provide Class 2 service, the transmitting and receiving N_Ports, and the Fabric shall obey the following rules:

1. An N_Port supporting Class 2 service is required to have logged in with the Fabric and the N_Port(s) with which it intends to communicate, either explicitly or implicitly. To login explicitly, the requesting N_Port shall use Fabric and N_Port Login protocols. (See "23 Login and Service Parameters"). **(Login)**
2. The Fabric does not establish a Connection between communicating N_Ports. The Fabric routes the frames through Connectionless Sub-Fabric. To obtain Class 2 service from the Fabric, the N_Port shall use the Class 2 Delimiters as specified in 22.2.3. **(Connectionless service)**
3. A given N_Port is allowed to send consecutive frames to different destinations and the Fabric is responsible to deliver them or inform the Source of its inability to deliver (See 22.2.3). This enables an N_Port to multiplex multiple Sequences to single or multiple destinations concurrently. **(demultiplexing and concurrent multiple Sequences)**
4. If the Fabric is unable to deliver the frame to the destination N_Port, then the Source is notified of each frame not delivered by an F_BSY or F_RJT frame from the Fabric with corresponding D_ID, S_ID, OX_ID, RX_ID, and SEQ_CNT. The Source is also notified of frames not processed by the destination N_Port by P_BSY or P_RJT. **(non-delivery)**
5. A given N_Port may receive consecutive frames from different Sources. Each Source is allowed to send consecutive frames for one or more Sequences. **(multiplexing)**
6. A destination N_Port shall provide a confirmation to the Source for each valid frame received, if mutually agreed to during the N_Port Login. As in Class 1, the destination N_Port shall use ACK for the confirmation (see 22.2.5). If unable to deliver ACK, the Fabric shall return a BSY or RJT. **(end-to-end confirmation)**
7. The transmitter shall increment SEQ_CNT field of each successive frame transmitted within a Sequence. However, the Fabric may not guarantee delivery to the destination in the same order of transmission. The receiving N_Port shall make the frames available to the ULP in the sequential order. (See 24.11.4) **(non-sequential delivery)**
8. The Frontend F_Port of the Fabric exercises buffer-to-buffer flow control with the transmitting N_Port and so does the Backend F_Port with the receiving N_Port. End-to-end flow control is performed by communicating N_Ports. ACK frames are used to perform end-to-end flow control and R_RDY is used for buffer-to-buffer flow control. **(dual flow control)**
9. A busy or reject can be issued by an F_Port or the destination N_Port with a valid reason code. (See 23.6, 23.5, 20.3.3.1, and 20.3.3.2) **(busy/reject)**
10. The Credit established during the Login protocol by interchanging Service Parameters shall be honored. (See 25.3 for more on Credit.) **(Credit)**
11. Frames are tracked on an S_ID||(OX_ID or RX_ID)||SEQ_ID||SEQ_CNT basis. **(tracking)**
12. An N_Port or F_Port shall be able to recognize **SOF** delimiters of all Classes of service, whether or not all Classes are supported by the N_Port, and provide appropriate responses for all Classes. A Class 2 N_Port shall issue a P_RJT or F_RJT for Class 1 frames and discard Class 3 frames. **(invalid Class)**

22.2.3 Class 2 delimiters

Sequences are initiated by transmitting a frame started by an **SOFi2**. Subsequent frames within a Sequence are started by an **SOFn2**. A Sequence is terminated with a frame ended by **EOFt**. All frames other than the last frame within the Sequence are terminated by an **EOFn**.

22.2.4 Class 2 frame size

The size of each frame transmitted is limited by the smaller value of the maximum Data Field size supported by the Fabric or by the receiving N_Port. Each frame is routed through the Fabric, if present, as a separate entity.

22.2.5 Class 2 flow control

Class 2 service uses both buffer-to-buffer and end-to-end flow controls. R_RDY (Receiver Ready) is used for buffer to buffer flow control. R_RDY is transmitted by the Fabric entry point to the N_Port originating the Class 2 frame to permit transmission of the next frame from the originating N_Port. R_RDY is transmitted by the destination N_Port to the Fabric exit point to permit delivery of the next frame to that destination N_Port.

ACK frames are used to perform end-to-end flow control. ACK frames shall begin with **SOFn2**. ACK response to the last Data frame of a Sequence shall end with **EOFt**. Otherwise it will end with **EOFn**.

All Class 2 frames shall follow the buffer-to-buffer flow control rules. (See 25.5.4).

22.3 Class 3 -- Datagram

This operating environment provides Connectionless service without any notification of non-delivery (BSY or RJT), delivery (ACK), or end-to-end flow control between two N_Ports. The Fabric is allowed to discard Class 3 frames without any notification to the transmitting N_Port. This service allows one N_Port to transmit consecutive frames to multiple destinations without establishing a "Dedicated Connection" with any specific N_Port. Conversely, this service also allows one N_Port to receive consecutive frames from one or more N_Ports without having established Dedicated Connections with any of them.

22.3.1 Class 3 function

A Class 3 service is requested by an N_Port on a frame by frame basis. The Fabric, if present, routes the frame to the destination N_Port. If the N_Port transmits consecutive frames to multiple destinations, the Fabric demultiplexes them to the requested destinations.

Class 3 Delimiters are used to indicate the requested service and to initiate and terminate one or more Sequences as described in 22.3.3. Since Class 3 is Connectionless, the question of service removal does not arise.

22.3.2 Class 3 rules

To provide Class 3 service, the transmitting and receiving N_Ports, and the Fabric shall obey the following rules:

1. An N_Port supporting Class 3 service is required to have logged in with the Fabric, either explicitly or implicitly. To login explicitly, the requesting N_Port shall use Fabric Login protocol. N_Port Login is optional for Class 3. (See "23 Login and Service Parameters") **(Login)**
2. The Fabric does not establish or retain a Connection between communicating N_Ports. The Fabric routes the frames through Connectionless Sub-Fabric. To obtain Class 3 service from the Fabric, the N_Port shall use the Class 3 Delimiters as specified in 22.3.3. **(Connectionless service)**
3. A given N_Port is allowed to send consecutive frames to different destinations. This enables an N_Port to multiplex multiple Sequences to single or multiple destinations concurrently. **(demultiplexing and concurrent multiple Sequences)**
4. If the Fabric is unable to deliver the frame to the destination N_Port, the frame is discarded and the Source is not notified. If the destination N_Port is unable to receive the frame, the frame is discarded and the Source is not notified. **(non-delivery)**
5. A given N_Port may receive consecutive frames from different Sources. Each Source is allowed to send consecutive frames for one or more Sequences. **(multiplexing)**
6. A destination N_Port shall not provide confirmation (ACK) to the Source for any valid frame received. **(end-to-end confirmation)**
7. The transmitter shall increment SEQ_CNT field of each successive frame transmitted within a Sequence. However, the Fabric may not guarantee delivery at the receiver in the same order of transmission. The receiving N_Port shall make the frames available to the ULP in the sequential order. (See 24.11.4). **(non-sequential delivery)**
8. The Frontend F_Port of the Fabric exercises buffer-to-buffer flow control with the transmitting N_Port and so does the Backend F_Port with the receiving N_Port. R_RDY is used for buffer-to-buffer flow control. (See 22.3.5) **(buffer to buffer flow control)**
9. Neither the F_Port nor N_Port shall issue busy or reject. **(busy/reject)**

10. The Fabric Credit is used for buffer to buffer flow control. N_Port Credit is not used. (See 25.3) **(Credit)**
11. Frames are tracked on an S_ID||(OX_ID or RX_ID)||SEQ_ID||SEQ_CNT basis. **(tracking)**
12. An N_Port or F_Port shall be able to recognize **SOF** delimiters of all Classes of service, whether or not all Classes are supported by the N_Port, and provide appropriate responses for all Classes. A Class 3 N_Port shall issue a P_RJT or F_RJT for Class 1 and Class 2 frames. **(invalid Class)**

22.3.3 Class 3 delimiters

Sequences are initiated by transmitting a frame started by an **SOFi3**. Subsequent frames within a Sequence are started by an **SOFn3**. A Sequence is terminated with a frame ended by **EOFt**. All frames other than the last frame within the Sequence are terminated by an **EOFn**.

22.3.4 Class 3 frame size

The size of each frame transmitted is limited by the maximum Data Field size supported by the Fabric and by the receiving N_Port. Each frame is routed through the Fabric, if present, as a separate entity.

22.3.5 Class 3 flow control

Class 3 uses only buffer to buffer flow control with R_RDY. End-to-end flow control is not supported.

All Class 3 frames shall follow the buffer-to-buffer flow control rules. (See 25.5.4).

22.4 Intermix

Intermix is an option of Class 1 service which allows interleaving of Class 2 and Class 3 frames during an established Class 1 Connection between two N_Ports. During a Class 1 Connection, an N_Port capable of Intermix may transmit and receive Class 2 and Class 3 frames interleaved with Class 1 frames. The N_Ports on Class 1 Connection may also transmit and receive Class 2 or Class 3 frames with Class 1. Support for the Intermix option of Class 1 service is specified during Login. An N_Port must be able to receive intermixed frames before it is allowed to transmit intermixed frames.

In a point-to-point topology, both interconnected N_Ports shall be required to support Intermix if Intermix is to be employed. In the presence of a Fabric, both the N_Port and the Fabric Port (F_Port) shall be required to support Intermix if Intermix is to be employed. Fabric support for Intermix requires that the full Class 1 bandwidth during a Dedicated Connection be maintained. Intermix permits the unused Class 1 bandwidth for transmission of Class 2 and Class 3 frames.

22.4.1 Fabric management

If an exit F_Port receives a Class 2 frame while the destination N_Port is engaged in a Dedicated Connection and either the F_Port or the destination N_Port do not support Intermix, an F_BSY is returned to the source N_Port of the frame. If an exit F_Port receives a Class 3 frame while the destination N_Port is engaged in a Dedicated Connection and either the F_Port or the destination N_Port do not support Intermix, the frame is discarded.

During an established Class 1 Connection with Intermix supported, Class 1 frames have priority over Class 2 or Class 3 frames. Class 2 and Class 3 frames shall be interleaved during a Class 1 Connection only if there is no backlog of Class 1 frames en route. Although an F_Port and attached N_Port both support Intermix, the F_Port may choose to transmit F_BSY to a Class 2 frame or discard a Class 3 frame while the N_Port is engaged in a Class 1 Connection.

The Fabric shall provide adequate buffering for an incoming Class 1 frame while a Class 2 or Class 3 frame is being transmitted during a Class 1 Connection. The extent of buffering needed is dependent on the manner in which a Class 2 or Class 3 frame in transit is managed:

- to successfully complete a Class 2 or 3 frame in transit, an incoming Class 1 frame needs to be buffered to the extent of the maximum Class 2 or Class 3 frame size, or
- if a Class 2 or Class 3 frame is being aborted (**EOFa**) on receipt of a Class 1 frame, the Class 1 frame needs to be buffered to the extent of the time required to append an **EOFa** to the Class 2 or 3 frame in transit.

22.4.2 Intermix rules

An N_Port pair shall have a Class 1 Connection established for the Intermix rules to be applicable to them. To provide an Intermix service, the Fabric and the receiving and transmitting N_Ports shall obey the following rules:

1. An F_Port or an N_Port shall provide the service parameter as a receiver to indicate its Intermix capability to the transmitting Port. **(receiver service parameter)**
2. An N_Port which supports Intermix may transmit Class 2 and Class 3 frames during a Class 1 Connection intermixed with Class 1 frames of that Connection if the Fabric entry F_Port supports Intermix. **(transmit Intermix)**
3. An N_Port which supports Intermix may receive Class 2 and Class 3 frames during Class 1 Connection intermixed with Class 1 frames of that Connection if the Fabric exit F_Port supports Intermix. **(receive Intermix)**
4. In a point-to-point topology, an N_Port which supports Intermix may transmit and receive Class 2 and Class 3 frames during Class 1 Connection intermixed with Class 1 frames if the other N_Port supports Intermix. **(point-to-point Intermix)**
5. If the destination is busy servicing Class 1 frames, an exit F_Port may issue a busy to a Class 2 frame. **(exit F_Port busy)**
6. During Class 1 Connection, if an entry F_Port receives a Class 2 or Class 3 frame from the attached N_Port but does not support Intermix, it shall reject or discard the frame. If the frame is Class 2, the reason code shall be provided in the reject frame. Class 3 frame shall be discarded without any response to the sender. **(entry F_Port reject)**
7. During Class 1 Connection, if an exit F_Port receives a Class 2 or Class 3 frame but does not support Intermix, it shall reject or discard the frame. If the frame is Class 2, the reason code shall be provided in the reject frame. Class 3 frame shall be discarded without any response to the sender. **(exit F_Port reject)**
8. The Fabric shall guarantee full Class 1 bandwidth during a Dedicated Connection. Class 2 and Class 3 frames shall flow only on the unused Class 1 bandwidth. **(bandwidth sharing)**
9. Class 1 frames have priority over Class 2 and Class 3 frames. Class 2 and Class 3 frames may be intermixed only when there is no backlog of Class 1 frames. The F_Port may issue F_BSY to a Class 2 frame and discard a Class 3 frame. **(Class 1 precedence)**
10. The Fabric may cause a delay and displace a Class 1 frame in time due to the Intermix. This delay is limited to the maximum Class 2 or 3 frame size. The latency due to this delay is transparent to the transmitting N_Port. The Fabric shall guarantee the integrity and delivery of Class 1 frames. **(Class 1 frame skew and integrity)**

22.4.3 Intermix delimiters

Intermix does not impose any additional delimiters.

22.4.4 Intermix frame size

The size of each Class 1, Class 2 or Class 3 frame is governed by the limitations of each Class individually. Intermix does not impose any additional limitation on the frame size.

22.4.5 Intermix flow control

The flow control for each Class is governed separately by individual Class. Intermix does not impose any additional rules on flow control.

23 Login and Service Parameters

23.1 Introduction

The Login procedure is a method by which an N_Port establishes its operating environment with a Fabric, if present, and other destination N_Ports with which it communicates. Fabric Login and destination N_Port Login are both accomplished with the same procedure using different Destination_Identifiers (D_IDs).

Explicit Login is accomplished using the LOGI Link_Data frame Sequence to transfer the Service Parameters (contained in the Data Field) of the N_Port initiating the LOGI Exchange. The Frame Content is the same for Fabric Login and destination N_Port Login except for the Destination Identifier (D_ID). The Accept (ACC) contains the Service Parameters of the Responder (contained in the Data Field).

Implicit Login is a method of defining and specifying the Service Parameters of destination N_Ports by means other than the explicit use of the Login Exchange. Specific methods of **implicit** Login are not defined in this document.

When Login is referred to throughout other sections of this document, either the **explicit** or **implicit** procedure is acceptable. Implicit Login is assumed to provide the same functionality as Explicit Login. Explicit Login permanently replaces previous Service Parameters and initializes the Credit_CNT and Fabric Credit_CNT.

Typically explicit Fabric Login is performed during the initialization process under the assumption that a Fabric is present. The first explicit Login is directed to the Fabric using the well-known Fabric address (ie. Fabric F_Port at hexadecimal 'FFFFFE'). It is mandatory for all Fabric types to support the Login procedure. An N_Port may use binary zeros if its physical address identifier is unknown, or it may use its last known physical address identifier in the LOGI frame as its S_ID.

Login with the Fabric provides the N_Port with Fabric characteristics for the entire Fabric as defined in the Fabric's Service Parameters. The Service Parameters specified by the N_Port provide the Fabric with information regarding the type of support the N_Port requires. The Service Parameters also include a 64-bit World_wide_Name. During the Fabric Login procedure the Fabric may optionally define the N_Port's physical address identifier within a system configuration.

Destination N_Port Login provides each N_Port with the other N_Port's Service Parameters. Knowledge of a destination N_Port's receive and transmit characteristics is required for data Exchanges. Service Parameters of destination N_Ports are saved and used when communication with those N_Ports is initiated. Saving the Service Parameters of destination N_Ports with which an N_Port communicates requires N_Port resources. These resources can be released using the destination N_Port Logout procedure.

If an N_Port desires to explicitly reLogin after a Login has been previously performed, the initiating N_Port shall quiese other active Sequences that it initiated with the destination N_Port prior to performing reLogin, otherwise, Credit_CNT values are unpredictable. If an N_Port receives a Login request while another Sequence is active which was initiated from the requesting N_Port, it may reject the Login request using an LA_RJT (Application Reject).

23.1.1 Applicability

Login with the Fabric is required for all N_Ports, regardless of Class of Service supported. For an N_Port which supports Class 1 or Class 2 Service, an N_Port is required to Login with each N_Port with which it intends to communicate (destination N_Port Login). Destination N_Port Login is optional for N_Ports supporting only Class 3 Service.

23.2 Fabric Login

Fabric Login accomplishes five functions:

1. Determines the presence or absence of a Fabric.
2. If a Fabric is present, it provides a specific set of operating characteristics associated with the entire Fabric.
3. If a Fabric is present, it may optionally assign or confirm the native Source Identifier

(S_ID) of the N_Port which initiated the Login.

4. If a Fabric is not present, a P_RJT response indicates that the requesting N_Port is attached in a point-to-point topology.

5. Initializes the buffer-to-buffer Credit_CNT.

23.2.1 Explicit Fabric Login

The explicit Fabric Login procedure requires transmission of a Login (LOGI) Link_Data frame by an N_Port with a Destination Identifier (D_ID) of the well-known Fabric F_Port address and a Source Identifier (S_ID) of binary zeros or its last known native address identifier (X). If the S_ID value used is zero, the F_Port shall assign a Fabric unique S_ID to the N_Port in the ACC reply Sequence. If the S_ID = X is invalid, the F_Port responds with an F_RJT indicating invalid S_ID. On receiving F_RJT in response to S_ID = X, the N_Port shall retry the LOGI Sequence with an S_ID of zero.

If the Fabric does not support native address identifier assignment, the N_Port shall assign its native address identifier by another method not defined in this Standard. The Data Field of this Link_Data frame contains the Service Parameters of the N_Port transmitting the LOGI frame. N_Port Service Parameters are defined in 23.5.

The normal reply Sequence to a LOGI Link_Data frame by a Fabric is an Accept (ACC) Link_Data frame Sequence with a Destination Identifier (D_ID) assigned to the N_Port by the Fabric and a Source Identifier (S_ID) of the well-known Fabric F_Port address. The Data Field of the ACC contains the Service Parameters of the Fabric. Fabric Service Parameters are defined in 23.6.

23.2.1.1 Table legend

The following meanings are associated with table 42:

- the Response/Reply Seq column identifies a Link_Control response, or a Link_Data frame transmitted in response to the LOGI frame directed to the Fabric Controller. More than one frame is possible in response.
- the Normal/Abnormal column identifies the frame response as expected (Normal) or unexpected (Abnormal). The Normal responses to LOGI for Class 1 and 2 is R_RDY primitive, ACK_1, and ACC. This assumes no busy or reject conditions in the presence of a Fabric (Abnormal). The Normal responses to LOGI for Class 3 is R_RDY primitive and ACC.
- the D_ID and S_ID columns specify the value of the corresponding field in the response frame received by the N_Port transmitting the LOGI frame.
- the Indication column provides a short summary of the possible conditions associated with a particular response.
- the N_Port Action column specifies the action for the N_Port transmitting the LOGI frame to take based on the response received.

23.2.2 Responses to Fabric Login

Table 42 describes the set of possible responses to Fabric Login. The LOGI frame transmitted contains a D_ID of Fabric F_Port and an S_ID of binary zeros or X. Further description of the response primitive or frame is found in clause 20. These responses are characterized by the following:

- Response 1 is possible from an N_Port or Fabric.
- Response 2 is from a Fabric. The D_ID and S_ID values correspond to the values in the LOGI frame (also for responses 4 and 5).
- Response 3 completes Fabric Login. The N_Port S_ID is either assigned or confirmed as X.
- Response 4 indicates that the Fabric is busy.
- Response 5 indicates a Fabric reject. If Class is not supported, retry Login with another Class. If S_ID is invalid, then retry Login with an S_ID of binary zeros. For other reject reasons, the N_Port shall respond accordingly.
- Response 6 indicates a point-to-point attachment without a Fabric present. The address identifiers for D_ID and S_ID may or may not be assigned. The N_Port shall perform destination N_Port Login using the assigned address identifiers, if present.
- Response 7 indicates a point-to-point attachment without a Fabric present. The N_Port shall respond according to the action and reason code of the P_RJT.

— Response 8 indicates a point-to-point attachment and a collision with a LOGI from the attached N_Port. The N_Port shall compare the values of the World_wide_Names contained in the Service Parameters of the Data Fields of its Parameters and that of the LOGI received. If its value is higher, then it shall retransmit the LOGI frame. If its value is lower, then it shall wait for a LOGI from the attached N_Port. See 23.5.10, "World_wide_Name" for a description of World_wide_Names. See 23.3.2.3 for a description of point-to-point destination N_Port Login.

— Response 9 indicates that a Link error has occurred.

Table 42. Responses to LOGI frame - Fabric Login

Response/ Reply Seq	Normal(N) Abnor(AB)	D_ID	S_ID	Indication	N_Port Action
1.R_RDY	(N)	N/A	N/A	- Class 1(**SOFc1**) - Class 2 or 3 - successful frame delivery to F_Port or N_Port	- wait for frame
2.ACK_1 or ACK_N	(N)	X or 0	F_Port	- LOGI frame has been received	- Wait for Reply Data frame Sequence
3.ACC	(N)	X	F_Port	- Fabric Login complete	- Perform destination N_Port Login
4.F_BSY	(AB)	X or 0	F_Port	- Fabric present - Fabric Busy	- retry later
5.F_RJT	(AB)	X or 0	F_Port	- Fabric present - reason code = invalid Class = invalid S_ID = other	 - select next Class - reLogin with S_ID = 0 - respond accordingly
6.P_BSY	(AB)	X or 0	Y or 0	- N_Port Busy - no Fabric	- perform N_Port Login
7.P_RJT	(AB)	X or 0	Y or 0	- no Fabric - reason code = invalid Class = invalid D_ID = other	 - select next Class - perform N_Port Login - respond accordingly
8.LOGI	(AB)	F_Port	Z or 0	- Collision with other N_Port	- respond with P_RJT (D_ID Addr error) - compare World_wide_Names - if xmit > rec'd retransmit LOGI - if xmit < rec'd wait for LOGI
9.None				- error	- retry after Sequence timeout

23.2.3 SOF delimiters

Since the Fabric may not support all three Classes of Service, the LOGI frame may require retransmission with a different **SOF** delimiter for each of the following Classes:

1. Class 1 - **SOFc1**
2. Class 2 - **SOFi2**
3. Class 3 - **SOFi3**

Selection of the **SOF** delimiter is based on the Classes of Service supported by the originating N_Port. The LOGI frame is transmitted and the appropriate action is specified in Table 42. When a Reject (F_RJT, P_RJT) has been received indicating incorrect Class, the next supported delimiter on the above list is attempted until the Login procedure is complete or all supported delimiter types have been attempted. If all supported delimiter types have been attempted and all have been rejected by the Fabric, the Fabric and N_Port are incompatible and outside intervention is required.

23.2.4 Frequency

The frequency of Fabric Login is installation dependent based on the frequency of configuration changes which may alter the N_Port addresses within an installation.

If a destination N_Port's address identifier has changed since the last Fabric Login, the source N_Port receives a Reject Link_Response frame with a reason code indicating an invalid source address identifier.

23.2.5 Fabric Login frame flow

See O.4 for examples of frame flow for Login for Classes 1, 2, and 3.

23.3 N_Port Login

Destination N_Port Login between two N_Ports is complete when each N_Port has received the Service Parameters of the other N_Port. This may be accomplished by either implicit or explicit destination N_Port Login.

23.3.1 Address Identifiers

An N_Port determines its own native source address identifier through explicit or implicit Login by:

- the Fabric, if present,
- implicit definition,
- assignment in the LOGI frame transmitted to a destination N_Port attached in a point-to-point topology, or
- assignment by the destination N_Port attached in a point-to-point topology in a response frame.

Address identifiers of other destination N_Ports with which an N_Port communicates may be collected from:

- the Fabric, if present,
- a name-server function,
- implicit definition, or
- an alternate initialization procedure.

23.3.2 Explicit N_Port Login

Explicit N_Port Login accomplishes two functions:

1. It provides a specific set of operating characteristics associated with the destination N_Port
2. Initializes the destination end-to-end Credit_CNT.

23.3.2.1 Fabric present

The destination N_Port explicit Login procedure requires transmission of a Login (LOGI) Link_Data frame with a Destination Identifier (D_ID = Y) of the destination N_Port and a Source Identifier (S_ID = X) of originating N_Port. The Data Field of this frame contains the Service Parameters of the N_Port originating the LOGI frame. N_Port Service Parameters are defined in 23.5.

The normal reply Sequence to a LOGI Link_Data frame by an N_Port is an Accept (ACC) Link_Data frame Sequence with a Destination Identifier (D_ID) of the originating N_Port (LOGI frame) and a Source Identifier (S_ID) of the responding N_Port. The Data Field of the ACC contains the Service Parameters of the responding N_Port.

Table 43. Responses to LOGI frame - N_Port Login (Fabric present)					
Response/ Reply Seq	**Normal(N) Abnor(AB)**	**D_ID**	**S_ID**	**Indication**	**N_Port Action**
1.R_RDY	(N)	N/A	N/A	- Class 1(**SOFc1**) - Class 2 or 3 only - successful frame delivery to F_Port	- wait for frame
2.ACK_1 or ACK_N	(N)	X	Y	- LOGI frame has been received	- Wait for Reply Data frame Sequence
3.ACC	(N)	X	Y	- Login complete	- End
4.F_BSY	(AB)	X	F_Port	- Fabric present - Fabric Busy	- retry later
5.F_RJT	(AB)	X	F_Port	- Fabric present - reason code = invalid D_ID = other	 - determine S_ID for Y - reattempt Login - respond accordingly
6.P_BSY	(AB)	X	Y	- N_Port busy	- retry later
7.P_RJT	(AB)	X	Y	- reason code = invalid Class = other	 - select next Class - respond accordingly
8.LA_RJT	(AB)	X	Y	- reason code = invalid parameters = other	 - correct parameters - respond accordingly

23.3.2.2 Responses to destination N_Port Login

See 23.2.1.1 for a description of the column meanings. The entries in table 43 are based on previous Login with a Fabric. Table 43 describes the set of possible responses to destination N_Port Login with a Fabric present.

23.3.2.3 Point-to-point

This procedure is based on the assumption that response 6, 7, or 8 in table 42 was received during an attempted Fabric Login. The destination N_Port explicit Login procedure requires transmission of a Login (LOGI) Link_Data frame with a Destination Identifier (D_ID) of the destination N_Port and a Source Identifier (S_ID) of originating N_Port. If the N_Port transmitting the LOGI frame wishes to assign N_Port native address identifiers, it does so by specifying non-reserved values with S_ID = X and D_ID = Y of the LOGI frame. If the N_Port wishes to defer native address identifier assignment to the destination N_Port, the N_Port transmitting LOGI specifies binary zeros for both S_ID and D_ID. The Data Field of this frame contains the Service Parameters of the N_Port originating the LOGI frame. N_Port Service Parameters are defined in 23.5.

The normal reply Sequence to a LOGI Link_Data frame by an N_Port is an Accept (ACC) Link_Data frame Sequence. If the LOGI frame contained zeros for S_ID and D_ID, then the destination N_Port shall assign S_ID = R and D_ID = Z. If the LOGI frame specified values for S_ID and D_ID, the destination N_Port shall use D_ID X and S_ID = Y in the ACC frame. The Data Field of the ACC contains the Service Parameters of the responding N_Port.

Table 44. Responses to LOGI frame - N_Port Login (point to point)					
Response/ Reply Seq.	Normal(N) Abnor(AB)	D_ID	S_ID	Indication	N_Port Action
1.R_RDY	(N)	N/A	N/A	- Class 1(SOFc1) - Class 2 or 3 only - successful frame delivery to N_Port	- wait for frame
2.ACK_1 or ACK_N	(N)	X or 0	Y or 0	- LOGI frame has been received	- Wait for Reply Data frame Sequence
3.ACC	(N)	X or R	Y or Z	- Login complete	- End
4.P_BSY	(AB)	X or 0	Y or 0	- N_Port busy	- retry later
5.P_RJT	(AB)	X or 0	Y or 0	- reason code = invalid Class = other	 - select next Class - respond accordingly
6.LA_RJT	(AB)	X or R	Y or Z	- reason code = invalid parameters = other	 - correct parameters - respond accordingly

23.3.2.4 Responses to destination N_Port Login

See 23.2.1.1 for a description of the column meanings. Table 44 describes the set of possible responses to destination N_Port Login for a point-to-point configuration.

For responses 2, 4, and 5 the responding N_Port shall use the address identifiers in the corresponding LOGI frame. For responses 3 and 6 the responding N_Port shall use X and Y in the LOGI frame, or it shall assign R and Z if the LOGI frame used binary zeros for address identifiers.

If a LOGI frame is received following transmission of LOGI, then the N_Port shall follow the N_Port action defined in table 42 for response 8.

23.3.3 SOF delimiters

Since the destination N_Port may support any of three Classes of Service, the LOGI frame may require retransmission with a different **SOF** for each Class in the same manner described for Fabric Login.

23.3.4 Frequency

The frequency of destination N_Port Login is installation dependent based on the frequency of configuration changes which may alter the N_Port addresses within an installation. Service Parameters of other N_Ports are retained until the next destination N_Port Login or until N_Port Logout is performed.

23.3.5 N_Port Login frame flow

See O.4 for examples of frame flow for destination N_Port Login for Classes 1, 2, and 3.

23.4 N_Port Logout

The destination Logout procedure provides a method for removing Service between two N_Ports. Logout releases resources associated with maintaining Service with a destination N_Port. There is no Logout procedure for the Fabric.

23.4.1 Explicit Logout

Logout is accomplished by transmitting a Logout (LOGO) Link_Data frame Sequence to a destination N_Port. The Logout procedure is complete when the responding N_Port transmits an ACC Link_Data frame reply Sequence.

If an N_Port desires to explicitly Logout, the initiating N_Port shall quiese other active Sequences that it initiated with the destination N_Port prior to performing Logout, otherwise, the state of other active Sequences is unpredictable. If an N_Port receives a Logout request while another Sequence is active which was initiated from the requesting N_Port, it may reject the Logout request using an LLA_RJT (Application Reject).

23.5 N_Port Service Parameters

16 Bytes	16 Bytes	16 Bytes	8 Bytes	8 Bytes
Class 1 Service Parameter	Class 2 Service Parameter	Class 3 Service Parameter	World_wide Name	r

Figure 51. LOGI or ACC Data Field

The first 48 bytes of the Data Field specify three sets of Service Parameters as shown in figure 51. The first 16-byte group specifies Service Parameters for Class 1. The second 16-byte group specifies the Service Parameters for Class 2. The third 16-byte group specifies the Service Parameters for Class 3. The next eight bytes contain the World_wide_Name of the N_Port transmitting the Service Parameters.

Within each group of 16-byte Service Parameters, fields are defined as shown in figure 52.

Word	Bits 31	0
0	Service Options VEEEEEEE EEEEEEEE	Transmitter Control DDDDDDDD DDDDDDDD
1	Receiver Control CCCCCCCC CCCCCCCC	Receive Data Field Size rrrrNNNN NNNNNNNN
2	Concurrent Sequences rrrrrrrr LLLLLLLL	Credit MMMMMMMM MMMMMMMM
3	Open Sequences per Exchange rrrrrrrr XXXXXXXX	Reserved rrrrrrrr rrrrrrrr

Figure 52. Service Parameters

Each group of sixteen byte Service Parameters are divided into the categories as specified in figure 52.

1. Class Validity (V)
2. Service Options (E)
3. Transmitter Control Flags (D)
4. Receiver Control Flags (C)
5. Receive Data Size (N)
6. Concurrent Sequences (L)
7. Credit (M)
8. Open Sequences per Exchange (X)

The parameters described are located in the Payload of the Login (LOGI) Link_Data request as well as the Accept (ACC) Link_Data reply Sequence sent in reply to the Login Link_Data request.

23.5.1 Applicability

The following table identifies Service Parameter fields and specifies the applicability of those parameters for Class 1, 2,and 3 of N_Ports and the Fabric.

Table 45. Service Parameter Applicability

			N_Port Class			Fabric Class		
Service Parameter	Word	Bits	1	2	3	1	2	3
Class Validity	0							
Valid = 1	0	31	y	y	y	y	y	y
Service Options	0	30-16						
Service Options - Class 1,2,3	0	30-28	y	y	y	y	y	y
FC_PH Version	0	23-16	y	y	y	y	y	y
Transmitter Ctl	0	15-0						
X_ID reassignment	0	15	y	y	n	n	n	n
Receiver Ctl	1	31-16						
N_Port/Fabric	1	31	y	y	y	y	y	y
ACK_N Support	1	30	y	y	n	n	n	n
X_ID transition interlock	1	29	y	y	n	n	n	n
Discard policy support	1	28	y	y	n	n	n	n
Process policy support	1	27	y	y	n	n	n	n
Reserved - Fabric Specific	1	23-16	y	y	y	y	y	y
Receive Data Field Size	1	15-0	y	y	y	y	y	y
Concurrent Sequences	2	31-16	y	y	n	n	n	n
Credit	2	15-0	y	y	n	y	y	y
Open Sequences per Exchange	3	31-16	y	y	n	y	y	y
Note: "y" indicates yes, applicable; "n" indicates no, not applicable								

23.5.2 Class Validity

- **Word 0, Bit 31 - Class Validity**

Bit 31	
0	Invalid (Class not supported)
1	Valid (Class supported)

The Class Validity bit indicates whether this Class is supported. If the Class Validity bit is zero, it indicates that this set of sixteen bytes shall be ignored. If the Class Validity bit is one, it indicates that this Class is supported. The Class is identified based on Class 1 = first sixteen-byte group, Class 2 = second sixteen-byte group, and Class 3 = third sixteen-byte group.

NOTE - Service Parameter options specify FC-2 capability. The Link Application may further limit values supplied during Login.

23.5.3 Service Options

Service Options (E) shall specify optional features of a Class of Service supported by the N_Port supplying the Service Parameters.

Word,Bits	Service Options (E)
0,30-28	Class Options
0,27-24	Reserved
0,23-16	FC_PH Version

For Class 1:

• **Word 0, Bits 30-28 - Class 1 Options**

Bits 3 2 2 0 9 8	
0 0 0	reserved
0 0 1	Class 1 Supported - Exclusive Connections
0 1 0	Class 1 Supported - Intermixed Connections

An N_Port supporting "Exclusive Connections" may only transmit and receive frames from the N_Port to which an existing Class 1 Dedicated Connection is established. "Exclusive Connections" require that the Fabric transmit an F_BSY frame, as appropriate, in response to frames issued by a third N_Port targeted for one of the two N_Ports engaged in a Class 1 Dedicated Connection.

An "Intermixed" Dedicated Connection specifies that the Fabric may insert Class 2 or Class 3 frames from a third N_Port while a Class 1 Dedicated Connection is established provided that sufficient Fabric buffering is present. Frames between the connected N_Ports may be displaced in time, but not discarded. All operating rules regarding Idle transmission shall be adhered to by the Fabric. If the Fabric must terminate an intermixed frame to a destination N_Port., an **EOFa** may be used to terminate the frame being transmitted.

Support for Intermix is optional by both an N_Port and a Fabric. If the N_Port supports intermixing of Class 2 and 3 frames during a Class 1 Dedicated Connection, it must request Fabric support for Intermix during Login with the Fabric. (See 22.4 for more discussion on Intermix.)

For Class 2:

• **Word 0, Bits 30-28 - Class 2 Options**

These bits are currently reserved since there are no Class 2 options.

For Class 3:

• **Word 0, Bits 30-28 - Class 3 Options**

These bits are currently reserved since there are no Class 3 options.

All Classes:

• **Word 0, Bits 23-16 - FC_PH Version**

These bits are encoded to represent a value which is mapped to a specific version of FC_PH.

23.5.4 Transmitter Control

Transmitter Control Flags (D) specify which protocols, policies, or functions the Transmitter function in the N_Port supplying the Service Parameters requires of the Receiver.

Word,Bits	Transmitter Ctl Flags (D)
0, 15	X_ID reassignment required
0, 14-0	Reserved

• **Word 0, Bit 15 - X_ID reassignment required**

— 0 no X_ID reassignment
— 1 X_ID reassignment required

X_ID reassignment required indicates that the N_Port supplying this parameter may reassign its X_ID value (either OX_ID or RX_ID) at certain Sequence boundaries (see 24.4). If X_ID reassignment is required, an N_Port shall transmit an Association Header at the beginning of the Exchange and at X_ID reassignment points within the Exchange. An N_Port which does not require X_ID reassignment shall use the Association Header as described in 24.4. If X_ID reassignment is required, then X_ID interlock also is required.

23.5.5 Receiver Control

Receiver Control Flags (C) shall specify which functions are supported by the N_Port supplying the Service Parameters when acting as a receiver of device frames. Receiver Control Flags specify the receiver functions supported by the N_Port.

Word,Bits	Receiver Ctl Flags (C)
1,31	N_Port/Fabric
1,30	ACK_N support
1,29	X_ID transition interlock
1,28	Discard policy supported
1,27	Process policy supported
1,26-24	Reserved
1,23-16	Reserved for Fabric specific

- **Word 1, Bit 31 - N_Port / Fabric**

— 0 N_Port supplying parameters
— 1 Fabric supplying parameters

- **Word 1, Bit 30 - ACK_N support**

— 0 ACK_N is not supported for ACK
— 1 ACK_N is supported for ACK

Bit 30 specifies that the Receiver does or does not support ACK_N for acknowledgement of good data transmission. ACK_N support is applicable to Class 1 and 2, but not Class 3. ACK_1 is mandatory, whereas support for the ACK_N form of ACK is optional.

If both N_Ports interchanging Service Parameters specify ACK_N support, ACK_N shall be used for ACK. If either N_Port does not support ACK_N, then both N_Ports shall use ACK_1. ACK_N support includes the processing capability to transmit as well as receive ACK_N frames.

NOTE - Depending on the Credit being managed, ACK_N may be difficult to manage by the Sequence Recipient when more than one Sequence is active. Since only the Sequence Initiator is aware of the Credit allocated to each Sequence, the Sequence Recipient is unable to choose an ACK_N window which corresponds to the Credit associated with a given Sequence.

- **Word 1, Bit 29 - X_ID transition interlock**

— 0 no X_ID interlock
— 1 X_ID interlock required

This bit indicates that the N_Port supplying this parameter requires that an interlock be used during X_ID assignment or reassignment. In X_ID assignment or reassignment, the Sequence Initiator sets the Recipient X_ID value to hexadecimal 'FFFF' in the first Data frame of a Sequence and the Recipient supplies its X_ID in the ACK frame corresponding to the first Data frame of a Sequence. The Sequence Initiator shall not transmit additional frames until the corresponding ACK is received. Following reception of the ACK the Sequence Initiator continues transmission of the Sequence using both assigned X_ID values.

X_ID interlock shall also be used during X_ID reassignment as described in 24.4.

- **Word 1, Bits 28-27 - Error Policy Supported**

These bits are set to specify the types of support possible for frame out of order conditions processed by the Receiver N_Port. The policy actively used for a given Exchange and Sequence is specified by the FC-4 at the Sequence Recipient.

Out of order detection

In Class 1, out of order detection is straightforward because frames arrive in the order of transmission. However, in Class 2 and 3, out of order frame delivery is possible. In order to simplify an N_Port's implementation, an N_Port may choose to define an "out of order window". An N_Port may choose a window value of W. If the highest SEQ_CNT received minus W is greater than the SEQ_CNT of a missing frame, then an out of order condition is detected. The value of W is only known internally to the N_Port.

In either **Discard or Process policy**, when an "out of order" error is detected, the "expected" sequence count is saved in the ERR_SEQ field of the appropriate Sequence Status Block and a Sequence error is posted in the ERR field in the same Sequence Status Block for a given Exchange (OX_ID, RX_ID). Only the first error is saved.

NOTE - The error status is reported by FC-2 to FC-4.

- **Word 1, Bit 28 = 1 - Discard Policy Supported**
  - — 0 Discard Policy not supported
  - — 1 Discard Policy is supported

Discard policy support specifies that the N_Port is able to discard Data frames received following detection of an out of order error condition including the frame at which the error is detected.

When the out of order condition is detected, the Sequence Recipient notifies the Sequence Initiator by indicating Abort Sequence in the ACK corresponding to the frame on which the error was detected. All subsequent frames continue to be discarded and abort Sequence indicated in the ACK frame. See 29.7.1.

- **Word 1, Bit 27 = 1 - Process Policy Supported**
  - — 0 Process Policy not supported
  - — 1 Process Policy is supported

Process policy support specifies that the N_Port is to continue to process valid Data frames following a detected out of order error condition in the normal manner including the frame at which the error is detected. See 29.7.1.

23.5.6 Receive Data_Field Size

- **Word 1, Bits 15-0 Receive Data_Field Size (N)**

The **Receive Data_Field Size** is a 12-bit binary value (bits 15-0) which specifies the largest Data_Field Size for an FT_1 frame that can be received by the N_Port supplying the Service Parameters. Values less than 64 or greater than 2112 are invalid. Values are required to be a multiple of four bytes.

The maximum Receive Data_Field Size is specified by FC-2.

23.5.7 Concurrent Sequences

- **Word 2, Bits 31-16 Concurrent Sequences (L)**

Concurrent Sequences shall specify the maximum number of Sequences that can be active at one time between a pair of N_Ports. It shall be equal to the number of Sequence Status Blocks available in the N_Port supplying the Service Parameters for tracking the progress of a Sequence. The maximum number of Concurrent Sequences that can be specified is 255 per N_Port which may be allocated across all three Classes. Bits 31-24 are reserved.

Bits 2222 1111 3210 9876	
0000 0000	Reserved
0000 0001	1 Sequence Status Register
....	
1111 1111	255 Sequence Status Registers

For Class 1:

The SEQ_ID values shall range from 0 to (L-1), inclusively, where L is the value of the Concurrent Sequence field.

For Class 2 and 3:

The SEQ_ID values shall range from 0 to 255.

23.5.8 Open Sequences per Exchange

- **Word 3, Bits 31-16 Open Sequences / Exchange (X)**

The value of Open Sequences per Exchange shall specify the maximum number of Sequences that can be open at one time between a pair of N_Ports for one Exchange. The value of X+2 specifies the number of Sequence Status Blocks that are maintained for a single Exchange. This value is used for Exchange and Sequence tracking. The value of X limits the link facility resources required for error detection and recovery. See clause

29. The value of X is specified in bits 23-16. Bits 31-24 are reserved.

23.5.9 Credit

- **Word 2, Bits 14-0 Credit (M)** - the maximum number of outstanding frames awaiting a response.

The minimum value of Credit is one. The Credit field specified is associated with the number of buffers available for holding the Data_Field of a frame and processing the contents of that Data_Field by the N_Port supplying the Service Parameters.

In order to insure data integrity, Credit provided in ACK_N should not exceed one-half of the maximum size sequence count.

Values in the Credit Field have the following meanings:

Bits 111 11 432 1098 7654 3210	
000 0000 0000 0000	Infinite Buffers
000 0000 0000 0001	1 Buffer
....	
111 1111 1111 1110	32766 Buffers
111 1111 1111 1111	32767 Buffers

Fabric F_Port Login
When an N_Port performs Login with the Fabric F_Port, Credit values provided are associated with buffer-to-buffer Credit. The buffer-to-buffer Credit (M) shall be a single value repeated for all three Classes of Service Parameters, otherwise a Reject frame is transmitted in response. An N_Port tracks buffer-to-buffer Credit_CNT as a single entity for all frames subject to buffer to buffer flow control (see clause 25).

Destination N_Port Login
When an N_Port performs Login with an N_Port, Credit values provided are associated with end-to-end Credit. The end-to-end Credit (M) shall be a unique and separate value for each of the three Classes of Service Parameters. An N_Port tracks end-to-end Credit_CNT as on a Class basis (see clause 25). Login with the Fabric Controller (hexadecimal 'FFFFFD') by an N_Port is processed in the same manner as N_Port Login.

From the view of the receiver of the Service Parameters, in point-to-point N_Port Login, buffer-to-buffer Credit_CNT is the sum of Class 2, Class 3, plus one (for Class 1). End-to-end Credit is the same as any N_Port Login.

23.5.10 World_wide_Name

The World_wide_Name is an eight byte field. Bits 63-60 specify the form of the name contained in bits 59-0. The World_wide_Name is assigned at the time of manufacture or may be locally assigned by a system administrator. If it is assigned at the time of manufacture, it shall be unique on a world_wide basis. Locally assigned values shall be unique for a given installation. See 19.4 for more information.

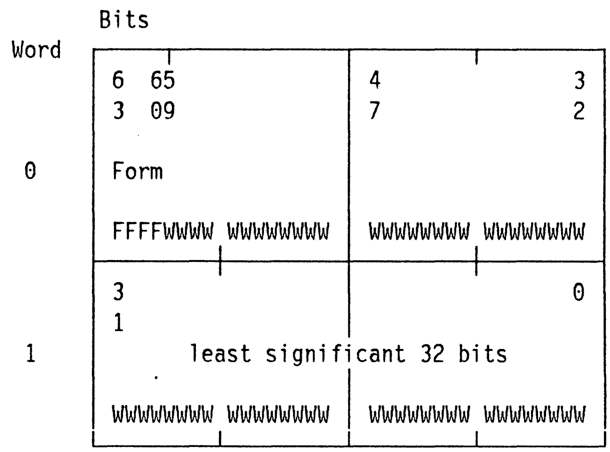


Figure 53. World_wide_Name

Form

Bits 6 6 6 6 3 2 1 0	
0 0 0 0	= Locally assigned (bits 59-0)
0 0 0 1	= CCITT 60-bit Address (bits 59-0)
0 0 1 0	= IEEE 48-bit Address (bits 47-0)
Others	= Reserved

23.6 Fabric Service Parameters

When a Fabric responds to a Login Link_Data frame Sequence, certain fields in the ACC Link_Data reply Sequence Data Field are ignored and other fields are interpreted in a different manner than if the reply Sequence came from another N_Port.

The first 48 bytes of the Data Field specify three sets of Service Parameters as shown in figure 51. The first sixteen-byte group specifies Service Parameters for Class 1. The second sixteen-byte group specifies the Service Parameters for Class 2 and the third sixteen-byte group specifies the Service Parameters for Class 3. The next six bytes specifies the World_wide_Name which identifies the Fabric.

Within each group of 16-byte Service Parameters, fields are defined as shown in figure 52.

23.6.1 Class Validity

- **Word 0, Bit 31 - Class Validity**

Bit
3
1

0 Invalid (Class not supported)
1 Valid (Class supported)

The Class Validity bit indicates whether this Class is supported. If the Class Validity bit is zero, it indicates that this set of sixteen bytes shall be ignored. If the Class Validity bit is one, it indicates that this Class is supported. The Class is identified based on Class 1 = first sixteen-byte group, Class 2 = second sixteen-byte group, and Class 3 = third sixteen-byte group.

23.6.2 Service Options

Service Options (E) shall specify Class of Service capabilities supported or required by the Fabric supplying the Service Parameters.

For Class 1:

- **Word 0, Bits 30-28 - Class 1 Options**

Bits
3 2 2
0 9 8

0 0 0 reserved
0 0 1 Class 1 Capable of Exclusive Connections
0 1 0 Class 1 Capable of Intermixed Connections
0 1 1 Class 1 Queued Requests supported

Support for Queued connect-requests allows an N_Port to transmit more than one connect-request while it is not involved in a Dedicated Connection. This allows an N_Port to request Class 1 Dedicated Connections with multiple destinations on a queued basis. While the N_Port is Connected to one destination, another connect-request may be processed by the Fabric in order to minimize connect latency. Caution should be observed since this technique effectively locks in the destination N_Port with a source N_Port which may be busy.

- **Word 0, Bits 30-28 - Class 2 Options**

These bits are currently reserved since there are no Class 2 options.

- **Word 0, Bits 30-28 - Class 3 Options**

These b=ts are currently reserved since there are no Class 3 options.

All Classes:

- **Word 0, Bits 23-16 - FC_PH Version**

These bits are encoded to represent a value which is mapped to a specific version of FC_PH.

23.6.3 Transmitter Control

This field is reserved.

23.6.4 Receiver Control

Receiver Control Flags (C) have the following designations for ALL classes:

- **Word 1, Bit 15 - N_Port / Fabric Type**

— 1 Fabric supplying parameters

- **Word 1, Bits 7-0 Reserved for Fabric specific use**

See Annex N for description of current definitions of these bits.

23.6.5 Receive Data_Field Size

This field specifies the largest Data_Field Size (N) that shall be transmitted to the Fabric.

For Class 1:

- the frame establishing a Class 1 Connection using an **SOFc1**.

For Class 2 and 3:

- all frames

This is the absolute capability of the Fabric. This field has the same format and requirements as the Service Parameters for an N_Port.

23.6.6 Concurrent Sequences

This field is reserved.

23.6.7 Buffer-to-buffer (Fabric) Credit

The buffer-to-buffer Credit (M) field specified is associated with the number of buffers available for holding Class 1 connect-request, Class 2, or Class 3 frames transmitted to the Fabric F_Port for routing.

The buffer-to-buffer Credit (M) shall be a single value repeated for all three Classes of Service Parameters, otherwise a Reject frame is transmitted in response. An N_Port tracks buffer-to-buffer Credit_CNT as a single entity for all frames subject to buffer to buffer flow control (see clause 25).

Values in the Credit field have the same format as in N_Port Service Parameters.

23.7 Estimate Credit

An estimate of the minimum Credit between an N_Port pair for a given distance helps achieve the maximum bandwidth utilization of the channel, by continuously streaming data. The Credit estimate procedure is defined to accomplish this purpose.

This procedure is preceded by the Login between this N_Port pair. Login determines for this N_Port pair the maximum frame size that can be transmitted.

Applicability
The procedure is applicable to both Class 1 and Class 2. In Class 2, the procedure and the continuous streaming function are limited by the Fabric Credit.

Users
The procedure shall be invoked by the ULP of the source N_Port and responded by the ULP of the destination N_Port.

Prerequisite
To perform this procedure for Class 1 or Class 1 Intermix, a Class 1 Connection shall have been established before the Estimate Credit procedure is performed. If no Class 1 connection exists, the source N_Port may establish a Connection by issuing a NOP frame with **SOFc1**.

23.7.1 Estimate Credit procedure

This procedure consists of following three request Sequences.

1. Establish Streaming Sequence
2. Estimate Credit Sequence
3. Advise Credit Sequence

The procedure is illustrated with these request Sequences and their respective reply Sequences in figure 54.

The procedure shall be performed for Class 1 or Class 2 with respective delimiters, as defined in clause "22 Classes of service."

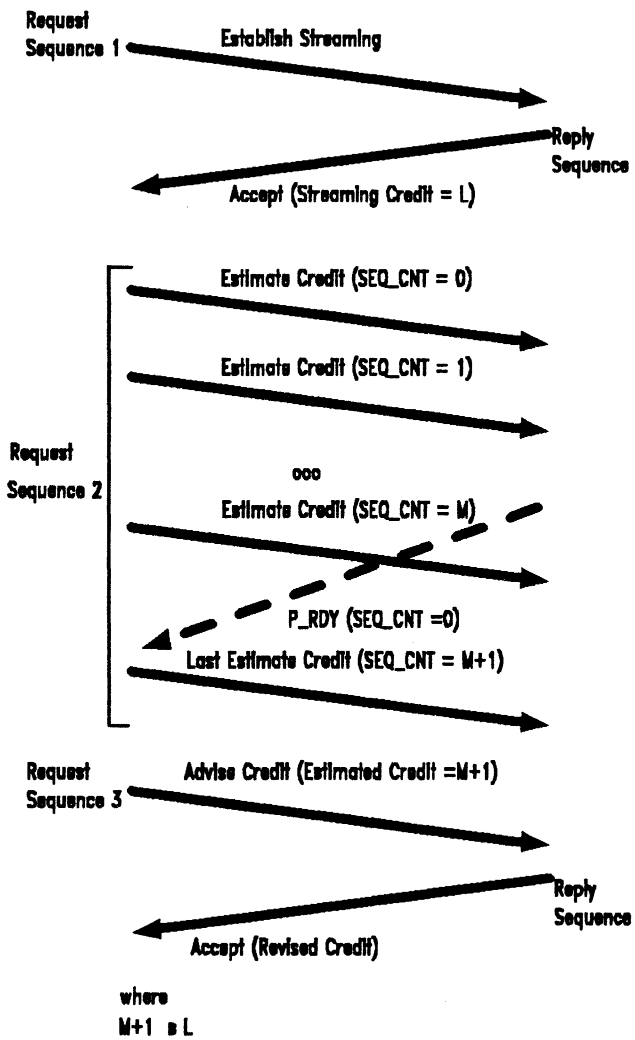


Figure 54. Establish Credit procedure

23.7.1.1 Establish Streaming Sequence

This Sequence shall be used to obtain a Credit large enough to perform continuous streaming from a source N_Port to a destination N_Port. This Sequence provides an opportunity for the destination N_Port to communicate the maximum Credit it can provide for the purposes of streaming and this temporary allocation is termed Streaming Credit (L).

This Sequence shall be used between an N_Port pair after the N_Port pair has logged in with each other. This Sequence may be initiated as a new Exchange or part of another Exchange. The Sequence may be initiated by the Originator or the Responder of the Exchange.

1. The source shall transmit the Establish Streaming (ESTS) frame.
2. The destination shall reply with an ACC frame.
3. The SOF delimiter of the ESTS frame identifies the Class.
4. The Data Field of ACC shall have the same format as the Service Parameters for Login. The Data Field shall contain Streaming Credit (L) allocated in Credit field of the appropriate Class of the Service Parameters for which the Estimate Credit procedure is being performed. The other fields shall be ignored by the receiver.

23.7.1.2 Estimate Credit Sequence

This Sequence shall be performed immediately following reception of the ACC in response to the Establish Streaming Sequence.

1. The source N_Port shall stream Estimate Credit (ESTC) frames consecutively until it receives the first ACK_1 from the destination N_Port. The source shall not exceed the Streaming Credit obtained during the Establish Streaming Sequence.
2. If the source does not receive ACK_1 after it has reached the limit imposed by the Streaming Credit value, it shall stop streaming and wait for the first ACK_1 to be received.
3. The size of the Data Field of the ESTC frame shall be the maximum allowed by the Service Parameters from Login.
4. The Data Field content shall be zeros.

5. The SEQ_CNT shall start with the appropriate value for the Exchange and Sequence in progress and monotonically increase with consecutive each ESTC frame transmitted.
6. The destination N_Port shall respond with ACK_1 to each ESTC frame received.
7. If the highest SEQ_CNT transmitted by the source N_Port at the time it receives the first ACK_1 is M, the number of outstanding frames (ie. Credit estimated for continuous streaming) shall equal M+1. If ACK_1 is received within the Streaming Credit limit (L ≥ M), this value of M+1 represents the minimum Credit required to utilize the maximum bandwidth of the Fibre. If the ACK_1 is received after reaching the Streaming Credit limit (L), this value is less than the optimal Credit required to utilize the maximum bandwidth of the Fibre.
8. The source N_Port shall follow all the rules in closing the Sequence, by sending the last frame of the Sequence and waiting for all the ACK_1s to be received.

23.7.1.3 Advise Credit Sequence

This Sequence shall be performed immediately following recpetion of the ACC in response to the Estimate Credit Sequence. The source N_Port which performed the Estimate Credit Sequence shall advise the destination N_Port of the Estimated Credit in ADVC Data Field. The destination N_Port shall reply using an ACC frame, with a revised Credit value in its Data Field. This value is determined by the destination N_Port based on its buffering scheme, buffer management, buffer availability and N_Port processing time. This is the final value to be used by the source N_Port for revised Credit.

This Sequence provides a complementary function to Login. In contrast to Login frame, the ADVC frame contains the Credit it would like to be allocated for continuous streaming.

If the Estimated Credit (M+1) is less than or equal to the Streaming Credit (L), the destination may choose to reallocate the estimated Credit. If the Streaming Credit (L) is smaller than needed for continuous streaming, the source N_Port is bound to run short of Credit and the source N_Port shall advice that value as the Estimated Credit.

1. The source N_Port shall transmit Advise Credit frame with the Estimated Credit (M+1).
2. The Data Field of the ACC shall have the same format as the Service Parameters for Login. The Data Field shall contain the Estimated Credit (M+1) in Credit field of the appropriate Class of the Service Parameters. The other fields shall be ignored by the receiver.
3. The destination N_Port shall determine the revised Credit value. The destination shall determine the value based on its buffer management, buffer availability and port processing time and may add a factor to the Estimated Credit value. This is the final value to be used by the source N_Port for Credit.
4. The destination N_Port replies with an ACC frame which successfully completes the Protocol. The ACC frame contains the Credit allocated to the source N_Port. The Data Field of ACC shall have the same format as the Service Parameters for Login. The Data Field shall contain the final Credit in Credit field of the appropriate Class of the Service Parameters. The other fields shall be ignored by the receiver.

Asymmetry
Since the maximum frame size is permitted to be unequal in forward and reverse directions, the Estimate Credit procedure shall be performed separately for each direction of transfer.

Frequency
The Estimate Credit procedure provides an approximation of the distance involved on a single path. If there are concerns that in a Fabric in which the length (and time) of the paths assigned can vary, the procedure may be repeated several times to improve the likelihood that the Estimated Credit value is valid.

Alternatively, a source may accept the Estimated Credit value. If at a later time, data transfers are unable to stream continuously the source may re-initiate the Estimate Credit Procedure, or arbitrarily request an increase in Estimated Credit by using an ADVC Link Data frame.

24 Exchange, Sequence, and Sequence Count Management

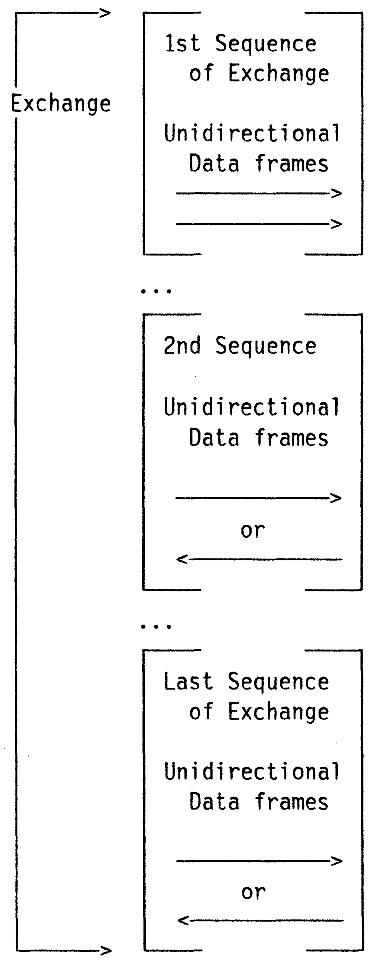
24.1 Introduction

An Exchange is a mechanism for coordinating the interchange of information and data between two N_Ports or nodes. This clause discusses Exchanges between N_Ports. However, Exchanges may be managed across multiple related N_Ports within a Node or single controlling entity. Additional control is required when multiple Sequences for a single Exchange are active across multiple N_Ports simultaneously.

Data frame transfer
Transfer of information between two N_Ports is based on transmission of a Data frame by a source N_Port with a corresponding ACK response frame from the destination N_Port receiving the Data frame to acknowledge Data frame delivery.

Sequence
A Sequence is a set of one or more related Data frames transmitted unidirectionally from one N_Port to another N_Port with corresponding ACK frames transmitted in response to each Data frame. A Sequence is started by an **SOFix** and terminated by an **EOFt**. A Sequence is assigned a SEQ_ID (Sequence_Identifier) by the Sequence Initiator. A Sequence shall only be initiated when an N_Port holds the Sequence Initiative for a given Exchange.

Sequence count
Each frame within a Sequence contains a sequence count (SEQ_CNT) which represents the sequential number of each Data frame transmitted within the Sequence specified by the SEQ_ID. The sequence count is incremented by one for each Data frame transmitted. The ACK response frame contains a Sequence count which is set equal to the Data frame sequence count to which it is responding.



... indicates time delay

Figure 55. Exchange - Sequence relationship

Exchange
An Exchange is a set of one or more related Sequences. Sequences for the same Exchange may flow in the same or opposite direction between a pair of N_Ports but not simultaneously (ie. Data flows in one direction at a time within an Exchange for a single N_Port pair). In Class 1, an Exchange may span multiple Dedicated Connections.

Therefore, an Exchange may be unidirectional or bidirectional. Within a single Exchange only one Sequence shall be active at any given time for a single N_Port. From the standpoint of the Sequence Initiator, a Sequence is active for the period of time from the transmission of the Data frame initiating the Sequence until the end of the last Data frame of the Sequence is transmitted. This allows an N_Port to initate a new Sequence following transmission of the last Data frame of a Sequence before the Sequence being terminated has been completely processed by the Sequence Initiator. The Sequence Initiator is required to use different SEQ_ID values for each of the streamed Sequences in order to properly track status.

For tracking purposes, the Sequence Recipient is required to ensure that consecutive Sequences are processed in the same order as received.

An Exchange is assigned an Originator Exchange_ID (OX_ID) by the Originator and a Responder Exchange_ID (RX_ID) by the Responder facilities within the N_Ports specified by the N_Port Identifiers involved in the Exchange. When an Exchange is originated, there is a binding of resources in both the Originator and Responder.

Since an Exchange utilizes link resources, an N_Port may choose to invalidate and, subsequently, reassign its X_ID value at certain Sequence boundaries during an Exchange. An N_Port indicates its requirement to reassign X_ID values though a Service Parameter during Login. The procedure for invalidating and subsequently, reassigning an N_Port's X_ID uses F_CTL bits in the Frame Header. X_ID's may be invalidated at the end of a Sequence and new X_ID values reassigned at the start of the next Sequence according to the procedure discussed in 24.4. When the X_ID value is invalidated, an N_Port locates its Exchange Status Block by using an Association Header (see 19.3).

Sequence Initiative
The initiative to transmit a Sequence begins with the first Sequence of an Exchange. At the end of each Sequence of the Exchange, the Initiator of the Sequence may transfer the initiative to transfer the next Sequence to the other N_Port, or it may retain the initiative to transmit the next Sequence.

If the Sequence Initiative is held, it is the responsibility of the Initiator to use a different SEQ_ID for the next consecutive Sequence in order to retain appropriate Sequence status information in both the Initiator and Responder.

Operation
An Operation is an optional grouping of one or more Exchanges which are associated by an Upper Level Protocol in order to complete a higher level function. An N_Port may support multiple concurrent Sequences at the same time where each Sequence is associated with a different Exchange. Multiple Exchanges for a given Operation over multiple N_Ports are allowed.

24.2 Applicability

Class 1 and 2:

For Class 1 and 2, FC-2 manages:

- Activation and deactivation of Exchanges,
- Initiation and termination of Sequences,
- Assignment of Exchange_IDs,
- Sequence Initiative
- Assignment of Sequence_IDs,
- Segmentation and Reassembly,
- Sequences,
- Sequence count of frames, and
- Detection and notification of frame Sequence errors.

For Class 1, the Sequence Initiator assigns SEQ_IDs from 0 to (L-1) where L is the number of Concurrent Sequences supplied by the destination N_Port. The Sequence Recipient may assign the SEQ_ID directly to a Recipient Sequence Status Block on an indexed basis.

A Class 1 Sequence requires a Class 1 Dedicated Connection for the duration of the Sequence. When the Connection is terminated, all active Class 1 Sequences are terminated.

For Class 2, the Sequence Initiator may assign SEQ_IDs from 0 to 255. The Sequence Recipient assigns the SEQ_ID to an available Recipient Sequence Status Block.

Class 3:

For Class 3, the Sequence Initiator may assign SEQ_IDs from 0 to 255. FC-2 manages the same functions as Class 1 and 2 excluding notification of frame Sequence errors to the Sequence Initiator.

NOTE - FC-2 defines a suite of functions available to an FC-4 Upper Level Protocol. There are more facilities than a given FC-4 may require.

Based on the needs of a particular protocol, FC-4 may choose only a subset of those functions available.

24.3 Exchange origination

The key facilities, functions, and events involved in the origination of an Exchange by both the Originator and Responder are diagrammed in figure 56.

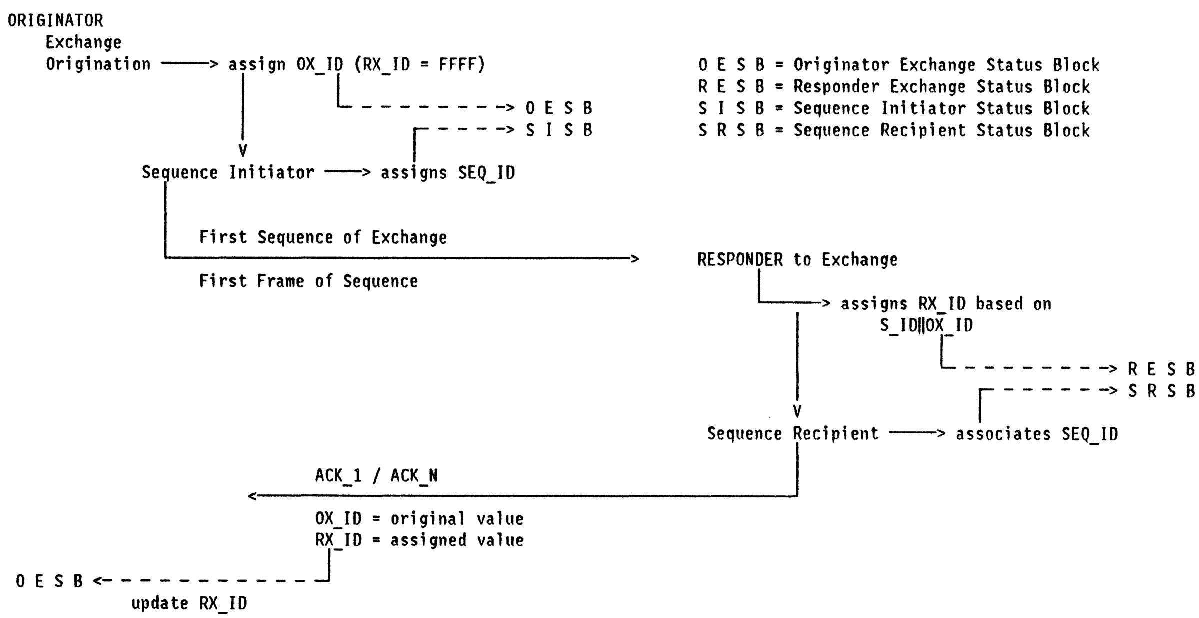


Figure 56. Exchange origination

An Exchange may be originated with a destination N_Port following destination N_Port Login. Login provides information necessary for managing an Exchange and Sequences such as: Class, the number of Concurrent Sequences, Credit, and Receive Data Field Size. Login and Service Parameters are discussed in 23.5. An Exchange is originated through the initiation of a Sequence.

A new Exchange may be originated if two conditions are met:

1. the originating N_Port has an Originator_Exchange_Identifier (OX_ID) available for use, and
2. the N_Port is able to initiate the first Sequence of an Exchange based on the following conditions being met:
   1. the initiating N_Port has a SEQ_ID available for use, and
   2. the total number of active Sequences initiated by this N_Port does not exceed the number of Concurrent Sequences specified by the destination N_Port in its Service Parameters.

Exchange Originator

When an Exchange is originated by an N_Port, that N_Port assigns an Originator Exchange_ID (OX_ID) unique to that N_Port Identifier. An Originator Exchange Status Block associated with the OX_ID is allocated and bound to the Exchange and other link facilities in that N_Port for the duration of the Exchange. All frames associated with that Exchange contain the assigned OX_ID unless the OX_ID is invalidated and reassigned by the Originator. The status of the Exchange is tracked by the Originator in the Originator Exchange Status Block.

Exchange context is identified in each frame transmitted by the Originator via the OX_ID and an F_CTL Bit designating the frame as Originator generated. The Originator Exchange_ID provides the mechanism for tracking request and reply Sequences for multiple concurrent

Exchanges which may be active at the same time.

NOTE - Since the Originator assigns the OX_ID, assignment may be organized to provide efficient processing within the N_Port.

Exchange Responder

The destination N_Port is designated as the Responder for the duration of the Exchange. When the destination N_Port receives the first Sequence of the Exchange, that N_Port assigns a Responder Exchange_ID (RX_ID) associated with the OX_ID from a given S_ID to a Responder Exchange Status Block (S_ID||OX_ID).

The assigned RX_ID shall be transmitted to the Originator on the ACK frame responding to the last Data frame of the Sequence or earlier, if possible. If X_ID interlock has been specified during Login by the Sequence Recipient, the RX_ID shall be assigned in the ACK to the first Data frame of the Sequence. The Originator shall withhold additional frame transmission for the Sequence until the ACK is received. The Responder Exchange_ID provides the mechanism for tracking request and reply Sequences for multiple concurrent Exchanges from multiple S_IDs.

NOTE - Since the Responder assigns the RX_ID, assignment may be organized to provide efficient processing within the N_Port.

Each frame within the Exchange transmitted by the Responder is identified with an F_CTL Bit designating the frame as Responder generated (Exchange context). Each frame within the Exchange transmitted by the Responder is identified with the assigned RX_ID unless the RX_ID is invalidated and reassigned by the Responder. The status of the Exchange is tracked by the Responder in the Responder Exchange Status Block.

Sequence Initiative

The Originator facility controls the origination of an Exchange. The origination of an Exchange establishes the initiative to transmit the first Sequence (Sequence Initiative). Thereafter, the Sequence Initiator facility in either the Exchange Originator or Responder controls holding or transferring the Sequence Initiative to the Sequence Recipient facility in the destination N_Port. The Sequence Initiator facility controls the initiation and termination of Sequences.

If an N_Port initiates a Sequence without holding the Sequence Initiative, the receiving N_Port responds with a P_RJT frame which terminates the Sequence and indicates a reason code of Exchange error.

24.3.1 Frame Characteristics

X_ID fields

In the first frame of an Exchange, the Originator sets the OX_ID to an assigned value and sets the RX_ID value to Hex 'FFFF' (unassigned). When the Responder receives the first Sequence of an Exchange, it assigns an RX_ID and returns the RX_ID in the ACK frame sent in response to the last Data frame in the Sequence, or in an earlier ACK.

If either N_Port has indicated during Login that an X_ID transition interlock is required at Exchange_ID reassignment transitions, then the Sequence Initiator shall not transmit the second Data frame of a Sequence until the corresponding ACK has been received with an assigned X_ID for the Sequence Recipient. The Originator sets the RX_ID to Hex 'FFFF' until the assigned RX_ID is received. When the Originator receives the assigned RX_ID, it sets the RX_ID field to the assigned value for all subsequent frames.

For all remaining frames within the Exchange, OX_ID and RX_ID fields retain these assigned values unless either the Originator or the Responder invalidates its X_ID value at the end of a Sequence.

Exchange delimiters

Since an Exchange is composed of one or more Sequences, the delimiters used for an Exchange are the same as those used to begin and terminate a Sequence. F_CTL Bits are used to indicate the beginning and ending Sequences of an Exchange.

24.4 X_ID reassignment

If an N_Port has indicated during Login that it requires X_ID reassignment, then that N_Port and any N_Port communicating with it shall follow a specific procedure in order to reassign its X_ID value during an Exchange. The procedure requires the use of X_ID transition interlock, an Association Header, X_ID invalidation, and the new X_ID indication.

24.4.1 X_ID transition interlock

When an N_Port inititates a Sequence with an N_Port which has specified during Login that X_ID transition interlock is required and the Recipient's X_ID is invalid, the initiating N_Port shall transmit the first frame of the Sequence with the Recipient's X_ID set to hexadecimal 'FFFF' and shall withhold transmission of the second frame until the corresponding ACK with an assigned X_ID has been received from the Recipient. The assigned X_ID is then used in all subsequent frames in the Sequence.

24.4.2 Association Header

When an N_Port inititates a Sequence with an N_Port which has specified during Login that X_ID reassignment is required and the Recipient's X_ID is unassigned, the Sequence Initiator shall set the Recipient's X_ID to hexadecimal 'FFFF' and shall include an Association Header. The Association Header shall contain the information known by the initiating N_Port at the time of transmission. (see 19.3)

NOTE - It is unnecessary for the Sequence Initiator to transmit an Association Header when the Initiator's X_ID is invalid.

NOTE - An implementation may require that both the Originator and Responder Association_IDs be known prior to initiation of an Exchange. Other implementations may allow dynamic assignment of Association_IDs during the Exchange.

24.4.3 New X_ID

When an N_Port initiates a Sequence and its X_ID is being assigned or reassigned, it indicates that the X_ID is new by specifying the New X_ID bit in F_CTL. When a Sequence Recipient assigns an X_ID in place of an unassigned X_ID (hexadecimal 'FFFF'), it indicates that its X_ID is new in an ACK frame being transmitted to the Sequence Initiator using the New X_ID bit in F_CTL.

24.4.4 Invalidating an X_ID

At the end of a Sequence (End_Sequence bit, F_CTL bit 19), the Sequence Initiator may invalidate its X_ID if it has indicated X_ID reassignment required during Login by setting the Invalidate X_ID bit in F_CTL. At this time the Sequence Recipient may also invalidate its X_ID using the Invalidate X_ID bit in F_CTL in the ACK corresponding to the last Data frame of the Sequence providing the Recipient has indicated this capability during Login. The Sequence Recipient shall only invalidate its X_ID if the Sequence Initiator has invalidated its X_ID in the last frame of a Sequence or the Sequence Initiative is transferred. Otherwise, the Invalidate X_ID bit in F_CTL is ignored by the Sequence Initiator in the corresponding ACK.

24.5 Sequence initiation

Sequence Initiator

The Sequence Initiator facility within an N_Port initiates a Sequence as one step within an Exchange under control of the Originator or Responder facility within the N_Port. When a Sequence is initiated by an N_Port, that N_Port assigns a Sequence_ID (SEQ_ID) uniquely associated with the destination N_Port. That assignment varies by Class as identified in 24.2.

The assigned SEQ_ID is bound to a Sequence Status Block and other link facilities in each N_Port for the duration of the Sequence. All frames associated with that Sequence contain the assigned SEQ_ID. The Initiator also retrieves the appropriate Service Parameters required for processing during the Sequence and associates those parameters with the Sequence Initiator Status Block for the duration of the Sequence.

Each frame transmitted by the Sequence Initiator is identified by the SEQ_ID and an F_CTL Bit designating the frame as Sequence Initiator. The SEQ_ID provides the mechanism for tracking Data and Link_Control frames within the Sequence.

If either N_Port has indicated during Login that an X_ID transition interlock is required, then the Sequence Initiator shall not transmit the second Data frame of a Sequence until the corresponding ACK has been received with an assigned Recipient X_ID when the Recipient's X_ID is invalid. If the Sequence Initiator has invalidated its X_ID in its previous Sequence, then the new X_ID assignment is indicated by setting the New X_ID bit in F_CTL on the first Data frame of a new Sequence. The status of the Sequence is tracked in the Sequence Status Block of the Sequence Initiator.

Sequence Recipient

The destination N_Port is designated as the Sequence Recipient for the duration of the Sequence. When the destination N_Port receives the first frame of the Sequence, that N_Port assigns a Sequence Recipient Status Block to the newly established Sequence. This Status Block is associated with the SEQ_ID and OX_ID received if Exchange Originator, or with the SEQ_ID and RX_ID received if Exchange Responder.

The Recipient also retrieves the appropriate Service Parameters required for processing during the Sequence and associates those Parameters with the Sequence Status Block for the duration of the Sequence. The SEQ_ID provides the mechanism for tracking the sequence count of individual frames transmitted and received during the Sequence.

The Sequence Recipient is the receiver of the Sequence and each frame transmitted by the Sequence Recipient is identified by the SEQ_ID and an F_CTL Bit designating the frame as Sequence Recipient.

If either N_Port has indicated during Login that an X_ID transition interlock is required, then the Sequence Initiator shall not transmit the second Data frame of a Sequence until the Sequence Recipient has responded with an ACK containing an assigned Recipient X_ID when the Recipient's X_ID is invalid. If the Sequence Recipient has invalidated its X_ID in the previous Sequence, then the new X_ID assignment is indicated by setting the New X_ID bit in F_CTL on the ACK to the first Data frame of a new Sequence. The status of the Sequence is tracked in the Sequence Status Blocks of the Sequence Recipient.

Active and Open Sequence

From the standpoint of the Sequence Initiator, a Sequence is active for the period of time from the transmission of the Data frame initiating the Sequence until the end of the last Data frame of the Sequence is transmitted. The Sequence Initiator considers the Sequence open until the ACK with **EOFt** (**EOFdt**) is received, or the Sequence is abnormally terminated.

From the standpoint of the Sequence Recipient, a Sequence is active and open from the time the initiating Data frame is received until the **EOFt** (**EOFdt**) is transmitted in the ACK to the last Data frame, or abnormal termination of the Sequence.

Sequence Initiative

The Sequence Initiative begins with the Originator of the Exchange. The Initiative is held or passed on the last Data frame of a Sequence. The Sequence Initiative is tracked in the Exchange Status Block in both the Originator and Responder facilities. The upper level protocol dictates when the Sequence Initiative is held or passed.

If the Sequence Initiative is held, the next Sequence transmitted shall use a different SEQ_ID in order to retain appropriate Sequence status information in both the Initiator and Responder.

End_Sequence

When the Sequence Initiator is ending the current Sequence, it shall set the End_Sequence bit in F_CTL to one on the last Data frame of the Sequence. It may also invalidate its X_ID if either N_Port has indicated that X_ID reassignment was required during Login by setting the Invalidate X_ID bit in F_CTL. If the Sequence Initiator has indicated that it is invalidating its X_ID or is transferring Sequence Initiative, the Sequence Recipient may also indicate that it is invalidating its X_ID by setting the Invalidate

X_ID bit in F_CTL in the ACK corresponding to the last Data frame of the Sequence.

24.5.1 Frame Characteristics

SOFxi

The first frame of a Sequence is identified by a Start_of_Frame Initiate. Each Class of Service has individual **SOF** delimiters.

When a Dedicated Connection must also be established for Class 1 Service, an **SOFc1** is used to indicate a connect-request. See clause 24 for a discussion on establishing and removing Class 1 Dedicated Connections. The **SOFc1** establishes a Dedicated Connection and also initiates a Sequence.

SEQ_ID

Each frame (Data frame and Link_Control frame) in a Sequence contains the SEQ_ID assigned by the Sequence Initiator.

SEQ_CNT

Each Data frame of a Sequence contains a sequence count (SEQ_CNT) which indicates the order of frame transmission for that Sequence. The SEQ_CNT on each subsequent frame is incremented by one. Each Link_Control frame transmitted in response to a specific frame contains the same SEQ_ID and SEQ_CNT as the frame to which it is responding.

The first frame of a Sequence shall contain a SEQ_CNT of binary zero. If back to back Sequences for the same X_ID occur, it is the responsibility of the Sequence Initiator to use a unique SEQ_ID for each consecutive Sequence. This is required since unique frame identification is based on X_ID||SEQ_ID||SEQ_CNT. Under certain error conditions, the Sequence Initiator is only able to determine where an error occurred by providing this unique frame identification.

Relative Offset

Each frame within a Sequence contains a Relative Offset in the Parameter field. The Relative Offset indicates the relative displacement of Payload portion of the Data Field content (excluding optional headers) within the entire data transmission stream of one or more Sequences within an Exchange. In contrast, frame sequence count represents the order of frame transmission, not necessarily consecutive, contiguous data elements.

If FC-2 encounters a wrap condition, it is detected as an error condition. The Exchange is terminated using the Abort Exchange Protocol and the appropriate FC-4 within this N_Port is notified of the ending status.

24.5.2 Busy or Reject condition

If destination link control facilities are not available when the frame is received which initiates a Sequence, the destination N_Port responds with a P_BSY frame. A P_RJT frame may also be transmitted in response based on invalid data in the content of the request. Busy reason codes are specified in 20.3.3.2. Reject reason codes are specified in 20.3.3.1.

24.6 Sequence management

Intermediate frames within a Sequence use the following F_CTL and Frame_Header fields.

Frame_Header fields

- OX_ID - as assigned, or Hex 'FFFF' if unassigned
- RX_ID - Hex 'FFFF', then as assigned
- SEQ_ID - as assigned
- SEQ_CNT - increments by one for each Data frame
- Relative Offset - appropriate values

24.6.1 Sequence timeout

A Sequence timeout is detected if an expected frame is not received within the timeout period. A Sequence timeout normally results from a lost or corrupted frame.

Sequence Initiator

The Sequence Initiator sets a timer value for each Data frame transmitted. If the corresponding ACK (ACK_1 or ACK_N), or a Link_Response frame is not received within the timeout period, a Sequence timeout is detected. The Sequence Initiator aborts the Sequence using the Abort Sequence Protocol. See clause 29 for a discussion on Sequence recovery.

Sequence Recipient

After a Sequence has been initiated by another N_Port, the Sequence Recipient sets a timer value upon reception of each Data frame. If no Data frame is received during the timeout period before the last Data frame of the Sequence is received, the Sequence Recipient detects a Sequence timeout. When a Sequence Recipient detects a Sequence timeout, the Sequence is terminated. The Sequence status is saved in the associated Exchange Status Block and the Sequence Status Block is reintialized and available for reuse.

If a Data frame is received for the terminated Sequence after a timeout, the Sequence Recipient responds with an ACK frame indicating that the Sequence was aborted by setting the Abort Sequence Condition bits in F_CTL.

24.7 Sequence termination

When a Sequence is terminated, the last Data frame transmitted by the Sequence Initiator is used to identify two conditions:

1. Sequence initiative, and
2. Sequence termination.

24.7.1 Sequence initiative

The Sequence Initiator controls which N_Port is allowed to initiate the next Sequence for the Exchange. The Sequence Initiator may hold the initiative to transmit the next Sequence of the Exchange or the Sequence Initiator may transfer the initiative to transmit the next Sequence of the Exchange. The decision to hold or transfer initiative is indicated by F_CTL Bit 16.

24.7.2 Termination

The last Data frame transmitted by the Sequence Initiator is indicated by setting F_CTL Bit 19 to one. The Sequence is terminated by either the Initiator or the Recipient transmitting a frame terminated by **EOFt**. The Sequence Initiator is in control of terminating the Sequence. Transmission of the **EOFt** may occur in two ways:

1. In Class 1 and 2, the Sequence Recipient transmits an ACK frame of ACK_1 or ACK_N in response to the last Data frame of the Sequence (F_CTL Bit 19 = 1) ended by an **EOFt**.
2. In Class 3, the Sequence Initiator transmits the last Data frame of the Sequence with **EOFt**.

Class 1

When **EOFt** has been transmitted or received by each N_Port, the appropriate Exchange Status Block associated with the Sequence is updated in each N_Port to indicate that the Sequence was completed and whether the Originator or Responder facility holds the Sequence Initiative. Link facilities associated with the Sequence (including Sequence Status Block) are released and available for other use.

Class 2

Since Class 2 frames may be delivered out of Sequence, Sequence processing is only completed after all frames (both Data and ACK) have been received and processed by the respective N_Ports. The Sequence Recipient shall withhold transmitting **EOFt** on the last ACK in response to the last Data frame of the Sequence until all preceeding Data frames have been received, processed, and corresponding ACK frames transmitted.

When the Sequence is completed by each N_Port, the appropriate Exchange Status Block associated with the Sequence is updated in each N_Port to indicate that the Sequence was completed and whether the Originator or Responder facility holds the Sequence Initiative. Link facilities associated with the Sequence (including Sequence Status Block) are released and available for other use.

Class 3

The Sequence Initiator transmits all Data frames and terminates the last Data frame of the Sequence with **EOFt**. Acknowledgment of Sequence completion is the responsibility of the Upper Level Protocol.

When the Sequence is completed by each N_Port, the appropriate Exchange Status Block associated with the Sequence is updated in each N_Port to indicate that the Sequence was completed and whether the Originator or Responder facility holds the Sequence Initiative. Link facilities associated with the Sequence (including Sequence Status Block) are released and available for other use.

24.7.3 Abnormal termination

A Sequence may also be terminated abnormally by either the Sequence Initiator or Recipient (also see 29.7.1). The Sequence Initiator abnormally terminates a Sequence by using the Abort Sequence Protocol for:

1. a malfunction or error detected in the Initiator N_Port facility,
2. failure to retransmit a frame with an **EOFa** delimiter,
3. F_CTL Bits 5-4 set to 0 1 or 1 0 on an ACK, or
4. Sequence Initiator Sequence timeout.

The Sequence Recipient abnormally terminates a Sequence when a Sequence timeout has been detected within the N_Port, or when directed to terminate the Sequence by an Abort Sequence frame from the Sequence Initiator.

If a Sequence Recipient has encountered a Sequence error prior to or at the termination of an active Sequence, it may also abnormally terminate a consecutive, streamed Sequence for the same Exchange.

24.8 Exchange management

An Exchange is managed as a series of unidirectional Data frame Sequences. The initial Sequence is transmitted by the Originator of the Exchange. Control and intermixing of Sequences within an Exchange are identified by CTL Bits within the Frame Header.

Following the initial Sequence, subsequent Sequences may be transmitted by either the Originator or the Responder facilities based on which facility holds the Sequence Initiative.

24.9 Exchange termination

An Exchange may be terminated by either the Originator or the Responder. The facility terminating the Exchange indicates Exchange termination on the the last Sequence of the Exchange by setting the Last_Sequence bit in F_CTL on each frame of the last Sequence of the Exchange.

The Sequence is terminated according to normal Sequence termination rules. When the last Sequence of the Exchange is terminated, the Exchange is also terminated. The OX_ID and RX_ID and associated Exchange Status Blocks are released and available for reuse.

24.9.1 Abnormal termination

An Exchange may be abnormally terminated by either the Originator or the Responder by using the Abort Exchange Protocol. In general, reception of a reject frame with action codes of 2 or 4 are not recoverable at the Sequence level and aborting of the Exchange is probable. Other reasons to abort an Exchange are FC-4 protocol dependent and not defined within this Standard.

24.10 Status Blocks

Exchange Status Block

An Exchange Status Block is a logical construct used to associate the OX_ID, RX_ID, D_ID and S_ID of an Exchange. The Status Block is used throughout the Exchange to track the progress of the Exchange and identify which N_Port (or node) holds the initiative to transmit Sequences. An Exchange Status Block is created in the Originator (OESB) and in the Responder (RESB) when an Exchange is originated. The Status Blocks are released when the Exchange is terminated.

Sequence Status Block

A Sequence Status Block is a logical construct used to track the progress of a single Sequence by an N_Port on a frame by frame basis. The Status Block is created in the Sequence Initiator (SISB) and in the Sequence Recipient (SRSB) when the Sequence is initiated. The Status Blocks are released when the Sequence is terminated.

24.11 Rules

24.11.1 Exchange origination

1. A new Exchange may be originated if three conditions are met:
   A. The originating N_Port has performed Login with the destination N_Port.
   B. The originating N_Port (or node) has an Originator_Exchange_Identifier (OX_ID) available for use.
   C. The N_Port is able to initiate a new Sequence.
2. Each frame within the first Sequence of an Exchange shall set F_CTL Bit 21 to one.
3. The Originator transmits the first Data frame of the Exchange with its assigned OX_ID and an unassigned RX_ID of hexadecimal 'FFFF' by setting the New X_ID bit in F_CTL.
4. The rules specified in Sequence initiation and termination specify the method for assigning and reassigning X_IDs.

24.11.2 Sequence Initiation

1. A new Sequence may be initiated if three conditions are met:
   A. The initiating N_Port holds the initiative to transfer (Sequence Initiative).
   B. The initiating N_Port has a SEQ_ID available for use.
   C. The total number of active Sequences initiated by this N_Port does not exceed the number of Concurrent Sequences specified by the destination N_Port in its Service Parameters.
2. The first Data frame of the Sequence is started by an **SOFix** (or an **SOFc1** for the first frame establishing a Class 1 Connection).
3. The Sequence Initiator assigns its X_ID (either OX_ID or RX_ID) to its assigned value and indicates if it is a new assignment by setting the New X_ID bit in F_CTL on the first frame of a Sequence.
4. A new X_ID shall not be assigned after the first Sequence of an Exchange unless the previous X_ID has been invalidated.
5. The Sequence Initiator uses the Recipient's X_ID value assigned by the Recipient or hexadecimal 'FFFF', if unassigned. If the Recipient's X_ID is unassigned and the Recipient requires X_ID reassignment, as specified during Login, the Initiator also transmits an Association Header.
6. When a Sequence is initiated, if the Recipient's X_ID is unassigned and X_ID transition interlock has been specified during Login, the Sequence Initiator withholds transmitting the second frame of the Sequence until the Recipient X_ID is received in the ACK corresponding to the first frame is received by the Initiator.
7. Frame transmission shall follow Flow Control Rules as specified in clause 25.

24.11.3 Sequence management

1. Each frame shall contain the assigned OX_ID, RX_ID, and SEQ_ID.
2. Hexadecimal 'FFFF' shall be used for unassigned X_ID values.
3. Each frame shall indicate the Exchange Context.
4. Each frame shall indicate the Sequence Context.
5. Each frame shall contain a SEQ_CNT which follows the Sequence Count rules as defined.
6. Frame transmission shall follow Flow Control Rules as specified in clause 25.
7. The Data Field size of each frame of the Sequence shall be less than or equal to:
   A. the maximum size specified by the Fabric (**SOFc1**, Class 2, or 3), if present, or
   B. the maximum size specified by the destination N_Port in the Service Parameters defined during Login, whichever is smaller.

24.11.4 Sequence count

Within a Data frame Sequence, SEQ_CNT is used to identify each Data frame for verification of delivery. The following rules specify the sequence count of each frame of a Sequence:

1. The sequence count (SEQ_CNT) of the first Data frame of a new Sequence is binary zero. (If back to back Sequences for the same X_ID occur, it is the responsibility of the Sequence Initiator to use a unique SEQ_ID for consecutive Sequences.)

2. The sequence count of each subsequent Data frame is incremented by one from the previous Data frame.

3. The sequence count of each Link_Response shall be set to the sequence count of the Data frame to which it is responding (Class 1 and 2).

4. The sequence count of each ACK_1 frame shall be set to the sequence count of the Data frame to which it is responding. (Class 1 and 2)

5. The sequence count of each ACK_N frame shall be set to the highest sequence count of a series of Data frames being acknowledged. (Class 1 and 2). See Annex M for ACK_N rules.

6. The sequence count shall not wrap within a Sequence.

24.11.5 Sequence termination

1. The Last Data frame of a Sequence shall be indicated by F_CTL Bit 19 set to one.
2. The Sequence Recipient shall transmit an ACK frame of ACK_1 or ACK_N in response to the last Data frame of the Sequence (F_CTL Bit 19 = 1) ended by an **EOFt** (or **EOFdt**) in Class 1 and Class 2.

   — Class 1 - the Sequence Recipient shall transmit the last ACK response after receiving the last Data frame.

   — Class 2 - the Sequence Recipient shall withhold transmission of the last ACK frame until all preceding Data frames have been received, processed, and corresponding ACK frames transmitted.

3. In the last Data frame of a Sequence, the Sequence Initiator sets the
   — Sequence Initiative bit in F_CTL to 0 to hold Sequence Initiative.
   — Sequence Initiative bit in F_CTL to 1 to transfer Sequence Initiative.
   — Invalidate X_ID bit in F_CTL to 0 to retain X_ID assignment.
   — Invalidate X_ID bit in F_CTL to 1 to invalidate its current X_ID.
4. If the last Data frame of a Sequence has the Invalidate X_ID bit set to one or transfers Sequence Initiative, the Sequence Recipient sets the
   — Invalidate X_ID bit in F_CTL to 0 to retain X_ID assignment in the ACK response.
   — Invalidate X_ID bit in F_CTL to 1 to invalidate its current X_ID in the ACK response.

24.11.6 Sequence delimiters

The following rules specify the management of frame delimiters within a Sequence:

1. A Sequence shall be initiated by transmitting the first frame with an **SOFix**, or **SOFc1**.

2. Intermediate frames within a Sequence shall be transmitted with **SOFnx** and **EOFn**.

3. The Sequence shall be complete when an **EOFt** (or **EOFdt**) has been transmitted or received for the appropriate Sequence_ID.

24.11.7 Exchange termination

1. The last Sequence of the Exchange shall be indicated by setting F_CTL Bit 20 to one.
2. The Exchange shall be terminated when the last Sequence is terminated by normal Sequence termination rules.

25 Flow control management

25.1 Introduction

Flow Control is a FC-2 control process to pace the flow of frames between N_Ports and between an N_Port and the Fabric to prevent overrun at the receiver. Flow Control is managed using end-to-end Credit, end-to-end Credit_CNT, ACK (ACK_1 or ACK_N), buffer-to-buffer Credit, buffer-to-buffer Credit_CNT, and R_RDY along with other frames.

Flow Control is managed between N_Ports (end-to-end) and/or between N_Port and F_Port (buffer-to-buffer). Flow Control management has variations dependent upon the Class of Service.

Applicability
Class 1 uses end-to-end flow control. Class 2 uses both end-to-end and buffer-to-buffer flow controls. Class 3 uses only buffer-to-buffer flow control. Table 47 shows the applicability of flow control mechanisms to each Class.

25.2 Physical flow control model

The physical flow control model is illustrated in figure 57. The model consists of following physical components:

- Each N_Port with Class 1 and/or Connectionless (Classes 2and 3 combined) receive buffers.
- Each F_Port to which an N_Port is attached, with its receive buffers (Classes 2 and 3 combined) for Connectionless Service. (Class 1 buffers internal to Fabric used for Class 1 service and Intemix are transparent to FC_2 flow control.)

Buffer participation
Buffering and transmission of Class 1 frames through the Fabric is transparent to FC-2. Class 1 buffering requirements during Intermix are specified in FC-2 (see section 22.4, "Intermix" form=numonly). The use of Class 1 buffers during Intermix is transparent to flow control. Class 2 and Class 3 buffers are shared by Class 2 and Class 3 frames. The following table summarizes the use of buffers for end-to-end and buffer-to-buffer flow controls.

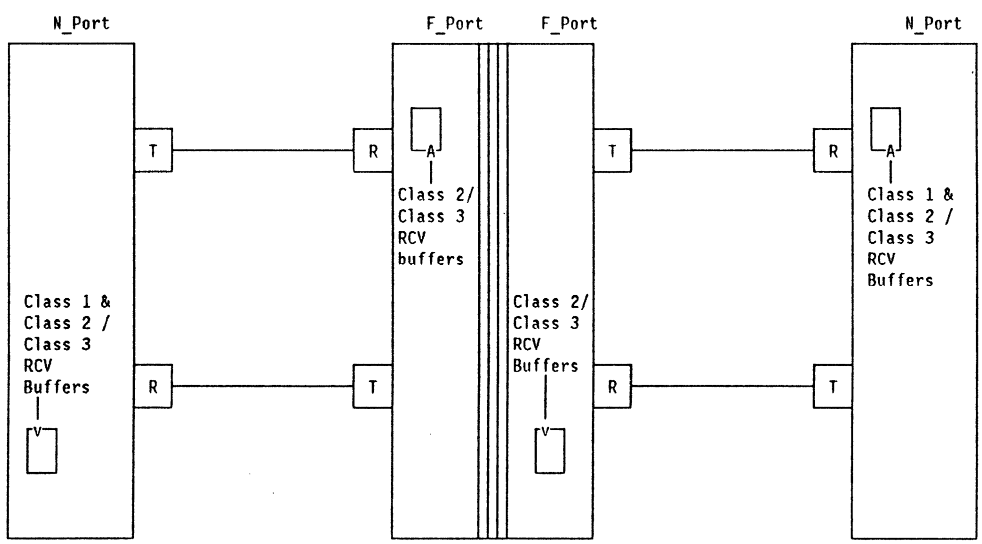


Figure 57. Physical flow control model

Table 46. Buffer participation		
Participating buffers	**End to end flow control**	**Buffer to buffer flow control**
Class 1 buffers*	Yes	NA
Connectionless buffers (Classes 2 and 3 combined)	Class 2	Yes
Note: * Participation of Class 1 buffers in the Fabric during Intermix is transparent to flow control.		

Naming convention
The F_Port attached to the Sequence Initiator N_Port is referred to as Frontend F_Port and the one attached to the Sequence Recipient N_Port as Backend F_Port.

25.3 Credit and Credit_Count

Credit is the number of receive buffers allocated and/or available to a transmitting port (an N_Port or an F_Port). Two types of Credits used in FC-2 flow control are:

1. End-to-end Credit
2. Buffer-to-buffer Credit

The Credit_Count is the running count of the Credit that the transmitting port (the N_Port or F_Port) uses to track the number of receive buffers occupied. The Credit_Count shall not exceed the value of Credit to avoid the possibility of overflow at the receiver. Corresponding to two types of Credits listed above, two types of Credit_Counts used in FC-2 are:

1. End-to-end Credit_Count
2. Buffer-to-buffer Credit_Count

Usage
The N_Port transmitting Data frames shall use the end-to-end Credit allocated by the receiving N_Port for end-to-end flow control and manage the corresponding end-to-end Credit_Count. When a port (an N_Port or an F_Port) is transmitting Data frames or Link_Control frames to the attached port (F_Port or N_Port respectively), the transmitting port shall use buffer-to-buffer Credit allocated by the receiving port for buffer-to-buffer flow control and manage the corresponding buffer-to-buffer Credit_Count.

25.3.1 End-to-end Credit

End-to-end Credit (EE_Credit) is the number of receive buffers in the Sequence Recipient N_Port that have been allocated to a given Sequence Initiator N_Port. End-to-end Credit represents the maximum number of unacknowledged or outstanding frames that can be transmitted without the possibility of overrunning the receiver at the Sequence Recipient N_Port. End-to-end Credit is defined per Class per Sequence Recipient N_Port and managed by the Sequence Initiator N_Port. Class 1 End-to-end Credit represents the number of Class 1 receive buffers and Class 2 End-to-end Credit the number of Class 2 buffers allocated to the Sequence Initiator N_Port. (EE_Credit is not applicable to Class 3.) The value of End-to-end Credit allocated to the Sequence Initiator N_Port is conveyed to this N_Port through the End-to-end Credit field of the Service Parameters. The minimum value of End-to-end Credit is one (1).

End-to-end Credit is used as a controlling parameter in end-to-end flow control.

Table 47. Flow Control applicability				
Flow Control methodology and mechanism	**Class 1**	**Connect request frame with SOFc1**	**Class 2**	**Class 3**
End-to-end	Yes	Yes	Yes	No
Buffer-to-buffer	No	Yes	Yes	Yes
ACK_1 or ACK_N	Yes	Yes	Yes	No
R_RDY	No	Yes	Yes	Yes
F_BSY/F_RJT	No*	Yes	Yes	No
P_BSY/P_RJT	No*	Yes	Yes	No

25.3.2 End-to-end Credit_Count

End-to-end Credit_Count (EE_Credit_CNT) is defined as the number of unacknowledged or outstanding frames awaiting a response and represents the number of receive buffers that are occupied at the Sequence Recipient N_Port. To track the number of frames transmitted and outstanding, the Sequence Initiator N_Port uses this variable called End-to-end Credit_Count (End-to-end Credit_CNT).

EE_Credit is obtained by a Sequence Initiator N_Port during N_Port Login from the N_Port to which it is logging into. EE_Credit allocated by the Sequence Recipient N_Port forms the maximum limit for the EE_Credit_CNT value. The EE_Credit_CNT value is initially set at zero (0), immediately after Login. The EE_Credit_CNT is incremented, decremented or left unaltered as specified by the flow control management rules (sections 25.4.2 and 25.5.4). The EE_Credit_CNT shall not exceed the EE_Credit value to avoid possible overflow at the receiver.

The Sequence Initiator shall allocate the total N_Port Credit associated with a Sequence Recipient among all active Sequences associated with that Recipient. The Sequence Initiator function may dynamically alter the Credit associated with each active Sequence as long as the total N_Port Credit specified for the Sequence Recipient is not exceeded. In the event of an abnormal termination of a Sequence using the Abort Sequence Protocol, the Sequence Initiator may reclaim the Sequence Credit allocation when the Accept response has been received to the Abort Sequence frame.

The N_Port is responsible for managing EE_Credit_CNT using EE_Credit as the upper bound on a per port basis.

25.3.3 End-to-end Class dependency

Allocation of EE_Credit and management of EE_Credit_CNT have some variations dependent upon Class of Service.

EE_Credit allocation

- Each Sequence Recipient N_Port may allocate the same Class 1 N_Port Credit value to each N_Port it is logging into. This Class 1 Credit value may be the maximum supportable by the Sequence Recipient N_Port.
- Each Sequence Recipient N_Port allocates some number of its receive buffers for Class 2 Service to N_Ports it is logging into. The sum of allocated Class 2 buffers may exceed the total number of Class 2 buffers supported at the Sequence Recipient N_Port. This excess buffer allocation may not necessarily result in overrun. Class 2 EE_Credit allocation depends upon system requirements which are outside the scope of this standard.

EE_Credit_CNT management

- Since Class 2 supports demultiplexing to multiple Sequence Recipient N_Ports, the Sequence Initiator N_Port manages a Connectionless EE_Credit_CNT for each Sequence Recipient N_Port currently active, with that Sequence Recipient N_Port's EE_Credit as the upper bound.
- An Class 3 N_Port does not perform EE_Credit_CNT management.

Note - Login ensures that appropriate values are interchanged.

25.3.4 EE_Credit management

EE_Credit management involves an N_Port establishing and revising EE_Credit with the other N_Port it intends to communicate with, for Class 1 or Class 2 or both. N_Port Login is used to establish and optionally revise these EE_Credit values. The Service Parameters interchanged during N_Port provide the Class 1 and/or Class 2 EE_Credit separately in their respective Credit fields.

25.3.5 Buffer-to-buffer Credit

Buffer-to-buffer Credit (BB_Credit) represents the number of receive buffers for Connectionless Service (Class 2 and/or Class 3) supported by a port. (N_Port or F_Port). BB_Credit values of the attached ports are mutually conveyed to each other during the F_Port Login thorough the Credit field of Service Parameters. Minimum value of BB_Credit that an F_Port shall support is one (1).

BB_Credit is used as the controlling parameter in buffer-to-buffer flow control.

25.3.6 Buffer-to-buffer Credit_Count

Buffer-to-buffer Credit_Count (BB_Credit_CNT) is defined as the number of unacknowledged or outstanding frames awaiting R_RDY responses from the directly attached port. It represents the number of receive buffers that are occupied at the attached port. To track the number of frames transmitted for which R_RDY responses are outstanding, the transmitting port uses this variable called BB_Credit_Count.

The transmitting port is responsible to manage BB_Credit_CNT with BB_Credit as its upper bound.

25.3.7 Buffer-to-buffer Class dependency

Allocation of BB_Credit and management of BB_Credit_CNT for Connectionless Service are described.

BB_Credit allocation
Each port allocates the total number of Connectionless buffers (Classes 1 and 2 combined) to the port it is directly attached to.

BB_Credit_CNT management
A port manages the BB_Credit_CNT with BB_Credit as the upper bound.

25.3.8 BB_Credit management

BB_Credit management involves a port receiving the BB_Credit value from the port it is directly attached to. F_Port Login is used to accomplish this. The Service Parameters interchanged during F_Port Login provide the number of combined Class 2 and Class 3 buffers. These combined values are provided in both Class 2 and Class 3 Credit fields.

25.4 End-to-end flow control

End-to-end flow control is an FC-2 control process to pace the frames between N_Ports. End-to-end flow control is used by an N_Port pair on Class 1 or Class 2.

End-to-end flow control is performed with EE_Credit_CNT with EE_Credit as the controlling parameter.

25.4.1 End-to-end flow control model

The end-to-end flow control model is illustrated in figure 58. The model includes flow control parameters, control variables and resources for a Data frame from a Sequence Initiator N_Port and ACK_1 or ACK_N or BSY/RJT in response from the Sequence Recipient N_Port.

- The Sequence Recipient N_Port provides a number of Class 1 and/or Class 2 receive buffers.
- The Sequence Initiator N_Port obtains the allocation of Class 1 and/or Class 2 receive buffers, as Class 1 and/or Class 2 EE_Credits, respectively. (That allocation is distributed among all the active Sequences for a specific Sequence Recipient.)
- The Sequence Initiator N_Port manages the end-to-end flow by managing Class 1 and/or Class 2 EE_Credit-CNT(s). (That management is distributed among all the active Sequences for a specific Sequence Recipient.)

The model illustrates all possible replies to the Data frame. The EE_Credit_CNT is decremented by one (1) or m depending upon the type of Link_Control frame received.

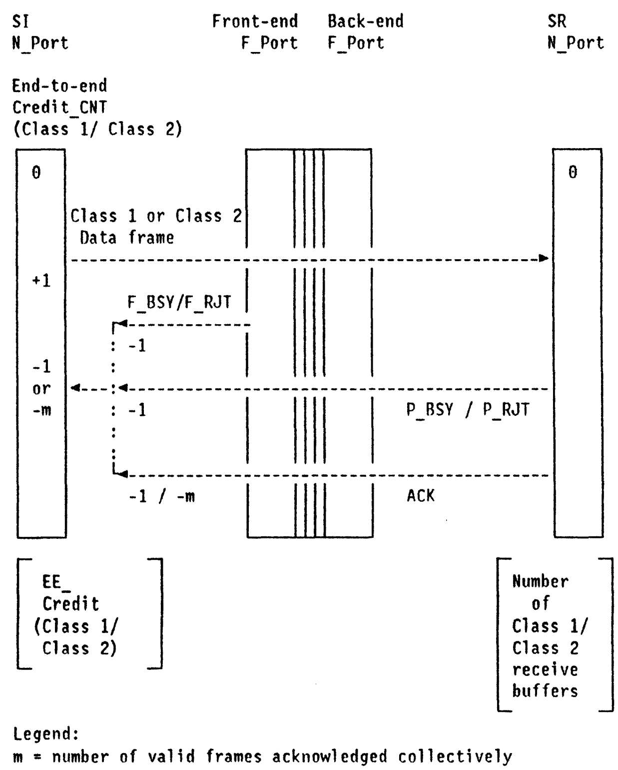


Figure 58. End-to-end flow control model

25.4.2 End-to-end management rules

End-to-End management rules are described in this section. Management of EE_Credit_CNT is summarized in table 49.

1. The Sequence Initiator N_Port is responsible for managing EE_Credit_CNT across all active Sequences.
2. The Sequence Initiator N_Port shall not transmit a Data frame unless the allocated Credit is greater than zero.
3. In Class 1 or Class 2, the initial value of the EE_Credit_CNT is set to zero (0) during Login or re-Login.
4. In Class 1, the EE_Credit_CNT is set to one (1), on transmitting the frame with SOFc1. It is incremented by one (1) for each subsequent Class 1 frame transmitted.
5. The EE_Credit_CNT incremented by one (1) for each Class 2 Data frame transmitted.
6. The Sequence Initiator N_Port decrements the EE_Credit_CNT by a value of one for each ACK_1 received and by a value of m for each ACK_N received where m is the number of valid frames received and acknowledged by the Sequence Recipient N_Port.
7. To avoid possible overrun at the receiver, the Sequence Initiator N_Port manages the EE_Credit_CNT not to exceed EE_Credit.

25.4.3 ACK_1 rules

1. The Sequence Recipient transmits an ACK_1 in response to each Data frame received and returns the SEQ_CNT of the Data frame to which it is responding.
2. The Sequence Recipient sets the discard bit in all ACK_1 frames transmitted after detection of a SEQ_CNT error condition, including the frame at which the error is detected if the discard error policy is specified.

25.4.4 ACK_N rules

1. The ACK_N window (maximum value of "m", the number of consecutive frames acknowledged collectively) chosen by the Sequence Recipient shall be less than Credit.

   Note - If the ACK_N window and the Credit are equal and a frame is lost, the receiver will be waiting for the frame and the sender will be waiting for the ACK resulting in a "hang" condition. "m less than Credit" helps prevent the "hang" condition unless multiple frames (equal to Credit minus m) are lost. Implementers are advised caution that "m less than Credit" does not help elimination of "hang" condition if Credit is collectively managed for all active Sequences. The hang condition is detected through timeout and rectified through Credit_Count recovery (see -- Heading 'TIMEOUT' unknown -- and 25.4.5).
2. ACK_N shall report the number of all consecutive frames received within the ACK_N window (in Credit acknowledged field).
3. ACK_N shall include the maximum sequence count of consecutive frames being reported (in SEQ_CNT field).
4. ACK_N shall be issued at the end of the Sequence.
5. ACK_N shall be sent on detection of a missing or a invalid frame.

(See ACK_N usage examples in appendix Annex M.)

25.4.5 EE_Credit recovery

1. In Class 1 and Class 2, EE_Credit can be recovered by Sequence Initiator when a Sequence is terminated, by the number of unacknowledged Data frames associated with the Sequence being terminated. Termination may be normal or abnormal.
2. In Class 1, EE_Credit may be recovered by the Sequence Initiator within the Sequence by detection of SEQ_CNT discontinuity in ACK_1.
3. Class 1 EE_Credit is also recovered when a Dedicated Connection is removed by either **EOFdt** or by the Link Recovery Protocol.
4. In Class 1 and 2, EE_Credit may also be recovered by an N_Port through re-Login.

25.5 Buffer-to-buffer flow control

Buffer-to-buffer flow control is an FC-2 staged control process to pace the flow of frames. The buffer-to-buffer stages are:

- Sequence Initiator N_Port to Front-end F_Port
- Back-end F_Port to Sequence Recipient N_Port

25.5.1 Buffer-to-buffer flow control model

The Buffer-to-buffer flow control model is illustrated in figure 59. The model includes flow control parameters, control variables for a Class 2 or Class 3 Data frame and R_RDY as its response, and the resources (Class 2 and/or Class 3 receive buffers) for Connectionless Service. All possible responses to a Class 2 or Class 3 Data frame are illustrated.

- Each N_Port and F_Port provides a number of receive buffers for Connectionless Service.
- Each N_Port obtains the allocation of receive buffers from the F_Port it is attached to, as BB_Credit. Each F_Port obtains the allocation of receive buffers from the N_Port it is attached to, as total BB_Credit for Connectionless Service.
- Each port manages the buffer-to-buffer flow by managing the BB_Credit_CNT for the Connectionless Service, with BB_Credit as the upper limit.

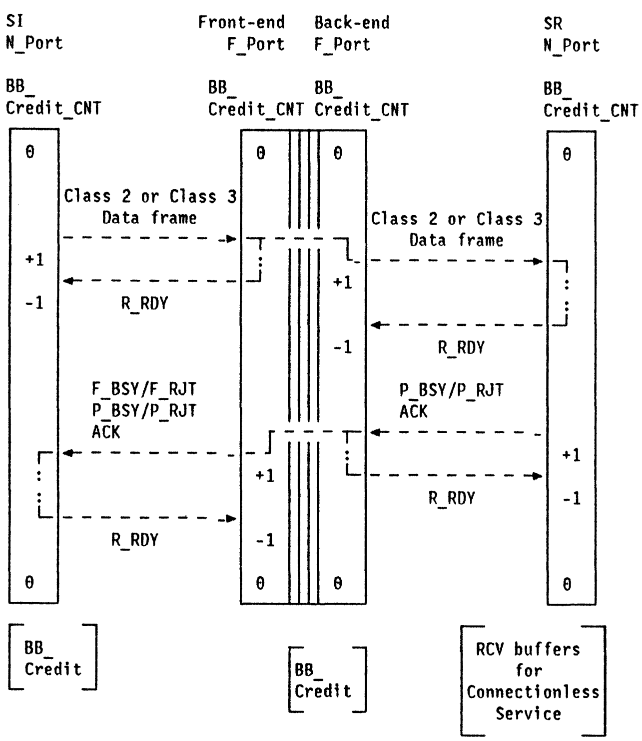


Figure 59. Buffer-to-buffer flow control model

25.5.2 Class dependent frame flow

Figures 60, 61 and 62 illustrate the flow of frames to accomplish buffer-to-buffer Control, respectively for the following cases:

- Class 2 with delivery or non-delivery to the Fabric.
  - Possible responses from the F_Port or an N_Port within the Fabric to a Class 2 Data frame are illustrated in figure 60.
- Class 2 with delivery or non-delivery to an N_Port.
  - Possible responses from the F_Port and the destination N_Port to a Class 2 Data frame are illustrated in figure 61.
- Class 3.
  - Possible responses from the F_Port and the destination N_Port to a Class 3 Data frame are illustrated in figure 61.

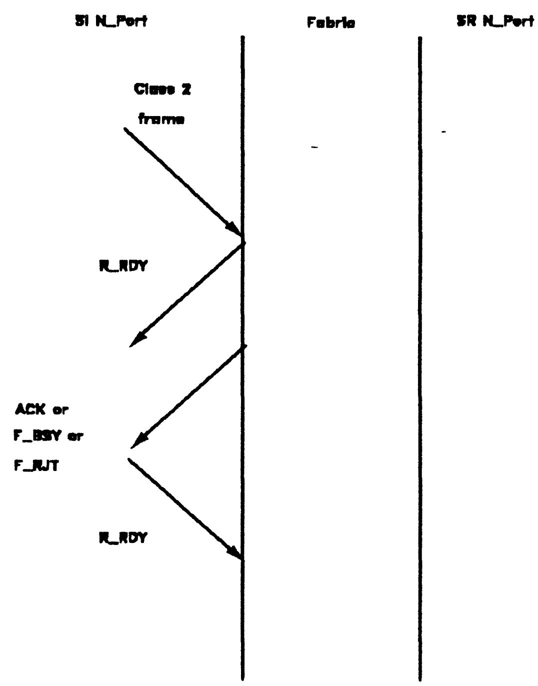


Figure 60. Buffer-to-buffer - Class 2 frame flow with delivery or non-delivery to the Fabric

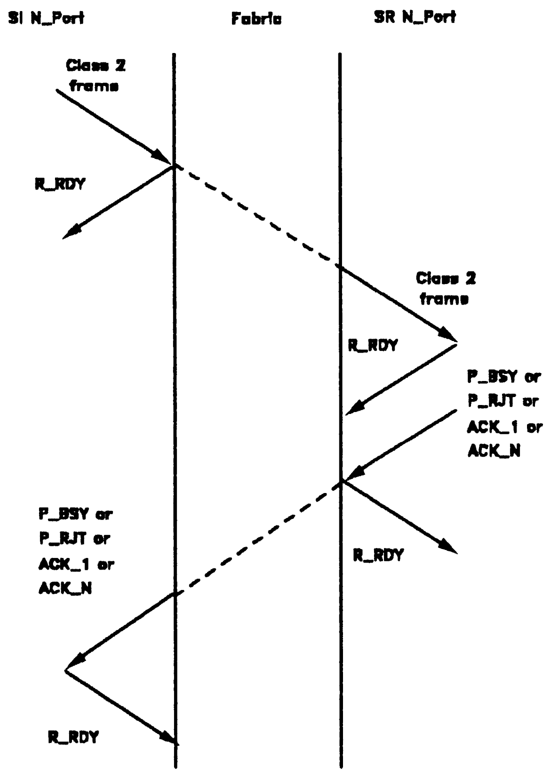


Figure 61. Buffer-to-buffer - Class 2 frame flow with delivery or non-delivery to an N_Port

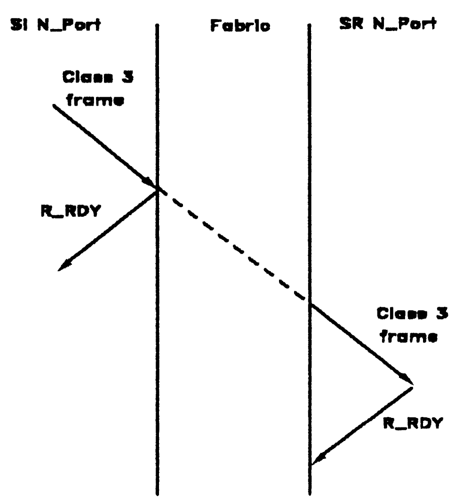


Figure 62. Buffer-to-buffer - Class 3 frame flow

25.5.3 R_RDY

R_RDY is the pacing mechanism exclusively used for buffer-to-buffer flow control. For any Class 2 or Class 3 or Class 1 frame with SOFc1 received at an F_Port or at an N_Port, each port issues an R_RDY primitive.

25.5.4 Buffer-to-buffer management rules

Buffer-to-buffer flow control rules are described in this section. Managing BB_Credit_CNT at an N_Port or an F_Port is summarized in table 50.

1. Each port is responsible for managing the BB_Credit_CNT.
2. Each port sets BB_Credit_CNT value to zero (0) during Login or re-Login.
3. Each port increments BB_Credit_CNT by one (1) for each Class 2 or 3 frame transmitted and decrements by one (1) for each R_RDY received.
4. Each port issues an R_RDY for each Class 2 or 3 frame received.
5. To avoid possible overrun at the receiver, each port manages BB_Credit_CNT not to exceed BB_Credit.

25.5.5 R_RDY rule

N_Port and F_Port transmits an R_RDY in response to each Class 2, Class 3 or Class 1 frame with SOFc1 received.

25.5.6 BB_Credit_Count reset

BB_Credit_Count is reset by an N_Port on re-Login and after the Link Recovery Protocol has been performed.

25.6 BSY / RJT in flow control

In Class 1 end-to-end flow control, F_BSY or P_BSY do not occur, except for a Class 1 Request with SOFc1. In Class 2 end-to-end flow control, F_BSY, F_RJT, P_BSY or P_RJT may occur for any Data frame. Each of these responses contributes to end-to-end and buffer-to-buffer flow controls.

Class 2 buffers at the Sequence Recipient N_Port are shared by multiple source N_Ports which may be multiplexing. This Class 2 multiplexing requires the distribution of Class 2 Credit to each source N_Port to be honored, to prevent BSY or RJT. Unless an adequate number of Class 2 buffers is available and Credit distribution is honored, a BSY or RJT may occur in Class 2.

25.7 Integrated Class 2 flow control

Integrated buffer-to-buffer and end-to-end flow controls applicable to Class 2 is illustrated in figure 63 for a Data frame from the Sequence Initiator N_Port and its response from the Sequence Recipient N_Port.

25.7.1 Management

Integrated Class 2 flow control management is summarized in table 48.

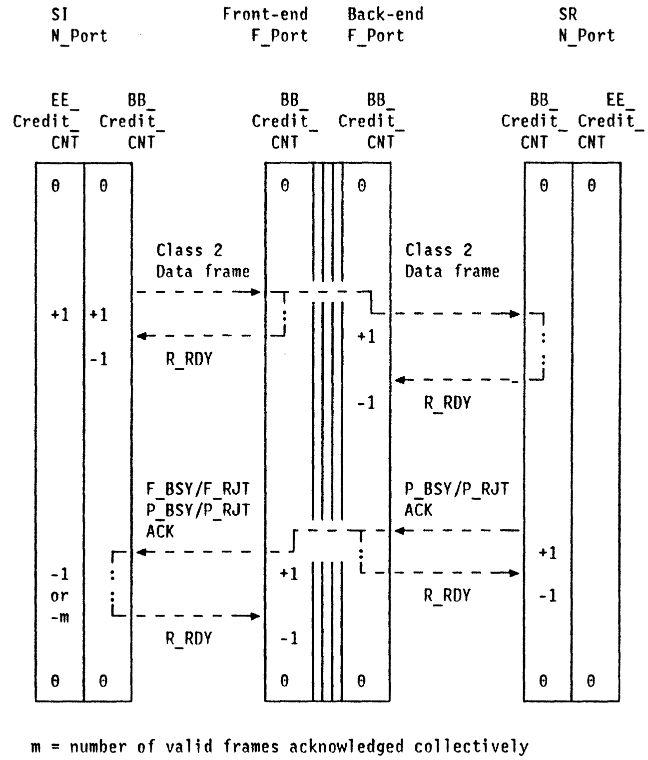


Figure 63. Integrated Class 2 flow control

Table 48. Integrated Class 2 flow control management

Activity	N_Port		F_Port
	EE_Credit_CNT	BB_Credit_CNT	BB_Credit_CNT
Port transmits a Class 2 Data frame	+1	+1	+1
Port receives R_RDY	NA	-1	-1
Port receives F_BSY, F_RJT, P_BSY, or P_RJT	-1	NA	NA
Port receives ACK_1	-1	NA	NA
Port receives ACK_N	-m	NA	NA
Port receives a Class 2 Data frame	NA	NA	NA
Port transmits R_RDY	NA	NA	NA
Port transmits F_BSY, F_RJT, P_BSY, P_RJT, or ACK	NA	+1	+1

Note: NA - Not Applicable
m - number of valid frames acknowledged collectively

Table 49. End-to-end flow control management	
Activity	**EE_Credit_CNT (N_Port only)**
N_Port transmits a Class 1 or Class 2 Data frame	+1
N_Port receives F_BSY, F_RJT, P_BSY, or P_RJT	-1
N_Port receives ACK_1	-1
N_Port receives ACK_N	-m
N_Port receives a Data frame (Class 1, Class 2 or Class 3)	NA
N_Port transmits a Class 3 Data frame	NA
N_Port transmits P_BSY or P_RJT	NA
N_Port trasnmits ACK	NA
Note: NA - Not Applicable m - number of valid frames acknowledged collectively	

Table 50. Buffer-to-buffer flow control management	
Activity	**BB_Credit_CNT (N_Port or F_Port)**
N_Port or F_Port transmits any Class 2, Class 3, or Class 1 frame with SOFc1	+1
N_Port or F_Port receives R_RDY	-1
N_Port or F_Port receives any Class 2 or Class 3 frame	NA
N_Port or F_Port transmits R_RDY	NA
Note: NA - Not Applicable	

26 Segmentation and Reassembly

26.1 Introduction

Segmentation and Reassembly provides a function to transfer blocks of data from a data source to a data target. The source data may consist of one or more blocks of data. Each block or sub-block of data is segmented, transferred, and recombined at the destination and may be scattered over an address space.

Mechanisms
FC-2 mechanisms to support this function are:

1. Relative Offset
2. Sequence
3. Exchange

Applicability
All these mechanisms are applicable to all Classes.

Offset
The Relative Offset field in the Frame_Header is used to indicate the relative displacement of first byte of Payload from a base address. The base address is specified by the ULP to FC-2. The base address is meaningful to the communicating ULPs and is not defined in this standard. The ULP's base address forms the Initial Offset to FC-2. The Initial Offset may be applicable to a block or a sub-block or multiple blocks of data as specified by the sending ULP.

26.2 Application data mapping

Mapping block(s) of application data to Sequence(s) at the sending end or Sequence(s) to block(s) of data at the receiving end by the ULPs is outside the scope of this standard but is performed within the N_Ports.

26.2.1 Mechanism usage

FC-2 mechanisms available for mapping are described and listed in Table 51.

1. A given block of data may be mapped by the sending end ULP for transfer by FC-2 as a single Sequence:
   — for N_Port to N_Port communication
   — with single Initial Offset for the block of data and
   — with a transmitting order beginning from the start address to end address of the block of data.
2. A given block of data may be split by the sending end ULP for transfer by FC-2 in multiple Sequences:
   — for N_Port to N_Port or Node to Node communication
   — with single Initial Offset for the block of data with no natural scattering at the receiver or
   — with an Initial Offset for each Sequence with natural scattering at the receiver and
   — with a transmitting order beginning from the start address to end address of each sub-block of data.
3. Multiple blocks of data may be gathered by the sending end ULP for transfer by FC-2 in a single Sequence:
   — for N_Port to N_Port communication
   — with an Initial Offset for each block of data with natural scattering at the receiver or
   — with a single Initial Offset for the Sequence with no natural scattering at the receiver and
   — with a transmitting order beginning from the start address to end address of each block of data.
4. Multiple blocks of data may be gathered by the sending end ULP for transfer by FC-2 in multiple Sequences:
   — for N_Port to N_Port or Node to Node communication
   — with an Initial Offset for each block of data with natural scattering at the receiver or
   — with a single Initial Offset for the Sequence with natural scattering at Sequence boundaries at the receiver or
   — with a single Initial Offset for the whole Exchange with no natural scattering at the receiver and
   — with a transmitting order beginning from the start address to end address of each block of data.

Table 51. FC-2 mechanisms mapping

Item	Number of blocks of data	Number of Sequences	Commu-nicating entities	Initial Offset coverage	Transmit order	Natural gather/scatter at send/receive
1	1	1	N_Port to N_Port	1 Initial Offset per block	Start to end address of each block	NA / NA
2	1	N	N_Port to N_Port OR Node to Node	1 Initial Offset per block OR	Start to end address of each sub-block	split/no scatter
				1 Initial Offset per Sequence OR		split/scatter
				1 Initial Offset per Exchange		split/no scatter
3	N	1	N_Port to N_Port	1 Initial Offset per block OR	Start to end address of each block	scatter
				1 Initial Offset per Sequence		gather/no scatter
4	N	M	N_Port to N_Port OR Node to Node	1 Initial Offset per block OR	Start to end address of each block	gather/scatter
				1 Initial Offset per Sequence OR		gather/scatter
				1 Initial Offset per Exchange		gather/no scatter

Note: M, N - integers NA - Not Applicable

26.3 Segmentation

A block or a sub-block of data to be transferred is segmented by FC-2 into multiple frames not to exceed the maximum size allowed as per service parameters interchanged (see 23.5 and 23.6). FC-2 allows equal or varied size of payload in each frame within this maximum. FC-2 transmits each block or sub-block of data sequentially from the start address to the end address of the block or sub-block. FC-2 supports transmission of a block of data as a single Sequence or multiple Sequences. FC-2 supports transmission of multiple blocks of data as a single Sequence.

During Segmentation, FC-2

1. determines the Size of Payload for each frame,
2. computes the Offset of the Payload from the initial Offset provided by the ULP, and
3. emdeds the Payload in the Data field and the computed Offset value in the Offset field.

26.3.1 Sequence Segmentation rules

Segmentation rules applicable to each Sequence are specified:

1. The sending end ULP is responsible to map multiple blocks or a single block or a sub-block of data to be transferred as a Sequence.
2. The ULP shall specify the Payload for the last frame in its entirety before FC-2 starts transfer of that frame.

3. Unless the ULP specifies the Initial Offset, it is assumed to be FFFFFFFF hex.
4. The Sequence Initiator is responsible for Segmentation management of the Sequence.
5. The Sequence Initiator subdivides the block or sub-block of data into Payloads of equal or varied sizes.
6. The Sequence Initiator shall compute Offset for Payload with reference to Initial Offset and embed it in each frame. The Offset shall indicate the relative address of the first byte of Payload in the Data field.
7. In Class 1, the maximum size of these frames is governed by service parameters interchanged between the communicating N_Ports.
   A. Class 1 connect request frame is however governed by service parameters of N_Ports as well as Fabric. (see 28.4.3).
8. In Class 2 and Class 3, the maximum size of these frames is governed by service parameters interchanged between the communicating N_Ports as well as Fabric.
9. The Sequence Initiator assigns a unique SEQ_ID which is common to all frames within the Sequence.
10. The Sequence Initiator always assigns SEQ_CNT sequentially by the order in which the frames are transmitted, beginning at the start address of the block or sub-block to its end address.
11. The Sequence Initiator uses Offset in each frame and with all Classes.
12. The block or sub-block of data is treated as a stream of bytes in FC-2.
13. The Sequence Initiator shall follow the Initial Offset coverage requested by the ULP.

26.3.2 Exchange Segmentation rules

Segmentation rules applicable to a multiple Sequences are specified:

1. An Exchange may be used to transfer over one or more N_Ports.
   A. Blocks or sub-blocks of data transferred (within an Exchange) over an N_Port shall be non-concurrent.
   B. Blocks or sub-blocks of data transferred (within an Exchange) over multiple N_Ports may be concurrent.
   C. Multiple Exchanges shall be used to transfer multiple concurrent blocks of data over a single N_Port.
2. The Originator or Responder shall perform non-concurrent transfer of multiple blocks of data over an N_Port as multiple non-concurrent Sequences within an Exchange.
3. The Originator or Responder shall perform multiple concurrent Sequences either as multiple Exchanges over an N_Port or one or more Exchanges over Multiple N_Ports.
4. Frames within the Exchange may be transferred through a single N_Port or over a multiple N_Ports.
   A. The Originator or Responder shall set the Exchange Reassembly bit (F_CTL Bit 2) to zero, indicating blocks or sub-blocks of data are transferred as an Exchange on a single N_Port.
   B. The Originator or Responder shall set the Exchange Reassembly bit (F_CTL Bit 2) to one, indicating blocks or sub-blocks of data are transferred as an Exchange on multiple N_Ports.
5. All Sequence Segmentation rules are applicable to individual Sequences within an Exchange.

26.4 Reassembly

The Recipient N_Port packs Payload of each frame at a displacement, indicated by the Offset specified in the Frame_Header.

26.4.1 Sequence Reassembly rules

Reassembly rules applicable to blocks or sub-blocks of data received as a Sequence are specified:

1. The Sequence Recipient is responsible for Reassembly.
2. The Recipient shall reassemble the blocks or sub-blocks of data on the S_ID || OX_ID or RX_ID ||SEQ_ID basis.
3. The first byte of each frame's Payload is received at the address indicated by Offset specified in the Frame_Header.

26.4.2 Exchange Reassembly rules

All Sequence Reassembly rules are applicable to Exchange Reassembly. Additional rules are:

1. If the Exchange Reassembly bit (F_CTL Bit 2) is set to zero, the Originator or Responder shall receive blocks or sub-blocks of data on a single N_Port.
2. If the Exchange Reassembly bit (F_CTL Bit 2) is set to one, the Originator or Responder shall receive blocks or sub-blocks of data on multiple N_Ports.

27 Primitive Sequence Protocols

27.1 Introduction

The interchange of frames is basically asynchronous in nature. The Data protocols defined in this standard allow multiple frames in flight in each direction at the same time. When serious link errors are detected or state changes occur within an N_Port, Primitive Sequence Protocols provide a synchronous mechanism to recover from those errors or state changes.

27.2 Primitive Sequences

Table 52. provides a summary of the Primitive Sequences defined, their meaning to the transmitter, and the corresponding response when received. A more complete description of Primitive Sequences is found in 16.4.

Table 52. Primitive Sequence Summary

Name	Meaning to Transmitter	Resp
NOS	Not-Operational - Link Failure - Internal failure in Port - Not operational state	OLS
OLS	Offline Sequence - Transmitter may power-down, perform diagnostics, or perform initialization. - Receiver shall ignore Link errors or Link Failure.	LR
LR	Link Reset - Remove Class 1 Connection, - Reset F_Port, or - OLS recognized	LRR
LRR	Link Reset Response - Link Reset recognized	Idles
Idles	Operational Link	Idles

27.3 Link Initialization

Link initialization is required after a Port has been powered-on or has been internally reset. The Ports involved may be an N_Port and an F_Port, or two N_Ports.

The following series of events defines the Link Initialization Protocol.

1. Transmit OLS.
2. When LR is recognized, transmit LRR.
3. When Idles are recognized, transmit Idles.

27.4 Online to Offline

An N_Port performs the Online to Offline protocol to enter the Offline State. The Offline State is entered prior to power-down or performing diagnostics. To exit the Offline State, an N_Port performs the Link Initialization Protocol.

The following series of events defines the Online to Offline Protocol.

1. Transmit OLS
2. When LR is recognized, or the timeout period has expired, the Port may:
   — perform diagnostics,
   — reinitialize, or
   — power-down

27.5 Link Failure

The Link Failure protocol is performed after a Port has either detected a loss of synchronization for a period of time greater than the timeout period, or has detected loss of signal, while not in the Offline State.

The Link Failure protocol is also performed after a Link Recovery timeout error is detected (see 27.6).

The following series of events defines the Link Failure Protocol.

1. Transmit NOS.
2. When OLS is recognized, transmit LR.
3. When LRR is recognized, transmit Idles.

27.6 Link Recovery

The Link Recovery protocol is performed following a Link timeout. If a Port performs step 1, 2, or 3 without receiving the appropriate response within the timeout period, a Link Failure is detected and the Link Failure protocol is performed.

The following series of events defines the Link Recovery Protocol.

1. Transmit LR.
2. If LR is recognized, transmit LRR.
3. If LRR is recognized, transmit Idles.
4. If Idles are recognized, transmit Idles.

At the conclusion of Link Recovery both Ports shall be transmitting and receiving Idles.

28 Connection Management

The procedures for establishing and removing Class 1 Dedicated Connections are specified in this clause. See Annex P for application examples for removing a connection as well as an N_Port State Diagram of the Connect and Disconnect Process.

28.1 Introduction

Class 1 Service is based on establishing a Dedicated Connection between a source N_Port and a destination N_Port. The Dedicated Connection guarantees that the full bandwidth of the Link is available to each N_Port.

Establishing a Connection:

In order to establish a Class 1 Dedicated Connection, the source N_Port transmits a Data frame to a destination N_Port with an **SOFc1** delimiter. The Data field size of the connect-request frame is limited by the maximum Receive Data_Field size specified by the Class 1 Service Parameters of the Fabric or by the maximum Receive Data_Field size specified by the destination N_Port, whichever is smaller.

The N_Port receives an R_RDY Primitive Signal to indicate that the connect-request frame was received successfully and a buffer in the F_Port or N_Port is available. When the N_Port transmitting the connect-request receives an ACK response frame (ACK_1 or ACK_N) with an **SOFn1** with the appropriate S_ID and D_ID fields of the connect-request frame, a Dedicated Connection is established.

When a Dedicated Connection is established:

1. the N_Port initiating the connection is known as the Connection Initiator and the N_Port responding to the connect-request is known as the Connection Recipient.
2. the Connection Initiator continues the initial Sequence, and
3. each N_Port may initiate new Sequences with an **SOFi1** delimiter.

Removing a Connection:

Removing a Dedicated Connection is accomplished by either N_Port transmitting a frame terminated by an **EOFdt**. Removing a Dedicated Connection shall be negotiated between the two N_Ports involved. Negotiation is required in order to avoid breaking the Dedicated Connection while frames are still flowing between the N_Ports.

The Fabric terminates a Dedicated Connection after an **EOFdt** or **EOFdti** passes through the F_Port in either direction. Any frames flowing in either direction at the time the Connection is removed may be corrupted.

End_Connection (E_C) is the control Bit in F_CTL which is used to perform the negotiation.

28.2 Applicability

Connection management applies to Class 1 Service. An N_Port supporting Class 1 Service may also support Class 2 and Class 3. Depending on the options supported by the Fabric, multiple class support by an N_Port may be complex. Because Class 1 involves Dedicated Connections, managing Class 1 usually overrides Class 2 or 3 management. The Standard specifies the allowable responses on a Class by Class basis.

28.3 Topology Models

An N_Port may be attached directly to another N_Port through a point to point connection or to a Fabric. The topology may be determined using the **explicit** Login procedure.

28.3.1 Fabric model

The N_Port model is based on an F_Port which acts as the control point for establishing and removing Class 1 connections on behalf of the attached N_Port. The N_Port relies on specific behavioral characteristics in order to base its operation.

The following terminology is used in the discussion of Connection management. N_Port (A) refers to the N_Port originating the connect-request and N_Port (B) refers to the destination N_Port of the connect-request. The side of the F_Port directly attached to the N_Port side is termed its "Link" side, whereas the side of the F_Port attached internally to the Fabric is termed its "internal" side.

The following F_Port characteristics are required behavior:

1. When an F_Port is in an inactive state or "not connnected", it may receive a connect-request and begin processing that request. The process of acting on that request is termed accepting the connect-request.
2. After an F_Port has accepted a connect-request from the Link, it reserves Fabric resources as it attempts to establish the requested Dedicated Connection.
   - the F_Port is Busy to other connect-requests from its internal side as destination N_Port.
   - the F_Port returns an F_BSY with **EOFdt** to the Link if a busy condition is encountered. (If a Fabric supports queued connect-requests, the period of time before issuing an F_BSY may be extended.)
   - the F_Port returns an F_RJT with **EOFdt** to the Link if a reject condition is satisfied.
3. After an F_Port has accepted a connect-request from the internal side:
   - it passes the connect-request to the Link as the destination N_Port.
   - it monitors its Link side for a proper confirmation response frame expected in response to the delivered connect-request.
   - it discards connect-request frames (**SOFc1**) received from its Link side. (If the Fabric supports queued connect-requests, the connect-request received from its Link side shall be queued for a establishing a Dedicated Connection at a later time.)
4. If an F_Port encounters a collision case wherein connect-requests from both the internal side and the Link side arrive simultaneously, the F_Port accepts the connect-request from the internal side and proceeds as in step 3 above.

28.3.2 Point to Point Model

A point to point topology is indicated during the Login procedure. Two N_Ports arranged in a point to point connection may choose to:

1. establish one Dedicated Connection for the duration of an operating period, or
2. establish and remove Dedicated Connections dynamically, as the need to communicate arises.

28.4 Establishing a Connection

A Dedicated Connection is established with an N_Port as the source of a connect-request (Connection Initiator) or as the destination N_Port (Connection Recipient) of a connect-request from another Class 1 N_Port.

28.4.1 Connection Initiator

When FC-2 receives a request from an upper level to initiate a Class 1 Sequence when a Dedicated Connection does not exist, the N_Port must also establish a Class 1 Connection with the destination N_Port as part of the Sequence initiation.

The N_Port (A) initiates the connect-request using a Data (Device_Data or Link_Data) frame with an **SOFc1** delimiter. The Data Field size is limited to the smaller of:

- the maximum Receive_Data_Field size specified by the Fabric, if present, or
- the maximum Receive_Data_Field size specified by the destination N_Port.

After the N_Port transmits the connect-request frame, the N_Port shall not transmit another frame for this Sequence until a response frame has been received. The N_Port receives an R_RDY Primitive in response to the connect-request to indicate successful delivery to the F_Port or N_Port and that a buffer is available for a connectionless frame. If an N_Port is not operating in Intermix mode, the N_Port shall not transmit Class 2 or 3 frames until the pending Dedicated Connection is removed. A Dedicated Connection is not established until a proper ACK frame is received from the destination N_Port. A proper ACK frame is defined in 28.4.3.1 item number 4.

Table 53 defines Event 1 as the connect-request and events 2 through 8 define the possible responses.

Table 53. Responses to connect-request (SOFc1)						
Event	SOF	D_ID	S_ID	CMD	EOF	N_Port Action
1.	**SOFc1**	B	A	Data frame	**EOFn**	-Transmit connect-request -Wait for confirmation frame
2.	**SOFn1**	A	FFFFFE	F_BSY	**EOFdt**	Connection failed, Busy in Fabric
3.	**SOFn1**	A	B	P_BSY	**EOFdt**	Connection failed, Busy in N_Port
4.	**SOFn1**	A	FFFFFE	F_RJT	**EOFdt**	Connection failed, Fabric Reject
5.	**SOFn1**	A	B	P_RJT	**EOFdt**	Connection failed, Port Reject
6.	**SOFn1**	A	B	ACK_1 or ACK_N	**EOFn**	-Dedicated Connection established -Continue transmitting Sequence
7.	**SOFc1**	A	X	Data frame	**EOFn**	A. PTP:if A > X in value, -discard X's frame and wait for response. or B. Fabric and PTP: if A < X : -Requeue request assoc with event 1 (**SOFi1**) -Respond with **SOFn1** on ACK_1 or ACK_N. -Dedicated Connection established with X.
8.						-Timeout, no response frame. -Perform Link Recovery Protocol. (see 27.6)

Event 1 A connect-request is transmitted by N_Port (A) with an **SOFc1** delimiter with a destination of N_Port (B).

Event 2 An F_BSY indicates that the Fabric is unable to access the destination N_Port due to a busy condition internal to the Fabric. Try again later.

Event 3 A P_BSY indicates that the destination N_Port link facility is temporarily occupied with other activity and unable to accept the connect-request. Try again later.

Event 4 An F_RJT indicates that the Fabric is unable to establish the Dedicated Connection. The reason code specifies the cause.

Event 5 A P_RJT indicates that the destination N_Port is unable to establish the Dedicated Connection. The reason code specifies the cause.

Event 6 A Dedicated Connection has been established. N_Port (A) is Connected to N_Port (B).

1. N_Port (A) is the Connection Initiator and N_Port (B) is the Connection Recipient.
2. N_Port (A) may continue transmitting the Sequence initiated.
3. N_Port (A) may initiate other Sequences with the same destination N_Port (B) up to the maximum number of Sequences defined by the Service Parameters obtained from (B) during Login.
4. The connected N_Port (B) may initiate Sequences using **SOFi1** to start each Sequence. The number of active Sequences is limited by the Service Parameters provided by N_Port (A) during Login.

Event 7 In the case of a point to point (PTP) topology, if (A) is greater than (>) (X) in absolute value, then N_Port (A) discards the connect-request and waits for a response (N_Port (X) requeues request with **SOFi1**).

In the case of a PTP topology where (A) < (X), or a Fabric is present, N_Port (A) terminates its own previous connect-request with the intent of retransmission at a later time (i.e. requeues Event 1 with **SOFi1** or **SOFc1** as appropriate). N_Port (A) responds as the destination of the connect-request with an appropriate ACK frame and becomes the Connection Recipient. N_Port (X) is the Connection Initiator.

Event 8 If a frame response is not received within the timeout period, a Link timeout is detected and the Link Recovery Protocol is performed (see 27.6).

See 28.4.3, "Connect-Request Rules" for the rules which a source N_Port of a connect-request shall follow.

28.4.2 Destination of Connect-Request

When N_Port (B) is not connected, but is available, and it receives a connect-request as the destination N_Port, N_Port (B) transmits the appropriate ACK frame (ACK_1 or ACK_N) to N_Port (A) which is requesting the connection. After the response frame has been transmitted with **SOFn1**, a Dedicated Connection is established with N_Port (A) as the Connection Initiator and N_Port (B) as the Connection Recipient. After a Connection has been established, N_Port (B) may initiate Sequences with the N_Port (A) using an **SOFi1** delimiter.

If N_Port (B) is not connected, but is busy, and it receives a connect-request as the destination N_Port from N_Port (A), it responds with P_BSY with an **EOFdt** delimiter.

See 28.4.3, "Connect-Request Rules" for the rules which a destination N_Port of a connect-request shall follow.

28.4.3 Connect-Request Rules

The following sections specify the rules governing the behavior of the source and destination of the connect-request.

28.4.3.1 Source of Connect-Request

The following rules specify the connect-request procedure as the source (A) of the connect-request:

1. An N_Port (A) initiates a connect-request using a Data (Device_Data or Link_Data) frame with an **SOFc1** delimiter directed to destination N_Port (B). The connect-request frame is formed as follows:
   — an **SOFc1** delimiter
   — a Data (Device_data or Link_Data) frame
   — an S_ID of (A) and a D_ID of (B)
2. The Data Field of the connect-request is limited to the smaller of:
   — the maximum Receive_Data_Field size specified by the Fabric, if present, or
   — the maximum Receive_Data_Field size specified by the destination N_Port.
3. After N_Port (A) transmits the connect-request frame, N_Port (A) shall wait for a response frame before transmitting another frame for this Sequence.
4. A Dedicated Connection is established when the connect-request frame has been responded to by a proper response frame. A proper response frame consists of:
   — a ACK_1 or ACK_N frame with
   — an **SOFn1** delimiter, and
   — an S_ID of (B), and a D_ID of (A)
5. An alternate response frame is also possible from the destination N_Port:
   — a P_BSY or P_RJT frame with
   — an **SOFn1** delimiter,
   — an S_ID of (B), and a D_ID of (A), and
   — an **EOFdt** ending delimiter.
6. An alternate response frame is also possible from the Fabric:
   — an F_BSY or F_RJT frame with
   — an **SOFn1** delimiter,
   — an S_ID of (FFFFFE), and a D_ID of (A), and
   — an **EOFdt** ending delimiter.
7. After a Dedicated Connection is established, N_Port (A) is the Connection Initiator and N_Port (B) is the Connection Recipient.
8. The Connection Initiator, N_Port (A), may continue transmitting its initial Sequence and initiate other Sequences with **SOFi1** up to N_Port (B)'s ability to support Concurrent Sequences.

28.4.3.2 Destination of Connect-Request

The following rules specify the connect-request procedure as the destination (B) of the connect-request:

1. If a Data frame started by **SOFc1** is received when N_Port (B) is not connected and N_Port (B) is busy, N_Port (B) responds with P_BSY with an **EOFdt** delimiter as specified in 28.4.3.1 item number 5.
2. If a Data frame started by **SOFc1** is received when N_Port (B) is not connected and not busy, N_Port (B) responds with the proper response frame. A proper response frame is defined in 28.4.3.1 item number 4.

   A Dedicated Connection is established with N_Port (A). N_Port (B) is the Connection Recipient and N_Port (A) as the Connection Initiator.
3. With a Fabric present, if N_Port (B) receives a connect-request frame from N_Port (X) after a connect-request has been transmitted, N_Port (B) requeues its own request for transmission at a later time and responds with a proper response frame to N_Port (X).

   A Dedicated Connection is established with N_Port (X) with N_Port (B) as Connection Recipient.
4. Without a Fabric present, N_Port (B) responds as follows:
   A. if A > B, in value,
      — N_Port (B) discards connect-request from (A), and
      — waits for a proper response frame.
   B. if A < B, in value,
      — N_Port (B) requeues its own request for transmission at a later time,
      — responds to (A) with a proper response frame,
      — a Dedicated Connection is established with N_Port (A) with N_Port (B) as Connection Recipient, and
      — N_Port (B) may initiate its connect-request Sequence using **SOFi1**.
5. After a Dedicated Connection is established, N_Port (B) may begin initiating Sequences with **SOFi1** up to the destination N_Port's ability to support Concurrent Sequences.

28.5 Connected

When an N_Port is in a Dedicated Connection, it may receive Sequences as the Sequence Recipient as well as initiate Sequences as the Sequence Initiator.

28.6 Removing a Connection

28.6.1 Connection_Resource control bit

This document does not specify the method to be employed in determining when to end a connection. However, a Connection_Resource bit in F_CTL (see 18.5) is defined to provide information between N_Ports.

28.6.2 End_Connection control bit

An E_C control Bit in F_CTL is used during the remove connection procedure. By monitoring both transmission of the E_C control Bit as well as reception of the E_C control Bit, each N_Port is able to determine the appropriate circumstances to transmit an **EOFdt** in order to remove the connection.

E_C is transmitted as zero in a frame to indicate:

1. This N_Port wishes to maintain the existing Dedicated Connection.
2. This N_Port may transmit another Sequence within this Connection.
3. This N_Port may wait for a reply Sequence within this Connection.

E_C is transmitted as one on a frame to indicate:

1. This N_Port is ready to remove the existing Dedicated Connection.
2. This N_Port has completed all active Sequences and agrees not to initiate another Sequence.
3. This N_Port is requesting the destination N_Port to complete active Sequences in progress and not initiate any new Sequences.

E_C may be transmitted set to one in two cases:

1. on an ACK response frame to indicate that this N_Port requests the receiving N_Port to complete active Sequences and not initiate any new Sequences, or

S

T

V

2. on the last Data frame of the last active Sequence for this Connection to indicate that this N_Port requests the receiving N_Port to transmit **EOFdt** on the ACK response frame if the receiving N_Port has completed all active Sequences.

28.6.3 EOFdt transmission

EOFdt is transmitted by either the Connection Initiator or the Connection Recipient on an ACK frame corresponding to the last Data frame of a Sequence with the E_C control Bit set to one if the N_Port receiving the Data frame has completed its last active Sequence of the existing Connection.

If both N_Ports have transmitted their last Data frame of a Sequence with E_C set to one but have not received the ACK frame to complete the Sequence, the Connection Initiator delays transmitting ACK until it has received the ACK for its last Sequence. After the Connection Initiator has completed its last Sequence, it transmits the final ACK to the Connection Recipient with the **EOFdt** delimiter to remove the Connection.

28.7 Remove Connection Rules

1. An active Sequence is complete when the corresponding ACK response has been transmitted by the Sequence Recipient from the Recipient's perspective and has been received by the Sequence Initiator from the Initiator's perspective in response to the last Data frame of the Sequence.
2. An N_Port transmits the E_C F_CTL Bit set to one to indicate:
   - it is ready to end the Connection,
   - it shall not initiate any new Sequences, and
   - it requests the other N_Port to complete its active Sequences and not initiate any new Sequences.
3. If an N_Port has transmitted the E_C F_CTL Bit set to one and it receives a Data frame initiating a new Sequence, it responds as though the Sequence had been initiated before the E_C Bit had been transmitted as one.
4. If either the Connection Initiator or Connection Recipient have completed its last active Sequence of the existing Connection and it receives a Data frame with E_C set to one, it transmits the corresponding ACK with an **EOFdt** delimiter.
5. If an N_Port encounters a collision case wherein a Data frame has been transmitted with E_C set to one and a Data frame is received with E_C set to one before receiving its ACK,
   - the Connection Recipient responds with an ACK with an **EOFn1** delimiter, whereas
   - the Connection Initiator withholds transmitting ACK until after its Sequence is complete.
   - the Connection Initiator transmits the ACK with the **EOFdt** delimiter.

28.8 Connection Recovery

In case of link errors, the state of the existing Dedicated Connection may not be known with certainty. The Dedicated Connection is removed and the two Ports are brought to a known state using the Link Recovery Protocol (see 27.6). Errors within a specific Sequence are processed according to the rules for handling Sequence errors.

28.8.1 Link timeout

When the last active Sequence during a Class 1 Dedicated Connection has detected a Sequence timeout, a Link Timeout is detected. The Link Recovery Protocol is performed as described in 27.6.

28.8.2 Corrupted connect-request

If an N_Port is not engaged in a Class 1 Dedicated Connection, and it receives a frame started by **SOFc1** and the frame is detected as invalid, the N_Port discards the frame and performs the Link Recovery Protocol.

29 Error detection/recovery

29.1 Introduction

Link integrity and Sequence integrity are the two fundamental levels of error detection in FCS. Link integrity focuses on the inherent quality of the received transmission signal. When the integrity of the link is in question, a hierarchy of Primitive Sequences are used to reestablish link integrity. When Primitive Sequence protocols are performed, additional data recovery on a Sequence basis may be required.

A Sequence within an Exchange provides the basis for ensuring the integrity of the block of data transmitted and received. Each frame within a Sequence is tracked on the basis of Exchange_ID, Sequence_ID, and a sequence count within the Sequence. Each frame is verified for validity during reception. Sequence retransmission is used to recover from any frame errors within the Sequence.

Credit loss is an indirect result of frame loss or errors. Credit loss is discussed in regard to methods available to reclaim apparent lost Credit resulting from other errors. See clause 25 for a more complete discussion on flow control, buffer-to-buffer Credit, and end-to-end Credit. Table 54 summarizes the discussion on link integrity.

Table 54. Link Error Detection/Recovery

LINK ERROR DETECTION	LINK ERROR RECOVERY Recovery Action	Effects on Sequences
Loss of Signal	Link Failure Protocol	possible - Sequence timeout (Class 2), certain - Terminate Sequence (Class1)
Loss of Sync > timeout	Link Failure Protocol	possible - Sequence timeout (Class 2), certain - Terminate Sequence (Class1)
Link Recovery timeout	Link Failure Protocol	possible - Sequence timeout (Class 2), certain - Terminate Sequence (Class1)
Link timeout	Link Recovery Protocol	possible - Sequence timeout (Class 2), certain - Terminate Sequence (Class1)
Code violation during Idles	Update LESB	No effect on frames

29.2 Link error detection

Link errors are detected at three levels.

29.2.1 Link failure

The first level of link error detection is at the receiver. Link failure is detected under the following conditions:

- Loss of Signal,
- Loss of Synchronization > timeout period (see 11.1 and -- Heading 'LOSYN' unknown --),
- Link Recovery timeout (see 27.6.)
- Reception of the Not Operational Primitive Seqüence (see 16.4.2).

Recovery from Link Failure is accomplished by performing the Link Failure Protocol (see 27.5).

29.2.2 Link timeout

The second level of errors is logically detected at the frame level with Link timeout detection.

In Class 1, a Link Timeout error is detected during a Dedicated Connection if Sequence timeouts have been detected for all active Sequences.

In Class 1 (**SOFc1**), 2, or 3, a Link Timeout error is detected if one or more R_RDY Primitive Signals is not received within the timeout period after the buffer-to-buffer Credit_CNT has reached zero.

Recovery from link timeout is accomplished by performing the Link Recovery Protocol (see 27.6).

29.2.3 Code violations

Code violations are link errors which result from an invalid transmission code point or disparity error. If a code violation occurs during Idles, it is recorded in the Link Error Status Block. If a code violation occurs during frame reception (see 29.10), the frame is discarded or processed based on the error policy being used.

29.3 Link error recovery

The Link recovery hierarchy is shown in figure 64.

The recovery protocols are nested and organized from the most serious to least serious link action.

- — link failure
- — link initialization
- — link recovery (reset)

Primitive Sequence Protocols provide two basic functions. The first function is to notify the other end of the link that a specific type of link error has occurred. The second function is to reset the link to a known state at both ends.

29.3.1 Link initialization and shutdown

Link initialization is accomplished by performing the Link Initialization Protocol. When this protocol is complete, the Port is synchronized, transmitting and receiving Idles. See 27.3.

Shutdown prior to power-off is accomplished by the Online to Offline Protocol. This protocol provides an attached Port with a graceful indication prior to loss of signal. This avoids logging an error event for a normal system function. See 27.4.

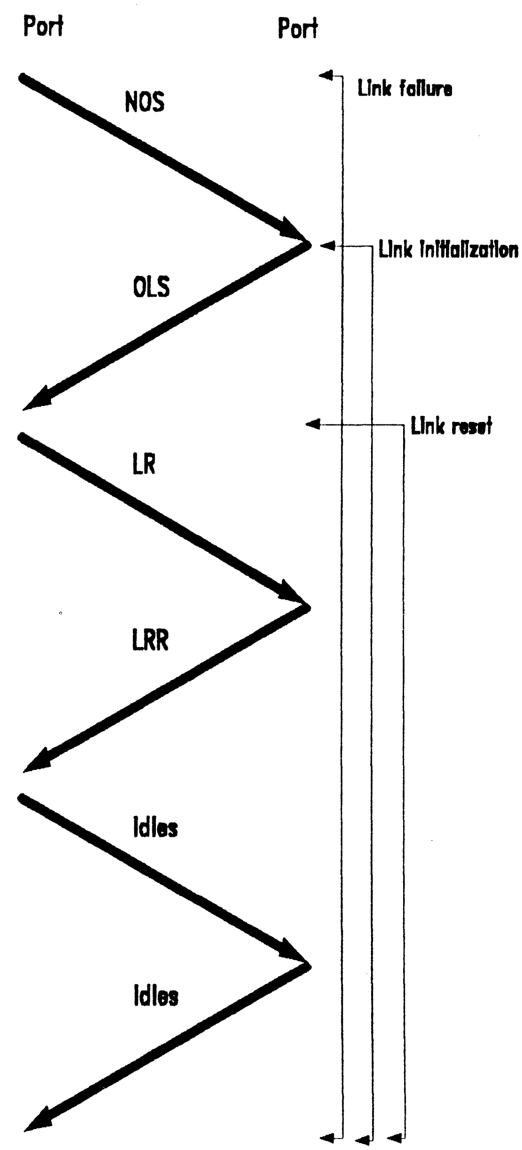


Figure 64. Link recovery hierarchy

29.4 Link recovery - secondary effects

If the recovery action from an error involving link integrity requires transmission of a Primitive Sequence, Sequences within an Exchange may be adversely affected based on Class of Service.

Class 1
During a Class 1 Dedicated Connection, transmission or reception of Primitive Sequences removes the connection. While an N_Port is transmitting a Primitive Sequence, it shall discard any Class 1 frames received. At the conclusion of the Primitive Sequence protocol, the end-to-end Credit and buffer-to-buffer Credit

are reset to their original Login values in both N_Ports and the F_Port's involved.

All frame Sequences that were active are terminated abnormally. Selective recovery on a Sequence by Sequence basis is required of the Sequence Initiator. When Sequences are abnormally terminated, the current status of each Sequence is posted in the corresponding Exchange Status Block and the Sequence Status Blocks are reset and released. Link resources associated with the active Sequences are released. If the Sequence Initiative is in doubt, the previous Initiator N_Port shall interrogate the other N_Port's Exchange Status Block to determine which N_Port has the Sequence Initiative (see 29.7.1).

Class 2
When Primitive Sequences are transmitted or received, the F_Port shall discard any Class 2 frames held in its buffers. After the Primitive Sequence protocol is completed, the buffer-to-buffer Credit is reset in an N_Port and an F_Port. Both the N_Port and F_Port may begin transmitting frames.

Active Sequences within an Exchange are not necessarily affected. Therefore, normal processing continues and selective Sequence recovery is performed as required.

Class 3
When Primitive Sequences are transmitted or received, the F_Port shall discard any Class 3 frames held in its buffers. After the Primitive Sequence protocol is completed, the buffer-to-buffer Credit is reset in an N_Port and an F_Port. Both the N_Port and F_Port may begin transmitting frames.

29.5 Sequence integrity

Applicability

Since Class 3 does not use Link_Control frames, Sequence integrity is verified at the Sequence Recipient on a Sequence by Sequence basis. In Class 3, only the Recipient is aware of an out of order condition and communication of that information to the Initiator is the responsibility of FC-4.

The remaining discussion concentrates on Class 1 and 2. Items applicable to Class 3 shall be specified explicitly.

29.5.1 Exchange processing

When an Exchange is originated, the Originator shall specify the required error policy of either discard or process using the Abort Sequence Condition bits in F_CTL. Process policy allows the receiving N_Port to process frame content regardless of detected error conditions. Process policy may override other actions specified.

The fundamental unit of data transfer within an Exchange is the Sequence. The integrity of a block of data is verified on a Sequence basis. If a Sequence is invalid, the Sequence is aborted and the entire Sequence is retransmitted. If an Exchange is being managed with a process error policy, then accounting for all data is not required. However, maintenance of a properly functioning Sequence is still required.

The Sequence Recipient shall process Sequences within an Exchange in the order received. If the order of Sequence processing is critical to the Initiator, the Initiator is responsible for controlling or verifying the delivery order of multiple, consecutive Sequences. The means for such control varies by Class of Service.

29.5.1.1 X_ID processing

During certain periods in the life of an Exchange, one or both X_ID fields are unassigned at the beginning or during the Exchange. Both the Sequence Initiator and the Sequence Recipient shall support the Abort Sequence and the Abort Exchange protocols at any point during the Exchange. The use of X_ID transition interlock and X_ID invalidation are options to help simplify this requirement.

29.5.2 Streaming Sequences

When a Sequence Initiator streams consecutive Sequences without waiting for the terminating ACK to the first Sequence, the Sequence Recipient may have encountered an error condition in the first Sequence when the second Sequence is initiated.

The number of open Sequences for a single Exchange between two N_Ports is limited by the

Open Sequences per Exchange value provided by the Recipient during Login. Therefore, the Sequence Initator is restricted from initiating more Sequences than allowed by the Recipient or an Exchange error is detected by the Recipient.

29.5.3 Sequence tracking within Exchange

Three facilities associated with the Exchange Status Block (see 29.11.3) are used to track Sequence status throughout an Exchange.

29.5.3.1 P_Mask

The P_Mask is a 256 bit vector (8 words) which corresponds to the SEQ-ID values of 0 to 255 where word 8, bit 31 corresponds to SEQ_ID = 0. A P_Mask is defined for each Exchange Status Block.

29.5.3.2 C_Mask

The C_Mask is a 256 bit vector (8 words) which corresponds to the SEQ-ID values of 0 to 255 where word 12, bit 31 corresponds to SEQ_ID = 0. A C_Mask is defined for each Exchange Status Block.

29.5.3.3 Mask processing

The following rules apply:

- When an Exchange is originated, both the P_Mask and C_Mask are reset to binary zeros.
- When a Sequence is initiated, the P_Mask bit corresponding to the SEQ_ID is set = 1.
- When a Sequence terminates normally, the C_Mask bit corresponding to the SEQ_ID is set = 1 and the corresponding P_Mask bit is set = 0.
- When a Sequence is terminated abnormally, the P_Mask bit corresponding to the SEQ_ID remains set = 1 and the corresponding C_Mask is set = 1.

The resulting status is derived from examination of the P_Mask and C_Mask in the Exchange Status Block for a specific SEQ_ID.

P_Mask=0 and C_Mask=0	Sequence never opened
P_Mask=0 and C_Mask=1	Sequence ended normally
P_Mask=1 and C_Mask=0	Sequence active or open
P_Mask=1 and C_Mask=1	Sequence terminated abnormally

A SEQ_ID is may be reused within the same Exchange if its current ending status indicates that it has been completed normally or abnormally. If the SEQ_ID is still open, a frame initiating a Sequence with the same SEQ_ID shall be rejected with a reason code of Exchange error.

29.5.3.4 Sequence Status Blocks

In addition to the P_Mask and C_Mask, both the Originator and Responder shall save a Sequence Status Block for each Sequence up to the limit of X+2 (where X=value of Open Sequences per Exchange). When the limit of X+2 is exceeded, the oldest normally completed Sequence Status Block is removed.

29.6 Sequence error detection

Table 55 summarizes the discussion on Sequence error detection.

Table 55. Sequence Error Detection/Recovery

SEQUENCE ERROR DETECTION	SEQUENCE ERROR RECOVERY Action	Effects/Recovery
Sequence timeout	Abort Sequence	Retransmit Sequence
Corrupted frame	Discard	Out of order detection, Abort Sequence,or Sequence timeout
Out of order detection	Abort Sequence	Retransmit or Interrogate Status
Frame rejected	Action 1 or 3 Action 2 or 4	Abort Sequence, retransmit Abort Exchange (FC-4)
Frame busied	Retransmit frame	
Code violation in frame	Discard frame	Out of order detection, Abort Sequence, or Sequence timeout
buffer-to-buffer Credit Loss	Link timeout	Link Recovery Protocol, Relogin
end-to-end Credit Loss	Sequence timeout, Link timeout (Class 1)	Abort Sequence, Reclaim end-to-end Credit

29.6.1 Sequence timeout

The basic mechanism for detecting errors within a Sequence is the Sequence timeout. Both the Sequence Initiator and the Sequence Recipient use a timer facility with a timeout period between expected events. The expected event for the Initiator to Data frame transmission is an ACK response. The expected event for the Recipient is another Data frame for the same Sequence. When a Sequence Recipient receives the last Data frame (End_Sequence) for the Sequence, it shall verify that all frames have been received before transmitting the final ACK for the Sequence.

If the timeout period expires before the Sequence is complete, a Sequence timeout is detected. The value used for the timeout period is local to an N_Port. Timeouts are detectable by both the Sequence Initiator and the Sequence Recipient. A Sequence timeout results in abnormal termination of a Sequence (see 29.7.1). Other mechanisms which detect frame errors within a Sequence are performance enhancements in order to detect an error sooner than the timeout period.

29.6.2 Out of order

In Class 1, sequential frame delivery is guaranteed. Therefore, an out of order frame is detected by a missing sequence count in consecutive frames received for the same Sequence.

In Class 2, sequential frame delivery is not guaranteed. Therefore, an N_Port shall use another method to detect missing frame sequence counts for a Sequence. In order to simplify an N_Port's implementation, an N_Port may choose to define an out of order window. An N_Port may choose a window value of W. If the highest SEQ_CNT received minus W is greater than the SEQ_CNT of a missing frame, then an out of order condition is detected. The value of W is only known internally to the N_Port.

In all Classes, when an out of order condition is detected, the N_Port shall save the sequence count of the missing frame in the Sequence Status Block. Only the first error is posted. Under the discard policy, the N_Port discards each successive frame delivered. However, ACK processing continues with the Abort Sequence Condition bits in F_CTL set appropriately, beginning with the frame on which the error was detected. Once the Abort Sequence bits are being set on the ACK frames, each Data frame shall be responded to with a separate

ACK (either ACK_1 or ACK_N with a one count). The Sequence Recipient does not transmit an **EOFt** in response to the last Data frame of the Sequence and does not recognize the transfer of Sequence Initiative, if present. When the Sequence Initiator detects the Abort Sequence indication, it transmits an Abort Sequence Link_Data request to terminate the Sequence (see 29.7.1).

Under the process policy, the N_Port processes remaining frames in the Sequence as though the error had never occurred. However, the ending status posted to the Exchange Status Block shall indicate the first error detected and completion ending status. The Abort Sequence bits in F_CTL are not set for missing frames. However, an N_Port may request an Abort Sequence for other reasons internal to the N_Port or Sequence processing.

29.6.3 Frame reject

All frame rejects result in the Sequence being terminated. For Action codes 1 or 2, the Sequence is terminated immediately. For action codes 3 or 4, the Sequence is terminated using the Abort Sequence protocol. Frame rejects with action codes of 1 or 3 are retryable if the condition indicated by the reason code is corrected. Frame rejects for action codes 2 and 4 indicate a non-retryable condition which probably results in Abort Exchange.

29.6.4 Frame busy

In Class 1, an F_BSY or P_BSY indicates a temporary busy condition which is retryable. The Sequence being initiated is immediately terminated. In Class 2, F_BSY or P_BSY is possible in response to any frame. The Sequence Initiator is responsible for retransmitting the frame and the Recipient is responsible for processing the retransmitted frame out of order. For Class 2, a busy response is not considered an error condition but is included for completeness as an abnormal condition.

29.7 Sequence recovery

29.7.1 Abnormal Sequence termination

The Abort Sequence protocol is one method by which a Sequence may be abnormally terminated. In the Abort Sequence protocol, the Initiator transmits an Abort Sequence Link_Data request to the Recipient to terminate the Sequence. If the Initiator detects an error in its Sequence processing, it performs the Abort Sequence protocol. However, if the Initiator has transmitted the last Data frame of the Sequence, the Sequence may have ended normally. The Initiator may choose to interrogate Sequence status before aborting the Sequence and retransmitting. Some applications may require interrogation and not allow duplicate transmission.

If a Sequence Recipient detects an out of order or other unusual condition within a Sequence, it requests the Initiator to perform the Abort Sequence protocol using the Abort Sequence Condition bits in F_CTL in an ACK frame within the Sequence which is in error. The Recipient may request that a Sequence be aborted and retransmitted, or that the Initiator stop the Sequence and read status (see 29.6.2).

However, if no Data frames are being received for the Sequence in error, the Sequence Recipient shall timeout the Sequence, abnormally terminate the Sequence, update the Sequence status in the Exchange Status Block, and release link facilities associated with the Sequence including the Sequence Status Block (Sequence timeout). If a frame for the abnormally terminated Sequence arrives after termination, the Recipient indicates that the Sequence has timed out by setting the appropriate Abort Sequence Condition bits in F_CTL in the ACK to the frame received for the terminated Sequence and the frame is discarded.

In the Abort Sequence protocol, after the Sequence Initiator has transmitted the Abort Sequence Link Sequence, it retires the SEQ_ID of the pending aborted Sequence. Sequence Initiative for the Exchange is passed to the Recipient. In Class 1, the Recipient (new Initiator) shall terminate the Sequence and post the appropriate Exchange Status Block. The Recipient transmits an Accept Link Reply Sequence returning Sequence Initiative for retransmission of the terminated Sequence. The Payload of the Accept contains the Exchange_IDs and Sequence_ID of the Aborted Sequence.

In Class 2, the Recipient shall account for all missing frames in the Sequence or wait a Sequence timeout period before responding with the Accept Link Reply.

29.7.1.1 End_Sequence

If the last Data frame of a Sequence has been transmitted and the Sequence Initiator detects a Sequence timeout, the Initiator shall initiate an Exchange with a Read Exchange Status Link_Data request to determine Sequence and Exchange status. If the Initiator is in the process of timing out a Sequence for a missing **EOFt** with Sequence Initiative transferred and it receives a new Sequence initiated by the Recipient (new Initiator), it may assume that the previous Sequence ended successfully.

From a Recipient view, if the last Data frame is lost, the Recipient terminates the Sequence when a Sequence timeout is detected.

29.7.2 Credit Loss

The Standard does not specify the method to be employed for Credit allocation to a destination N_Port. If destination N_Port end-to-end Credit is allocated on a Sequence basis, then that portion of end-to-end Credit is reclaimed when the Sequence is aborted. When a Sequence is aborted, any outstanding ACK frames associated with the Sequence being aborted may be reclaimed. This applies to both Class 1 and Class 2.

In addition, with the discard policy in effect in Class 1, the Sequence Initiator may reclaim end-to-end Credit for missing ACK frames if abort is not requested by the Recipient for an ACK following the missing ACK (i.e. this indicates a lost ACK frame and that the Data frame arrived successfully). A similar technique may be applied to Class 2 using an out of order Sequence window. This may be accomplished dynamically, or at the normal end of a Sequence.

29.8 Timeout period

There are four timeouts defined:

- Link
- Sequence
- Link Recovery
- OLS transmit

Each timeout period associated with different events may specify a different value or a value which represents a common timeout period for all events which are timed. The expiration of the timeout period may be logically derived from the use of one or more physical timers within a specific implementation. A specific timeout period is not specified by the Standard.

29.8.1 Timer

FC-2 uses a 32-bit programmable timer (resolution to be determined). The programmable timer is logically used to notify a link control facility if a timeout period has been exceeded by a particular procedure or event in process.

29.9 Link Error Status Block

Detected errors shall be accumulated in a Link Error Status Block.

The layout is specified in figure 65.

Link Error Status Block

Byte	0	1	2	3
Bits	31 24	23 161718	15 8	7 0
	rrrr rrrr	rrrr rrrr	rrrr rrrr	rrrr rrrr

Figure 65. Link Error Status Register

Specific Bit assignments have not been made.

29.10 Frame reception

Frame reception starts with detection the Start_of_Frame delimiter. Characters are received until normal frame reception is terminated by detection of the End_of_Frame delimiter. Frame reception is abnormally terminated by recognition of any recognized or unrecognized ordered set. A frame received with a proper **SOF**, **EOF**, and valid codepoint translation is then checked for a valid CRC applied to the 8-Bit character codes. A frame with a valid **SOF**, **EOF**, and verified CRC value is designated a valid frame and further processing is performed.

Once frame reception is started, a single code violation within the Frame_Content is detected as an error (code violation is defined by FC-1). If the CRC value is incorrect, the frame is considered in error and no further processing is performed. The **EOF** type is noted.

A frame in error is discarded and the appropriate error bit is posted. No further processing of error frames is attempted.

A frame started by **SOFc1**, **SOFx2**, or **SOFx3** shall be responded to by transmission of the R_RDY Primitive Signal, regardless of frame validity.

29.11 Detailed Error detection / actions

29.11.1 Errors detected

Table 56 lists 10 categories of errors which are detectable. The categories specified relate directly to link integrity or data integrity as previously discussed.

Table 56 (Page 1 of 2). Detailed Errors and Actions

Error Category	Specific Error	Seq Init Action	Seq Recp Action
Link Failure	1. Loss of Signal	1. 12	1. 12
	2. Loss of Sync > timeout period	2. 12	2. 12
Link Errors	1. Code Violation during frame reception	1. 1,11	1. 1,11
	2. Code Violation outside frame reception	2. 11	2. 11
	3. Loss of Sync	3. 11	3. 11
	4. Invalid Ordered Set	4. 11	4. 11
Link Timeout	1. Class 1: all Sequences timed out	1. 6	1. 6
	2. Class 2 and 3: missing R_RDY	2. 6	2. 6
Link Recovery timeout	1. missing LRR response to LR transmission	1. 12	1. 12
	2. missing Idle response to LRR transmission	2. 12	2. 12
Sequence Timeout	1. timeout during Sequence	1. 8	1. 9
	2. timeout at end of Sequence	2. 8,14	2. 9
Delimiter Errors	1. Class not supported	1. 2	1. 2
	2. Delimiter usage error (**SOFc1** while connected)	2. 2	2. 2
Address Identifier errors	1. incorrect D_ID	1. 2	1. 2
	2. incorrect S_ID	2. 2	2. 2
Frame_Content errors	1. CRC	1. 1	1. 1
	2. Busy frame received	2. 5	2. 5
	3. Reject frame received	3. 3	3. 3
	4. Invalid Type	4. 2	4. 2
	5. Invalid Type Modifier	5. 2	5. 2
	6. Invalid R_CTL	6. 2	6. 2
	7. Invalid F_CTL	7. 2	7. 2
	8. Invalid OX_ID	8. 2	8. 2
	9. Invalid RX_ID	9. 2	9. 2
	10. Invalid SEQ_ID	10. 2	10. 2
	11. Invalid SEQ_CNT	11. 2	11. 2
	12. Exchange Error	12. 2	12. 2
	13. Protocol Error	13. 2	13. 2
	14. Data Field too large	14. 2	14. 2
	15. Unexpected Link_Continue	15. 2	15. 2
	16. Unexpected Link_Response	16. 2	16. 2
Data Frame errors	1. buffer not available - Class 1	1. NA	1. 2,7
	2. buffer not available - Class 2	2. NA	2. 4
	3. buffer not available - Class 3	3. NA	3. 1
	4. ABTS frame received	4. NA	4. 9
	5. Out of order error detected	5. NA	5. 13,7
	6. Missing frame	6. NA	6. 7
ACK_1, ACK_N frame errors	1. Out of order error detected	1. 13	1. NA
	2. Abort sequence indicated	2. 8	2. NA
	3. Missing frame	3. 8	3. NA

29.11.2 Actions by Initiator or Recipient

1. **Discard frame**

   When a frame is being received which contains code violation errors, the frame reception logic continues to process the characters within the frame until frame reception is terminated by an **EOF** delimiter, or another Ordered Set. The Frame_Content is discarded, however, the delimiters are noted and appropriate action taken. The Link Error Status Register is updated to record the error.

   If a frame is received with proper delimiters but an incorrect CRC is obtained, the frame is discarded. The delimiters are noted and appropriate action is taken. The Link Error Status Register is updated to record the error.

   If the discarded frame was started by an **SOFc1** and a Dedicated Connection did not exist, Link Recovery is performed as defined in 27.6.

   If the discarded frame was a Class 1 frame which was terminated by an **EOFdt** or **EOFdti**, all active Class 1 Sequences are terminated abnormally. The appropriate Exchange Status Blocks are updated and the Sequence Status Blocks are released for reuse.

   If a Class 3 frame is received and no buffers are available, the frame is discarded.

2. **Transmit P_RJT frame**

   If a frame other than a P_RJT or F_RJT frame is received which contains information in the Frame_Header which is invalid or incorrect, a P_RJT frame is transmitted with the appropriate reason code as specified in 20.3.3.1. Reason codes are prioritized such that the first error encountered is returned as the reason code.

   If a P_RJT or F_RJT frame is received which contains information in the Frame_Header which is invalid or incorrect, the frame is discarded and the Link Error Status Register is updated.

   During a Class 1 Dedicated Connection, if a Data frame is received and no buffer is available, this indicates that the transmitter has an end-to-end Credit tracking problem. The Frame_Header is retained in order to generate a P_RJT while the remainder of the Frame_Content is discarded. The reason code in P_RJT shall indicate "protocol error" and the Abort Sequence Bits shall be set in order to request the Sequence Initiator to Abort the Sequence.

3. **Process Reject**

   When a P_RJT or F_RJT frame is received in response to a frame transmission, it shall be passed to the appropriate Upper Level Protocol in order to process. Certain errors are recoverable by taking an appropriate action, such as Login. The Sequence shall terminated or aborted using the Abort Sequence Protocol, regardless of possible recovery actions. For most non-retryable errors the Exchange shall also be aborted.

4. **Transmit P_BSY frame**

   An N_Port shall track the status of its buffers such that if a Class 2 Data frame is received and no buffer is available, a P_BSY is returned to the transmitter of the frame. Portions of the frame other than the Frame_Header are discarded. The Frame_Header is captured in order to generate a proper P_BSY Link_Response frame.

   If an N_Port receives a Class 1 connect-request frame and its internal link facilities are busy or unavailable, the N_Port responds by transmitting a P_BSY frame with the appropriate reason code (see 20.3.3.2.) with **EOFdt** and discards the connect-request frame.

5. **Process Busy**

   When an F_BSY or P_BSY is received in response to a Class 1 connect-request, the N_Port may:

   — attempt a connect-request to a different destination N_Port if such a request is pending,
   — delay a period of time before reissuing the connect-request, or
   — immediately reissue the connect-request.

   The decision as to which action to take is dictated based on the conditions in the N_Port and the period of time lost if another busy condition is returned.

When an F_BSY or P_BSY is received in response to a Class 2 frame, the N_Port shall retransmit the frame which was busied. In order to avoid reissuing a frame for an extended number of retries an N_Port may choose to count the number of retries and decide to shutdown communication with a specific N_Port.

6. Perform Link Recovery Protocol

When an N_Port has reached a buffer-to-buffer Credit_CNT of zero and has exceeded the Link timeout period, a Link timeout is detected. In Class 1, if an N_Port has timed out all active Sequences, a Link timeout is detected. When a Link timeout is detected, the N_Port or F_Port begins the Link Recovery Protocol.

In addition, in Class 1 an N_Port may not know whether a Dedicated Connection exists or not. An N_Port may initiate the Link Recovery Protocol in order to reach a known state.

7. Set Abort Sequence Bits

When a Sequence Recipient detects a Sequence error from out of order detection or other internal processing errors, the Recipient sets the appropriate Abort Sequence in F_CTL bits 5-4.

```
0 0   Continue Sequence
0 1   Abort Sequence and retransmit
1 0   Stop Sequence and read Status
1 1   Sequence timeout by Recipient
```

The SEQ_CNT of the frame in which the "out of order" condition was detected is saved in the Sequence Status Block. Only the first error is saved in the Sequence Status Block. This information may or may not be required by the Sequence Initiator for recovery purposes.

8. Perform Abort Sequence Protocol

When a Sequence Initiator detects a Sequence error or receives an appropriate Abort Sequence indication in F_CTL bits 5-4 in an ACK for an active Sequence, the Initiator transmits an Abort Sequence Link_Data request to the Recipient and transfers Sequence Initiative in order to complete Abort Sequence processing. See 29.7.1.

9. Abnormally terminate Sequence

When a Sequence Recipient detects a Sequence timeout and no Data frames are being received for the Sequence, the Recipient shall terminate the Sequence and update the Exchange Status Block.

When a Class 1 Dedicated Connection is removed by an **EOF** delimiter or a Primitive Sequence, the N_Port shall abnormally terminate any active Sequences. See 29.7.1.

10. Retry Sequence

When a Sequence has been abnormally terminated, the Sequence Initiator shall retransmit the Sequence.

11. Update LESB

The Link Error Status Block is updated to track errors not directly related to an Exchange.

12. Perform Link Failure Protocol

The Link Failure Protocol is initiated by transmission or reception of the Not Operational Primitive Sequence.

13. Error Policy processing

When an error is detected within a Sequence, the Sequence is either processed normally (process policy), or discarded (discard policy). See 29.6.2.

14. Read Status Block

Read Status Block (either Sequence or Exchange) is accomplished a Sequence using the appropriate Link_Data request.

29.11.3 Exchange Status Block

When a Read Exchange Status Block request is received, the following Payload is transmitted in the Accept Reply Sequence. The content is specified in the arrangement shown in figure 66. The Exchange Status Block is a logical construct and does not require specific hardware facilities.

Following transmission of word 15, a list of Sequence Status Blocks up to a count of X+2 is provided. The ordering of the list of SSBs is from the oldest to the newest.

Word	Bits 31 .. 0
0	OX_ID | RX_ID
1	rrrrrrrr | D_ID
2	E_STAT
3	reserved
4	Word 0-Class Service Parameters
5	Word 1-Class Service Parameters
6	Word 2-Class Service Parameters
7	Word 3-Class Service Parameters
8	P_Mask (SEQ_ID =0 at bit 31)
9	P_Mask
10	P_Mask
11	P_Mask (SEQ_ID =255 at bit 0)
12	Q_Mask (SEQ_ID =0 at bit 31)
13	Q_Mask
14	Q_Mask
15	Q_Mask (SEQ_ID =255 at bit 0)
16	Oldest Sequence Status Block: SEQ_ID | S_STAT | ERROR SEQ_CNT
17	reserved | HIGHEST SEQ_CNT
	...
	Newest Sequence Status Block: SEQ_ID | S_STAT | ERROR SEQ_CNT
	reserved | HIGHEST SEQ_CNT

Figure 66. Exchange Status Block

29.11.3.1 E_STAT

Bit 31 - Exchange owner

- 0 Originator
- 1 Responder

Bit 30 - Sequence Initiative

- 0 Other Port holds initiative
- 1 This Port holds initiative

Bit 29 - Class 1

- 0 not Class 1
- 1 Class 1

Bit 28 - Class 2

- 0 not Class 2
- 1 Class 2

Bit 27 - Class 3

- 0 not Class 3
- 1 Class 3

Bit 26 - Completion

- 0 active
- 1 complete

Bit 25 - Ending Condition

- 0 normal
- 1 abnormal

Bit 24 - Error type

- 0 aborted by initiator
- 1 timed out by recipient

Bit 23 - Error Policy

- 0 discard
- 1 process

Bit 22 - Originator X_ID invalid

- 0 Originator X_ID valid
- 1 Originator X_ID invalid

Bit 21 - Responder X_ID invalid

- 0 Responder X_ID valid
- 1 Responder X_ID invalid

Bits 20-0 - Reserved

29.11.4 Sequence Status Block

Word	Bits 31		0
0	SEQ_ID	S_STAT	ERROR SEQ_CNT
1	reserved		HIGHEST SEQ_CNT

Figure 67. Sequence Status Block

29.11.4.1 S_STAT

Bit 23 - Sequence context

- 0 Initiator
- 1 Recipient

Bit 22 - Completion

- 0 active
- 1 complete

Bit 21 - Ending Condition

- 0 normal
- 1 abnormal

Bit 20 - Error type

- 0 aborted by initiator
- 1 timed out by recipient

Bit 19 - Error Policy

- 0 discard
- 1 process

Bits 18-16 - Reserved

Annex A. Test methods (informative)

This annex is not part of the standard but is included for information purposes only. This section is referenced by and applies to FC-0.

This annex defines terms, measurement techniques, and conditions for testing jitter and optical waveshapes. This annex deals with issues specific to Fiber Channel and is not intended to supplant standard test procedures referenced in the specifications. In cases where there are conflicts between this annex and the referenced standard this annex shall take precedence. The annex directly applies to verifying node performance related to the Optical Interface Specifications.

These same procedures may be used to measure a single component of the system. Component performance is outside the scope of Fiber Channel compliance but it is useful from a design viewpoint. A later annex provides exemplary information on how to interpret component measurements.

A.1 Active output interface

A.1.1 Optical power measurement

Section 6, specifies the average optical power launched into a fibre conforming to section 8.

The measurement shall be made with the node transmitting an idle sequence. The measurement shall be made by the methods of EIA/TIA-526-2 "OFSTP-2, Effective Transmitter Output Power Coupled Into Single-Mode Fibre Optic Cable".

A.1.2 Optical Spectrum Measurement

The center wavelength and spectral width (FWHM or RMS) value of the Active Output Interface can be measured as appropriate using an optical spectrum analyzer per FOTP-127. The patch cable used to couple the light from the Active Output Interface to the spectrum analyzer should be short to minimize spectral filtering by the patch cable. The output signal during the measurement shall be any valid 8B/10B code pattern.

A.1.3 Optical waveform

A.1.3.1 Mask of the eye diagram (laser)

The mask of the eye diagram for the laser transmitters may be measured using a receiver with a fourth-order Bessel-Thompson transfer function given by:

$$H_p = \frac{105}{105 + 105y + 45y^2 + 10y^3 + y^4}$$

$$\text{With } y = 2.114p\,, \quad p = \frac{j\omega}{\omega_r}\,, \quad \omega_r = 2\pi f_r$$

$$f_r = 0{,}75 \times \text{Bit Rate}$$

Note - This filter is not intended to represent the noise filter used within an optical receiver but is to provide a uniform measurement condition.

The nominal attenuation at the reference frequency, f_r, is 3 dB. The corresponding attenuation and group delay distortion at various frequencies is given below:

Table 57. Transmit pulse noise filter

f/f_0	f/f_r	Attenuation (dB)	Distortion (UI)
0,15	0,2	0,1	0
0,3	0,4	0,4	0
0,45	0,6	1,0	0
0,6	0,8	1,9	0,002
0,75	1,0	3,0	0,008
0,9	1,2	4,5	0,025
1,0	1,33	5,7	0,044
1,05	1,4	6,4	0,055
1,2	1,6	8,5	0,10
1,35	1,8	10,9	0,14
1,5	2,0	13,4	0,19
2,0	2,67	21,5	0,30

The mask of the eye diagram is intended to measure the waveshape properties of the optical signal such as rise/fall time, overshoot, undershoot, and ringing. As such this test applies to the deterministic portion of the waveform and does not include random noises in either time or amplitude.

In order to set up the mask the magnitude of the one and zero logic levels is required. These may be determined from the average value of the signals at the 50% time point in a bit cell. If these values are obtained from a measurement system which preserves the amplitude information these values may be used to determine the extinction ratio.

A.1.3.2 Pulse Parameters (LED)

Optical waveform measurements shall be shall be measured using a DC coupled wide bandwidth opto-electronic receiver and an oscilloscope. The signal during the measurement shall consist of a sequence of K28.7 data bytes. This pattern corresponds to a square wave at 10% of the Bit Rate. It is important that the receiver's frequency response, gain flatness and linearity over the range of optical power being measured be sufficient to provide accurate measurement of the 0% and 100% levels and to yield accurate optical rise and fall times and minimum distortion of the optical pulse. A minimum frequency response of five times the data rate is required. In the event of a noisy optical waveform, optical detector, or oscilloscope front end the use of a digital sampling oscilloscope operated in the averaging mode to reduce the noise is recommended.

The Active output interface extinction ratio is a measure of the modulation depth of the optical waveform exiting the node. The extinction ratio is the ratio of voltage corresponding to the 0% level (low light) to the voltage corresponding to the 100% (high light) level.

A.1.4 Jitter measurements

The Active Output Interface jitter specifications apply in the context of a 10^{-12} bit error rate (BER). Jitter may be measured with a digital oscilloscope, or a bit error rate tester (BERT) as described in sections A.3 and A.4.

A.2 Active input interface

The Active Input Interface sensitivity shall be measured by the methods of EIA/TIA-526-3, "OFSTP-3, Fibre Optic Terminal Equipment Receiver Sensitivity and Maximum Receiver Input". The source of the Active Input Interface test signal shall be any optical source conforming to the worst case limits of the Active Input Interface specifications of the media under test.

The test shall be performed with traffic consisting of frames of data so that the receiving equipment may perform its normal synchronizing operations. The frame contents shall consist of an encoded pseudo random pattern whose length is at least 1000 bytes long. The purpose of this pattern is to provide frequency components in the data which span the full frequency range of the clock recovery system. The contents of the frame shall end with a positive running disparity so that the K28.5 character in the first position of the EOF delimiter will be present in the complement form from the K28.5 character in the first position of the SOF delimiter.

A compliant node shall receive the test signal over the range of conditions specified with a frame error rate that corresponds to a BER less than or equal to 10^{-12}. The requirements in section 6 were written in terms of BER to facilitate to the specification of components to be used in a particular implementation.

The characteristics of the test signal may be measured with the methods outlined in sections A.1, A.3, and A.4. Components used in a particular implementation may also be measured with these methods.

A.3 Distortion and jitter contributions

Optical waveform distortion and jitter are measured as the deviation from the ideal time position of the signal at the 50% point of the signal. The 50% is identified as the zero crossing of the AC coupled signal. The zero level is established in absence of the signal.

There are two types of jitter specified in the FC specifications. The definitions and contributing factors are given below and the test methods are given in section A.4.

Deterministic jitter (DJ) Timing distortions caused by normal circuit effects in the transmission system. Deterministic jitter is often subdivided into duty cycle distortion (DCD) caused by propagation differences between the two transitions of a signal and data dependant jitter (DDJ) caused by the interaction of the limited bandwidth of the transmission system components and the symbol sequence.

Random jitter (RJ) RJ is due to noise contributions in the optical receiver and mode partition noise effects on long single mode links. RJ is modeled as a Gaussian process which will average to zero. It shall be characterized by the peak to peak value at a 10^{-12} probability. The peak to peak value is 14,0 times the RMS jitter value.

A.4 Distortion and jitter measurement

In this standard there are two methods used to specify jitter. The LED based systems specify DJ and RJ separately. The laser based systems combine the two components into a single measurement of the eye opening. This is the interval in the bit time which is error free to BER<10^{-12}.

A.4.1 Measurement system

The opto-electronic measurement system described in section A.1.3 shall be used for jitter measurement.

A.4.2 Active output interface eye opening measurement

The Active Output Interface (AOI) optical eye opening measurement involves measuring the open eye of the AOI on a bit by bit basis using a BERT (BIT Error Rate Test) test set. The BER is measured at various T_d's (decision points) within the eye pattern to insure conformance to the eye opening specification in tables 5 and 6 of the standard.

The eye opening is given by:

$$EW = |T_d(\max) - T_o| + |T_o - T_d(\min)|$$

Where:

T_o

= center of the eye pattern,

T_d

= BER decision point as referenced from T_o,

$T_d(\max)$

= rightmost decision point, and

$T_d(\min)$

= leftmost decision point.

For each position of T_d from T_d(min) to T_d (max) a BER measurement is taken, giving the probability of error at the T_d position. In effect T_d is swept across the eye pattern, measuring the probability of error at each point in the eye. The range of T_d values that result in a BER $\leq 10^{-12}$ establishes the eye opening and the smallest range from T_o must be $\geq$ half the appropriate eye opening specification.

In practice, a BER Test set is used to generate and sweep the decision point (using the BERT clock in conjunction with a precise delay generator), to make the bit by bit error count, and to calculate the measured BER. The center of the eye (T_o) pattern is the midpoint between positioning T_d to the left and right edges of the eye to achieve a BER $> 10^{-2}$. The measured BER at T_o, $T_d(\max)$, $T_d(\min)$ must be $\leq 10^{-12}$ and the values of both ($T_d(\max) - T_o$) and ($T_o - T_d(\min)$) must be greater than or equal to half the appropriate eye-window specification in tables 5 and 6 of the standard. All measurements are made with respect to a low pass forth-order Bessel-Thompson filter (described elsewhere in this annex) or equivalent. It is important that the BERT retiming data latch be significantly faster than the timing resolution of interest.

A common practice used to save time is to measure the eye opening at higher probabilities (e.g. 10^{-6}) and then extrapolate to the eye opening at a 10^{-12} probability.

A.4.3 DJ measurement

A digital sampling oscilloscope may be used to measure DJ with a test pattern consisting of repeating K28.5 data bytes. Synchronize the oscilloscope to the pattern repetition frequency and average the waveforms to remove the effects of random jitter and noise in the measurement system. In the 20 bit pattern there will be five positive transitions and 5 negative transitions. Measure the time of the 50 % crossing of all 10 of the transitions. The time of each crossing is then compared to the mean expected time of the crossing and a set of ten timing variations are determined. The DJ is the range of the timing variations.

A.4.4 RJ measurements

If a digital oscilloscope is available which has the capability of measuring time statistics the RJ may be determined by applying a sequence of K28.7 data bytes. The oscilloscope is used to determine the standard deviation of the 50% crossings. The Random Jitter at 10^{-12} will be the +/- 7 sigma limit points of the distribution.

An alternate method is to measure the eye opening by the method of section A.4.2 and subtract the DJ contribution as measured in section A.4.3.

A.5 Relative intensity noise (RIN) measuring procedure

A.5.1 Test objective

When lasers which are subject to reflection induced noise effects are operated in a cable plant with a low optical return loss the lasers will produce an amount of noise which is a function of the magnitude and polarization state of the reflected light.

The magnitude of the reflected light tends to be relatively constant. However, the polarization state varies significantly as a function of many cable parameters, particularly cable placement. In a cable plant which is physically fixed in place the variation is slow. If the fibre is subject to motion, such as occurs in a jumper cable, the change may be sudden and extreme. The effect is unpredictable changes in the noise from the laser with the result that the communication link may exhibit sudden and unexplainable bursts of errors.

The solution to this is to assure that the lasers used do not generate excessive noises under conditions of the worst case combination of polarization and magnitude of reflected optical signal.

The noise generated is a function of the return loss of the cable plant. For the Fiber Channel the specified return loss is 12 dB resulting in the notation of RIN_{12} for the relative intensity noise.

A.5.2 General test description

The test arrangement is shown in Figure 68 The test cable between the Device Under Test (DUT) and the detector forms an optical path having a single discrete reflection at the detector with the specified optical return loss. There must be only one reflection in the system as the polarization rotator can only adjust the polarization state of one reflection at a time.

Two measurements are made by the photodetector. The average optical power is determined by measuring the average current, Ipd, through the detector.

The second measurement is the noise, which is measured by AC coupling the detector into the high frequency electrical power meter. If needed, an amplifier may be used to boost the signal to the power meter.

A low pass filter is used between the photodetector and the power meter to limit the noise measured to the passband appropriate to the data rate of interest.

In order to measure the noise the modulation to the DUT must be turned off. If the laser is modulated the modulation power will be measured by the power meter rather than the noise power.

A.5.3 Component descriptions:

Test Cable The test cable and detector combination must be configured for a single dominate reflection with an optical return loss of 12dB. (The Optical return loss may be determined by the method of FOTP-107) If multiple lengths of cable are required to complete the test setup they should be joined with splices or connectors having return losses in excess of 30 dB. The length of the test cable is not critical but should be in excess of 2 meters.

Polarization Rotator The polarization rotator must be capable of transforming an arbitrary orientation elliptically polarized wave into a fixed orientation linearly polarized wave. A polarization rotator consisting of two quarter wave retarders will have the necessary flexibility. A possible con-

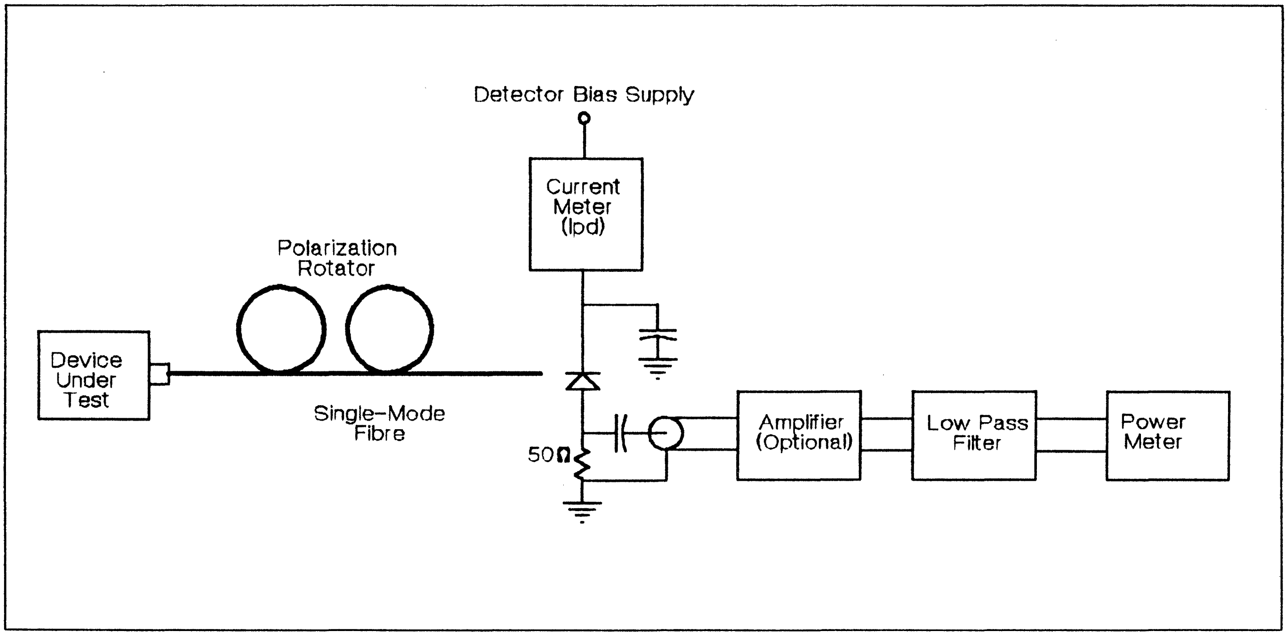


Figure 68. RIN Test Setup

struction technique is given in H.C.LeFever, "Single Mode Fibre Fractional Wave Devices and Polarization Controllers," Electronics Letters, Vol. 16, No. 10, pp 778-780, September 25, 1980.

Detector and Detector/Amplifier The detector may be of any type which is sensitive to the wavelength range of interest. The frequency response of the detector must be higher than the cut-off frequency of the low pass filter.

The detector shall be capable of determining the average and variation of the optical signal. Typically the average optical power is determined by the current, Id, through the diode. The detector assembly shall have an effective output impedance of 50 Ω.

If necessary, the noise may be amplified to a level consistent with accurate measurement by the power meter.

Filter The low pass filter shall have a 3dB bandwidth of approximately 70% of the bit rate. Recommended values are shown in table 178. The total filter bandwidth used in the RIN calculation must take the low frequency cut-off of the DC blocking capacitor into consideration.

Table 58. Filter 3 dB point

Bit Rate	Filter 3 dB Point
265,625 MBit/s	200 MHz
531,25 MBit/s	400 MHz
1,062 5 GBit/s	750 MHz

The filter shall be placed in the circuit as the last component before the power meter so that any high frequency noise components generated by the detector/amplifier are eliminated. If the power meter used has a very wide bandwidth care should be taken in the filter selection to ensure that the filter does not lose its rejection at extremely high frequencies.

Power Meter The power meter should be an RF type designed to be used in a 50 Ω coaxial system. The meter must be capable of being zeroed in the absence of input optical power to remove any residual noise from the detector or its attendant amplifier, if used.

A.5.4 Test Procedure

1. Connect and turn on the test equipment. Allow the equipment to stabilize for the manufacturers recommended warm up time.
2. With the DUT disconnected zero the power meter to remove the contribution of any noise power from the detector and amplifier, if used.
3. Connect the DUT, turn on the laser and verify that the optical output is within normal operation range by means of the detector current. The laser should not be modulated.
4. Operate the polarization rotator while observing the power meter output to maximize the noise read by the power meter
5. Calculate RIN from the observed detector current and electrical noise by use of the equation:

$$RIN = 10 \log \frac{P_e}{BW\, 25\, I_{pd}^2} - G \; dB/Hz$$

Where:
RIN = Relative Intensity Noise

I_{pd}

= Current through the detector (A)

P_e

= Electrical noise power in Watts

BW

= Bandwidth of the measuring system (Hz)
= Low Pass Bandwidth of filter - High Pass Bandwidth of DC blocking capacitor.

G

= Gain in dB of any amplifier in the Noise Measurement path

Annex B. Alternative cable plant usage (informative)

This annex is not part of the standard but is included for information purposes only.

In some cases, it will be desirable to use an alternative multi-mode cable plant to those described in sections 8.2 and 8.3. This may be due to the need for extended distances or for operation in locations where alternative cable plants are presently installed. These fiber types have not been studied and details for their use is not provided for in the main body of the specification. Therefore, using these fibre types may change the maximum achievable distance between nodes.

Table 59. Alternative fibre types

Nomenclature	Primary Clause	Fibre Type	Distance
50-M6-SL-I	6.2	62.5μm	2m-1Km
25-M6-SL-I	6.2	62.5μm	2m-2Km
25-M5-LE-I	6.3	50μm	Note
12-M5-LE-L	6.4	50μm	Note

Note: These cases will have to be treated on an individual basis. See Annex D for methods of computing distances.

Annex C. Electrical interface example (informative)

This annex is not part of the standard but is included for information purposes only.

This annex describes an example implementation of the electrical interfaces for optoelectronic modules to meet the requirements of the Fibre Channel. This is a minimum set of requirements and it is recognized that individual implementers may desire to provide features in excess of this minimum set to accomplish additional system functions or as an aid to testing.

A block diagram of the modules is shown in Figure 69. It is option of individual implementers to place all or part of the function in a product. Because of this individual products may not have all of the interfaces exposed. Any unexposed interface is under no obligation to conform to this example.

This is an example only. Conformance to this example is not required for Fibre Channel conformance. Fibre Channel conformance is obtained by conformance to the requirements of the main body of this standard.

It is realized that differing communications media will require differing controls. For this reason this example limits itself to consideration of only the data interface and those control functions which will be common over all media. Therefore, the block labeled "Media Control" in Figure 69 will not be described in this example along with it's related signals Media_Control and Media_Status.

The example includes the majority of the function of the FC-0 layer and a portion of the FC-1 layer.

C.1 Communications Levels

Two communications levels are employed. The high speed, full data or clock rate, signals are implemented in differential ECL. The lower speed lines associated with the parallel data transfers and function controls are implemented in TTL compatible voltage levels.

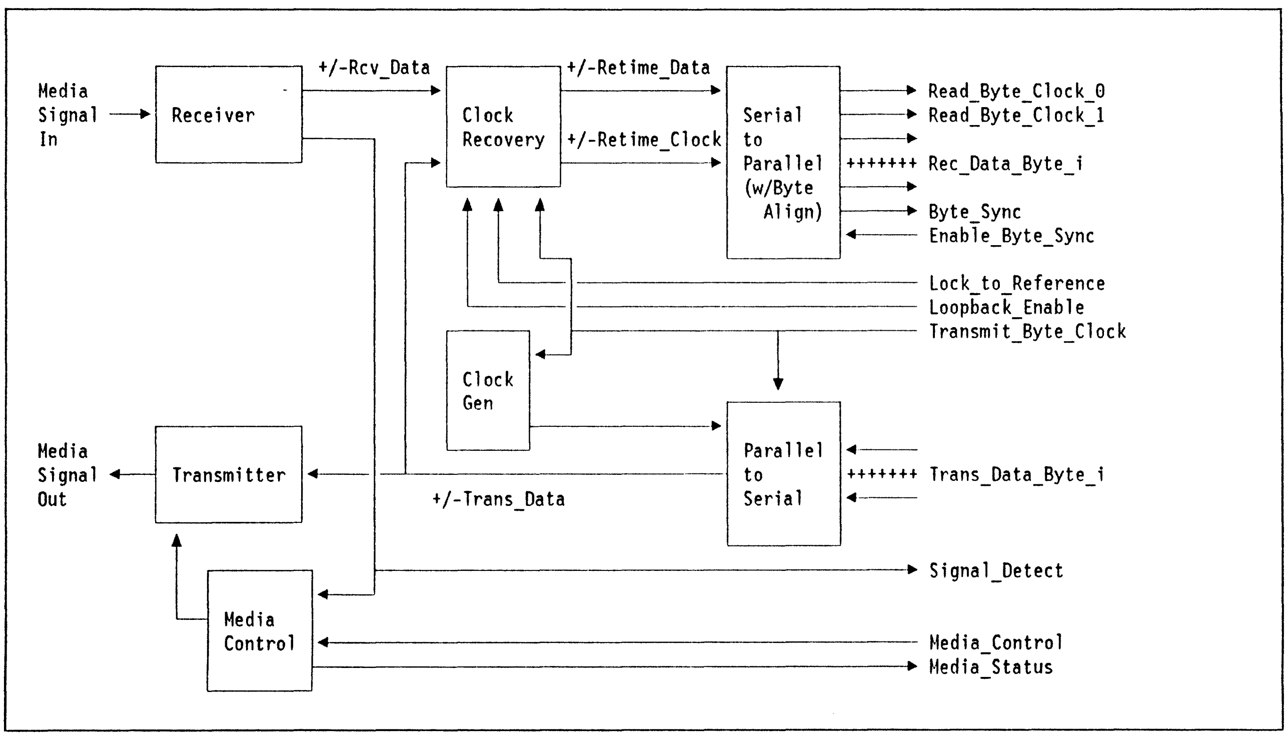


Figure 69. Fibre Channel Implementation Block Diagram

C.1.1 ECL

The high speed communication of the serial data and clock are implemented in differential ECL. The signal swings are shown in Figure 70 referenced to the most positive voltage driving the ECL interface circuits. This voltage could be either ground, in the case of conventional negative ECL, +5,0 volts, in the case of positive referenced ECL. These outputs should be capable of driving 50 Ω transmission lines terminated in 50 Ω to 2,0 volts below the reference level. The drivers and receivers are intended to operate only within the package and are not required to have the electrostatic discharge protection required for driving box to box cables.

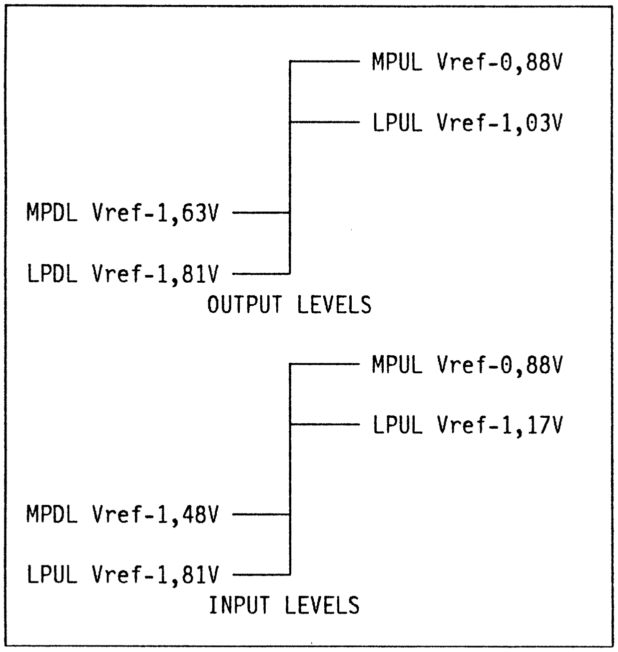


Figure 70. ECL Communications Levels

C.1.2 TTL

The parallel communications and control lines are implemented in TTL compatible voltage swings as shown in Figure 71 in order to provide compatibility with the widest number of logic technologies in the host system. Due to the relatively high speeds of the signals, especially the byte clocks in the 1,0625 GBit/s systems, the use of series resistors in the driver outputs is strongly recommended to minimize ringing. Because of this, the current capability of the drivers has been reduced to allow adequate output voltages in the presence of an output resistor.

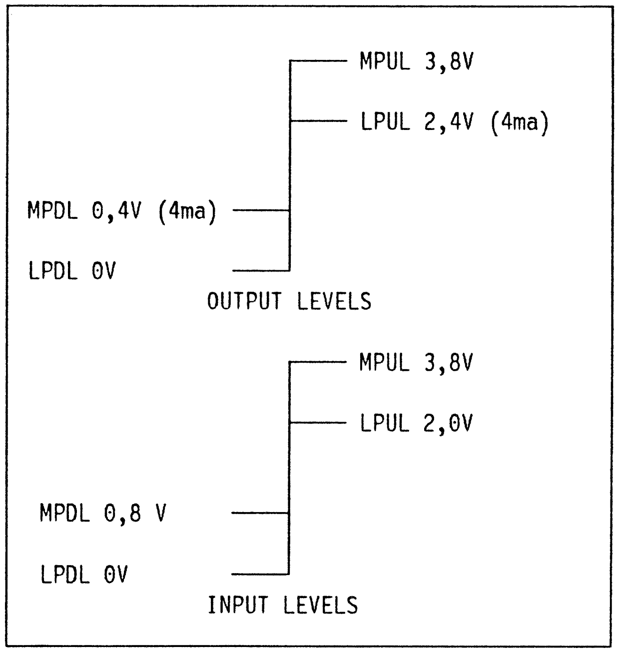


Figure 71. TTL Compatible Communications Levels

C.2 Serial Interfaces

All serial interfaces communicate on differential ECL communication levels.

C.2.1 Transmit Serial Interface

The transmit serial interface communicates the data from the serializer to the communications media. It corresponds to the FC-0_Data.Request primitive of the FC-0 Service Interface.

C.2.2 Receive Serial Interface

The receive serial interface communicates the received data from the media interface circuits to the clock extraction and retiming circuits.

C.2.3 Receive Retimed Serial Interface

The receive retimed data carries the data and recovered clock from the clock recovery circuits to the deserializer.

C.2.3.1 Retimed_Data

The Retimed_Data lines are an implementation of the FC-0_Data.Indication primitive of the FC-0 service interface.

C.2.3.2 Retimed_Clock

The Retimed_Clock lines carry the clock signal recovered from the Receive Serial Interface to the deserializer. This line is the implementation of the FC-0_Clock_Out.Indication primitive of the FC-0 service interface.

C.2.3.3 Retimed Serial Interface Timing

The relative timing between the Retimed_Data and the Retimed_Clock are shown in Figure 72.

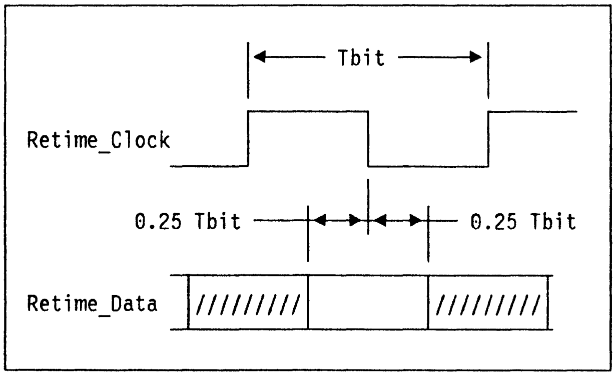


Figure 72. Retimed Interface Signal Timing

C.3 Parallel Interfaces

The parallel interfaces are implemented using TTL compatible voltage levels.

C.3.1 Bus width

The example allows for bus widths of either one or two bytes depending on data rate. The 265,625 MBit/s rate utilized a one byte wide bus, while the 1,062 5 GBit/s rate utilizes a two byte bus.

C.3.2 Transmit Parallel Interface

The transmit parallel interface timings are shown in Figure 73. The data transfers are timed by the host system supplied Transmit_Byte_Clock which also acts as a frequency reference for the transmit serial data and the receiver clock extraction circuitry. Data transfers are timed from the rising edge of the clock. A one byte interface would use the Trans_Data_Byte_0 timings and a two byte interface would use both timings.

In the case of the two byte interface the two data fields are to be clocked into the optoelectronic module alternately. The module timing is arranged to allow the system to switch both bytes together if desired.

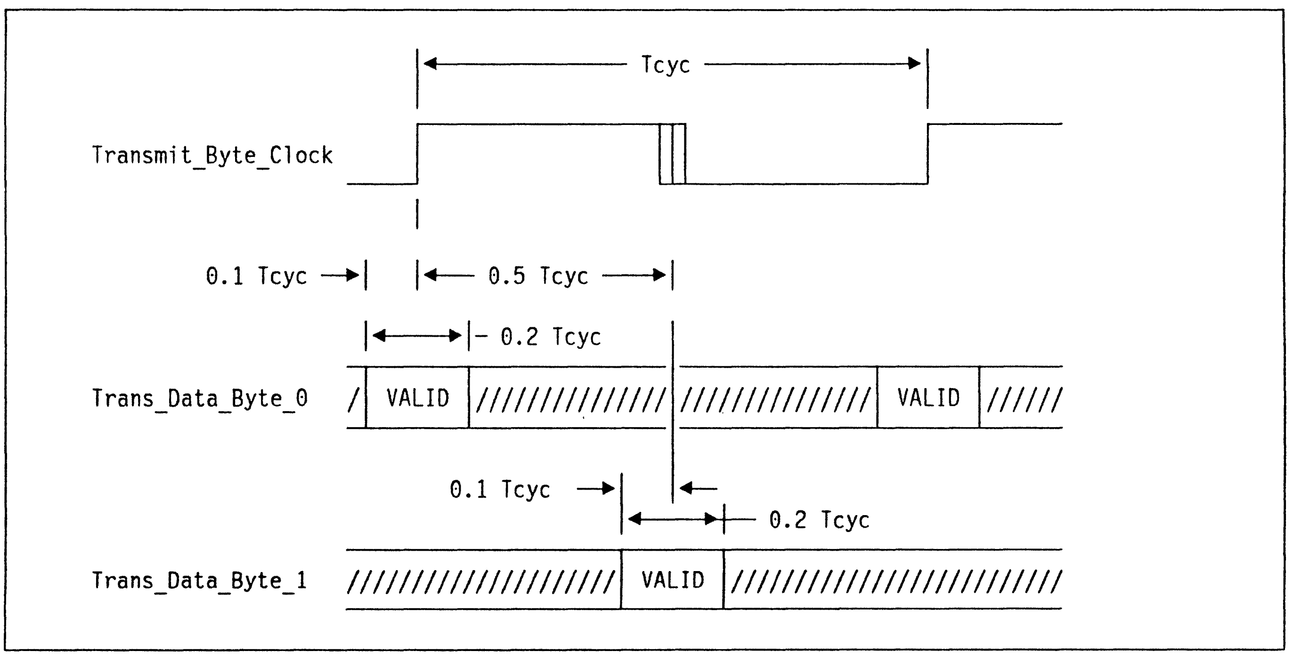


Figure 73. Parallel Transmit Data Interface Timing

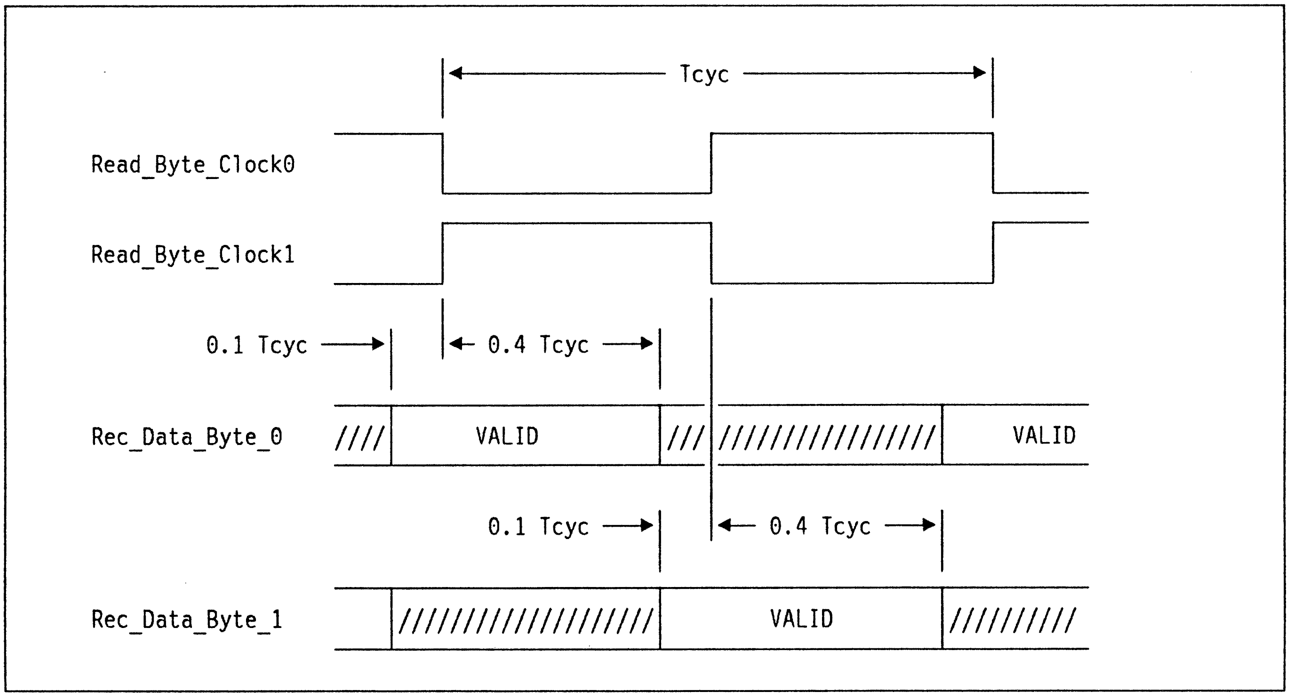


Figure 74. Parallel Receive Data Interface Timing

C.3.3 Receive Parallel Interface

The receive parallel interface timings are shown in Figure 74. In the case of a two byte interface two receive byte clocks are generated by the optoelectronic module to simplify system timings. The bytes are presented alternately to reduce simultaneous switching noise.

In the case of a one byte interface only Read_Byte_Clock_0 and Rec_Data_Byte_0 are required.

The data bytes are valid about the negative edge of their respective Read_Byte_Clocks.

C.4 Transmit Clock Reference

The Transmit_Byte_Clock forms the reference for the data rate synthesizer. The timing is performed from the leading edge which must have low jitter. This line performs the function of the FC-0_Clock_Reference.Request primitive of the FC-0 Service Interface.

C.5 Control and Status Interface

The control and status lines are implemented in TTL compatible communications levels.

C.5.1 Receive Bit Sync Acquire

The receive interface bit synchronization is under control of the host system. When the FC-1 layer wishes to acquired bit synchronization the following sequence is followed:

1. Bring Lock_to_Reference line positive.
2. Wait 2,500 bit times for the PLL to lock to the reference frequency provided by the Transmit_Byte_Clock.
3. Bring Lock_to_Reference line negative.
4. Wait 2 500 bit times for the PLL to lock to the incoming data.

Should synchronization be lost while the PLL is operating near to the data frequency the clock recovery will take less than 2 500 bits to reacquire synchronization. This may occur as a result of a phase jump in the incoming data or activation of the loopback function.
This bit synchronization activity results in establishing the frequency and timing of the Retime_Clock which in turn drives the deserializer and hence the Read_Byte_Clocks. In the event that the incoming signal is lost the activity of the Retime_Clock is undefined. Under these conditions the frequency or other activity of the Read_Byte_Clocks is undefined.

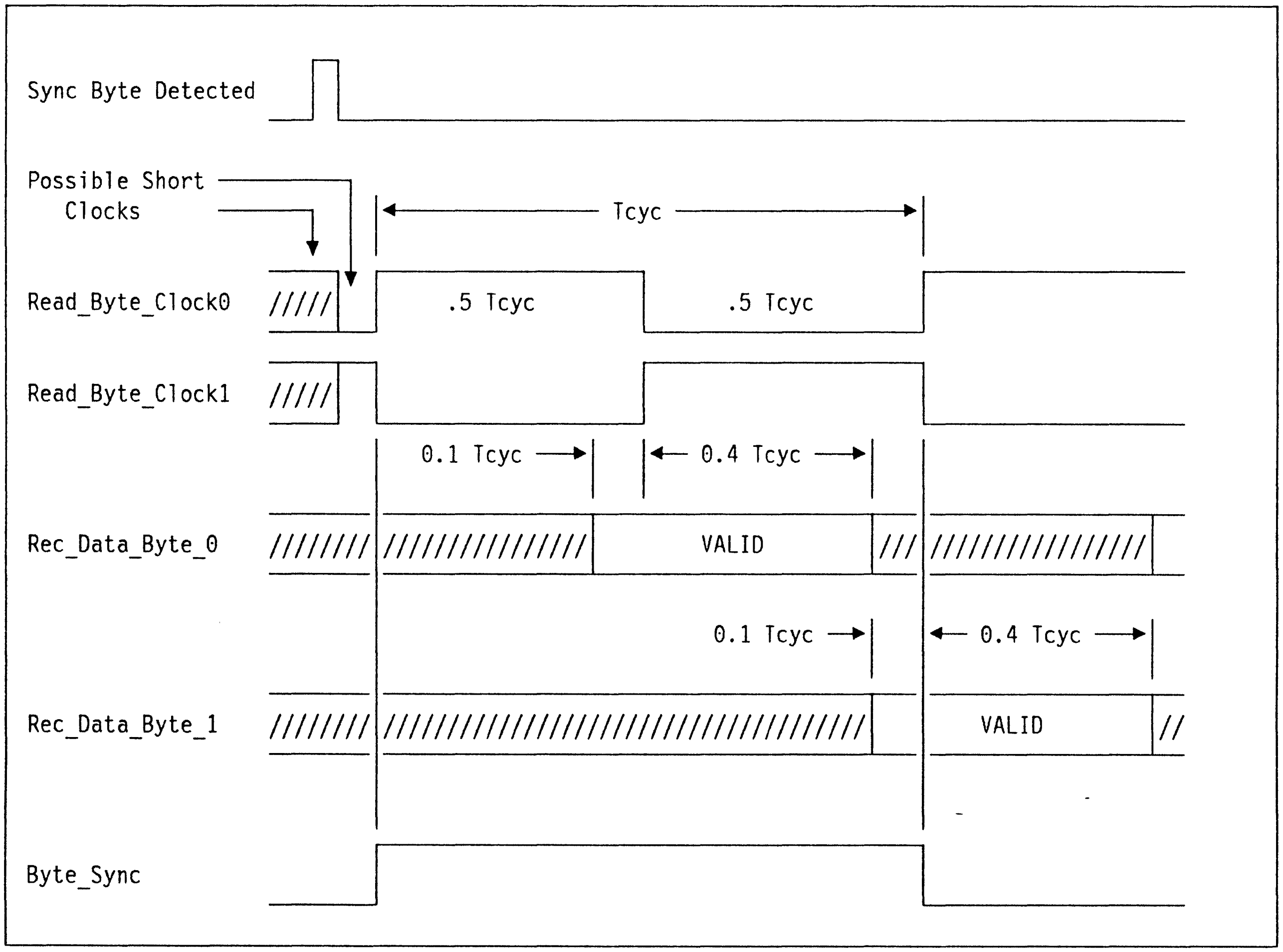


Figure 75. Parallel Receive Data Interface Timing

The Lock_to_Reference line is the implementation of FC-0_Resync.Request primitive of the FC-0 service interface.

C.5.2 Receive Byte Alignment

The optoelectronic module contains circuitry to provide byte alignment. The circuitry will cause the output to be aligned on any occurrence of the alignment pattern in the incoming serial data stream. The alignment pattern is defined as the COMMA sequence of the 8B/10B code. This is either of the ten bit sequences 0011111XXX or 1100000XXX. These sequences contain the first seven bits of the K28.5 special character in both disparities.

When alignment occurs the byte of the incoming data which triggered the alignment will be presented in uncorrupted form on the parallel data out lines. In the case of a two byte interface the byte which triggered the alignment data will be presented as Rec_Data_Byte_0.

When the byte alignment pattern is detected in the incoming data stream the Read_Byte_Clocks are forced to an initial state. The state is undefined except for the following: When the clock sequence resumes there shall be at least one full half width of the Read_Byte_Clock prior to the clock edge about which the data is valid. This is illustrated in Figure 75.

The occurrence of a byte alignment is signaled by a pulse on the Byte_Sync line. The pulse will have a nominal leading edge position coincident with the nominal rising edge timing of Read_Byte_Clock0. The falling edge of the pulse will have a nominal timing coincident with the following rising edge location of Read_Byte_Clock0. Due to implementation dependant timing skews the actual relative timing between Read_Byte_Clock0 and Byte_Sync will vary with the implementation.

This line signals the using system that the presented byte is the first byte of an ordered set as defined by FC-1 and that the previous clocks may have been shortened.

The alignment is enabled by holding the Enable_Byte_Sync input high. This allows the byte alignment to be disabled for test purposes. Alternately, the byte alignment may be enabled in single acquire mode by lowering the Enable_Byte_Sync input in response to a Byte_Sync output.

It should be noted that if the byte sync is enabled the internal circuits will respond to any occurrence of the alignment pattern in the input data stream. This may result in two anomalous situations. A single bit error may cause the alignment pattern to be falsely detected out of position. If the byte sync is enabled an alignment to a false boundary will occur with the resulting destruction of all data until a properly located alignment pattern is detected. Secondly, under an open link condition the thermal noise of the receiver may cause many false locks to occur. The Byte_Sync line will be pulse randomly and the Read_Byte_Clocks will be reset randomly.

C.5.3 Loopback Function

When the Loopback_Enable Line is brought positive the loopback function is enabled. In this operating mode the serial data stream from the transmitter is routed to the input of the retiming latch. After the time required for bit and byte synchronizations this data will be available on the receive data output lines. During loopback the operation of the communications media is not defined.

The loopback function is defined at the FC-1 rather than the FC-0 level.

C.5.4 Signal Detect

The Signal_Detect line shall go positive when an incoming signal is detected. This line is the implementation of the FC-0_Signal_Detect.Indication primitive of the FC-0 service interface.

Annex D. Cable plant usage (informative)

This annex is not part of the standard but is included for information purposes only.

The data links described in section 6 provide maximum optical power budgets. For planning purposes these data links will support the communication distances indicated in tables 5 and 6. Cable plant specification in section 8 defines the cable plant optical attenuation ranges for the data links.

This section documents the assumptions leading to the nominal communications distances and describes the cable plant in terms of number of connectors and splices and assumed cable parameters. Operation within this combination of parameters will provide for operation at the nominal communications distance. Other combinations will support greater or lesser distances.

D.1 Analysis method, loss limited

Using a statistical approach, the system loss budget constraint requires that the loss budget exceed the mean loss of the cable plant by at least three standard deviations.[1]

Therefore:

$$LB - \mu_L > 3\sigma_L \quad \text{(E.1)}$$

Where:

LB = Loss Budget

μ_L = Mean link loss

σ_L = Standard deviation of link loss

The link losses are given by:

$$\mu_L = l_t\mu_c + N_s\mu_s + N_{CON}\mu_{CON} + \mu_{cl} \quad \text{(E.2)}$$

and

$$\sigma_L^2 = l_R l_t \sigma_c^2 + N_s\sigma_s^2 + N_{CON}\sigma_{CON}^2 + \sigma_{cl}^2 \quad \text{(E.3)}$$

Where:

l_t = Total length of the cable plant

l_R = Average reel length of the cable

μ_c = Mean cable loss

σ_c = Standard deviation of cable loss

N_s = Number of splices

μ_s = Mean splice loss

σ_s = Standard deviation of splice loss

N_{CON} = Number of connectors

μ_{CON} = Mean connector loss

σ_{CON} = Standard deviation of connector loss

μ_{cl} = Mean excess coupling loss due to mode field mismatch in the cable plant

σ_{cl} = Standard deviation of excess coupling loss

A cable splice is assumes to be located at each end of the trunk and at one per Km over the

[1] This requirement is more stringent than the two sigma requirement found in TI.106, "American National Standard for Telecommunications Digital Hierarchy Optical Interface Specification: Single-Mode," March 7,1988. This additional requirement is in anticipation of a more diverse, multivendor, cable which is more difficult to control over it's life.

length of the trunk. In addition two splices are assumed for future repair and change activity. This results in an effective reel length of 1 Km. The number of splices in the link is given by:

$$N_s = 3 + \frac{l_t}{1}$$

where Ns is an integer rounded upward.

D.2 Analysis method, bandwidth limited

For multimode fiber length estimates, the total distance limitation can be loss limited (see D.1 above) or bandwidth limited. There are two cases for the bandwidth limitation: short wavelength laser datalinks and LED datalinks.

For the short wavelength datalink, divide the cable manufacturer's specified bandwidth, given in MHz•Km, by a constant factor of 0.8 to obtain the cable bandwidth in units of MBits/s•km. This 0.8 factor is a net result of combining commonly accepted cable bandwidth to system baud rate conversion factor of 0.7 with the additional effect of chromatic dispersion present at the 780 nm wavelength region. Then divide this number by the data rate, given in MBits/s to determine the maximum cable plant distance.

For the LED data link, an effective cable bandwidth is calculated from the modal bandwidth and chromatic dispersion (intramodal) contributions. The length limit is then calculated from this effective cable bandwidth.

The Effective System Bandwidth can be represented by the relationship of intramodal and modal bandwidth of a given span as shown below:

$$\frac{1}{BW_{effect}^{\;2}} = \frac{1}{BW_{modal}^{\;2}} + \frac{1}{BW_{intra}^{\;2}} \qquad \text{(E.4)}$$

where:

BW modal
: = The bandwidth of a fiber limited by the pulse broadening due to optical power traveling via different waveguide modes. This is the specified bandwidth of the cable.

BW intra
: = The bandwidth of a fiber limited by the pulse broadening due to chromatic dispersion (the speed of an optical pulse changes as its wavelength changes).

The chromatic dispersion of a fiber can be calculated by using the equation below:

$$D(\lambda) = \frac{S_0}{4}\left[\lambda - \frac{\lambda_0^{\,4}}{\lambda^3}\right] \text{ [ps/(nm•km)]} \qquad \text{(E.5)}$$

Where

$D(\lambda)$
: = Chromatic dispersion in ps/nm•km

S_0
: = Dispersion function slope at λ_0

λ_0
: = The zero dispersion wavelength

λ
: = The wavelength of operation

For LEDs operating near 1300 nm:

$$BW_{intra} = \frac{187000}{\sigma_{intra}} \text{ [MHz]} \qquad \text{(E.6)}$$

$$\sigma_{intra} = L\sqrt{D(\lambda)^2(\Delta\lambda \times k)^2 + \frac{S_0^{\,2}(\Delta\lambda \times k)^4}{2}} \qquad \text{(E.7)}$$

Where:

σ_{intra}
: = RMS pulse broadening in ps

k
: = 0.425 (the conversion factor from FWHM to RMS)

$\Delta\lambda$
: = FWHM source spectral width in nm

L
: = Length in km

S_0
: = Dispersion slope in ps/(nm
2•km)

D.3 Single mode cable plant usage

For the single mode cable length estimates consider the data links of section 6.1, "SM data links." The data links with a nominal distance capability of 10 Km have 14 dB loss budgets and the data links of with a nominal distance of 2 Km have 6 dB loss budgets. The connector at each end is accounted for in the interface power specifications of section 6.1 and is not included in the cable plant. Two (2) dB has been removed from the difference between the transmitter output power and the receiver sensitivity specifications to account for reflection and dispersion power penalties.

The cable parameters for the single mode loss budget are specified in table 60.

Table 60. Single mode cable plant parameters

FC-0 Parameter	Sym	Value
Cable	μ_c σ_c	0,50 dB/Km 0 dB/Km
Splice	μ_s σ_s	0,15 dB/spl 0,10 dB/spl
Connector	μ_{con} σ_{con}	0,60 dB/con 0,20 dB/con
Additional coupling loss	μ_{cl} σ_{cl}	0,4 dB 0 dB

Substituting in Equations (E.2) and (E.3) gives:

$$\mu_L = 0{,}50 l_t + 0{,}15(3 + l_t) + 0{,}6 N_{CON} + 0{,}40$$
$$= 0{,}65 l_t + 0{,}85 + 0{,}6 N_{CON}$$

and

$$\sigma_L^2 = l_R l_t (0)^2 + (3 + l_t)(0{,}10)^2 + N_{CON}(0{,}20)^2 + (0)^2$$
$$= 0{,}01 l_t + 0{,}04 N_{CON} + 0{,}03$$

The maximum lengths estimated by iteratively solving Equation (E.1) found in table 61.

Table 61. Calculated max. single mode length

FC-0	LB	N_{con}	Length
100-SM-LL-L 50-SM-LL-L 25-SM-LL-L	16 - 2 = 14	8	9,8 Km
100-SM-LL-I 25-SM-LL-I	8 - 2 = 6	4	2,1 Km

D.4 Multi-mode cable plant usage

The data link in section 6.3 provides a maximum optical power budgets of 12 dB for SW laser links and 6 dB for LW LED links. The connector at each end is accounted for in the interface power specifications and is not included in the cable plant. In SW laser or LED based systems, there are no additional loss to account for reflection and dispersion power penalties. For planning purposes the data link will support distances shown in table 63 based on the assumption noted in table 6. Other combinations will support greater or lesser distances.

The cable parameters for the multi-mode loss budget are specified in table 62.

Table 62. Multi mode cable plant parameters

FC-0 Parameter	Sym	Value
Cable Versions 1) 50/125, λ=780nm 2) 62,5/125, λ=780nm 3) 50/125 and 62,5/125 λ=1 300nm	μ_{c1} μ_{c2} μ_{c3} σ_c	4,0 dB/Km 4,5 dB/Km 1,5 dB/Km 0 dB/Km
Splice	μ_s σ_s	0,15 dB/spl 0,10 dB/spl
Connector	μ_{con} σ_{con}	0,5 dB/con 0,2 dB/con
Additional coupling loss	μ_{cl} σ_{cl}	0 dB 0 dB

The maximum lengths estimated by iteratively solving equation (E.1) are:

Table 63. Calculated max. multi-mode length

FC-0	LB	N_{con}	Length	Note
50-M5-SL-I	12	4	1,0 Km	1
50-M6-SL-I	12	4	350 m	2
25-M5-SL-I	12	4	2,0 Km	
25-M6-SL-I	12	4	700 m	3
25-M5-LE-I				5
25-M6-LE-I	6	4	1.0 Km	4
12-M5-LE-I				5
12-M6-LE-I	6	4	500 m	

Notes:

1. This length limit of 1,0 Km is due to fibre bandwidth constraints (see table 20), the length limit due to loss budget (LB) would be 2,0 Km.
2. This length limit of 350 m is due to fibre bandwidth constraints (see table 17), the length limit due to loss budget (LB) would be 1,8 Km.
3. This length limit of 700 m is due to fibre bandwidth constraints (see table 17), the length limit due to loss budget (LB) would be 1,8 Km.
4. This length limit of 1,0 Km is due to fibre bandwidth constraints (see table 17), the length limit due to loss budget (LB) would be 1,4 Km.
5. For these cases the distance must be determined on a case by case basis.

Annex E. Coaxial cable examples (informative)

This annex is not part of the standard but is included for information purposes only.

E.1 Example of CATV coax cable characteristics

Table 64. Typical characteristics of 75 Ω, 0,27″ O.D. CATV coax (e.g.: RG-6/U)

Category Electrical	Impedance 75±5 Ω	Capacitance 17 pF/ft nom.	Attenuation 11dB/50m @531MHz	Velocity 82%
Category Conductor	Material Bare Copper	Size AWG 18	Construction Solid	Outer diameter 0,96 mm nom.
Category Insulation	Material Foamed polyethylene	Wall thickness 1,77 mm nom.	Dielectric Constant 1,50 nom.	Outer diameter 4,57 mm nom.
Category Shield	Material Tin plated Cu braid over foil		Coverage 100% min.	Outer diameter 5.84 mm nom.
Category Jacket	Material PVC	Wall thickness 0,51 mm nom.	Color --------	Outer diameter 6,86 mm nom.

E.2 Example of plenum rated coaxial cable characteristics

Table 65. Typical plenum rated coaxial cable characteristics (e.g.: RG-6/U)

Category Electrical	Impedance 75±5 Ω	Capacitance 16,5 pF/ft nom.	Attenuation 11dB/50m @531MHz	Velocity 82%
Category Conductor	Material Bare copper covered steel	Size AWG 18	Construction Solid	Outer diameter 1,01 mm nom.
Category Insulation	Material Foamed Teflon	Wall thickness 1,65 mm nom.	Dielectric Constant 1,50 nom.	Outer diameter 4,37 mm nom.
Category Shield	Material Tin plated Cu braid over foil		Coverage 100% min.	Outer diameter 5,59 mm nom.
Category Jacket	Material Teflon	Wall thickness 0,203 mm nom.	Color --------	Outer diameter 6,00 mm nom.

E.3 Example of miniature coax cable characteristics

Table 66. Typical characteristics of 75 Ω, miniature coax (e.g.: RG-179 B/U)

Category Electrical	Impedance 75±5 Ω	Capacitance 20 pF/ft nom.	Attenuation 7dB/10m @531MHz	Velocity 70%
Category Conductor	Material Silver coated copper covered steel	Size AWG 30	Construction 7-38 stranded	Outer diameter 0,305 mm nom.
Category Insulation	Material Extruded TFE Teflon	Wall thickness 0,635 mm	Dielectric Constant 2,10 nom.	Outer diameter 1,60 mm nom.
Category Shield	Material Silver coated Cu braid		Coverage 95% min	Outer diameter 2,11 mm nom.
Category Jacket	Material FEP	Wall thickness 0,203 mm nom.	Color -----------	Outer diameter 2,54 mm nom.

E.4 Example of double-shielded miniature cable characteristcs

Table 67. Typical characteristics of 75 Ω, double-shielded miniature coax (e.g.: miniature RG-59)

Category Electrical	Impedance 75±5 Ω	Capacitance 17 pF/ft nom.	Attenuation 7dB/10m @531MHz	Velocity 78%
Category Conductor	Material Copper covered steel	Size AWG 26	Construction Solid	Outer diameter 0,508 mm nom.
Category Insulation	Material Foamed polyethelene	Wall thickness 0,711 mm	Dielectric Constant 1,50 nom.	Outer diameter 1,90 mm nom.
Category Shield #1	Material Bonded Alum. Tape		Coverage 96% min	Outer diameter 2,59 mm nom.
Category Shield #2	Material Aluminum braid		Coverage 60% min	Outer diameter 2,00 mm nom.
Category Jacket	Material PVC	Wall thickness 0,355 mm nom.	Color -----------	Outer diameter 3,71 mm nom.

Annex F. Preferred connector requirements (informative)

This annex is not part of the standard but is included for information purposes only.

- Application usage; machine interface only (FCS compliant connector on Printed Circuit, Patch Panel or Bulk Head encouraged but not required)
- Single connector design for Multi-mode and Single mode
- Single channel (2-fiber) granularity
- Duplex & polarized
- Keyed to prevent cross plugging
- Color differentiation (multi-mode / singlemode)
- Push/pull insertion/withdrawal
- Audible/tactile insertion feedback
- Low insertion/withdrawal force
- Easy ferrule access for inspection and cleaning
- The Single mode FCS receptacle and associated Single mode connector plug will have to be unique and different from the Multimode receptacle/connector plug design to insure the Multimode connector plug cannot be engaged in a Singlemode receptacle. This is required for LASER Safety Compliance.

Table 68. Preferred connector characteristics

FERRULE CHARACTERISTICS	MULTI-MODE	SINGLE MODE
Diameter	2,5 mm	2,5 mm
FIBRE		
Size	50/125 μm 62,5/125 μm	9/125 μm
FUNCTIONAL/OPERATIONAL		
Connector optical repeatability (1) Connector optical cross plug repeatability (2) Return loss (fibre-fibre)	< 0,3 dB < 1,0 dB 26 dB min	< 0,3 dB < 1,3 dB 30 dB min
MECHANICAL CHARACTERISTICS		
Axial pull force (3) Low insertion/withdrawal force Cable/connector axial pull strength (min)(4) Resistance to off axis pull	90 N (20 lbs) 20 N (4,5 lbs) 140N (30 lbs)	90 N (20 lbs) 20 N (4,5 lbs) 140N (30 lbs)
CONNECTOR SIZE		
Ferrule to ferrule spacing (maximum) Lateral connector pitch (maximum) Low profile component height (maximum)	12.7mm (0,5") 25mm (0,98") 12mm (0,47")	12.7 mm (0,5") 25mm (0,98") 12mm (0,47")
ENVIRONMENTAL		
Operating temperature range Relative humidity range Storage/shipping temperature Storage/shipping relative humidity Cable flammability Connector assembly flammability	0 to 60 ° C 5 to 95% RH -40 to 85 ° C 5 to 100% RH (5) (5)	0 to 60 ° C 5 to 95% RH -40 to 85 ° C 5 to 100% RH (5) (5)
RELIABILITY		
Insertion/withdrawals Product life (minimum)	250 min 100K hours	250 min 100K hours

NOTES -

1. **Connector optical repeatability**: is the fibre-to-fibre optical loss repeatability encountered with multiple insertions in the same mating connector pair.
2. **Connector optical cross plug repeatability**: is the fibre-to-fibre optical loss between any two ANSI FCS compliant connectors.
3. **Axial pull force on the cable & connector latched in its mating receptacle**: is the receptacle-to-connector minimum retention force.
4. **Cable/connector axial pull force**: is the minimum connector-to-cable retention force (measure of cable strain relief strength).
5. **All applicable safety codes**: are to apply and are dependent on application/facility usage.

Annex G. FC-0 Service interface (informative)

This annex is not part of the standard but is included for information purposes only.

This annex defines the interfaces of the communications media controls and services that are valid for all FC-0 data links. The controls and services are described in terms of logical primitives rather than physical signals to allow for the widest range of physical implementations.

The interface between FC-0 and FC-1 is intentionally structured to be technology and implementation transparent. That is, the same set of commands and services may be used for all signal sources and communications media. The intent is to allow for the the interface hardware to be interchangeable at the system level without regard to the technology of a particular implementation. As a result of this, all safety or other operational considerations which may be required for a specific communications technolgy are to be handled by the FC-0 level associated with that technology. Such safety features are provided by by the Link Control block of Figure 76.

G.1 General Description

As an aid in visualizing the FC-0 services refer to Figure 76. In this figure the general function performed by the FC-0 level are illustrated along with the logical control associated with the functions. This diagram is meant to represent the most complex implementation of the FC-0 level. In some cases individual implementation, or some types of media, may not require all of these function or logical communications.
For example, LED implementations will not require the FC-0_Signal_Detect to be communicated to the link control function. In the case of a SAW filter type of clock recovery the FC-0_Resync and the FC-0_Clock_Reference will not be required. The basic function of the primitive controls and services are:

FC-0_Data.Request
The serial data to be transmitted over the communications media.

FC-0_Data.Indication
The retimed serial data stream received from the communications media.

FC-0_Transmit
The command to turn the transmitter on and off.

FC-0_Transmit_State
The present internal state of the transmitter.

FC-0_Signal_Detect
An indication of the presence or absence of a signal on the communications media.

FC-0_Clock_Out
The timing clock recovered from the incoming data.

FC-0_Clock_Reference
A frequency reference to be used as an aid in acquiring bit synchronisation.

FC-0_Resync
A command from the FC-1 level to attempt to reacquire bit synchronisation.

G.2 FC-0 States

G.2.1 Transmitter States

The transmitter is controlled by the FC-1 level. Its function is to convert the serial data received from the FC-1 level into the proper signal types associated with the operating media.

- **Transmitter Not-Enabled State**: A not-enabled state is defined as the optical output off for optical communication media and a logical zero for coaxial media. This is the state of the transmitter at the completion of power on sequence unless the transmitter is specifically directed otherwise by the FC-1 level.
- **Transmitter Enabled State**: The transmitter shall be deemed to be in an enabled state when the transmitter is capable of operation within its specifications while sending valid data patterns.

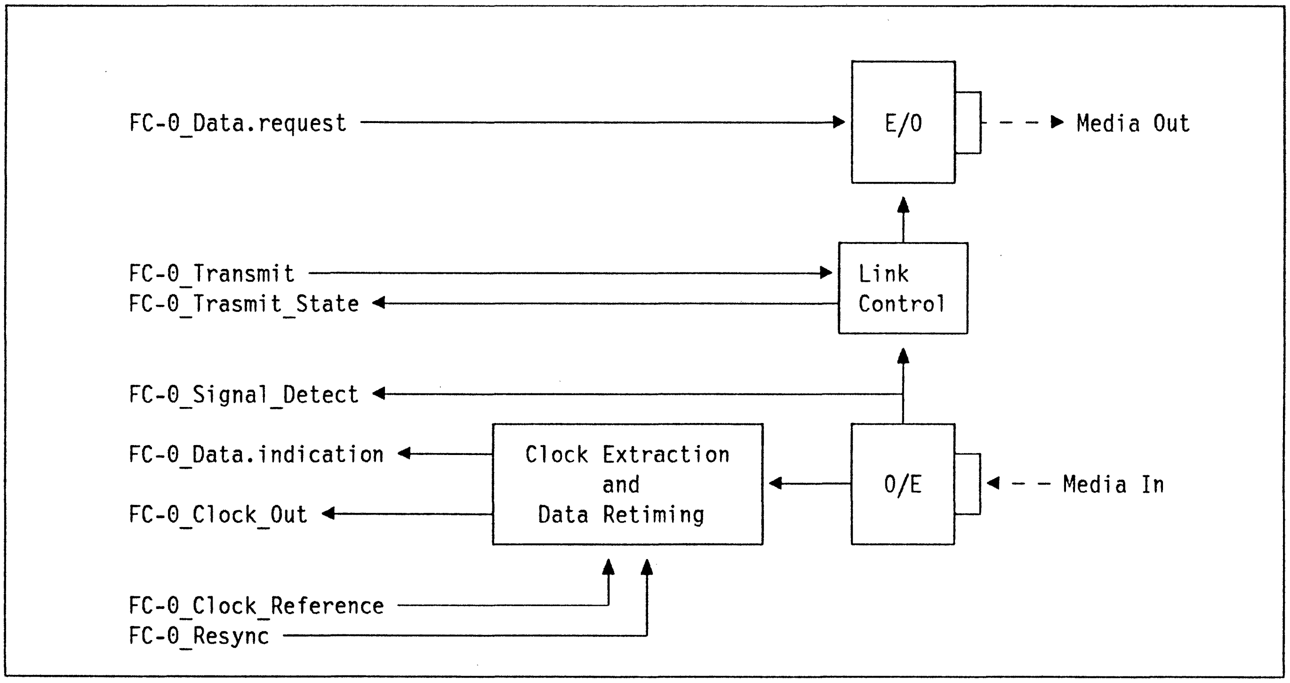


Figure 76. FC-0 logical structure

- **Transition Between Not-Enabled and Enabled States**: The sequence of events required for the transition between the not-enabled and enabled states will be media dependant, both as to the time period required and the optical or logical activity on the media interface.
- **Transmitter Failure State**: Some types of transmitters are capable of monitoring themselves for internal failures. Examples are laser transmitters where the monitor diode current is may be compared against a reference to determine proper operating point. Other transmitters, such as Light Emitting Diodes and coaxial transmitters do not typically have this capability. If the transmitter is capable of performing this monitoring function then a detection of a failure shall cause entry into the failure state.

G.2.2 Receiver States

The function of the receiver interface is to convert the incoming data from the form required by the communications media employed, retime the data, and present the data and an associated clock to the FC-1 level.

The receiver has no states.

G.3 FC-1 Services

Figure 77 graphically portrays the Open Systems Interface (OSI) model as applied to the FC-0 service interface.

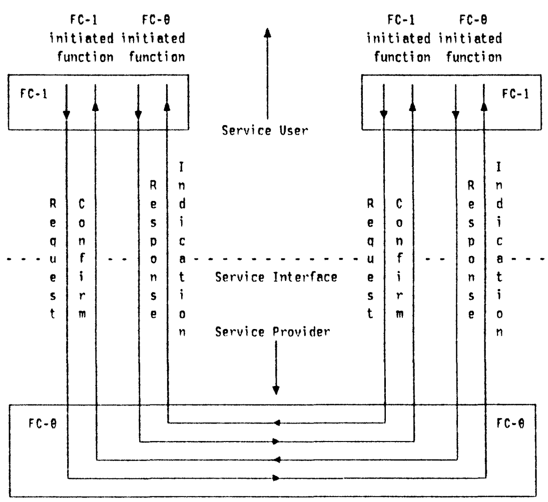


Figure 77. FC-0 service interface

Primitives are of four generic types:

Request The request primitive is passed from the FC-1 level to the FC-0 level to request that a service be initiated.

Indication The indication primitive is passed from the FC-0 level to the FC-1 level to indicate an internal FC-0 event which is significant to the FC-1 level. This event may be logically related to a remote service request, or may be caused by an event internal to the FC-0 level.

Response The response primitive is passed from the FC-1 level to the FC-0 level to complete a procedure previously invoked by an indication primitive.

Confirm The confirm primitive is passed from the FC-0 level to the FC-1 level to convey the results of one or more associated previous service request(s).

Figure 78 illustrates the primitive combinations that are supported by the FC-0 service interface. Each primitive described is this section if the standard is correlated to one of the subfigures (a), (b), (c), or (d) of Figure 78. This figure also indicates the logical relationships of the primitive types. Primitive types that occur earlier in time and are connected by dotted lines in the diagrams are the logical antecedents of subsequent primitive types.

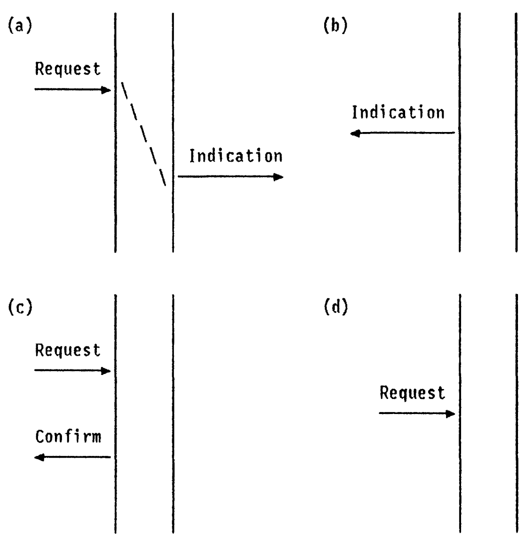


Figure 78. FC-0 Service Interface Primitive Combinations

The following service primitives are defined by the FC-0 service interface:

- FC-0_Transmit
  - request
- FC-0_Transmit_State
  - indication
- FC-0_Data
  - request
  - Indication
- FC-0_Clock_Out
  - indication
- FC-0_Clock_Reference
  - request
- FC-0_Resync
  - request
- FC-0_Signal_Detect
  - indication

G.4 FC-0_transmit

G.4.1 FC-0_transmit.request

G.4.1.1 Function

This primitive is used by the FC-1 level to control the operational state of the FC-0 transmitter.

G.4.1.2 Semantics

The FC-0_transmit.request service primitive is defined as follows:

FC-0_transmit.request(state)

state The requested state of the FC-0 transmitter. The allowable states of this parameter are enable, and disable. The format of this parameter is not defined by the standard.

G.4.1.3 When Generated

The FC-1 level issues the FC-0_transmit.request with state = enable to request that the transmitter begins its transition to an enabled state. During this time the FC-1 level must be supplying valid encoded information to the FC-0 level via the FC-0_data.request primitive.

The FC-1 level issues the FC-0_transmit.request with state = disable to request that the transmitter begin a transition to a not-enabled state.

G.4.1.4 Effect on Receipt

The receipt of this primitive with state = enable causes the FC-0 level to attempt to go to the enabled state. If the FC-0 level is in the Not-Enabled State the Transition state is entered. If the FC-0 level is in the Transition State the state is not affected. If the FC-0 level is in the Failure State no action is taken.

The receipt of this primitive with state = disable causes the FC-0 level to go to the not-enabled state from all states except the failure state. If the FC-0 level is in the failure state no action is taken.

G.4.1.5 Additional Comments

It is anticipated that for FC-0 media that do not have safety requirements the transition state will be a null state. For these cases the FC-0 level will go directly from the Not-Enabled state to the Enabled State.

When this primitive is issued with state=enable the FC-1 level must issue FC-0_data.request primitives continuously untill this primitive is again issued with state=disable.

This primitive is illustrated by Figure 78 (d).

G.5 FC-0_transmit_state

G.5.1 FC-0_transmit_state.indication

G.5.1.1 Function

This primitive is used by the FC-0 level to indicate to the FC-1 level that the transmitter is active and capable of transmitting data on the media interface.

G.5.1.2 Semantics

The FC-0_transmit_state.indication service primitive is defines as follows:

FC-0_transmit_state.indication(status)

status The state of the FC-0 transmitter. The allowable states of this parameter are active, inactive, and fault. The format of this parameter is not defined by the standard.

G.5.1.3 When Generated

The FC-0 level issues the FC-0_transmit_state.indication primitive with status = active continuously while the transmitter is in the enabled state and is capable of transmitting data on the media interface.

The FC-0 level issues the FC-0_transmit_state.indication primitive with status=fault when the transmitter is in an internal fault state as determimed by any internal fault monitoring which it may possess. If the transmitter contains no internal fault detection circuits this is a null state.

The FC-0 level issues the FC-0_transmit_state.indication primitive with status = inactive continuously when the transmitter is not in the enabled or failure states. This will occur when the transmitter is in any of the following states:

— Transmitter not-enabled;

— Transmitter in transition between enabled and not-enabled states in either direction; or

— Transmitter not operational due to an internal fault which is not detected by internal fault monitoring circuitry.

G.5.1.4 Effect on Receipt

Upon receipt of this primitive with the status = active the FC-1 level may assume that any issued FC-0_data.request primitives will be placed on the communications media.

Upon receipt of this primitive with the status = inactive the FC-1 level will be informed that the response of the FC-0 level to any FC-0_data.request primitives is undefined.

Upon receipt of this primitive with the status = fault the FC-1 level will be informed that a repair activity is required.

G.5.1.5 Additional Comments

This primitive is illustrated by Figure 78 (b).

G.6 FC-0_data

G.6.1 FC-0_data.request

G.6.1.1 Function

This primitive is used by the FC-1 level to pass transmission requests to an associated FC-0 level.

G.6.1.2 Semantics

The FC-0_data.request service primitive is defined as follows:

FC-0_data.request(bit)

bit The information which is to be transmitted on the media. The allowable values of this parameter are one and zero.

G.6.1.3 When Generated

This primitive is generated by the FC-1 level. It is the serialized form of the data to be transmitted as formatted and coded by the FC-1 level.

G.6.1.4 Effect on Receipt

Upon receipt of this primitive the FC-0 level will change the communications media to the state appropriate to the parameter value, provided the FC-0 tramsmitter is in the enabled state as indicated by the FC-0_transmit_state.indication primitive with status = active. The response to this parameter will be undefined if the FC-0 level is not in the enabled state as indicated by the FC-0_transmit_state.indication primitive with status = active.

bit = one shall be represented by a optical high power state in the case of optical media and a logical one in the case of coaxial media. bit = zero shall be represented by an optical low power state in the case of optical media and a logical zero in the case of coaxial media.

G.6.1.5 Additional Comments

This primitive shall be presented at a rate appropriate to the media in use. It must be presented continuously when FC-1 level has requested the transmitter to be in the enabled state by issuing the FC-0_transmit.request primitive with state=enable. Issuing of this primitive is optional when the FC-0_transmit.request primitive has been issued with state=disable.

This primitive is illustrated by the Request portion of Figure 78 (a).

G.6.2 FC-0_data.indication

G.6.2.1 Function

This primitive is used by the FC-0 level to pass data received from the media to the associated FC-1 level.

G.6.2.2 Semantics

The FC-0_data.indication service primitive is defined as follows:

FC-0_data.indication(bit)

bit The information which was received from the media. The allowable values of this parameter are one and zero.

G.6.2.3 When Generated

This primitive is generated synchronously with the FC-0_clock_out primitive. When the FC-0_clock_out.indication is issued the FC-1 level may receive the FC-0_data.indication primitive. When the FC-0_clock_out.indication is not present the value of the FC-0_data.indication is not guaranteed and the FC-1 level is advised to not receive the FC-0_data.indication primitive.

The parameter value of bit = one is generated by a high level of optical power in the case of optical media or a logical one in the case of coaxial media.

The parameter value of bit = zero is generated by a low level of optical power in the case of an optical media or a logical zero in the case of coaxial media.

G.6.2.4 Effect on Receipt

The receipt of an FC-0_data.indication primitive by the FC-1 level allows the FC-1 level to receive data.

G.6.2.5 Additional Comments

In the event that there is no input to the FC-0 from the media this output is undefined.

This primitive is illustrated by the Indication portion of Figure 78 (a).

G.7 FC-0_clock_out

G.7.1 FC-0_clock_out.indication

G.7.1.1 Function

This primitive is used by the FC-0 level to provide synchronisation information for the FC-0_data.indication to the FC-1 level.

G.7.1.2 Semantics

The FC-0_clock_out.indication service primitive is defined as follows:

FC-0_clock_out.indication

No parameters are defined for this primitive.

G.7.1.3 When Generated

An FC-0 level issues the FC-0_clock_out.indication primitive to provide a data transmission synchronisation signal to the FC-1 level. When this primitive is issued the FC-0_data.indication primitive will be in a valid state. At this time the FC-1 level may choose to receive FC-0_data.indication. At times between issuing the FC-0_clock_out.indication the FC-0_data.indication may not be in a valid state and it is recommended that the FC-1 level not receive the FC-0_data.indication primitive at these times.

G.7.1.4 Effect on Receipt

Receipt of this primitive may be used by the FC-1 level to synchronize data transmission between the FC-0 and FC-1 levels.

G.7.1.5 Additional Comments

In the event that there is no input to the FC-0 from the media the output frequency and timing information is undefined.

This primitive is illustrated by Figure 78 (b).

G.8 FC-0_Clock_Reference

G.8.1 FC-0_clock_reference.request

G.8.1.1 Function

This primitive is issued by the FC-1 level as a aid in initializing the clock recovery functions of the FC-0 level.

G.8.1.2 Semantics

The FC-0_clock_reference.indication service primitive is defined as follows:

FC-0_clock_reference.indication(boundary)

boundary An indication of the transmission intervals of the reference clock. The allowable values of boundary are word and null.

G.8.1.3 When Generated

The FC-1 level issues this primitive with value boundary = word to indicate the reference times of the clock signal. This primitive has a value of boundary = null at intervals between these reference times. The FC-1 level must issue this primitive with boundary = word continuously at a rate appropriate to the physical implementation.

G.8.1.4 Effect on Receipt

This primitive is used in the FC-0 bit synchronisation to initialize the clock recovery to near the bit frequency if a phase lock loop (PLL) form of clock recovery is used.

G.8.1.5 Additional Comments

In the event that the FC-0 implements the clock recovery using a method (ex. SAW filters) which does not require a frequency reference this primitive will not be required.

This primitive is illustrated by Figure 78 (d).

G.9 FC-0_signal_detect (optional)

G.9.1 FC-0_signal_detect.indication

G.9.1.1 Function

This primitive is issued by the FC-0 level to inform the FC-1 level of the presence or absence of a signal on the communications media.

G.9.1.2 Semantics

The FC-0_signal_detect.indication service primitive is defined as follows:

FC-0_signal_detect.indication(state)

state The result of the signal detect circuit. The allowable values of this parameter are present and absent.

G.9.1.3 When Generated

The FC-0_signal_detect.indication will have state = present when the receiver has determined that the signal on the input is above a predetermined level. In the case of optical media the activation level will lie in a range whose upper bound is the minimum specified sensitivity of the receiver and whose lower bound is the lower of either the point where the bit error rate will exceed 10^{-2} or 7 dB below the minimum receiver sensitivity. If the signal is below this range the primitive will be issued with state = absent.

Implementation of a signal detect function is optional. In the event that signal detect is not implemented the FC-0 shall issue this primitive with state = present.

While there is no defined hysteresis for this primitive the primitive shall undergo a single transition between the values of the state parameter for any monotonic increase or decrease in the optical power. The optical power at the transition points shall lie within the previously defined bounds.

For optical links that employ a link control function, such as section Annex H, then this "Signal Detect" is replaced by the "Link Status" signal, because it provides the service primitive described in this section.

In the case of optical media the reaction time of this primitive to a change in the input optical power shall be less than 12 seconds.

In the case of coaxial media this primitive shall be issued with state = absent within 12 seconds of the occurence of the fault. Otherwise State=Present will be issued.

G.9.1.4 Effect on Receipt

The FC-1 level uses this primitive to determine if a receiver which has been in a loss-of-synchronisation state for more than one second should enter the loss-of-synchronisation-failure state or the loss-of-signal-failure state.

G.9.1.5 Additional Comments

This primitive is illustrated by Figure 78 (b).

G.10 FC-0_resync

G.10.1 FC-0_resync.request

G.10.1.1 Function

This primitive is used by the FC-1 level to request that the FC-0 attempt to acquire bit synchronisation.

Some bit synchronisation methods do not require an initialization procedure. In the event that one of these methods is employed by the FC-0 level this primitive is not required.

G.10.1.2 Semantics

The FC-0_resync.request service primitive is defined as follows:

FC-0_resync.request

No parameters are defined for the FC-0_resync.request primitive.

G.10.1.3 When Generated

This primitive is issued by the FC-1 level to request that the FC-0 level attempt to acquire bit synchronisation. This may occur as a result of an FC-1 reset condition.

G.10.1.4 Effect on Receipt

When this primitive is received the FC-0 level will begin any bit synchronisation procedure appropriate to the specific implementation. The method of acquiring bit synchronisation is not a subject of this standard.

The synchronisation procedure shall take less than 1 millisecond from the receipt of this primitive.

G.10.1.5 Additional Comments

The time required to achieve bit synchronisation is highly variable depending on the particular bit synchronisation method employed.

This primitive is illustrated by Figure 78 (d).

Annex H. The open fibre control interface (informative)

This annex is not part of the standard but is included for information purposes only.

The Open Fibre Control (OFC) safety interlock system was introduced in section 6.2.3, "The open fibre control safety system"; in addition, the four-state state machine and handshake timings required for compatibility were specified. The details of a recommended implementation of this OFC system are described in the first part of this annex. Much of the information presented in section 6.2.3 is repeated here for purposes of completeness. The second part of this annex presents a justification of the maximum power values and OFC timing values specified for the system.

H.1 OFC Interface Description

H.1.1 System overview

The OFC system functions as a safety interlock by detecting whenever the data link is disrupted (i.e. cut fibre or disconnected connector) and forcing the transceiver into a repetitive pulsing mode of operation with very low duty cycle. The link can return to normal data traffic only after the OFC system detects that the link has been repaired and the proper reconnection handshake has taken place between the two transceivers in the link.

H.1.1.1 Input lines

Input lines to the OFC system consist of a system clock and two independent loss-of-light (LOL) detector lines that indicate whether-or-not an optical signal is being received by the photodetector/receiver. At least one of the two LOL detectors is AC coupled to the photodetector/receiver and therefore sensitive only to modulated optical signals (see figure 21 in section 6.2.3).

H.1.1.2 Output lines

The output lines consist of two laser driver control lines and an optional link status line. The two laser driver control lines are of opposite polarity to prevent voltage control problems from accidentally activating the laser. They are also each independently capable of disabling the laser drive circuitry (via separate control paths). The link status line is used to signal the user system when the link is inactive due to a loss-of-light condition detected by the OFC system.

H.1.1.3 Additional control lines

In addition to the input and output lines, there is a power-on-reset line and two optional user system control lines that interface with the OFC system. The power-on-reset line is used to synchronize the counters and state machines in the system. The two user system control lines, link control and loop-back enable, interact with the OFC system by forcing the OFC to disable the laser drive circuitry and turn off the laser. Once the laser is turned off, only the OFC system can turn it back on by performing a link reconnection handshake.

H.1.1.4 Reconnection handshake timings

The repetitive pulsing and reconnection handshake are controlled by logic in two state machines contained in the OFC system. The two state machines are independent and identical. This redundancy found throughout the OFC system is required by the safety standards which state that the safety interlock system must remain functional during single fault conditions.

The OFC timings used during a reconnection attempt depend upon the maximum (worst case) power accessible from a transmitter receptacle port, the circuit components and technology, and the restrictions imposed by the laser safety standards. The timings chosen for this standard are used for both the 531 Mbit/s and 266 Mbit/s SW laser data links, however, they are based on the maximum launched power for the 531 Mbit/s link since it is larger than the 266 Mbit/s link. The following power and timing specifications were given in section 6.2.3:

1. Transmitter receptacle power

- maximum (worst case) = 2,3 dBm (1,7 mW)

2. Pulse repetition time, T, (Disconnect state)
   - maximum = 11,9 sec
   - minimum = 11,7 sec
3. Pulse duration, D1=D3=t, (Disconnect and Reconnect states)
   - maximum = 410 microsec
   - minimum = 400 microsec
4. D2 stop time (Stop state)
   - maximum = 1 260 microsec
   - minimum = 1 250 microsec

H.1.2 Block Diagram description of the OFC System

A block diagram of a module that implements the OFC system control is shown in Figure 79. The discussion which follows explains the function of each of the blocks.

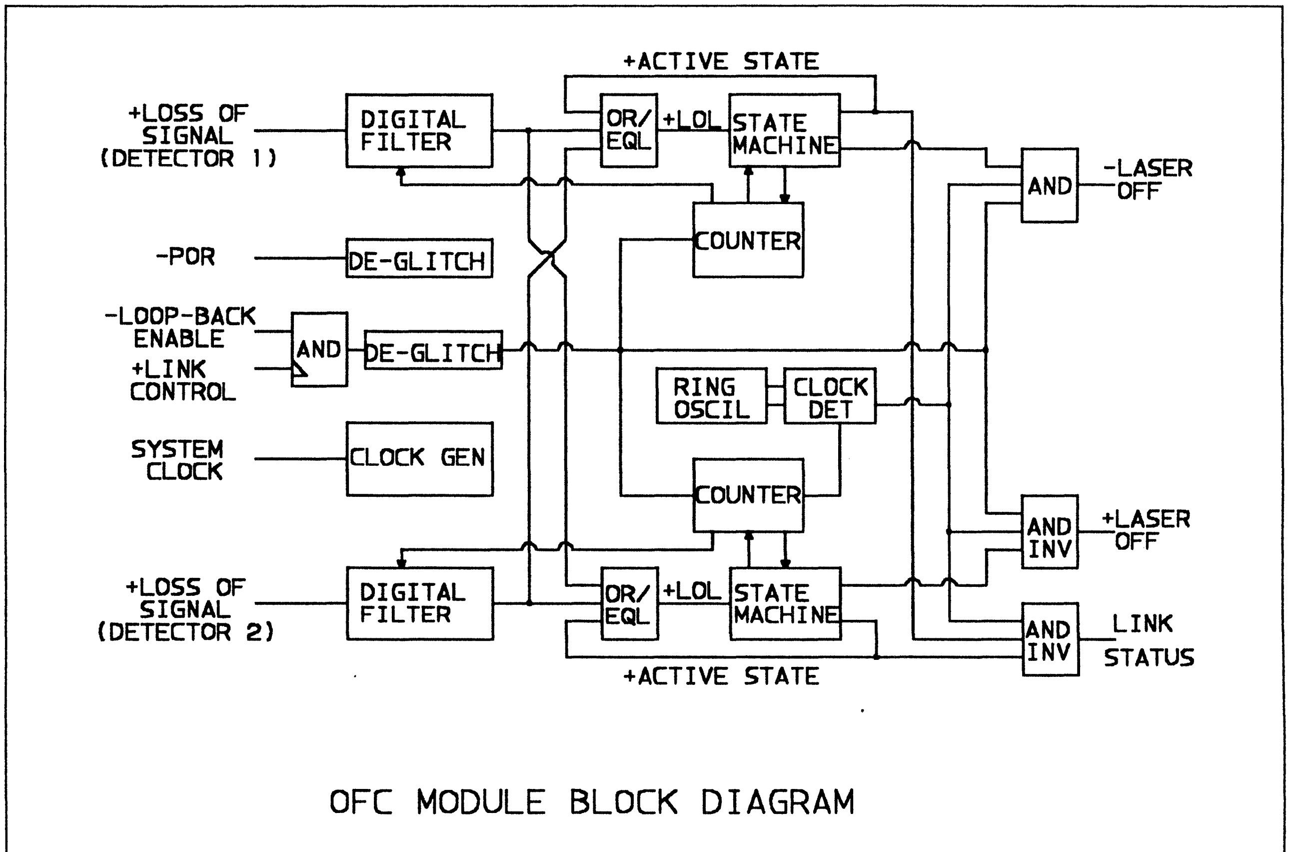


Figure 79. Possible implementation of the OFC system

The OFC module provides two control paths that must be satisfied before the laser can be activated. This provides the redundancy required by the laser safety standards. Each path has a digital filter, state machine and a counter. The internal redundancy is complemented externally, by two loss-of-light detectors and two control paths in the laser drive circuitry.

The two loss-of-light detectors each feed a digital filter, the output of each filter is 'OR/EQUAL'd to form an internal Loss-of-Light (LOL) signal. The two filter outputs must be 'EQUAL' during the Disconnect, Stop and Reconnect states to activate the laser and bring the link up to a normal operating mode. But once the system is in the Active state, the 'OR' is implemented so that only one light detector is required to bring the link down and deactivate the laser.

The digital filters integrate the incoming signals to improve their reliability. The filters sample at a faster rate when acquiring a light present signal and at a slower rate when dropping a light present signal (i.e. setting LOL high), while still maintaining the correct handshake timings.

The internal LOL signals are used to synchronize the counters and state machines. The state machines control the handshake algorithm implemented in the OFC system. Each state machine controls a 'Laser Off' output that connects to separate laser drive control circuits. The counters control the pulse repetition, pulse duration and stop times. The counters also provide the low frequency sampling clock to the digital filters.

The OFC module contains a ring oscillator. The ring oscillator drives a clock detector that monitors the 'System Clock' coming into the module. If the 'System Clock' gets stuck high or low the clock detector will cause the laser to be deactivated. This provides a back up safety feature to the single clock coming into the module. The 'System Clock' only has to slow down sufficiently for the clock detector to detect a fault. If the 'System Clock' speeds up, all of the timings scale proportionately so that the ratio of the pulse duration time to the pulse repetition time remains constant.

The response of the module to the external control signals, -Loop Back Enable, +Link Control and -POR, must not cause a pulse to be emitted when they are toggled. The logic must ensure that a safe time period exists between pulses. Hence any disruption of the OFC system must cause the state machine to be reset to the Disconnect state.

H.1.3 The OFC State Machine

The OFC state machine is the heart of the control system since it contains the logic that detects when the optical link becomes open due to a disconnection or break in the fibre and presides over the link reconnection handshake when it detects that the link is once again closed. A state diagram of the algorithm implemented in the OFC system is shown in Figure 80. A description of all the states and transitions follows.

The state machine has five variables that control the transitions from state to state. Loss of Light (LOL) is the "EQUAL" of the two light detector lines during the disconnect state, stop state and reconnect state, and the "OR" of the two light detector lines in the active state. In other words, both light detectors must agree to activate the link, but once activated either light detector sensing no light will cause the state machine and optical link back to return to the disconnect state. The "master of link reconnection" (MAS) flag is used to insure that if a transceiver responds to receiving an incoming light pulse while in the disconnect state, then it can only respond to receiving a light pulse in the reconnect state and it cannot initiate a reconnect state light pulse on its own. This prevents problems that could arise from timer variations and link synchronization. The three decodes are generated by the system clock and counter in the OFC system. The decodes are used to ensure that no ON-OFF-ON sequence generated by the physical insertion of a fibre into the connector can accidentally satisfy the safety algorithm.

The following list describes each of the four states of operation of the OFC system (refer to Figure 80 as needed):

1. Disconnect state:

   While in the disconnect state the OFC system on each shortwave laser transceiver is operating the laser at a very low duty cycle to conform to laser safety emission limits. It activates the laser driver circuitry for 405 usec every 11,8 sec to check for a closed optical link between itself and the transceiver at the other end of the link. As long as the LOL flag remains high, the OFC system keeps the transceiver in this state. To exit from the disconnect state, light must be both sent and received by the transceiver. This condition can be satisfied in two ways:

   A. When the 11,8 sec timer expires, decode D1 is set high (D1=1), the MAS flag is set high (MAS=1), and the laser is activated for the duration of the decode 1 period (405 usec). If during this decode period an optical signal is received from the transceiver at the other end of the link (i.e. LOL=0), then the OFC system proceeds to the stop state for the remainder of the decode 1 period. The MAS flag set high implies that this transceiver unit initiated the link reconnection check by sending light first and receiving light second; it is considered the "master" of the link reconnection.

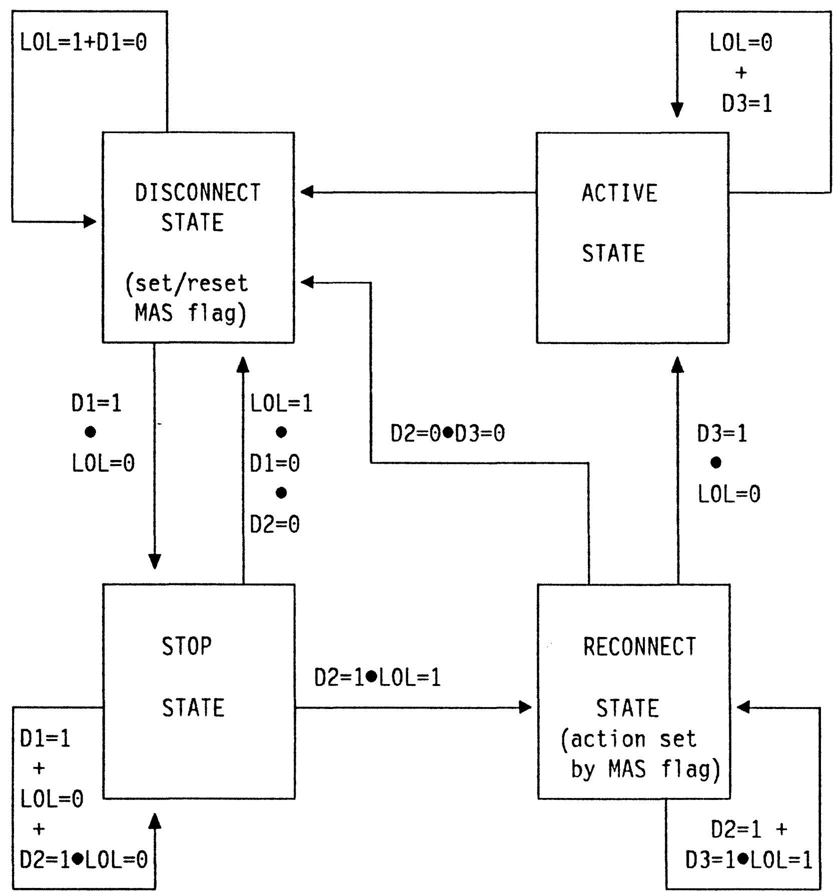
   B. If an optical signal is received (LOL=0) sometime during the 11,8 sec wait period, then the counters controlling the timing are reset, decode D1 is set high

(D1=1), the MAS flag is set low (MAS=0), and the laser is activated for the duration of the decode 1 period (405 usec). This also causes the OFC system to exit to the stop state. The MAS flag set low implies that this transceiver unit is responding to a link reconnection attempt from the unit at the other end of the link. By receiving light first and sending light second, it is considered the "slave" of the link reconnection attempt.

2. Stop state:

In the stop state the OFC system forces the laser off. However, the laser is not disabled until after the decode 1 (D1) time period is complete. This ensures that both lasers are on long enough to satisfy the disconnect state exit condition at each transceiver unit. When the decode 1 time period ends, D1 is reset low (D1=0), D2 is set high (D2=1) and the laser driver circuitry is disabled for the duration of the decode 2 time period (1255 usec). The OFC system remains in the stop state for as long as light is received (i.e. LOL=0); this could be for an indefinite period of time. There are two exit paths from the stop state:

A. One exit from the stop state is to continue on to the reconnect state. This occurs when light is no longer received



DEFINITIONS:
LOL = Loss of light flag
MAS = Master of link reconnection flag
D1 = Decode 1 - 1st time period flag (405 usec) = Link check
D2 = Decode 2 - 2nd time period flag (1255 usec) = Disable laser
D3 = Decode 3 - 3rd time period flag (405 usec) = Link check
(note: usec means microseconds)

Figure 80. Open Fibre Control module State Diagram

(LOL=1) and the decode 2 time period has not expired (D2=1). This is the normal chain of events during a reconnect handshake.

B. The other exit from the stop state takes the OFC system back to the disconnect state. This happens when the loss-of-light condition (LOL=1) occurs after the decode 2 time period has expired (i.e. D1=0 and D2=0). Since an optical signal remained present throughout the decode 2 time period (1255 usec), the OFC system assumes that the other end of the link does not have a compatible safety control system and therefore rejects the link connection attempt.

3. Reconnect state:

In the reconnect state each of the transceiver units must once again send and receive an optical signal to proceed to the active state. This verifies that a closed and safe link exists between the two transceivers. The function of the OFC system is different for the master (MAS=1) and slave (MAS=0) in this state. If a transceiver unit responded to an optical signal in the disconnect state (i.e. MAS=0), it is important that it also responds in the reconnect state and does not attempt to initiate the send/receive exchange. When the OFC system enters the reconnect state, decode 2 is high (D2=1) and the laser is disabled. The OFC system continues to keep the laser disabled until the decode 2 time period expires, then one of the following sequences of events occur:

A. If the transceiver is master of the connection attempt (i.e. MAS=1), then decode 3 is set high (D3=1) and the laser is activated for the duration of the decode 3 time period (405 usec). If during the decode 3 time period an optical signal is received in response to this initiating light pulse (i.e. D3=1 and LOL=0), then the OFC system exits to the active state. Otherwise, when the decode 3 time period ends (D2=0 and D3=0), the OFC system disables the laser and exits to the disconnect state.

B. If the transceiver is slave of the connection attempt (i.e. MAS=0), then decode 3 is set high (D3=1) but the laser is not activated. If during the decode 3 time period (405 usec) an initiating optical signal is received from the master (i.e. D3=1 and LOL=0), then the OFC system activates the laser in order to send a response and exits to the active state. Otherwise, when the decode 3 time period ends (D2=0 and D3=0), the OFC system keeps the laser disabled and exits to the disconnect state.

4. Active state:

In the active state the OFC system allows the laser to function continuously and monitors the two loss-of-light detectors. The OFC system will remain in the active state for as long as an optical signal is received by both of the detectors (i.e. LOL=0). If either of the detectors sense a loss-of-light condition, the laser is disabled, the loss-of-light flag is set high (LOL=1) and the OFC system exits to the disconnect state.

H.2 Laser Safety Standards and OFC Timing Specifications

H.2.1 Laser Safety Standards

Although there is a considerable amount of similarity between the laser safety standards and regulations that exist throughout the world, some requirements differ, especially with respect to labeling and certification. Within the U.S., all laser products must be certified by the manufacturer to conform to the requirements contained in the F.D.A. regulation 21 CFR subchapter J. In addition, it may be a business requirement to conform to the Z136.2 laser safety standard produced by the American National Standards Institute (ANSI). Outside of the U.S., many countries base their laser safety regulations on the International Electrotechnical Commission (IEC) 825 laser safety standard. The time values used for the OFC system in this Fibre Channel standard are based on the emission requirements for Class 1 laser products contained in the IEC 825 standard (1984 plus amendment 1, 1990). The reason this standard was used over the other two is that the IEC emission requirements for a Class 1 system are more restrictive and the goal was to specify an OFC system interface which satisfies worldwide Class 1 emission requirements.

H.2.2 The OFC Timing Specifications

An optical fibre transmission system is a closed system (i.e. no accessible laser emissions) during normal link operating conditions and therefore a Class 1 system. It is only during maintenance and service conditions when the optical path has been accidentally or purposefully broken that access to laser emissions is possible. The point in the transmission system that will always have the largest emission level is at the transmitter receptacle of a transceiver. Classification is therefore based on the maximum emission level at the transmitter receptacle. This maximum value is a worst case value and includes variations due to temperature effects, lifetime effects and single faults.

The OFC system makes use of a repetitive pulsing technique (t microsec on every T sec) during the time that a link is open (i.e. the disconnect state) in order to reduce the maximum possible exposure to a value which is below the level set by the safety standard IEC 825 for Class 1 operation. The maximum power level per pulse is a function of the wavelength, pulse duration (t), and pulse repetition frequency (PRF = 1/T) which determines the number of pulses (N) that occur during the time base used for classification. Note that the use of the word "pulse" refers to the time, t, during which the laser is powered on and being modulated with a valid full rate data pattern. From a laser safety point of view, this "pulse" can be thought of as a CW pulse of duration t and power level equal to the average power of the modulated signal.

The OFC system described above contains a natural potential for a two pulse emission every T seconds when only one of the two linking fibres is disconnected. This condition will occur **if** the optical transmission path from transceiver A to B is open, but the path from B to A is closed, **and** the T sec timer on transceiver A is sufficiently faster than the T sec timer on transceiver B. Transmitter A will emit a t usec pulse when its timer expires (i.e. D1=1) in an attempt to link up as master. This will fail since the link is open. A second t usec pulse will be emitted by transmitter A in response to transmitter B sending a pulse to receiver A along the closed part of the link. Since the timer on transceiver A is faster than that on B, this pattern will be repeated until the link is repaired. Hence the worst case scenario is one that includes two (t usec) pulses every T seconds; this must be taken into consideration in any safety calculations.

The accessible emission limit (AEL) is defined as the maximum accessible laser emission level for a particular classification. For the repetitively pulsed situation found in the OFC system, the AEL for the pulse train is

$$AEL_{train} = AEL_{single} \times N^{-0,25}$$

where:

AEL_{train}

= exposure from any single pulse in the train

AEL_{single}

= AEL for a single pulse

N

= number of pulses during the applicable time base

The Class 1 AEL for a single pulse in the wavelength range 700 to 1050 nm and emission duration, t, between 1.8×10^{-5} and 1 000 sec is given by the equations:

$$AEL_{single} = 7 \times 10^{-4}\, t^{0,75}\, C_4\ J$$
$$= 0.7\, t^{-0,25}\, C_4\ mW$$

and

$$C_4 = 10^{(\lambda - 700)/500}$$
$$= 1,38 \quad (\lambda = 770\, nm)$$

The change from energy units to power units was used since the power level is essentially constant during the emission duration. For a maximum pulse duration of t = 410 usec, the single pulse AEL is

$$AEL_{single} = 6,79\ mW \quad (t = 410\, \mu\, \text{sec}).$$

For wavelengths greater than 400 nm and situations where intentional viewing is not inherent in the design of the product, the time base is 1000 sec. Thus the worst case number of pulses during the time base is

$$N = \left(\frac{1000}{T}\right) \times 2 = 171\ pulses \quad (T = 11,7\ \text{sec})$$

where T is the minimum pulse repetition time (T = 11,7 sec) and the two pulse potential has been included. The worst case (i.e. smallest) AEL for the train of pulses can now be calculated and is found to be

$$AEL_{train} = 6{,}79 \times (171)^{-0{,}25} = 1{,}88\ mW$$

Comparing this value to the worst case (i.e. maximum) transmitter receptacle power, 1,70 mW, we find that the specified maximum value is less than the AEL for the train of pulses and allows for a 10% guard band. Thus, a point-to-point optical fibre transmission system that implements the OFC system as described in this standard should be classifiable as a Class 1 laser system with respect to the IEC 825 standard (1984 plus amendment 1, 1990).

Annex I. Raw Data Mode

In order to provide complete diagnostic capability within a Port implementation, a Port may choose implement an optional means by which an Upper Level Protocol (FC-4) may transmit raw 10B data. This raw data could include frames, Primitive Signals, Idles, or Primitive Sequences.

In order to test a Port's error detection logic, it maybe helpful to transmit code violations, invalid ordered sets, and other alterations of the encoded bit stream. In order to avoid damage to FC-0 components, FC-0 shall define a set of rules which limits the deviation allowed from a set of minimum operating characteristics.

This list of rules is to be defined and contained in this appendix for guidelines to an implementation which desires implementing such an option.

Annex J. FC-1 Service interface (informative)

FC-1 presents a decoded-Transmission-Word-oriented service interface to FC-2. This interface uses the same primitive types as those defined by the OSI model.

The FC-1 service interface provides a conceptual view of FC-1 function from the perspective of the FC-2 level and does not restrict FCS implementation flexibility. The following figure graphically portrays the Fiber Channel conceptual model as applied to the FC-1 service interface:

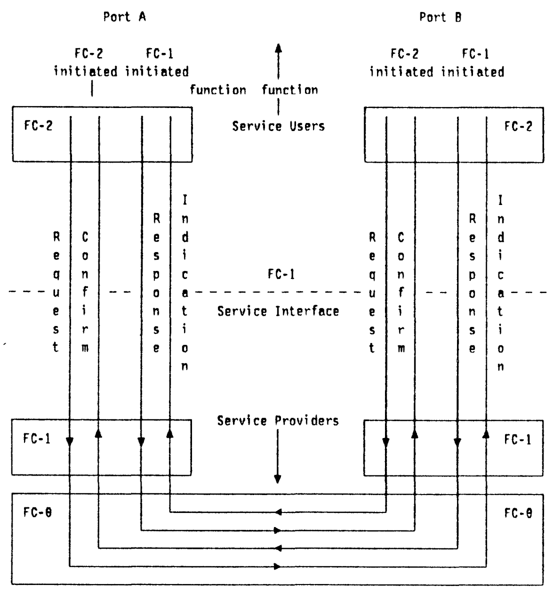


Figure 81. FC-1 Service Interface

Primitives are of four generic types:

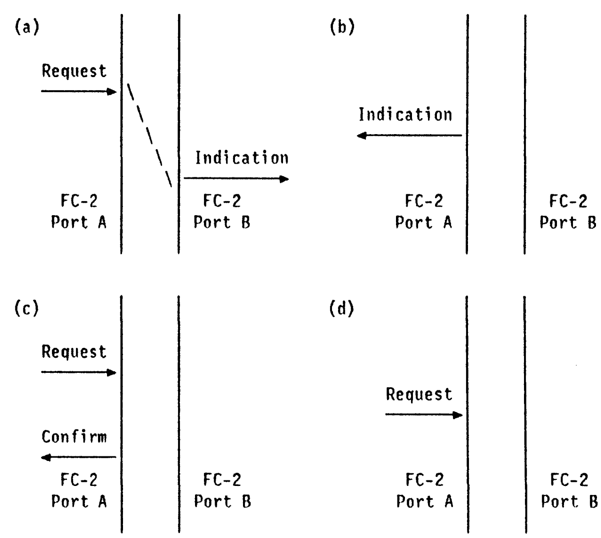
Request The request primitive is passed from the FC-2 level to the FC-1 level to request that a service be initiated.

Indication The indication primitive is passed from the FC-1 level to the FC-2 level to indicate an internal FC-1 event which is significant to the FC-2 level. This event may be logically related to a remote service request, or may be caused by an event internal to the FC-1 level.

Response The response primitive is passed from the FC-2 level to the FC-1 level to complete a procedure previously invoked by an indication primitive.

Confirm The confirm primitive is passed from the FC-1 level to the FC-2 level to convey the results of one or more associated previous service request(s).

Figure 82 illustrates the primitive combinations that are supported by the FC-1 service interface. Each primitive described in this section of the standard is correlated to one of the subfigures (a), (b), (c), or (d) of figure 82. This figure also indicates the logical relationship of the primitive types. Primitive types that occur earlier in time and are connected by dotted lines in the diagrams are the logical antecedents of subsequent primitive types.



NOTE - The Response primitive is not used by the FC-1 service interface.

Figure 82. FC-1 Service Interface Primitive Combinations

The following service primitives are defined by the FC-1 service interface:

- FC-1_Receiver_State
  - Indication
- FC-1_Transmitter_State
  - Indication
- FC-1_Data_Path_Width *
  - Request
  - Confirm

- FC-1_Transmitter_Clock
  - Indication
- FC-1_Data
  - Request
  - Indication
- FC-1_Receiver_Control
  - Request
- FC-1_Transmitter_Control
  - Request
- FC-1_Loopback_Control
  - Request
  - Confirm

*: optional primitive

J.1 FC-1_Receiver_State

J.1.1 FC-1_Receiver_State.Indication

J.1.1.1 Function

This primitive is used by the FC-1 level to indicate the state of its Receiver to an associated FC-2 level.

J.1.1.2 Semantics

The FC-1_Receiver_State.Indication service primitive is defined as follows:

FC-1_Receiver_State.Indication(State)

State The functional state of the FC-1 level's Receiver. The allowable values of this parameter are Loss_Of_Signal_Failure, Loss_Of_Synchronization_Failure, Loss_Of_Synchronization, Synchronization_Acquired, Offline, and Reset. The format of this parameter is not defined by the standard.

J.1.1.3 When generated

An FC-1 level issues the FC-1_Receiver_State.Indication primitive with State = Loss_Of_Signal_Failure to indicate to the FC-2 level that its Receiver has been unable to achieve Synchronization (i.e., has been in the Loss-Of-Synchronization state for an extended period of time as defined by FC-0) and is currently not receiving a signal from its attached Fibre. This state may be entered from the Loss-Of-Synchronization and Loss-Of-Synchronization-Failure states. While in the Loss-Of-Signal-Failure state, the FC-1 Receiver is considered Not Operational and does not provide FC-1_Data.Indication primitives to the FC-2 level.

An FC-1 level issues the FC-1_Receiver_State.Indication primitive with State = Loss_Of_Synchronization_Failure to indicate to the FC-2 level that its Receiver has been unable to achieve Synchronization (i.e., has been in the Loss-Of-Synchronization state for an extended period of time as defined by FC-0) but is currently receiving a signal from its attached Fibre. (See 11.1.2.2 for a description of Synchronization.) This state may be entered from the Loss-Of-Synchronization and Loss-Of-Signal-Failure states. While in the Loss-Of-Synchronization-Failure state, the FC-1 Receiver is considered Not Operational and does not provide FC-1_Data.Indication primitives to the FC-2 level.

An FC-1 level issues the FC-1_Receiver_State.Indication primitive with State = Loss_Of_Synchronization to indicate to the FC-2 level that its Receiver has lost Synchronization on the signal received from its attached Fibre. (See 11.1.2.2 for a description of Synchronization.) This state may be entered from the Synchronization-Acquired or Reset states. While in the Loss-Of-Synchronization state, the FC-1 Receiver is considered Operational but does not provide FC-1_Data.Indication primitives to the FC-2 level.

An FC-1 level issues the FC-1_Receiver_State.Indication primitive with State = Synchronization_Acquired to indicate to the FC-2 level that its Receiver has achieved Synchronization on the signal received from its attached Fibre. (See 11.1.2.2 for a description of Synchronization.) This state may be entered from the Loss-Of-Synchronization or Offline states. While in the Synchronization-Acquired state, the FC-1 Receiver is considered Operational and provides FC-1_Data.Indication primitives to the FC-2 level according to the rules described in J.5.

An FC-1 level issues the FC-1_Receiver_State.Indication primitive with State = Offline to indicate to the FC-2 level that it has received and processed an FC-1_Receiver_Control primitive from FC-2 with

Control = Enter_Offline while in the Synchronization-Acquired state or that it has completed a previously requested reset and is returning to the Offline state. This state may be entered from the Synchronization-Acquired or Reset states. While in the Offline state, the FC-1 Receiver is considered Operational and provides FC-1_Data.Indication primitives to the FC-2 level according to the rules described in J.5. The Receiver continues to process received Transmission Words under the rules of the Loss-of-Synchronization procedure specified in "Invalid Transmission Word rules," but completion of the procedure does not cause the Receiver to enter the Loss-Of-Synchronization state.

An FC-1 level issues the FC-1_Receiver_State.Indication primitive with State = Reset to indicate to the FC-2 level that it has initiated a reset condition at its Receiver. This state may be entered from any other state. While in the Reset state, the FC-1 Receiver is considered Not Operational and does not provide FC-1_Data.Indication primitives to the FC-2 level.

J.1.1.4 Effect on receipt

Receipt of this primitive with State = Synchronization_Acquired causes the FC-2 level to prepare to accept FC-1_Data.Indication primitives from the FC-1 level. Receipt of this primitive with State = Loss_Of_Synchronization notifies the FC-2 level that FC-1_Data.Indication primitives will not be forthcoming from the FC-1 level. Receipt of this primitive with State = Loss_Of_Synchronization_Failure or Loss_Of_Signal_Failure notifies the FC-2 level that the FC-1 level is incapable of reception and may cause the FC-2 level to initiate maintenance activity. Receipt of this primitive with State = Offline notifies the FC-2 level that the FC-1 level has processed the FC-2 request to enter the Offline state or has completed the previously requested reset and presents FC-1_Data.Indication primitives to the FC-2 level. It also continues to process received Transmission Words under the rules of the Loss-of-Synchronization procedure specified in "Invalid Transmission Word rules," but completion of the procedure does not cause the Receiver to enter the Loss-Of-Synchronization state. Receipt of this primitive with State = Reset notifies the FC-2 level that the FC-1 level is incapable of reception because it has had a Receiver reset condition imposed upon it, either internally or externally.

J.1.1.5 Additional comments

The Loss-Of-Synchronization state is the default state of an FC-1 level's Receiver upon completion of Initialization.

This primitive is illustrated by figure 82 (b).

J.2 FC-1_Transmitter_State

J.2.1 FC-1_Transmitter_State.Indication

J.2.1.1 Function

This primitive is used by the FC-1 level to indicate the state of its Transmitter to an associated FC-2 level.

J.2.1.2 Semantics

The FC-1_Transmitter_State.Indication service primitive is defined as follows:

FC-1_Transmitter_State.Indication(State)

State The functional state of the FC-1 level's Transmitter. The allowable values of this parameter are Failure, Not_Enabled, and Working. The format of this parameter is not defined by the standard.

J.2.1.3 When generated

An FC-1 level issues the FC-1_Transmitter_State.Indication primitive with State = Failure to indicate to the FC-2 level that its Transmitter has entered the Failure state as described by 11.3.4. This state may be entered from the Working or Not-Enabled states. While in the Failure state, the FC-1 Transmitter is considered Not Operational, does not accept FC-1_Data.Request primitives from the FC-2 level, and does not provide FC-1_Transmitter_Clock.Indication primitives to the FC-2 level.

An FC-1 level issues the FC-1_Transmitter_State.Indication primitive with State = Not_Enabled to indicate to the FC-2 level that its Transmitter is Operational but is prevented from transmitting a signal onto its associated Fibre. This state may be entered from the Working state. It may result from the processing of an

FC-1_Transmitter_Control.Request primitive or from detection by FC-0 laser safety procedures of a laser safety condition (i.e., a condition which requires that the Transmitter cease transmission). While in the Not-Enabled state, the FC-1 Transmitter is considered Operational and accepts FC-1_Data.Request primitives from the FC-2 level in synchronization with the FC-1_Transmitter_Clock.Indication primitives provided by it to the FC-2 level. However, no signals resulting from the accepted FC-1_Data.Request primitives are transmitted onto its associated Fibre. If the FC-1 level is in Loopback mode, the encoded bit stream resulting from the accepted FC-1_Data.Request primitives is provided to the FC-1 Receiver.

An FC-1 level issues the FC-1_Transmitter_State.Indication primitive with State = Working to indicate to the FC-2 level that its Transmitter is operating normally according to the error-monitoring procedure described by 11.3.4. This state may be entered from the Not-Enabled state. While in the Working state, the FC-1 Transmitter is considered Operational and accepts FC-1_Data.Request primitives from the FC-2 level in synchronization with the FC-1_Transmitter_Clock.Indication primitives provided by it to the FC-2 level. Signals resulting from the accepted FC-1_Data.Request primitives are transmitted onto its associated Fibre. If the FC-1 level is in Loopback mode, the encoded bit stream resulting from the accepted FC-1_Data.Request primitives are also provided to the FC-1 Receiver.

J.2.1.4 Effect on receipt

Receipt of this primitive with State = Working causes the FC-2 level to continue issuance of FC-1_Data.Request primitives to the FC-1 level (synchronous to FC-1_Transmitter_Clock.Indication primitives from the FC-1 level). Receipt of this primitive with State = Not_Enabled causes the FC-2 level to begin or continue issuance of FC-1_Data.Request primitives to the FC-1 level (synchronous to FC-1_Transmitter_Clock.Indication primitives from the FC-1 level). No transmitted signals result from the receipt of FC-1_Data.Request primitives. The Transmitter remains Operational. Receipt of this primitive with State = Failure notifies the FC-2 level that FC-1_Transmitter_Clock.Indication primitives will not be forthcoming from the FC-1 level and that FC-1_Data.Request primitives will not be accepted by the FC-1 level. The FC-2 level may initiate maintenance activity.

J.2.1.5 Additional comments

The Not-Enabled state is the default state of an FC-1 level's Transmitter upon completion of Initialization.

This primitive is illustrated by figure 82 (b).

J.3 FC-1_Data_Path_Width

J.3.1 FC-1_Data_Path_Width.Request

J.3.1.1 Function

This primitive is used to establish the path width of a variable-width data interface between an FC-1 and an FC-2 level.

J.3.1.2 Semantics

The FC-1_Data_Path_Width.Request service primitive is defined as follows:

FC-1_Data_Path_Width.Request(Width)

Width The data path width of the interface between the FC-1 and FC-2 levels. The allowable values of this parameter are Byte, Half_Word, and Word. The format of this parameter is not defined by the standard.

J.3.1.3 When generated

An FC-2 level issues the FC-1_Data_Path_Width.Request primitive to specify to the FC-1 level the width of the data path to be used (Byte, Half_Word, or Word) between the FC-1 and FC-2 levels.

J.3.1.4 Effect on receipt

Receipt of the FC-1_Data_Path_Width.Request primitive causes an FC-1 level to attempt to establish the requested path width for its data interface to FC-2.

J.3.1.5 Additional comments

This primitive is not required for those implementations which use an FC-1 level with a fixed data path width. For those implementations which use an FC-1 level with a variable data path width, the attached FC-2 level issues this primitive to indicate the preferred path width.

An FC-1_Data_Path_Width.Request primitive is accepted by an FC-1 level or a fixed data path width is defined before FC-1_Data and FC-1_Transmitter_Clock primitives can be issued.

The Half_Word value of Width corresponds to a 2-byte interface. The Word value of Width corresponds to a 4-byte interface.

This primitive is illustrated by the request portion of figure 82 (c).

J.3.2 FC-1_Data_Path_Width.Confirm

J.3.2.1 Function

This primitive is used to accept or reject the establishment of the path width of a variable-width data interface between an FC-1 and an FC-2 level.

J.3.2.2 Semantics

The FC-1_Data_Path_Width.Confirm service primitive is defined as follows:

FC-1_Data_Path_Width.Confirm(Status)

Status The results of the FC-1_Data_Path_Width.Request primitive. The allowable values of this parameter are Accept and Reject. The format of this parameter is not defined by the standard.

J.3.2.3 When generated

An FC-1 level issues the FC-1_Data_Path_Width.Confirm primitive to an FC-2 level to indicate the acceptance or rejection of a previously issued FC-1_Data_Path_Width.Request primitive.

J.3.2.4 Effect on receipt

Receipt of the FC-1_Data_Path_Width.Indication primitive with Status = Accept informs the FC-2 level that path width establishment was successful. Receipt of the FC-1_Data_Path_Width.Indication primitive with Status = Reject informs the FC-2 level that path width establishment was unsuccessful.

J.3.2.5 Additional comments

This primitive is not required for those implementations which use an FC-1 level with a fixed data path width. For those implementations which use an FC-1 level with a variable data path width, the attached FC-1 level must issue this primitive in response to an FC-1_Data_Path_Width.Request primitive.

This primitive is illustrated by the confirm portion of figure 82 (c).

J.4 FC-1_Transmitter_Clock

J.4.1 FC-1_Transmitter_Clock.Indication

J.4.1.1 Function

This primitive is used by the FC-1 level to provide data transmission synchronization information, including an indication of Transmission Word alignment boundaries, to the FC-2 level.

J.4.1.2 Semantics

The FC-1_Transmitter_Clock service primitive is defined as follows:

FC-1_Transmitter_Clock.Indication(Boundary)

Boundary An indication of the transmission boundary presented by the Transmitter clock. The allowable values of this parameter are Word and Null. The format of this parameter is not defined by the standard.

J.4.1.3 When generated

An FC-1 level with a Transmitter in the Working or Not-Enabled states issues the FC-1_Transmitter_Clock.Indication primitive to provide a data transmission synchronization signal to the FC-2 level. Synchronous to the receipt of an FC-1_Transmitter_Clock.Indication primitive, the FC-2 level may choose to issue an

FC-1_Data.Request primitive to the FC-1 level to present data to be transmitted (see J.5 for a description of the rules associated with data transmission).

The FC-1_Transmitter_Clock.Indication primitive is issued periodically to the FC-2 level with a frequency based upon the FC-0 transmission frequency and the data path width between the FC-1 and FC-2 levels. When a byte-wide data path is provided, the frequency of issuance = FC-0 transmission frequency / 10. When a half-word-wide data path is provided, the frequency of issuance = FC-0 transmission frequency / 20. When a word-wide data path is provided, the frequency of issuance = FC-0 transmission frequency / 40.

The Boundary parameter is used by the FC-1 level to indicate whether the transmission data presented on an FC-1_Data.Request primitive that is associated with the FC-1_Transmitter_Clock.Indication primitive may be of the type that requires a Transmission-Word boundary. When Boundary = Word, any information type (Valid_Data_Byte, Delimiter, or Primitive_Signal) may be presented on the Type1 parameter of the associated FC-1_Data.Request primitive. When Boundary = Null, only the type Valid_Data_Byte may be presented on the Type1 parameter of the associated FC-1_Data.Request primitive.

Transmission-Word alignment is established by an FC-1 Transmitter at the time it enters the working state and begins to transmit idle words as specified in J.5. The established Transmission-Word alignment is indicated by the FC-1 level to the FC-2 level through the FC-1_Transmitter_Clock.Indication primitive. When a byte-wide data path width has been specified, the FC-1 level indicates a Transmission-Word boundary on every fourth FC-1_Transmitter_Clock.Indication primitive. When a half-word-wide data path width has been specified, the FC-1 level indicates a Transmission-Word boundary on every second FC-1_Transmitter_Clock.Indication primitive. When a word-wide data path width has been specified, the FC-1 level indicates a Transmission-Word boundary on every FC-1_Transmitter_Clock.Indication primitive.

J.4.1.4 Effect on receipt

Receipt of this primitive informs an FC-2 level that a request for information transmission can occur via issuance of an FC-1_Data.Request primitive.

J.4.1.5 Additional comments

This primitive is illustrated by figure 82 (b).

J.5 FC-1_Data

J.5.1 FC-1_Data.Request

J.5.1.1 Function

This primitive is used by the FC-2 level to pass transmission requests to an associated FC-1 level.

J.5.1.2 Semantics

The FC-1_Data service primitive is defined as follows:

FC-1_Data.Request(Type1, Code1
Type2, Code2
Type3, Code3
Type4, Code4)

Type The type of transmission information that is to be encoded and transmitted by the FC-1 level. The allowable values of this parameter are Valid_Data_Byte, Non_Repeating_Ordered_Set, and Repeating_Ordered_Set. The format of this parameter is not defined by the standard.

When a Non-Repeating Ordered Set or Repeating Ordered Set is indicated, a single Type field (Type1) is provided regardless of data path width. When a Valid Data Byte is indicated, a separate Type field is provided for each byte of the data path (i.e., Type1 is provided when a byte-wide data path width is specified, Type1 and Type2 are provided when a half-word-wide data path width is specified, and Type1, Type2, Type3, and Type4 are provided when a word-wide data path width is specified).

The values Non_Repeating_Ordered_Set and Repeating_Ordered_Set are restricted to the Type1 parameter. When present, the Type2, Type3, and Type4 parameters are set to the Valid_Data_Byte value.

When multiple Type parameters are presented in an FC-1_Data.Request primitive, the corresponding Transmission Characters are transmitted in the order given (i.e., Type1 corresponds to the first character received, followed by Type2, Type3 (if present), and Type4 (if present)).

Code The Valid Data Byte or Ordered Set that is to be encoded and transmitted by the FC-1 level. The contents of the Code parameter are defined as follows:

- When Type = Valid_Data_Byte, the Code parameter contains information in a format not defined by the standard which indicates the Valid Data Byte.

- When Type = Non_Repeating_Ordered_Set the Code parameter contains information in a format not defined by the standard which indicates the Non-Repeating Ordered Set. All Ordered Sets specified in 10.3, "Ordered Sets" may be specified as a Non-Repeating Ordered Set.

- When Type = Repeating_Ordered_Set, the Code parameter contains information in a format not defined by the standard which indicates the Repeating Ordered Set. Only Primitive Sequences and the Idle Primitive Signal may be specified as Repeating Ordered Sets.

When a Repeating Ordered Set or Non-Repeating Ordered Set is indicated, a single Code field (Code1) is provided regardless of data path width. When a Valid Data Byte is indicated by Type, a separate Code field is provided for each byte of the data path (i.e., Code1 is provided when a byte-wide data path width is specified, Code1 and Code2 are provided when a half-word-wide data path width is specified, and Code1, Code2, Code3, and Code4 are provided when a word-wide data path width is specified).

When multiple Code parameters are presented in an FC-1_Data.Request primitive, the corresponding Transmission Characters are transmitted in the order given (i.e., Code1 corresponds to the first character received, followed by Code2, Code3 (if present), and Code4 (if present)).

J.5.1.3 When generated

An FC-2 level issues the FC-1_Data.Request primitive to the FC-1 level synchronous to a received FC-1_Transmitter_Clock.Indication primitive when it wishes to specify information to be encoded and transmitted by the FC-1 level. Information to be encoded is passed by the FC-2 level to the FC-1 level according to the following rules:

1. When the data path width between the FC-1 and FC-2 levels is defined to be Byte, the information passed by the FC-1_Data.Request primitive may be a Valid Data Byte, a Non-Repeating Ordered Set, or a Repeating Ordered Set. If a Repeating Ordered Set or Non-Repeating Ordered Set is passed by the FC-1_Data.Request primitive, the FC-2 level does not issue an FC-1_Data.Request primitive synchronous to the following three received FC-1_Transmitter_Clock.Indication primitives.
2. When the data path width between the FC-1 and FC-2 levels is defined to be Half_Word, the information passed by the FC-1_Data.Request primitive may be two Valid Data Bytes, a Non-Repeating Ordered Set, or a Repeating Ordered Set. If a Repeating Ordered Set or Non-Repeating Ordered Set is passed by the FC-1_Data.Request primitive, the FC-2 level does not issue an FC-1_Data.Request primitive synchronous to the following received FC-1_Transmitter_Clock.Indication primitive.
3. When the data path width between the FC-1 and FC-2 levels is defined to be Word, the information passed by the FC-1_Data.Request primitive may be four Valid Data Bytes, a Repeating Ordered Set, or a Non-Repeating Ordered Set.

The Boundary parameter of the FC-1_Transmitter_Clock.Indication primitive may restrict what an FC-2 level is allowed to present on an FC-1_Data.Request primitive. When

Boundary = Word, any information type (Valid_Data_Byte, Non_Repeating_Ordered_Set, or Repeating_Ordered_Set may be presented on the Type1 parameter of the associated FC-1_Data.Request primitive. When Boundary = Null, only Valid_Data_Byte may be presented on the Type1 parameter of the associated FC-1_Data.Request primitive. When information types other than Valid_Data_Byte are presented, unpredictable Transmitter behavior results and the Transmitter enters the Failure state.

J.5.1.4 Effect on receipt

Receipt of an FC-1_Data.Request primitive by an FC-1 Transmitter in the Not-Enabled state causes the FC-1 level to attempt to encode the data indicated by the primitive. If the FC-1 level is in Loopback mode, data are provided to the FC-1 Receiver for decoding and presentation to the FC-2 level. No signal is transmitted onto the attached Fibre as the result of a FC-1_Data.Request primitive received by an FC-1 Transmitter in the Not-Enabled state.

Receipt of an FC-1_Data.Request primitive by an FC-1 Transmitter in the Working state causes the FC-1 level to attempt to encode and transmit the data indicated by the primitive onto its attached Fibre. If the FC-1 level is in Loopback mode, this data is also provided to the FC-1 Receiver for decoding and presentation to the FC-2 level.

Encoded information is processed by an FC-1 Transmitter in the Working or Not-Enabled states according to the following rules:

1. Upon entering the Working state, the FC-1 Transmitter continuously transmits and/or provides encoded idle words until receipt of an FC-1_Data.Request primitive. (See table 24 for the definition of the idle word.)
2. Upon receipt of an FC-1_Data.Request primitive specifying Repeating_Ordered_Set, the FC-1 Transmitter begins transmitting and/or providing the specified encoded Ordered Set and continues to transmit and/or provides this Ordered Set until receipt of a subsequent FC-1_Data.Request primitive.
3. Upon receipt of an FC-1_Data.Request primitive specifying Valid_Data_Byte or Non_Repeating_Ordered_Set the FC-1 Transmitter encodes and transmits and/or provide the specified information. Receipt of such an FC-1_Data.Request primitive ends the continuous transmission and/or provision of Repeating Ordered Sets if such transmission and/or provision has been previously established by a prior FC-1_Data.Request primitive.
4. If an FC-1_Data.Request is not received by the FC-1 level synchronous to an FC-1_Transmitter_Clock.Indication primitive representing a word boundary (i.e., Boundary = Word) and the FC-1_Data.Request received synchronous to the immediately previous FC-1_Transmitter_Clock.Indication primitive representing a word boundary specified Valid_Data_Byte or Non_Repeating_Ordered_Set the FC-1 level continuously transmits and/or provides encoded idle words until the subsequent receipt of an FC-1_Data.Request primitive.
5. If an FC-1_Data.Request is not received by the FC-1 level synchronous to an FC-1_Transmitter_Clock.Indication primitive not representing a word boundary (i.e., Boundary = Null) and the FC-1_Data.Request received synchronous to the immediately previous FC-1_Transmitter_Clock.Indication primitive representing a word boundary specified Valid_Data_Byte, unpredictable FC-1 level behavior results and the Transmitter enters the Failure state.
6. If an FC-1_Data.Request is received by the FC-1 level synchronous to an FC-1_Transmitter_Clock.Indication primitive not representing a word boundary (i.e., Boundary = Null) and the FC-1_Data.Request received synchronous to the immediately previous FC-1_Transmitter_Clock.Indication primitive representing a word boundary specified Non_Repeating_Ordered_Set or Repeating_Ordered_Set, unpredictable FC-1 level behavior results and the Transmitter enters the Failure state.
7. If an FC-1_Data.Request is received by the FC-1 level synchronous to an FC-1_Transmitter_Clock.Indication primitive not representing a word boundary (i.e., Boundary = Null) and the FC-1 level is continuously transmitting and/or providing a Repeating Ordered Set, unpredictable FC-1 level behavior results and the Transmitter enters the Failure state.

Receipt of an FC-1_Data.Request primitive not conforming to the rules described previously results in unpredictable FC-1 level behavior and the Transmitter enters the Failure state.

Encoded information is not transmitted and/or provided and FC-1_Data.Request primitives are not accepted by an FC-1 Transmitter in the Failure state.

J.5.1.5 Additional comments

This primitive is illustrated by the request portion of figure 82 (a).

J.5.2 FC-1_Data.Indication

J.5.2.1 Function

This primitive is used by the FC-1 level to pass received and decoded data to an associated FC-2 level.

J.5.2.2 Semantics

The FC-1_Data.Indication service primitive is defined as follows:

FC-1_Data.Indication(Type1, Code1
Type2, Code2
Type3, Code3
Type4, Code4)

Type The type of transmission information that has been received and decoded by the FC-1 level. The allowable values of this parameter are Valid_Data_Byte, Ordered_Set, Special_Code, Code_Violation, Invalid_Special_Code_Alignment, and Reserved_Special_Code_Violation. (The latter three values are associated with the detection of invalid Transmission Words as specified by "Invalid Transmission Word rules." Special Codes and Valid Data Bytes may be reported in conjunction with these values as defined by the rules specified in J.5.2.3.) The format of this parameter is not defined by the standard.

When an Ordered Set is indicated, a single Type field (Type1) is provided regardless of data path width. When information other than an Ordered Set is indicated, a separate Type field is provided for each byte of the data path (i.e., Type1 is provided when a byte-wide data path width is specified, Type1 and Type2 are provided when a half-word-wide data path width is specified, and Type1, Type2, Type3, and Type4 are provided when a word-wide data path width is specified).

The values Ordered Set and Reserved_Special_Code_Violation are restricted to the Type1 parameter. When present, the Type2, Type3, and Type4 parameters are set to the Valid_Data_Byte, Code_Violation, Special_Code, or Invalid_Special_Code_Alignment value.

When multiple Type parameters are presented in an FC-1_Data.Indication primitive, the corresponding Transmission Characters were received in the order given (i.e., Type1 corresponds to the first character received, followed by Type2, Type3 (if present), and Type4 (if present)).

Code The information (Valid Data Byte, Ordered Set, or invalid information) that has been received and decoded by the FC-1 level. The contents of the Code parameter are defined as follows:

- When Type = Valid_Data_Byte, the Code parameter contains information in a format not defined by the standard which indicates the Valid Data Byte.

- When Type = Special_Code, the Code parameter contains information in a format not defined by the standard which indicates the Special Code.

- When Type = Ordered_Set, the Code parameter contains information in a format not defined by the standard which indicates the Ordered Set.

- When Type = Invalid_Special_Code_Alignment, the Code parameter contains information in a format not defined by the standard which indicates the Special Code that was detected in an invalid position in the second, third, or fourth character of the Transmission Word.

- When Type = Reserved_Special_Code_Violation, the Code parameter contains information in a format not defined by the standard which indicates the reserved Special Code that was detected in the first character of the Transmission Word.

- When Type = Code_Violation, the Code parameter is not meaningful and is ignored.

When an Ordered Set is indicated, a single Code field (Code1) is provided regardless of data path width. When information other than an Ordered Set is indicated by Type, a separate Code field is provided for each byte of the data path (i.e., Code1 is provided when a byte-wide data path width is specified, Code1 and Code2 are provided when a half-word-wide data path width is specified, and Code1, Code2, Code3, and Code4 are provided when a word-wide data path width is specified). (Note that the Code field associated with a Type field indicating Code_Violation, while present, is not meaningful.)

When multiple Code parameters are presented in an FC-1_Data.Indication primitive, the corresponding Transmission Characters were received in the order given (i.e., Code1 corresponds to the first character received, followed by Code2, Code3 (if present), and Code4 (if present)).

J.5.2.3 When generated

An FC-1 Receiver in the Synchronization-Acquired or Offline state issues the FC-1_Data.Indication primitive to pass received and decoded Transmission Words to the FC-2 level. When the FC-1 level is in normal mode, Transmission Words are received from the Fibre attached to the FC-1 Receiver. When the FC-1 level is in Loopback mode, received Transmission Words are provided by the FC-1 Transmitter.

Decoded Transmission Words that are determined to be valid by the FC-1 level are passed to the FC-2 level according to the following rules:

1. When the data path width between the FC-1 and FC-2 levels is defined to be Byte, the information passed by the FC-1_Data.Indication primitive may be a Valid Data Byte, a Special Code, or an Ordered Set. (A Special Code is passed only when an Unrecognized Ordered Set is received and decoded by the Receiver.)
2. When the data path width between the FC-1 and FC-2 levels is defined to be Half_Word, the information passed by the FC-1_Data.Indication primitive may be two Valid Data Bytes, a Special Code followed by a Valid Data Byte, or an Ordered Set. (The Special Code / Valid Data Byte pair is passed only when an Unrecognized Ordered Set is received and decoded by the Receiver.)
3. When the data path width between the FC-1 and FC-2 levels is defined to be Word, the information passed by the FC-1_Data.Indication primitive may be four Valid Data Bytes, a Special Code followed by three Valid Data Bytes, or an Ordered Set. (The Special Code / Valid Data Byte / Valid Data Byte / Valid Data Byte string is passed only when an Unrecognized Ordered Set is received and decoded by the Receiver.)

Decoded Transmission Words that are determined to be invalid by the FC-1 level is passed to the FC-2 level according to the following rules:

1. When the data path width between the FC-1 and FC-2 levels is defined to be Byte, the information passed by the FC-1_Data.Indication primitive may be one of the following: a Code Violation, an invalid Special Code alignment, or a reserved Special Code Violation. Preceding or subsequent FC-1_Data.Indication primitives associated with the invalid Transmission Word may contain Valid Data Bytes and/or Special Codes. (Note that an invalid Transmission Word as specified by "Invalid Transmission Word rules" may contain one or more valid Transmission Characters; it is these characters that are passed as Valid Data Bytes or Special Codes in the four FC-1_Data.Indication primitives which represent the invalid Transmission Word.)
2. When the data path width between the FC-1 and FC-2 levels is defined to be Half_Word, the information passed by the FC-1_Data.Indication primitive may be one or more of the following: a Code Violation, an invalid Special Code alignment, or a

reserved Special Code Violation. A reserved Special Code Violation may be specified only in the Type1 parameter. Code Violations and invalid Special Code alignments may be specified in either or both of the Type1 and Type2 parameters. These values may be combined with a Valid Data Byte or Special Code value in one of the Type1 and Type2 parameters according to the contents of the received and decoded Transmission Word and the usage rules described in the definition of the Type parameter. Preceding or subsequent FC-1_Data.Indication primitives associated with the invalid Transmission Word may also contain Valid Data Bytes and/or Special Codes. (Note that an invalid Transmission Word as specified by "Invalid Transmission Word rules" may contain one or more valid Transmission Characters; it is these characters that are passed as Valid Data Bytes or Special Codes in the two FC-1_Data.Indication primitives which represent the invalid Transmission Word.)

3. When the data path width between the FC-1 and FC-2 levels is defined to be Word, the information passed by the FC-1_Data.Indication primitive may be one or more of the following: a Code Violation, an invalid Special Code alignment, or a reserved Special Code Violation. A reserved Special Code Violation may be specified only in the Type1 parameter. Code Violations may be specified in one or more of the Type1, Type2, Type3, and Type4 parameters. Invalid Special Code alignments may be specified in one or more of the Type2, Type3, and Type4 parameters. These values may be combined with Valid Data Byte and Special Code values in one or more of the Type1, Type2, Type3, and Type4 parameters according to the contents of the received and decoded Transmission Word and the usage rules described in the definition of the Type parameter. (Note that an invalid Transmission Word as specified by "Invalid Transmission Word rules" may contain one or more valid Transmission Characters; it is these characters that are passed as Valid Data Bytes or Special Codes in the FC-1_Data.Indication primitive which represents the invalid Transmission Word.)

When information other than Ordered Sets is received and decoded by the FC-1 level, the FC-1_Data.Indication primitive is issued periodically to the FC-2 level with a frequency based upon the FC-0 transmission frequency and the data path width between the FC-1 and FC-2 levels. When a byte-wide data path is provided, the frequency of issuance = FC-0 transmission frequency / 10. When a half-word-wide data path is provided, the frequency of issuance = FC-0 transmission frequency / 20. When a word-wide data path is provided, the frequency of issuance = FC-0 transmission frequency / 40. When Ordered Sets are received by the FC-1 level, the FC-1_Data.Indication primitive is issued according to the rules described below.

Received and decoded information is passed to the FC-2 level by an FC-1 Receiver in the Synchronization-Acquired state according to the following rules:

1. Upon entering the Synchronization-Acquired state, the FC-1 Receiver indicates receipt of the Ordered Set which allowed it to enter the Synchronization-Acquired state. When the Ordered Set is recognized, this indication is completed by issuing an FC-1_Data.Indication with Type = Ordered_Set and Code indicating the appropriate value. When the Ordered Set is unrecognized, this indication is completed by issuing one or more FC-1_Data.Indication primitives sufficient to represent the Unrecognized Ordered Set according to the defined data path width between the FC-1 and FC-2 levels.

2. Upon receipt of an incoming bit stream representing a recognized Ordered Set, the FC-1 Receiver indicates receipt of this Ordered Set by issuing an FC-1_Data_Indication with Type = Ordered_Set and Code set to the appropriate value.

3. Upon receipt of an incoming bit stream representing an Unrecognized Ordered Set, the FC-1 Receiver indicates receipt of this Ordered Set by issuing one or more FC-1_Data.Indication primitives sufficient to represent the Unrecognized Ordered Set according to the defined data path width between the FC-1 and FC-2 levels.

4. Upon receipt of an incoming bit stream representing a data character that is not part of an Ordered Set, the FC-1 Receiver indicates receipt of this character by issuing an FC-1_Data.Indication with Type = Valid_Data_Byte and Code set to the appro-

priate value. Additional Valid_Data_Byte entries are provided when the data path width between the FC-1 and FC-2 levels is Half_Word or Word.

5. Upon receipt of an incoming bit stream representing an invalid Transmission Word as specified by "Invalid Transmission Word rules," the FC-1 Receiver indicates receipt of this word by issuing an FC-1_Data.Indication with Type indicating the appropriate condition or conditions according to the rules described previously in this section.

Decoded information is not passed to the FC-2 level and FC-1_Data.Indication primitives are not issued by an FC-1 Receiver in the Loss-Of-Signal-Failure, Loss-Of-Synchronization-Failure, Loss-Of-Synchronization, or Reset states.

J.5.2.4 Effect on receipt

Receipt of an FC-1_Data.Indication primitive by an FC-2 level causes the FC-2 level to accept the received and decoded data indicated by the primitive from the FC-1 level.

J.5.2.5 Additional comments

This primitive is illustrated by the indication portion of figure 82 (a).

J.6 FC-1_Receiver_Control

J.6.1 FC-1_Receiver_Control.Request

J.6.1.1 Function

This primitive is used by the FC-2 level to request that the FC-1 Receiver be reset, enter the Offline state, or exit the Offline state.

J.6.1.2 Semantics

The FC-1_Receiver_Control.Request service primitive is defined as follows:

FC-1_Receiver_Control.Request(Control)

Control The Receiver control request made by the FC-2 level. The allowable values of this parameter are Reset, Enter_Offline, and Exit_Offline. The format of this parameter is not defined by the standard.

J.6.1.3 When generated

An FC-2 level issues the FC-1_Receiver_Control.Request primitive with Control = Reset to request that the FC-1 level Receiver be reset. This request may result from the FC-1 level Receiver entering a failure state, although the FC-2 level may issue the FC-1_Receiver_Control.Request primitive with Control = Reset at any time.

An FC-2 level issues the FC-1_Receiver_Control.Request primitive with Control = Enter_Offline to request that the FC-1 level Receiver enter the Offline state.

An FC-2 level issues the FC-1_Receiver_Control.Request primitive with Control = Exit_Offline to request that the FC-1 level Receiver exit the Offline state.

J.6.1.4 Effect on receipt

Receipt of this primitive with Control = Reset causes the FC-1 level to initiate a reset condition in its Receiver. If the Receiver is not already in the reset state, the FC-1 level concurrently issues an FC-1_Receiver_State.Indication primitive to the FC-2 level with State = Reset. The FC-1 level is incapable of providing FC-1_Data.Indication primitives to the FC-2 level while a reset condition exists in its Receiver. When the Receiver reset condition is exited, the FC-1 level issues an FC-1_Receiver_State.Indication primitive to the FC-2 level with State = Loss_Of_Synchronization or Offline depending upon the state of the Receiver prior to the reset request.

Receipt of this primitive with Control = Enter_Offline causes the FC-1 level Receiver to enter the Offline state. The Receiver continues to provide FC-1_Data.Indication primitives to the FC-2 level. It also continues to process received Transmission Words under the rules of the Loss-of-Synchronization procedure specified in "Invalid Transmission Word rules," but completion of the procedure does not cause the Receiver to enter the Loss-Of-Synchronization state.

Receipt of this primitive with Control = Exit_Offline causes the FC-1 level Receiver to enter the Synchronization_Acquired state.

J.6.1.5 Additional comments

An FC-1 reset condition may result in a request to the FC-0 Receiver to attempt to reacquire bit synchronization.

This primitive is illustrated by figure 82 (d).

J.7 FC-1_Transmitter_Control

J.7.1 FC-1_Transmitter_Control.Request

J.7.1.1 Function

This primitive is used by the FC-2 level to request that the FC-1 Transmitter be enabled or disabled.

J.7.1.2 Semantics

The FC-1_Transmitter_Control.Request service primitive is defined as follows:

FC-1_Transmitter_Control.Request(Control)

Control The enable control request made by the FC-2 level. The allowable values of this parameter are Enable and Disable. The format of this parameter is not defined by the standard.

J.7.1.3 When generated

An FC-2 level issues the FC-1_Transmitter_Control.Request primitive with Control = Enable to request that the FC-1 level Transmitter be enabled.

An FC-2 level issues the FC-1_Transmitter_Control.Request primitive with Control = Disable to request that the FC-1 level Transmitter be disabled.

J.7.1.4 Effect on receipt

Receipt of this primitive with Control = Enable causes the FC-1 level to enable its Transmitter unless the Transmitter is in one of the following conditions:

- The Transmitter is in the Failure state.
- The Transmitter has been placed in the Not-Enabled state as a result of detection by FC-0 laser safety procedures of a laser safety condition (i.e., a condition which requires that the Transmitter cease transmission).

If the Transmitter is not already enabled and it is not in one of the above conditions, the FC-1 level concurrently issues an FC-1_Transmitter_State.Indication primitive to the FC-2 level with State = Working. Otherwise, receipt of this primitive with Control = Enable by a Transmitter does not result in a Transmitter state change.

Receipt of this primitive with Control = Disable causes the FC-1 level to disable its Transmitter. If the Transmitter is in the Working state at the time of receipt, the FC-1 level concurrently issues an FC-1_Transmitter_State.Indication primitive to the FC-2 level with State = Not_Enabled. (The Transmitter is already disabled when in the Not-Enabled or Failure state; receipt of this primitive with Control = Disable by a Transmitter in one of these states does not result in a Transmitter state change.)

J.7.1.5 Additional comments

This primitive is illustrated by figure 82 (d).

J.8 FC-1_Loopback_Control

J.8.1 FC-1_Loopback_Control.Request

J.8.1.1 Function

This primitive is used by the FC-2 level to request that the FC-1 Loopback function be enabled or disabled.

J.8.1.2 Semantics

The FC-1_Loopback_Control.Request service primitive is defined as follows:

FC-1_Loopback_Control.Request(Control)

Control The Loopback control request made by the FC-2 level. The allowable values of this parameter are Enable and Disable. The format of this parameter is not defined by the standard.

J.8.1.3 When generated

An FC-2 level issues the FC-1_Loopback_Control.Request primitive with Control = Enable to request that the FC-1 Loopback function be enabled.

An FC-2 level issues the FC-1_Loopback_Control.Request primitive with Control = Disable to request that the FC-1 Loopback function be disabled.

J.8.1.4 Effect on receipt

Receipt of this primitive with Control = Enable causes the FC-1 level to enable its Loopback function unless the FC-1 Transmitter or Receiver is in a state not compatible with the Loopback function. When the Loopback function is enabled, information passed to the FC-1 level via FC-1_Data.Request primitives is shunted to the FC-1 Receiver and reflected in FC-1_Data.Indication primitives issued by the Receiver after Synchronization is reestablished (if necessary), overriding any information received by the FC-1 Receiver.

Receipt of this primitive with Control = Disable causes the FC-1 level to disable its Loopback function. When the Loopback function is disabled, information passed to the FC-1 level via FC-1_Data.Request primitives is encoded and transmitted by the FC-1 Transmitter. Information received and decoded by the FC-1 Receiver is indicated to the FC-2 level by FC-1_Data.Indication primitives after Synchronization is reestablished (if necessary).

J.8.1.5 Additional comments

The FC-1 Loopback function is disabled upon completion of Initialization of an FC-1 Transmitter and Receiver.

Behavior of a Transmitter in the Working state is unpredictable when the FC-1 Loopback function is enabled.

Loss of Synchronization may occur whenever the Loopback function is enabled or disabled.

This primitive is illustrated by the request portion of figure 82 (c).

J.8.2 FC-1_Loopback_Control.Confirm

J.8.2.1 Function

This primitive is used by the FC-1 level to accept or reject the request that the FC-1 Loopback function be enabled or disabled.

J.8.2.2 Semantics

The FC-1_Loopback_Control.Confirm service primitive is defined as follows:

FC-1_Loopback_Control.Confirm(Status)

Status The results of the FC-1_Loopback_Control.Request primitive. The allowable values of this parameter are Accept and Reject. The format of this parameter is not defined by the standard.

J.8.2.3 When generated

An FC-1 level issues the FC-1_Loopback_Control.Confirm primitive to an FC-2 level to indicate the acceptance or rejection of a previously issued FC-1_Loopback_Control.Request primitive. Acceptance is indicated only when valid information is presented by the Receiver via FC-1_Data.Indication primitives.

J.8.2.4 Effect on receipt

Receipt of the FC-1_Loopback_Control.Indication primitive with Status = Accept informs the FC-2 level that the enabling or disabling of the FC-1 Loopback function was successful or that the FC-1 Loopback function was already enabled or disabled.

Receipt of the FC-1_Loopback_Control.Indication primitive with Status = Reject informs the FC-2 level that the enabling of the FC-1 Loopback function was unsuccessful. A request to enable the FC-1 Loopback function is rejected whenever the FC-1 Receiver is not in the Synchronization-Acquired or Loss-Of-Synchronization state and the FC-1 Transmitter is not in the Working or Not-Enabled state. (A request to disable the FC-1 Loopback function is always accepted.)

J.8.2.5 Additional comments

This primitive is illustrated by the confirm portion of figure 82 (c).

Annex K. Communication models (informative)

This annex provides communication model examples.

K.1 Half-duplex communication model

When one port of the channel only transmits Data frames and the other port only transmits Link level (Link_Control) responses, the channel is said to be performing half-duplex communication.

For example in figure 83,

- N_Port A(1) transmits a Data frame (command or data) on its outbound Fibre and N_Port B(1) receives the Data frame on its inbound Fibre.
- In response, N_Port B(1) transmits a Link_Control frame concurrently on its outbound Fibre and N_Port A(1) receives it on its inbound Fibre.
- Multiple Data frames from N_Port A(1) and the respective Link_Control frames from N_Port B(1) results in similar flow.

Figure 83 depicts the FC-2 half-duplex communication model with its logical structure and components. The model is composed of the following components:

- Node A and B
- N_Ports A(1) and B(1)
- Link_Control_Facility (LCF) in each N_Port
- Originator and Originator status in one N_Port
- Responder and Responder status in the other N_Port

The Originator is the logical function in an N_Port A(1) which initiates an Exchange, whereas the Responder is the logical function in an N_Port which is the destination of the Exchange. Either N_Port can be an Originator and the other N_Port a Responder in figure 83 at any given time. The role of N_Ports as Originator and Responder shall not change during a given Exchange.

K.2 Duplex communication model

When both ports of a channel transmit Data frames and Link_Control frames (responses to Data frames) in directions opposite to their respective Data frames, the channel is said to be performing duplex communication.

For example in figure 84,

- N_Port A(1) transmits a Data frame on its outbound Fibre and N_Port B(1) receives it on its inbound Fibre.
- Simultaneously N_Port B(1) transmits its Data frame to N_Port A(1) on its outbound Fibre and N_Port A(1) receives it on its inbound Fibre.
- In response to the Data frame from N_Port A(1), N_Port B(1) transmits a Link_Control response on its outbound Fibre and N_Port A(1) receives the response on its inbound Fibre.
- In response to the Data frame from N_Port B(1), N_Port A(1) transmits a Link_Control

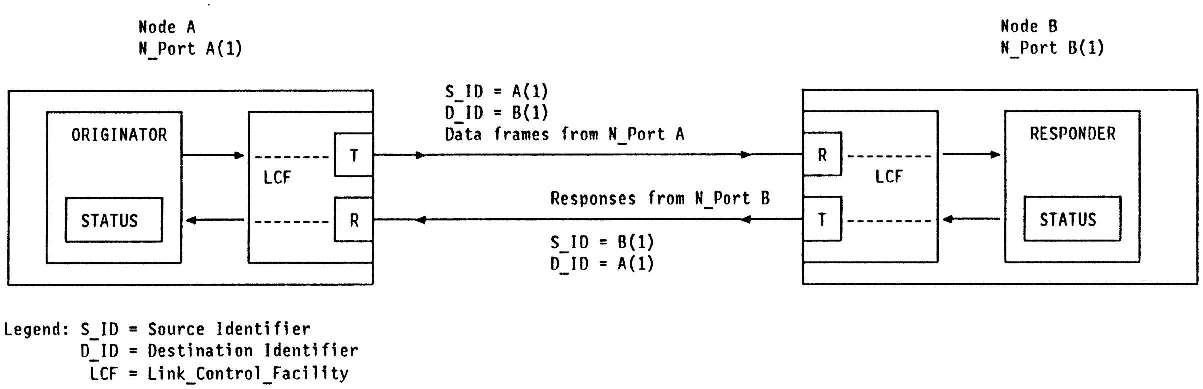


Figure 83. FC-2 half-duplex communication model example

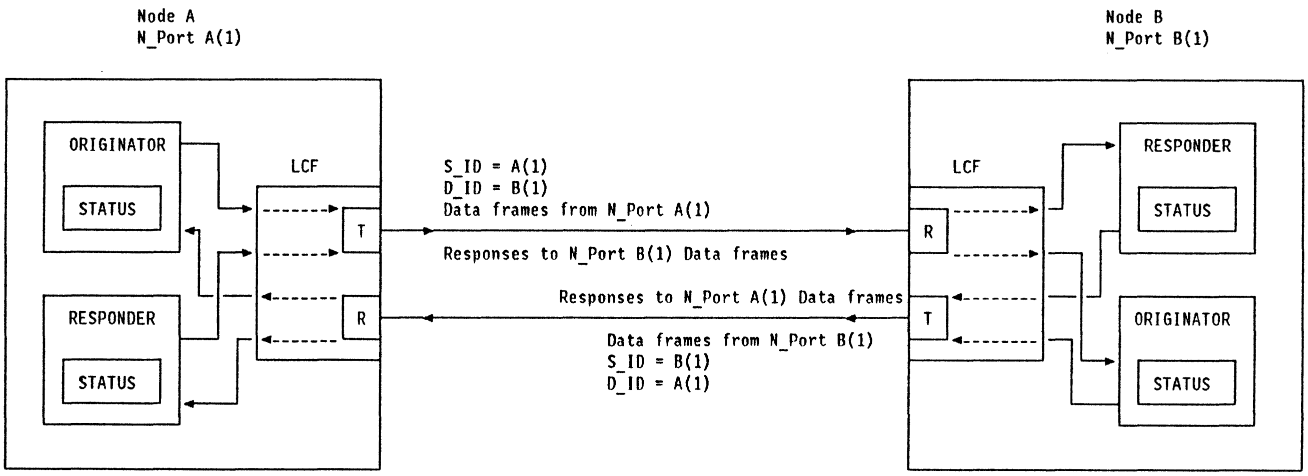


Figure 84. FC-2 duplex communication model example

response on its outbound Fibre and N_Port B(1) receives the response on its inbound Fibre.

— Multiple Data frames from N_Port A(1), and N_Port B(1) and the respective responses from N_Port B(1), and N_Port A(1) results in similar flow.

Figure 84 depicts the FC-2 duplex communication model with its logical structure and components. The FC-2 duplex communication model is bidirectionally symmetrical and both communicating N_Ports have equivalent abilities.

The FC-2 duplex communication model consists of an Originator and a Responder in each of the communicating N_Ports.

From the perspective of a given N_Port, the Outbound Fibre transmits a mixture of Data frames according to the activity generated by the N_Port and Link level responses to the Data frames received on the Inbound Fibre from the remote N_Port.

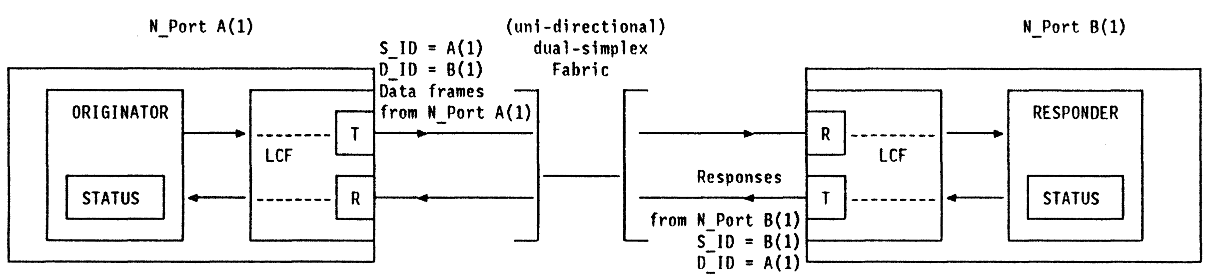
K.3 Dual-simplex communication model

When the channel transmits Data frames in one direction with the Link_Control frames arriving in the opposite direction, and limits the flow of Data frames and Link_Control frames at any given time to a single direction, the channel is said to be performing dual-simplex communication.

For example in figure 85,

— N_Port A(1) transmits a Data frame on its outbound Fibre and N_Port B(1) receives the Data frame on its inbound Fibre through the unidirectional Fabric. (simplex from A(1) to B(1)).

— In response to the Data frame, N_Port B(1) transmits a Link_Control response on its outbound Fibre and, due to unidirectional limitation of the Fabric, N_Port A(1) receives the response non-concurrently on its inbound Fibre. (simplex from B(1) to A(1)).



Legend: S_ID = Source Identifier
D_ID = Destination Identifier
LCF = Link_Control_Facility

Figure 85. FC-2 dual-simplex communication model example

— Multiple Data frames from N_Port A(1) and the respective responses from N_Port B(1) results in similar flow.

Figure 85 depicts the FC-2 dual-simplex communication model with its logical structure and components. The FC-2 dual-simplex communication model is functionally similar to the half-duplex model with the following difference. While the half-duplex communication model supports simultaneous bidirectional flow of frames, the dual-simplex model allows only unidirectional flow at any given time.

Annex L. CRC Generation and Checking (informative)

This appendix is an EXTRACT from Fiber Distributed Data Interface (FDDI) Media Access Control (MAC). (Document reference: ISO # 9314-2/x3.139-1987). FDDI's Frame Check Sequence (FCS) methodology, polynomials and equations are used by Cyclic Redundancy Check (CRC) in FC-2. The term FCS is unique to FDDI and not used by Fibre Channel. Note that CRC coverage of FC-2 fields is specified in FC-2 standard.

4.3.6 Frame Check Sequence (FCS)

This section specifies the generation and checking of the FCS field. This field is used to detect erroneous data bits within the frame as well as erroneous addition or deletion of bits to the frame. The fields covered by the FCS field include the FC, DA, SA, INFO, and FCS fields.

4.3.6.1 Definitions

F(x) - A degree k-1 polynomial which is used to represent the k bits of the frame covered by the FCS sequence (see section 4.2.2). For the purposes of the FCS, the coefficient of the highest order term shall be the first bit transmitted.

L(x) - A degree 31 polynomial with all of the coefficients equal to one, ie.,

$$L(x) = X^{31} + X^{30} + X^{29} + \cdots + X^{2} + X + 1$$

G(x) - The standard generator polynomial

$$G(x) = X^{32} + X^{26} + X^{23} + X^{22} + X^{16} + X^{12} + X^{11} + X^{10} + X^{8} + X^{7} + X^{5} + X^{4} + X^{2} + X + 1$$

R(x) - The remainder polynomial which is of degree less than 32

P(x) - The remainder polynomial on the receive checking side which is of degree less than 32

FCS - The FCS polynomial which is of degree less than 32

Q(x) - The greatest multiple of G(x) in

$$[X^{32}F(x) + X^{k}L(x)]$$

Q*(x) - $X^{32}Q(x)$

M(x) - The sequence which is transmitted

M*(x) - The sequence which is received

C(x) - A unique polynomial remainder produced by the receiver upon reception of an error free sequence. This polynomial has the value

$$C(X) = X^{32}L(X)/G(X)$$

$$C(X) = X^{31} + X^{30} + X^{26} + X^{25} + X^{24} + X^{18} + X^{15} + X^{14} + X^{12} + X^{11} + X^{10} + X^{8} + X^{6} + X^{5} + X^{4} + X^{3} + X + 1$$

4.3.6.2 FCS Generation Equations

The equations which are used to generate the FCS sequence from F(x) are as follows:

$$FCS = L(X) + R(X) = R\$(X) \quad (1)$$

where R$(X) is the one's complement of R(X)

$$[X^{32}F(x) + X^{k}L(X)]/G(X) = Q(X) + R(X)/G(X) \quad (2)$$

$$M(x) = x^{32}F(x) + FCS \quad (3)$$

NOTE:

All arithmetic is modulo 2.

In equation (1), note that adding L(x) (all ones) to R(x) simply produces the one's complement of R(x); this equation is specifying that the R(x) is inverted before it is sent out.

Equation (3) simply specifies that the FCS is appended to the end of F(x).

4.3.6.3 FCS Checking

The received sequence M*(x) may differ from the transmitted sequence M(x) if there are transmission errors. The process of checking the sequence for validity involves dividing the received sequence by G(x) and testing the remainder. Direct division, however, does not yield a unique remainder because of the possibility of leading zeros. Thus a term L(x) is prepended to M*(x) before it is divided. Mathematically, the received checking is shown in equation (4).

$$X^{32}[M^*(X) + X^{k}L(X)]/G(X) = Q^*(X) + P(X)/G(X) \quad (4)$$

In the absence of errors, the unique remainder is the remainder of the division

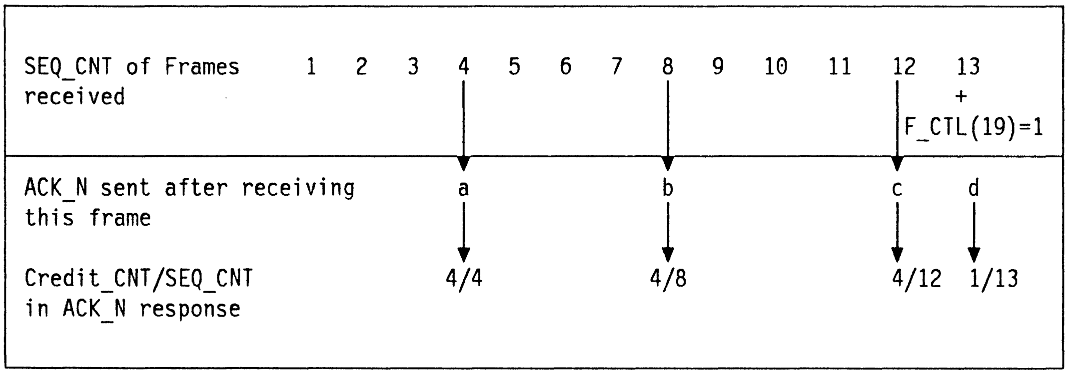
$$P(X)/G(X) = X^{32}L(X)/G(X) = C(X) \quad (5)$$

Annex M. ACK_N Usage (Informative)

Three examples are shown on the usage of ACK_N. Example 1 illustrates a case where all Data Frames are received in sequential order.

Examples 2 and 3 correspond to cases when the Frames are received out of order.

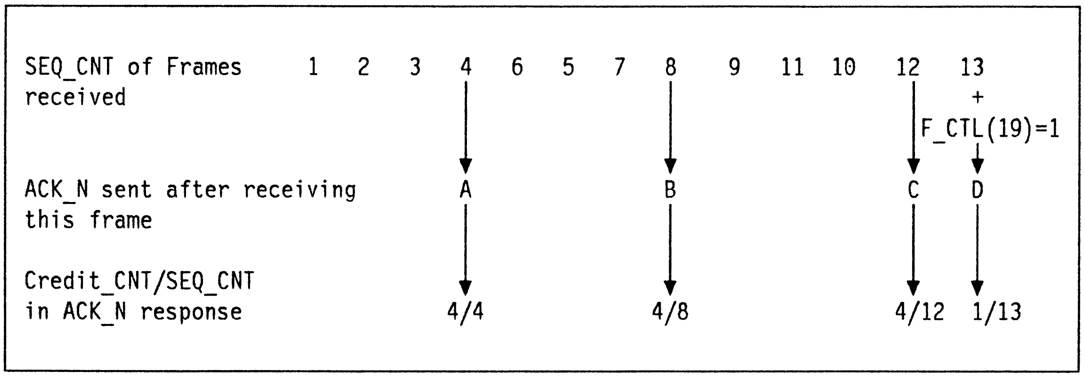
ACK_N usage example 1:



ACK_N sent at	ACK_N content: Credit_Count	ACK_N content: Sequence Count	SEQ_CNT of Frames responded to
a	4	4	1,2,3 and 4
b	4	8	5,6,7 and 8
c	4	12	9,10,11 and 12
d	1	13	13

Figure 86. ACK_N usage example 1

ACK_N usage example 2:



ACK_N sent at	ACK_N content		SEQ_CNT of Frames responded to
	Credit_Count	Sequence Count	
A	4	4	1,2,3 and 4
B	4	8	5,6,7 and 8
C	4	12	9,10,11 and 12
D	1	13	13

Figure 87. ACK_N usage example 2

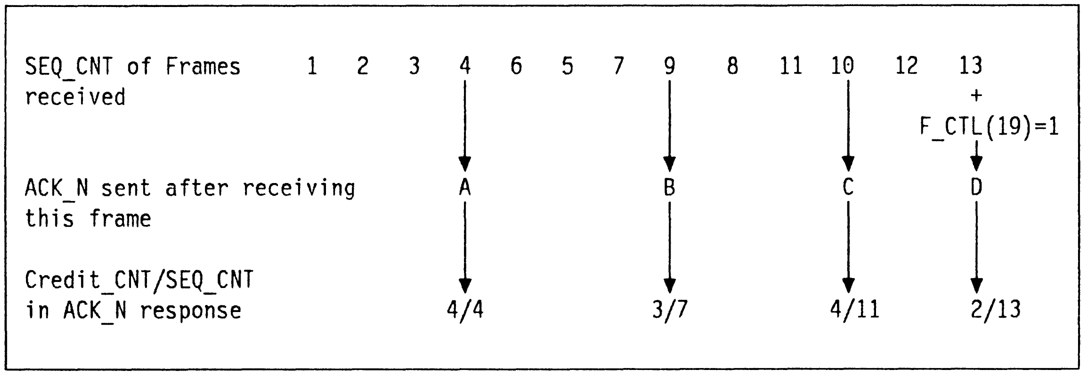
ACK_N usage rules:

1. ACK_N always reports the number of ALL consecutive frames received within a window (in Credit_CNT field).
2. ACK_N always includes the maximum sequence count of the consecutive frames being reported (in SEQ_CNT field).

Implementation Notes:

1. Receiver tracks the max. SEQ_CNT reported in the previous window.
2. Receiver tracks the expected SEQ_CNT which is max. SEQ_CNT reported plus 1.
3. Receiver does not track more than n frames (n=4 in the examples 2 and 3) before sending an ACK_N.

ACK_N usage example 3:



ACK_N sent at	ACK_N content		SEQ_CNT of Frames responded to
	Credit_Count	Sequence Count	
A	4	4	1,2,3 and 4
B	3	7	5,6 and 7
C	4	11	8,9,10 and 11
D	2	13	12 and 13

Figure 88. ACK_N usage example 3

Annex N. FC-2 Fabric Requirements (informative)

N.1 N_Port requirements

N.1.1 Receiver Control

Receiver Control Flags (C) shall specify which protocols, or functions are supported by the N_Port supplying the Service Parameters when acting as a receiver of device frames. Receiver Control Flags are specified by the FC-2 level.

NOTE - FC-4 can further limit FC-2 capabilities.

Bits	Receiver Ctl Flags (C)
42	Echo Detect/Retry
41	Alternating Type A/B
40	Reserved

Figure 89. Receiver control flags

- **Bit 42 - Echo Detect/Retry**
  - 0 Echo Detect/Retry not supported by N_Port
  - 1 Echo Detect/Retry supported by N_Port

Bit 42 specifies that the N_Port supplying this information does or does not support the Echo Detect/Retry function within the N_Port Link Control Facility. Echo Detect/Retry is activated if the Fabric (or other N_Port, in point to point) requires this function.

If Echo Detect/Retry is active, the N_Port receiver monitors for frames which are the "Echo" of frames transmitted by the N_Port by comparing the D_ID of the received frames with the address of the destination N_Port, or by comparing the S_ID of the received frames with the address of its N_Port.

If a match is found, the frame has been successfully transmitted to the destination N_Port. If the "Echo" is not detected by the N_Port receiver within a predetermined time ("retry" time), the frame has not been successfully transmitted and must be retransmitted. The "retry" attempts are repeated until the N_Port successfully detects the "Echo" of the transmitted frame.

Detection of an "Echo" is equivalent to receiving an R_RDY Link_Continue response.

- **Bit 41 - Alternating frame type required by Fabric**
  - 0 Alternating frame type A/B not supported by N_Port
  - 1 Alternating frame type A/B supported by N_Port

Bit 41 specifies that the N_Port supplying this information does or does not support Alternating frame type A/B function within the N_Port Link Control Facility. This function is activated based on Fabric requirements specified in the Fabric Service Parameters.

If the Alternating frame type A/B function is active, the N_Port transmitter marks the frame with type A or B and then transmits. After the frame is successfully transmitted ("Echo detected"), the N_Port transmitter changes the frame type before transmitting the next frame. Alternating frame type A/B is a method which attempts to provide "fairness" in a Broadcast type of Fabric.

Bit 41 is ignored if Bit 42 is zero. NOTE - Frame type "A" for a Broadcast type of Fabric has been specified as a "normal" Idle word immediately preceding the **SOF** for a frame. Type "B" would be a different encoding of an Idle word which precedes the **SOF** for a frame. These definitions need to be added to FC-1 and FC-2 for completeness.

N.2 F_Port Requirements

N.2.1 Receiver Control

Receiver Control Flags (C) have the following designations for ALL classes:

- **Bit 42 - Echo Detect/Retry**

— 0 Echo Detect/Retry not required by Fabric
— 1 Echo Detect/Retry required by Fabric

- **Bit 41 - Alternating frame type required by Fabric**

— 0 Alternating frame type A/B not required by Fabric
— 1 Alternating frame type A/B required by Fabric

Bit 41 specifies that the Fabric supplying this information does or does not require the use of the Alternating frame type A/B function within the N_Port Link Control Facility attached directly to the Fabric.

Bit 41 is ignored if Bit 42 is zero.

N.3 Fabric Characteristics

The following list of characteristics describes a fabric as viewed from an FC-2 Port.

End to End communication

FC-2 is based on endpoint to endpoint communication. An optional intervening fabric may be present within the physical configuration. Features are provided within the FC-2 definition which allows several different modes or classes of service between two endpoints which are interconnected by means of a fabric.

FC-2 -- Reply to every Request/Command

FC-2 protocol is based on receiving a Reply to every Request. Depending on the options chosen and the intervening fabric present, multiple replies may be received by an FC-2 Port for a single Request.

Sequential frame transmission

Sequential transmission is required of the transmitting Port. The order of delivery is in the same sequence in some environments and non-sequential in others.

6 Idle sequences transmitted after a frame

During frame transmission, 6 Idles are transmitted after one frame and before the start of the next frame. Up to 4 Idles may be removed by an intervening fabric for internal communication and clock skew management. However, 2 Idles are guaranteed preceding the start of each frame at the receiving N_Port.

InBand Addressing

Frames are routed by the fabric based on the destination address contained within the frame contents.

Control Port

A fabric shall have an addressable control Port in order to support various control functions.

Service Classes

Class 1 - Dedicated Connection
Provides guaranteed service between two Ports as though the two Ports were physically interconnected on both transmit and receive fibres by a circuit.

Class 2 - Multiplex
Provides connectionless service with non-delivery notification between two Ports as though the two Ports were logically interconnected. Operates on a frame-switched basis.

Class 3 - Datagram
Provides connectionless service between two Ports as though the two Ports were logically interconnected. Operates on a frame-switched basis without non-delivery notification.

Limited Frame size

Frame Data Field size is always limited by the maximum size which the destination Port may receive. The frame establishing a Class 1 Connection, and all Class 2 and 3 frames may also be limited by the maximum size supported by the intervening fabric buffers.

Login

Login with Fabric and the destination Port is always required.

N.4 F_Port R_RDY processing

```
** F_Port receives frame from an attached N_Port **

If (Frame_Header in Error)
   Then
      Begin
      F_Port returns F_RJT (S_ID)
      F_Port returns R_RDY (S_ID)
      End
Else
   Begin
   While (Elapsed_time < Frame_timeout_period)
     Begin
     F_Port attempts to send frame to N_Port (D_ID)
     If (F_Port sends frame)
        Then
        Begin
        F_Port returns R_RDY (S_ID)
        Exit
        End
     End
   End_While
   ** Timeout reached without sending frame **
   F_Port returns F_BSY (S_ID)
   F_Port returns R_RDY (S_ID)
   End
```

Annex O. Data transfer protocols and examples (informative)

This annex summarizes Data transfer protocols with examples and provided as information.

O.1 Introduction

FC-2 provides Data transfer protocols to transfer data between N_Ports or Nodes at two levels:

1. Sequence level protocol
2. Frame level protocol

Sequence level
Sequence level protocol is used by Originator and Responder pair of an Exchange. The Originator issues a Request Sequence to the Responder who may issue a Reply Sequence in response. The Originator may in turn respond with another Reply Sequence and the communication may continue until the end of the Exchange. The Originator or the Responder may issue multiple consecutive Sequences if it chooses to. A Sequence Initiative control bit in F_CTL determines who (the Originator or the Responder) is allowed to transmit a Sequence at a given time. Control bits in F_CTL are used to indicate the first, intermediate, or the last Sequence in the Exchange.

Frame level
Frame level protocol is used by a Sequence Initiator (SI) and a Sequence Recipient (SR) pair. The Sequence Initiator issues one or more Data frames and the Sequence Recipient issues a Link_Control frame for each of these Data frames. Control bits in F_CTL are used to indicate the first, intermediate, or last frame in the Sequence. (see "24 Exchange, Sequence, and Sequence Count Management" for the definition of Exchange, Sequence, SI, and SR.)

Users
Sequence level protocol is used by the ULP at its discretion and needs. Frame level protocol is used by FC-2 itself to communicate successful or unsuccessful frame delivery and frame level flow control.

Applicability
Sequence level protocol is Class independent. Frame level protocol is significantly Class dependent.

Directions
If an Exchange is made up of one or more Sequences flowing in a single direction (i.e., from Originator to Responder), the Exchange is said to be unidirectional. If an Exchange is made up of multiple Sequences flowing in both directions, the Exchange is bidirectional.

O.2 Sequence level protocol

Sequence level protocol is specified for Bidirectional Exchange. (Unidirectional Exchange is considered a special case of generic Bidirectional Exchange and not specified separately).

The Originator starts the first Sequence of an Exchange. It may act as the Sequence Initiator and initiate one or more Sequence(s) before it passes the Sequence Initiative to the Responder. The Responder then may perform one or more Sequences and may pass the Sequence Initiative to the Originator. A Sequence is initiated by embedding SOFix (or SOFc1) in a Data frame.

The Exchange may be terminated either by the Originator or the Responder. One who terminates the Exchange shall own the Sequence Initiative and act as the Sequence Initiator for the Sequence in which it terminates the Exchange. The Exchange is terminated with the termination of the last Sequence.

Figure 90 illustrates the Sequence level protocol with a request and a reply Sequence.

Login and Estimate Credit procedures are examples of Sequence level protocols. In Login procedure, LOGI is a single frame request Sequence and the ACC to LOGI is a single frame reply Sequence. In Estimate Credit procedure, ESTS and ADVC are single frame request Sequences and ESTC is a request Sequence with three or more frames. The related ACCs are single frame reply sequences. Initiative to transmit is passed on completion of each request or reply Sequence.

0.2.1 Protocol description

A Sequence flows from the Sequence Initiator to the Sequence Recipient. A Sequence is initiated by the Sequence Initiator with a SOFix (or SOFc1 if establishing a Class 1 Connection as well) in the first Data frame of the Sequence. A Sequence is terminated by the Sequence Initiator by indicating the last Data frame in the F_CTL. If ACK is used in the protocol, the Sequence Recipient embeds an EOFt (or EOFdt if removing the Connection as well). If ACK is suppressed in the protocol, the Sequence Initiator embeds an EOFt (or EOFdt) in the last Data frame.

An Exchange is performed between an Originator and a Responder. The Originator assigns an Exchange with an OX_ID = x and starts the Exchange by initiating the first Sequence with a SEQ_ID = a.

1. The Originator sends the first Data frame which
   A. may establish a Class 1 Connection,
   B. initiates an Exchange, and
   C. initiates the first Sequence of the Exchange.
2. The Originator owns the Sequence Initiative when it initiates the Exchange. The Originator acts as the Initiator and the Responder as the Recipient of this first Sequence.
3. The Responder as the Sequence Recipient returns an RX_ID in its ACK within the first Sequence.
4. On terminating the first Sequence, the Originator may choose to initiate another Sequence or pass the Sequence Initiative to the Responder in the last Data frame of the Sequence.
5. On receiving the Sequence Intiative from the Originator, the Responder may perform one or more Sequences as the Sequence Initiator. The Originator performs the function of the Sequence Recipient.
6. As driven by the ULP, the Originator or the Responder terminates the Exchange when it is acting as the Sequence Initiator. The termination of the Exchange is indicated by specifying the Sequence as the last Sequence and the last Data frame of the Exchange as the last frame in that Sequence.
7. The termination of the last Sequence also marks the end of the Exchange.

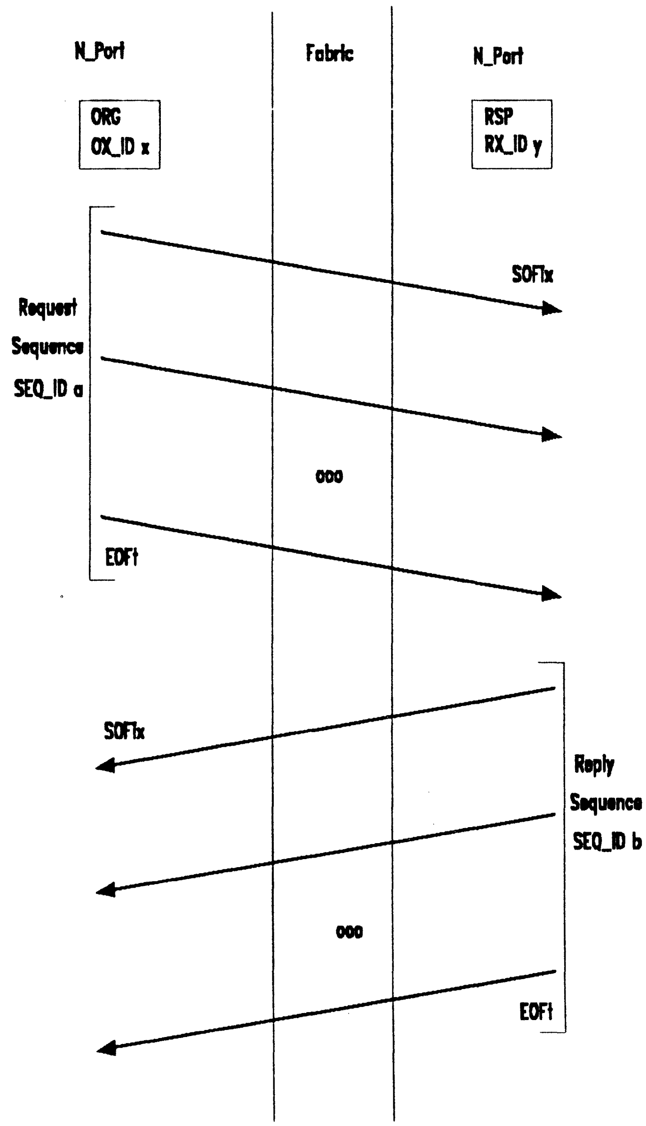


Figure 90. Sequence level protocol

0.3 Frame level protocol

Frame level protocol is specified for a Sequence flowing from the Originator to the Responder and is also applicable for a Sequence from the Responder to the Originator with appropriate value for Exchange Context and Sequence Context bits in F_CTL field. (Also see 0.4 for examples.)

Frame level protocol is Class dependent and is separately specified for each Class. Class dependent factors are:

1. SOF delimiters
2. Place of usage of EOFt/EOFdt
3. ACK usage
4. R_RDY usage

0.3.1 Class 1 frame level protocol

Class 1 frame level protocol employs

1. Data frame
2. ACK

Class 1 frame level protocol is illustrated in figure 91.

1. The Sequence is initiated by the Originator with a Data frame embedded with SOFc1 or SOFi1. SOFc1 is used to indicate a Connect request and the Sequence initiation. Within an established Connection SOFi1 indicates the Sequence initiation.
2. Next Data frame is not sent until the Connect request is accepted by the destination.
3. The Originator streams multiple Data frames and the Responder responds with ACK.
   - ACK returns following information contained in F_CTL of the Data frame to which it is responding unaltered.
     - First Sequence
     - Last Sequence
     - Last Data frame
     - Sequence transmit initiative
   - ACK toggles following information contained in F_CTL of the Data frame.
     - Exchange Context
     - Sequence Context

   F_CTL usage for the Sequence is described in table 69.
4. SOFn1 is used to indicate the Sequence in progress.
5. The end of Sequence is indicated to the Sequence Recipient by the Last Data frame bit in F_CTL. The Last Data frame contains EOFt or EOFdt, if ACK is suppressed. The last ACK is embedded with EOFt or EOFdt if ACK is used.
6. The end of Exchange is indicated to the Sequence Recipient by the Last Sequence and the Last Data frame bits in F_CTL. The Last Sequence and the EOFt or EOFdt indicate to either N_Port the end of Exchange.

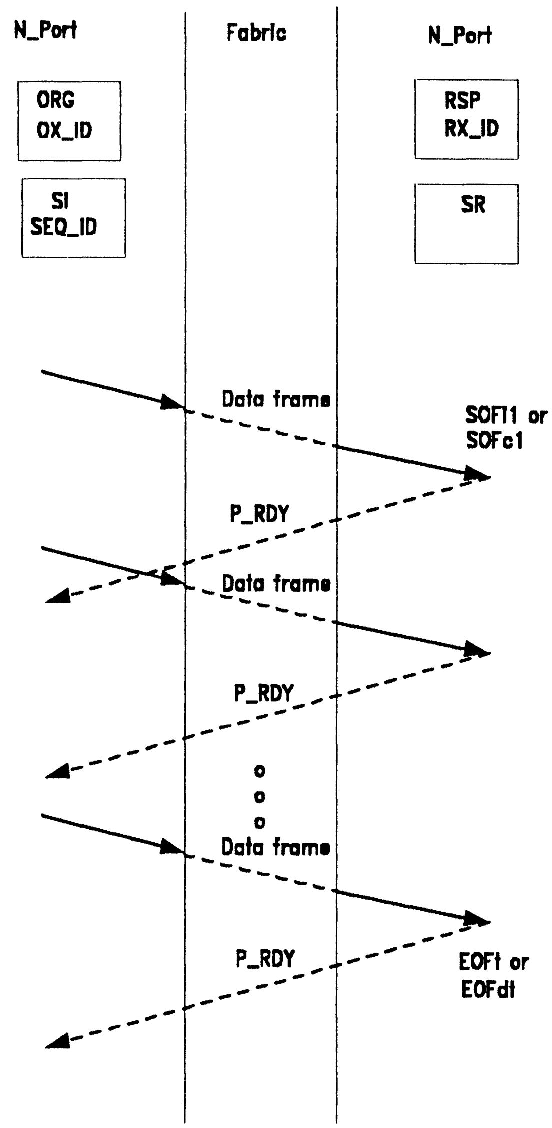


Figure 91. Class 1 Frame level protocol

0.3.2 Class 2 frame level protocol

Class 2 frame level protocol employs

1. Data frame
2. ACK
3. R_RDY

Class 2 frame level protocol is illustrated in figure 92.

1. The Sequence is initiated by the Originator with a Data frame embedded with SOFi2.
2. The F_Port if present responds with an R_RDY and forwards the Data frame to the destination.
3. The destination responds with an R_RDY, in addition to ACK.
4. The F_Port and the N_Port respond with R_RDY each on receipt of ACK.
5. The Originator streams multiple Data frames and the Responder responds with ACK.
6. For each of these frames received except R_RDY, each N_Port or F_Port, returns a R_RDY.
   - ACK returns some information contained in F_CTL of the Data frame to which it is responding unaltered.
     - First Sequence
     - Last Sequence
     - Last Data frame
     - Sequence transmit initiative
   - ACK toggles some information contained in F_CTL of the Data frame.
     - Exchange Context
     - Sequence Context

   F_CTL usage for the Sequence is described in table 69.
7. SOFn1 is used to indicate the Sequence in progress.
8. The end of Sequence is indicated by the Sequence Initiator by the Last Data frame bit in F_CTL. However, the Sequence ends in the perspective of Sequence Recipient, only when all Data frames are received or accounted for.
9. The Sequence Initiator places in the final frame, EOFt or EOFdt only after all other frames are acknowledged, if ACK is suppressed. If ACK is used, the Sequence Recipient transmits EOFt or EOFdt only in the final ACK after all Data frames are received or accounted for.

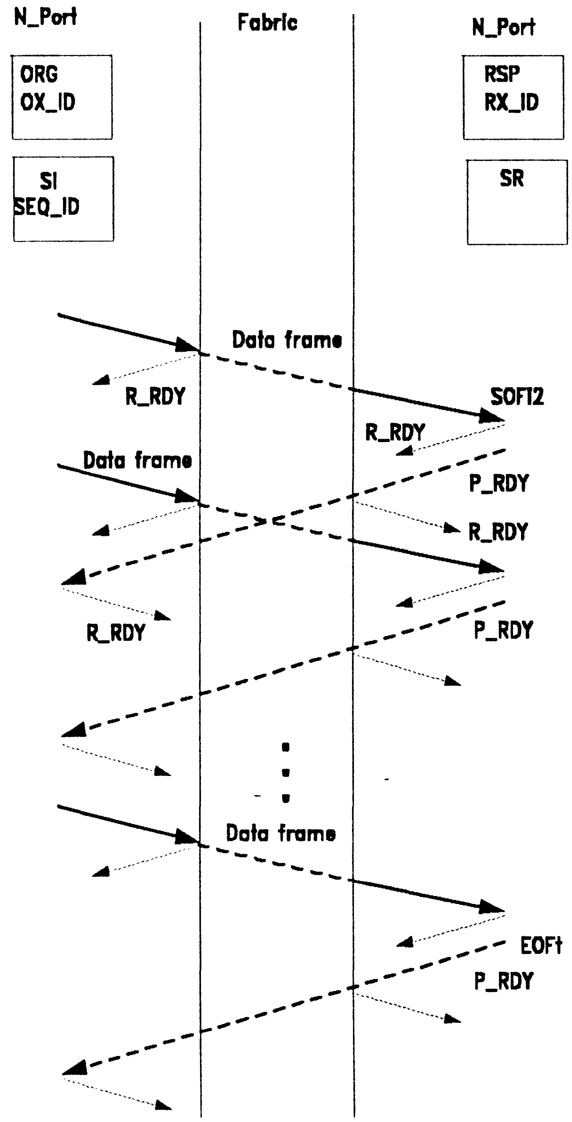


Figure 92. Class 2 Frame level protocol

O.3.3 Class 3 frame level protocol

Class 3 frame level protocol employs

1. Data frame
2. R_RDY

Class 3 frame level protocol is illustrated in figure 93.

1. The Sequence is initiated by the Originator with a Data frame embedded with SOFi3.
2. The F_Port if present responds with an R_RDY and forwards the Data frame to the destination.
3. The destination responds with an R_RDY.
4. The F_Port and the N_Port respond with R_RDY each.
5. The Originator streams multiple Data frames. For each of these frames received except R_RDY, each N_Port or F_Port, returns a R_RDY. F_CTL usage for the Sequence is described in table 70.
6. SOFn3 is used to indicate the Sequence in progress.
7. The end of Sequence is indicated to the Sequence Recipient by the Last Data frame bit in F_CTL and EOFt.

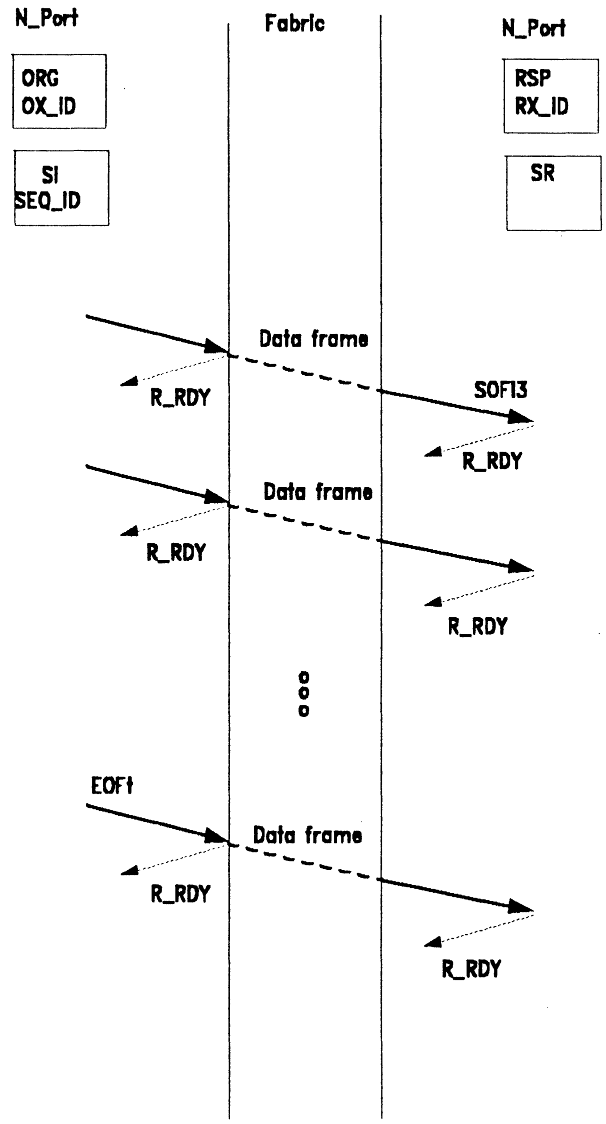


Figure 93. Class 3 Frame level protocol

Table 69. F_CTL for Class 1 or Class 2 frame level protocol

Description	Exchange Context	Sequence Context	First Sequence of Exchange	Last Sequence of Exchange	Last Data frame of Sequence	Sequence transmit initiative
F_CTL Bits	23	22	21	20	19	16
First Data frame	0 (ORG)	0 (SI)	1 (First	0 Sequence)	0	0 (hold Sequence Initiative)
P_RDY or ACK	1 (RSP)	1 (SR)	1 (First	0 Sequence)	0	1 (transfer Sequence Initiative)
Interme-diate Data frame(s)	0	0	1	0	0	0
P_RDY or ACK	1	1	1	0	0	1
Last Data frame	0	0	1	0	1	0
P_RDY or ACK	1	1	1	0	1	1

Table 70. F_CTL for Class 3 Frame level protocol

Description	Exchange Context	Sequence Context	First Sequence of Exchange	Last Sequence of Exchange	Last Data frame of Sequence	Sequence transmit initiative
F_CTL Bits	23	22	21	20	19	16
First Data frame	0 (ORG)	0 (SI)	1 (First	0 Sequence)	0	0 (hold Sequence Initiative)
Interme-diate Data frame	0	0	1	0	0	0
Last Data frame	0	0	1	0	1	0

O.4 Sequence level protocol example

Sequence level protocol is illustrated with a three Sequence Exchange in figure 90. The first Sequence is a "read" request. The second Sequence transfers the "data". The third Sequence transfers "ending status" and ends the Exchange.

Frames 1,2, and 3 represent the first Sequence of an Exchange. In this example a Command Request for a Read operation is sent as a request Sequence. Note that Sequence Initiative is transferred to the Sequence Recipient.

Frames 4,5 and 6 represent the first, intermediate and last frames of the data transferred in response to the Read request. Note that the Sequence Initiative is retained in order to start a Sequence with ending status.

Frames 7,8 and 9 represent the ending status for the preceding data transfer and end the Exchange. Depending on the Upper Level Protocol, the Responder may not be allowed to end the Exchange, but transfer the Sequence initiative to the Originator to complete the Exchange.

F_CTL usage

Use of F_CTL bits for these example Sequences are shown in table 71.

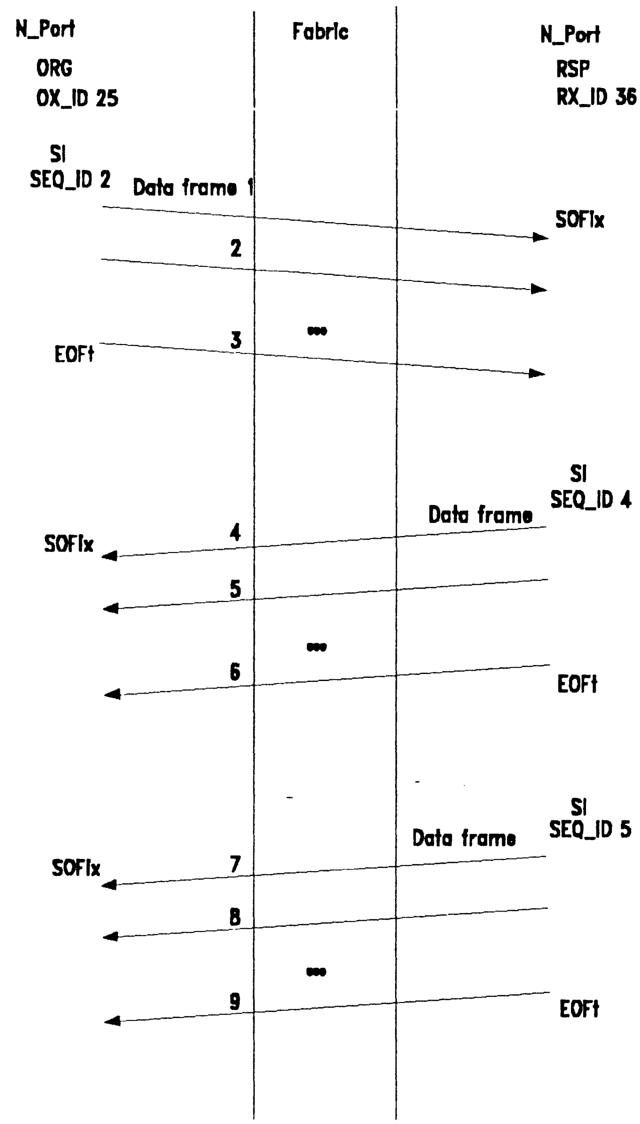


Figure 94. Sequence level protocol example

Table 71. F_CTL for Bidirectional Exchange example

Description	Exchange Context	Sequence Context	First Sequence of Exchange	Last Sequence of Exchange	Last Data frame of Sequence	Sequence transmit initi-ative
F_CTL Bits	23	22	21	20	19	16
1. First Data frame (**SOF**ix) of the Exchange and of the first Sequence (a Read Request Sequence)	0	0	1	0	0	0
2. Intermediate Data frame of first Sequence	0	0	1	0	0	0
3. Last Data frame of first Sequence	0	0	1	0	1	1
4. First Data frame (**SOF**ix) of intermediate Sequence (Reply Sequence)	1	0	0	0	0	0
5. Intermediate Data frame of intermediate Sequence	1	0	0	0	0	0
6. Last Data frame of intermediate Sequence	1	0	0	0	1	0
7. First Data frame (**SOF**ix) of the Last Sequence (Reply Status Sequence)	1	0	0	1	0	0
8. Intermediate Data frame of the Last Sequence	1	0	0	1	0	0
9. Last Data frame of the Last Sequence and of the Exchange	1	0	0	1	1	0

O.5 Class 1 Frame level protocol example

N_Port Login is used as an example to illustrate Class 1 frame flow.

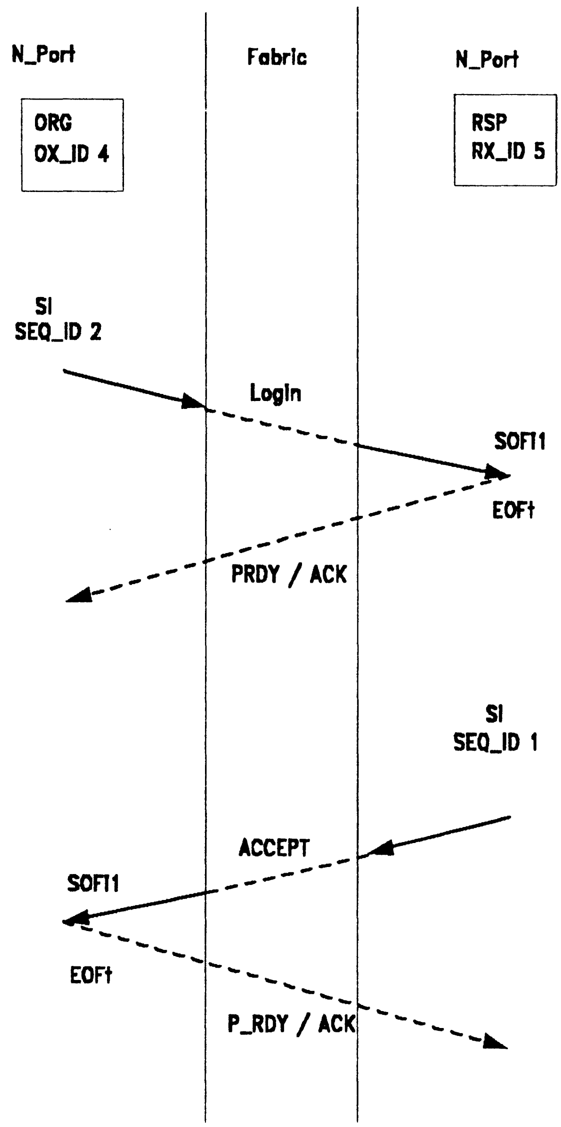


Figure 95. Class 1 Frame level protocol - Login example

O.6 Class 2 Frame level protocol example

N_Port Login is used to illustrate Class 2 frame flow.

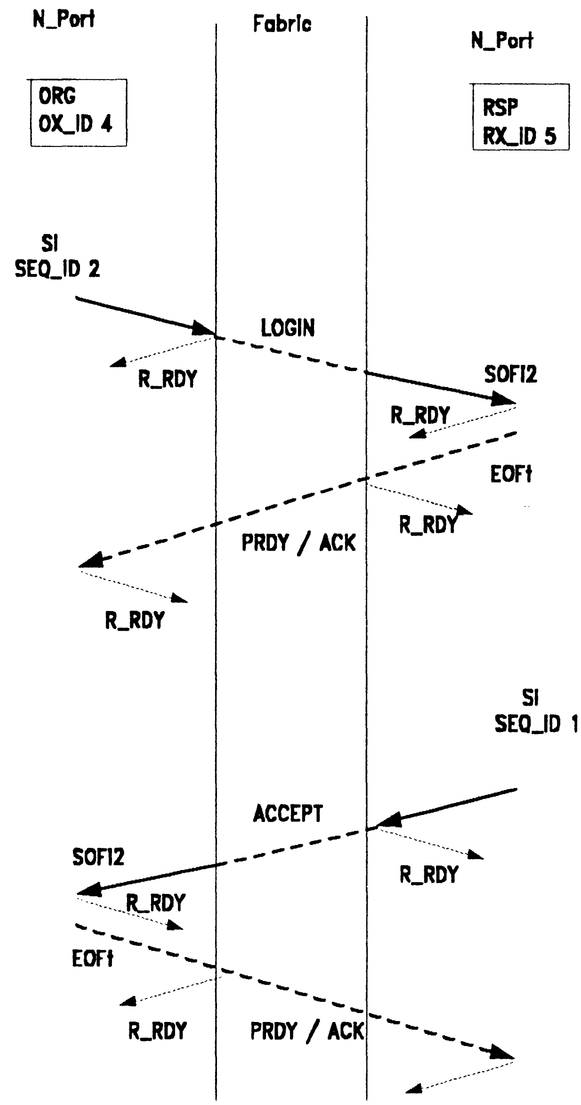


Figure 96. Class 2 frame level protocol for Login example

O.7 Class 3 Frame level protocol example

N_Port Login is used to illustrate Class 3 frame flow.

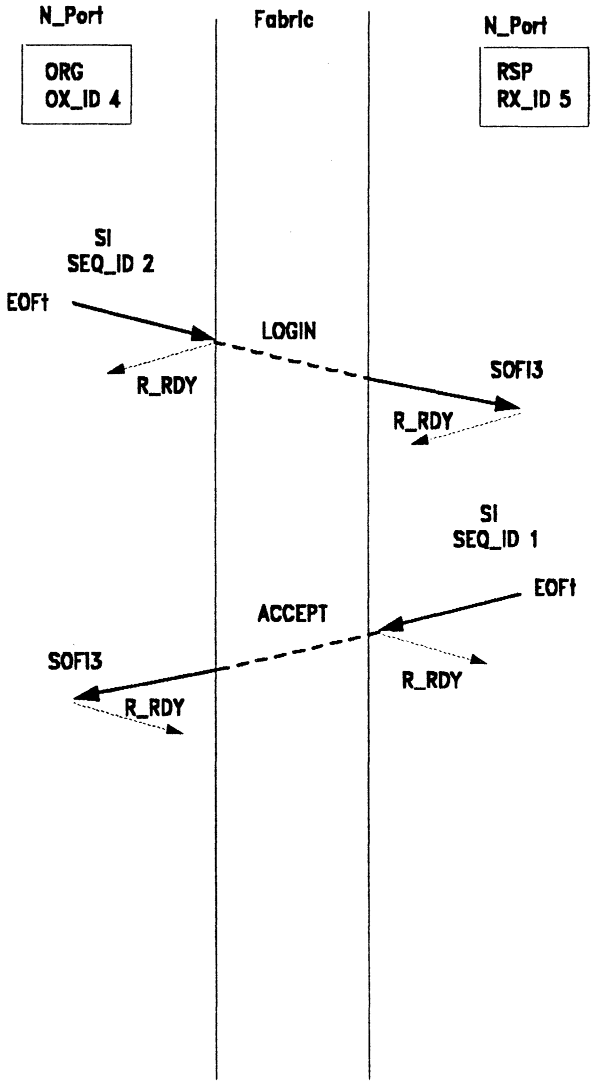


Figure 97. Class 3 frame level protocol for Login example

Annex P. Connection Management applications (informative)

P.1.1 Case 1

Table 72 shows a Case where N_Port (A) transmits its last Sequence without N_Port (B) transmitting any Sequences.

Table 72. Case 1

xmit E_C (A)	recv E_C (A)	N_Port Actions A = this N_Port B = other connected N_Port
0	0	A transmitting Sequences
1	0	A transmits last Data frame of last Sequence
0	0	A receives the ACK_1 or ACK_N to the last Data frame with **EOFdt** -- Connection Removed

P.1.2 Case 2

Table 73 shows a Case where N_Port (A) completes its last Sequence before N_Port (B).

Table 73. Case 2

xmit E_C (A)	recv E_C (A)	N_Port Actions A = this N_Port B = other connected N_Port
0	0	Both A and B transmitting Sequences
1	0	A transmits last Data frame of last Sequence
0	0	B completes Sequences in progress, A responds with Link_Continues
0	1	A receives the last Data frame of last Sequence from B
0	0	A transmits **EOFdt** on ACK_1/ACK_N -- Connection Removed

P.1.3 Case 3

Table 74 shows a Case where N_Port (B) transmits its last Sequence before N_Port (A).

Table 74. Case 3

xmit E_C (A)	**recv E_C (A)**	**N_Port Actions** **A = this N_Port** **B = other connected N_Port**
0	0	Both A and B transmitting Sequences
0	1	A receives last Data frame of last Sequence from B
0	0	A completes Sequences in progress
1	0	A transmits last Data frame of last Sequence
0	0	A receives **EOFdt** on ACK_1/ACK_N -- Connection Removed

P.1.4 Case 4

Table 75 shows a Case where N_Port (A) transmits its connect-request with E_C set to one.

Table 75. Case 4

xmit E_C (A)	**recv E_C (A)**	**N_Port Actions** **A = this N_Port** **B = other connected N_Port**
1	0	A transmits connect-request with E_C set to one
0	0	A receives the ACK_1 or ACK_N with **EOFdt** -- Connection Removed

P.1.5 Case 5

Table 76 shows a Case where N_Port (A) transmits its last Data frame of its last Sequence simultaneously with receiving the last Data frame of N_Port (B)'s last Sequence.

Table 76. Case 5

xmit E_C (A)	recv E_C (A)	N_Port Actions A = this N_Port B = other connected N_Port
0	0	Both A and B transmitting Sequences (A is Connection Initiator)
1	0	A transmits last Data frame of last Sequence
0	1	A receives Data frame from B before receiving Link_Continue for its last Data frame with E_C = 1
0	0	A waits for its Link_Continue response
0	0	A receives its ACK_1/ACK_N from B with $\textbf{EOF}_{n1}$
0	0	A transmits ACK_1/ACK_N with $\textbf{EOF}_{dt}$ -- Connection Removed

P.2 Ending Sequence and Connection

Table 77 shows an example of F_CTL Bit settings for key Data frames within an example Exchange.

The first Sequence is a "read" request. The second Sequence transfers the "data". The third Sequence transfers "ending status" and ends the Exchange and terminates a Class 1 Dedicated Connection.

Table 77. F_CTL for example Exchange

Description	Exc Owner	Seq Owner	First Seq of Exc	Last Seq of Exc	Last Seq Data frame	E_C	Seq xmit initiative
Bits	23	22	21	20	19	18	16
1.First Data frame of Exchange (**SOFix**) - a Read request(xmit)	0	0	1	0	0	0	0
2.Intermediate Data frame of first Sequence (xmit)	0	0	1	0	0	0	0
3.Last Data frame of first Sequence (xmit)	0	0	1	0	1	0	1
4.First Data frame of reply "data" Sequence (**SOFix**)(recv)	1	0	0	0	0	0	0
5.Last Data frame of reply data Sequence (recv)	1	0	0	0	1	0	0
6.First Data frame of reply Status Sequence (**SOFix**)(recv)	1	0	0	1	0	0	0
7.Last Data frame of Exchange (recv)	1	0	0	1	1	1	1
8.Last ACK_1 of Exchange (xmit) with **EOFdt**	0	1	0	1	0	0	0

Frames 1,2, and 3 represent the first Sequence of an Exchange. In this example a Command Request for a Read operation is sent as a request Sequence. Note that Sequence Initiative is transferred to the Sequence Recipient.

Frames 4 and 5 represent the first and last frames of the data transfer associated with the Read operation. Note that the Sequence Initiative is retained in order to start a Sequence with ending status.

Frames 6 and 7 represent the ending status for the preceding data transfer and ends the Exchange. Frame 7 ends the Sequence, ends the Exchange, and requests termination of the Connection. The ACK_1 response to frame 7 (frame 8) removes the Dedicated Connection with **EOFdt**. (This example assumes the other N_Port never transmitted a Sequence and, therefore, never transmitted E_C set to one.

P.3 N_Port States

With respect to Class 1 Service, the establishment and removal of Class 1 connections can be viewed as a number of separate and distinct states within an N_Port

Inactive (IN)

An N_Port is in the Inactive state when it is not in any other state. In the Inactive state, an N_Port is available to initiate a Dedicated Connection request or to respond to a request to establish a Dedicated Connection.

Busy (BS)

When an N_Port is in the Busy state because of internal reasons, the N_Port is unable to initiate or respond to a request to establish a Dedicated Connection. The Busy state is a normal, temporary condition.

Connect_Try (CT)

This state within the N_Port results when this N_Port has initiated a connect-request (**SOFc1**) and the destination N_Port has not yet responded.

This state within the N_Port also results when another N_Port has initiated a connect-request (**SOFc1**) which this N_Port has received but not yet responded to.

Connected (CN)

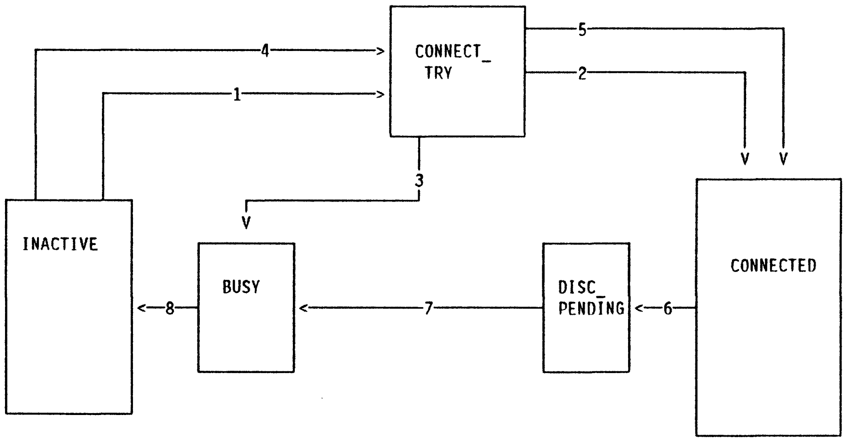
This state results after this N_Port (A) has transmitted a connect-request (**SOFc1**) to a destination N_Port (B) and this N_Port (A) has received a response frame from the destination N_Port (B) started by **SOFn1** or by **SOFi1** (ie. N_Port (A) is the source of the connect-request). A Dedicated Connection is now established.

This state also results after this N_Port (A) has received a connect-request (**SOFc1**) from another N_Port (B) and this N_Port (A) has transmitted a response frame started by **SOFn1** (ie. N_Port (A) is the destination of the connect-request). A Dedicated Connection is now established.

Disconnect_Pending (DP)

The state results after this N_Port has transmitted a frame with the End_Connection Bit in F_CTL set to one.

P.4 State Diagram



1. transmit **SOF$_{c1}$**
2. receive **SOF$_{n1}$** - Dedicated Connection as Source
3. receive BSY or RJT with **EOF$_{dt}$**
4. receive **SOF$_{c1}$** - Dedicated Connection as Destination
5. transmit **SOF$_{n1}$** - response to **SOF$_{c1}$** from another N_Port
6. transmit or receive frame with E_C set to one
7. transmit or receive **EOF$_{dt}$**
8. Internal Busy condition cleared

Index

W

End of Document
May 25, 1991
8:50 a.m.