Copies of this document may be purchased from:
Global Engineering, 15 Inverness Way East,
Englewood, CO 80112-5704
Phone: (800) 854-7179 or (303) 792-2181 Fax: (303) 792-2192

XXX-199X
T11/Project xxxx-xx/Rev 1.0

FIBRE CHANNEL

Virtual Interface

(FC-VI)

REV 1.0

NCITS working draft proposed
American National Standard
for Information Technology

February 3, 1999

Secretariat: Information Technology Industry Council

NOTE:
This is a working draft American National Standard of Accredited Standards Committee NCITS. As such this is not a completed standard. The T11 Technical Committee or anyone else may modify this document as a result of comments received anytime, or during a future public review and its eventual approval as a Standard. Use of the information contained herein is at your own risk.

POINTS OF CONTACT:

Kumar Malavalli (T11 Chairman)
Brocade Communications
1901 Guadalupe Parkway
San Jose, CA 95131
Voice: (408) 487-8156
Fax: (408) 487-8101
EMail: kumar@brocade.com

Ed Grivna (T11 Vice Chairman)
Cypress Semiconductor
2401 East 86th Street
Bloomington, MN 55425
Voice: (612) 851-5046
Fax: (612) 851-5087
E-Mail: elg@cypress.com

Dave Ford (Facillitator)
Orca Systems, Inc.
77 Union Street
Watertown, MA 02172
Voice: (617) 926-1399
Fax: (617) 926-1495
E-mail: dford@orca.com

Mark Farley (Technical Editor)
Finisar Corporation
274 Ferguson Drive
Mountain View, CA 94043
Voice: (650) 691-4000
Fax: (650) 691-4010
E-mail: mfarley@finisar.com

Release Notes for Version 0.3:

Document re-write to incorporate consolidated proposal for Device Headers in all FC-VI frames.

Release Notes for Version 0.6:

Incorporate comments from previous FC-VI working group meetings. Edit Definitions to clarify ‘VI Message’ and ‘VI Connection’. Add ladder diagrams for all FC-VI Exchange operations. Significant rewrite of connection management section. Remove tables from ‘FC-VI IU Formats’ section unless they define Payload data, and add information on Device Header applicibility. Update FARP section.

Release notes for Version 0.7:

Add more information about FC2 Sequence Count and Sequence ID control. Add new section for previously defined FC Definitions. Change FCVI_CONNECT_RESP and FCVI_CONNECT_DONE IU names to be Response IUs. Add FCVI_DISCONNECT_RESP IU and useage rules. Add connection state machine diagrams. Added Annex B about VI QOS.

Release Notes for Version 1.0:

Added section for CS_CTL definition. Added more detail to connection setup operation in ‘FC-VI Structure and Concepts’ section. General document word cleaning.

ANSI ®
dp ANS XXX-199X

draft proposed American National Standard
for Information Technology

Fibre Channel —
Virtual Interface (FC-VI)

Secretariat

Information Technology Industry Council

Approved,199x

American National Standards Institute, Inc.

Abstract

This standard defines the Fibre Channel mapping protocol for the Virtual Interface (VI) Architecture (FC-VI).

.

American National Standard

Approval of an American National Standard requires verification by ANSI that the requirements for due process, consensus, and other criteria for approval have been met by the standards developer.

Consensus is established when, in the judgement of the ANSI Board of Standards Review, substantial agreement has been reached by directly and materially affected interests. Substantial agreement means much more than a simple majority, but not necessarily unanimity. Consensus requires that all views and objections be considered, and that a concerted effort be made towards their resolution.

The use of American National Standards is completely voluntary; their existence does not in any respect preclude anyone, whether he has approved the standards or not, from manufacturing, marketing, purchasing, or using products, processes, or procedures not conforming to the standards.

The American National Standards Institute does not develop standards and will in no circumstances give interpretation on any American National Standard. Moreover, no person shall have the right or authority to issue an interpretation of an American National Standard in the name of the American National Standards Institute. Requests for interpretations should be addressed to the secretariat or sponsor whose name appears on the title page of this standard.

CAUTION NOTICE: This American National Standard may be revised or withdrawn at any time. The procedures of the American National Standards Institute require that action be taken periodically to reaffirm, revise, or withdraw this standard. Purchasers of American National Standards may receive current information on all standards by calling or writing the American National Standards Institute.

PATENT STATEMENT

The developers of this Standard have requested that holder's of patents that may be required for the implementation of the Standard, disclose such patents to the publisher. However, neither the developers nor the publisher have undertaken a patent search in order to identify which, if any, patents may apply to this Standard.

As of the date of publication of this Standard and following calls for the identification of patents that may be required for the implementation of the Standard, no such claims have been made. No further patent search is conducted by the developer or the publisher in respect to any Standard it processes. No representation is made or implied that licenses are not required to avoid infringement in the use of this Standard.

Published by

American National Standards Institute
11 W. 42nd Street, New York, New York 10036

Printed in the United States of America

Draft American National Standard for Information Technology—

Fibre Channel — Virtual Interface Architecture Mapping Protocol (FC-VI)

1 Scope

This standard defines the Fibre Channel mapping protocol for the Virtual Interface (VI) Architecture (FC-VI). FC-VI defines the Fibre Channel Information Units in accordance with the VI Architecture model. FC-VI additionally defines how Fibre Channel services are used to perform the services required by the VI Architecture Model of its network transport.

2 Normative references

The following American National Standards contain provisions which, through reference in this text, constitute provisions of FC-VI. 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.

Copies of the following documents can be obtained from ANSI: Approved ANSI standards, approved and draft international and regional standards (ISO, IEC, CEN/CENELEC, ITUT), and approved and draft foreign standards (including BSI, JIS, and DIN). For further information, contact ANSI Customer Service Department at 212-642-4900 (phone), 212-302-1286 (fax) or via the World Wide Web at http://www.ansi.org.

Reference is made to these documents by their standard designations.

Additional availability contact information is provided below as needed.

2.1 Approved references

ANSI X3.230-1994*, Fibre Channel Physical and Signaling Interface (FC-PH)*

ANSI X3.297-1997*, Fibre Channel Second Generation Physical Interface (FC-PH-2)*

ANSI X3.272-1996, *Information Technology - Fibre Channel - Arbitrated Loop (FC-AL)*

NCITS TR-19*, Fibre Channel Private Loop, SCSI Direct Attach (FC-PLDA)*

NCITS TR-20*, Fibre Channel Fabric Loop Attachment (FC-FLA)*

2.2 References under development

At the time of publication, the following referenced standards were still under development by X3T10. For information on the current status of the document, or regarding availability, contact the NCITS

Secretariat, Information Technology Industry Council (ITI) at 202-737-8888 (phone) or by mail at 1250 Eye Street NW, Suite 200, Washington, DC 20005-3922.

ANSI X3.303-199X, *Fibre Channel Third Generation Physical Interface (FC-PH-3)*

NCITS Project 1311D, *Fibre Channel Framing and Signaling Interface (FC-FS)*

Copies of these X3T10 draft documents are available for purchase from Global Engineering Documents. For further information, contact Global Engineering Documents at 800-854-7179 (phone) or 303-792-2181 (phone) or by mail at 15 Inverness Way East, Englewood, CO 80122-5704

2.3 Other references

The following reference for VI Architecture is the product of Intel, Microsoft, and Compaq. The VI Architecture Specification is completely defined in these two documents. For information on the current status and availability of the document, contact the Intel VI Architecture Web Site at developer.intel.com/design/servers/vi, or the general VI Web site at www.viarch.org.

VI Architecture Specification, V1.0

VI Developer's Guide, V1.0

3 Definitions and abbreviations

Definitions, conventions, abbreviations, acronyms and symbols applicable to this standard are provided, unless they are identical to that described in any referenced standard, in which case they are included by reference. Some definitions from the glossary or body of other standards are included here for easy reference.

3.1 FC-VI Definitions

3.1.1 VI Connection - A virtual connection that is established and maintained between two FC-VI Ports.

3.1.2 VI Message - Application data that is transferred between FC-VI Ports over a previously established VI Connection.

3.1.3 VI Message Transfer - An FC-VI operation that consists of one or more FC-VI IU's to transfer a VI Message between FC-VI Ports.

3.1.4 VI Connection Setup - An FC-VI operation that consists of a sequence of FC-VI Connection IUs which establishes a connection between a pair of FC-VI Ports.

3.1.5 VI End Point - The context for a VI within a FC-VI Port. Each end of a VI Connection is a VI End Point.

3.1.6 VI Connection Point - The context used to listen for VI Connection requests and responses within a FC-VI Port. It is bound to an IP address and a Discriminator.

3.1.7 FC-VI Provider - The hardware and software services that implement a VI Provider over a Fibre Channel transport conforming to this specification.

3.2 VI Definitions

The following VI Terms used in this specification are defined in the VI Architecture Specification and the VI Developer's Guide.

Send

RDMA Read

RDMA Write

Unreliable Delivery

Reliable Delivery

Reliable Reception

Client-Server

Peer-to-Peer

VI Application

Discriminator

3.3 FC-PH Definitions

3.3.1 Information Unit - An organized collection of data specified by FC-4 to be transferred as a single Sequence by FC-2.

3.4 Abbreviations

D_ID	Destination_Identifier[ANSI X3.230]
ELS	Extended Link Service [ANSI X3.330-1994]
FC	Fibre Channel [ANSI X3.230]
FC-PH	The architecture specified by the Fibre Channel standard. [ANSI X3.230]
FCP	Fibre Channel Protocol for SCSI [ANSI X3.269]
FC-4	Fibre Channel Layer 4 mapping layer. [ANSI X3.230]
IU	Information Unit [ANSI X3.230]
S_ID	Source_Identifier[ANSI X3.230]
ULP	Upper Layer Protocol [ANSI X3.230]

3.5 Editorial conventions

In this Standard, a number of conditions, mechanisms, sequences, parameters, events, states, or similar terms are printed with the following conventions:

– the first letter of each word in uppercase and the rest lowercase (e.g., Exchange, Class, etc.).

All terms and words not conforming to the conventions noted above have the normal technical English meanings.

Numbered items in this International Standard do not represent any priority. Any priority is explicitly indicated.

In case of any conflict between text, figure, table and state diagram, the state diagram, then table, then figure, and finally, text takes precedence. Exceptions to this convention are indicated in the appropriate subclauses.

The term “shall” is used to indicate a mandatory rule. If such a rule is not followed, the results are unpredictable unless indicated otherwise.

The fields or control bits which are not applicable shall be reset to zero.

If a field or a control bit in a frame is specified as not meaningful, the entity which receives the frame shall not check that field or control bit.

Hexadecimal notation

Hexadecimal notation is used to represent fields. For example, a four-byte VI_Handle field containing a binary value of 00000000 11111111 10011000 11111010 is shown in hexadecimal format as ‘0x00FF98FA’.

4 Structure and Concepts

4.1 Fibre Channel Structure and Concepts

Fibre Channel (FC) is logically a point-to-point serial data channel. The architecture has been designed so that it can be implemented with high performance hardware that requires little real-time software management. The Fibre Channel Physical (FC-PH) layer described by X3.230-1994 performs those functions required to transfer data from one N_Port to another. The FC-PH layer can be treated as a very powerful delivery service with information grouping and multiple defined classes of service.

A switching fabric allows communication among more than two N_Ports.

The Fibre Channel Arbitrated Loop (FC-AL) is an alternate FC Port interconnect architecture that uses the FC mechanisms to transfer data between an NL_Port selected by the arbitration process and any of the other NL_Ports on the loop. Once communication is opened between two NL_Ports, standard FC frames are used to provide an FC-PH compliant delivery service.

An FC-4 mapping layer uses the services provided by FC-PH to execute the steps required to perform the functions defined by the ULP. The protocol is defined in terms of the stream of FC IU's generated by a pair of N_Ports or NL_Ports that support the FC-4. In this document, N_Ports and NL_Ports capable of supporting FC-VI transactions are collectively referred to as FC-VI Ports.

The number of Exchanges that may simultaneously be open between two FC-VI Ports is defined by the FC-PH implementation. The architectural limit for this value is 65,535. The maximum number of active Sequences that can simultaneously be open between two FC-VI Ports is defined by the FC-PH SEQ_ID as 256. To allow management Exchanges to be originated, a certain number of extra Exchanges and at least one extra Sequence_ID should always be available.

FC-PH allows management protocols above the FC-PH interface to perform link data functions. The standard FC-PH primitive sequences, link management protocols, and basic and extended link services are used as required by FC-VI endpoints.

4.2 FC-VI Structure and Concepts

FC-VI defines the mapping of VI Messages and VI Connections to FC-PH. FC-VI is based on the concept of first establishing a VI Connection between two FC-VI Ports, and then sending VI Messages over the VI Connection. FC-VI defines the Information Units (IU) that are required to establish and remove VI Connections, and the IU's that are required to send VI Messages.

Each VI Message is transferred in one Exchange, and each VI Connection is established in one or two Exchanges. Each FC-VI Information Unit (IU) is constructed from one Sequence. One or more IUs are grouped together to form one Exchange. FC-VI defines Connection IU's to establish and remove VI Connections, and defines Message IU's to send VI Messages.

FC-VI defines the concept of a VI_Handle to specify the destination VI for the VI Message. FC-VI Ports will send FC-VI Connection IU's between them to establish the VI_Handles. An FC-VI Port is allowed to choose its own VI_Handle for each of its VI End Points. The FC-VI Port is required to use the VI_Handle specified by the other FC-VI port when transmitting a VI message.

FC-VI also defines the concept of a VI Message ID (VI_MSG_ID). This field is assigned by the FC-VI Port that sends the first IU of the VI Message. The FC-VI Port that receives the VI Message must echo the same VI_MSG_ID in any Response IU. Each FC-VI Port must increment the VI_MSG_ID value by one for every VI Message sent within a VI to ensure detection of lost VI Messages at the receiver. A separate VI_MSG_ID count is kept for each direction of a VI.

VI reliability levels are defined with respect to the VI Message delivery. FC-VI supports all three levels of reliability (Unreliable Delivery, Reliable Delivery, Reliable Reception) and all connection models (Client-Server, Peer-to-Peer) defined by the VI Architecture. FC-VI supports all VI data transfer models (Send, RDMA Write and RDMA Read). FC-VI supports Class 2 and Class 3 Classes. FC-VI supports Arbitrated-Loop, Fabric, Loop attached Fabric, and Point-to-Point topologies.

VI does not specify how to resolve names to addresses. FC-VI maps an FC-VI Port Identifier to an IP Address. A Name Server is the preferred method to map an FC-VI Port Identifier to an IP Address. FC-VI additionally specifies the Fibre Channel FARP model to resolve IP Addresses to VI Port Identifiers for configurations that do not support a Name Server.

Figure XX illustrates the addressing objects associated with a FC-VI Port.

FC-VI Ports are assigned IP addresses. The method of assigning IP addresses is outside the scope of this standard. The format of an IP address is IPv6.

Before VI Messages are exchanged, a VI Connection must be established between a pair of VI End Points. A VI End Point must be created before it is connected. A FC-VI Port may need to discover the Port Identifier of the FC-VI Port it wishes to send a connection request to. This is accomplished by querying the Name Server or issuing a FARP-REQ.

A VI Application listens for incoming connection requests on a Connection Point by executing a VipConnectWait for a Client-Server connection or VipConnectPeerRequest for a Peer-to-peer connection. A Connection Point is bound to an IP address and a Discriminator. A Discriminator is typically unique per Connection Point, but it is not required to be so.

An incoming connect request, VI_CONNECT_RQST, contains both the local and remote IP addresses and Discriminators. A connection match exists if the IP address and Discriminator in the VI_CONNECT_RQST payload match the IP address and Discriminator in the Connection Point. See Section 6.7 in the VI Developer's Guide for the exact matching rules for connection setup. Note that the attributes of the VI End Point - VI_ATTRIBUTES - must also match.

During VI Connection Setup, each VI End Point comminicates a VI_HANDLE to the other VI End Point. Once the VI Connection is established, the VI_HANDLE is used to route all VI Messages to the proper VI End Point within a FC-VI Port. IP addresses and Discriminators are no longer used.

This process of establishing a VI Connection is defined as a VI Connection Setup.

Implementor's Note: VI Messages are typically processed completely in hardware, while VI Connection Setup is typically processed in software.

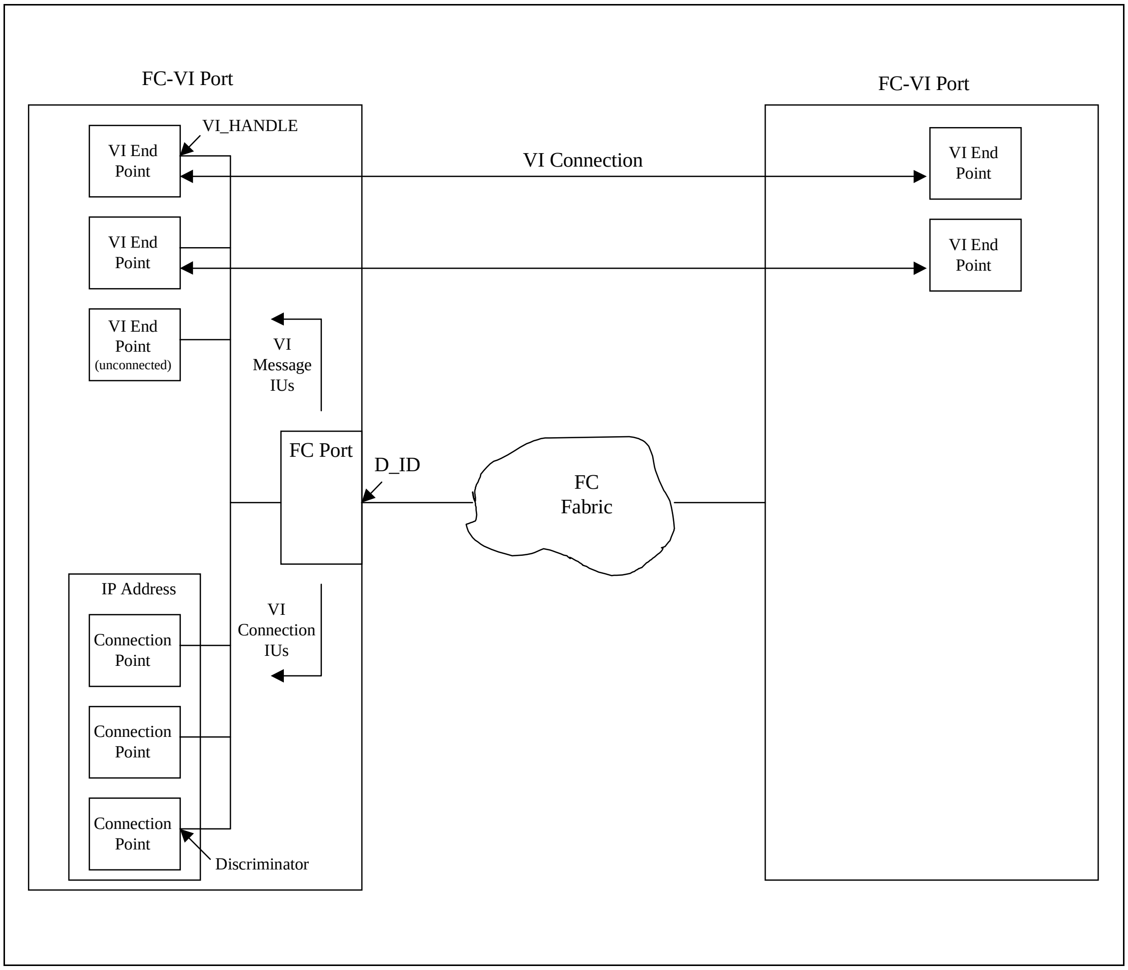

Figure 1 – FC_VI Addressing Objects

5 FC-VI protocol overview

5.1 FC-VI Information Units

The information units used by FC-VI Ports and their characteristics are shown in Table 1 below. All FC-VI IUs shall be composed of Device_Data type frames. Different R_CTL encodings are used so that FC-2 hardware may appropriately route the IUs without needing to look at the VI_OPCODE. It is important for FC-2 HBA hardware assists to make is a clear distinction between the Message Request/Response IU's and the Connection Request/Response IU's. Therefore all of the Connection Request/Response IU's are either Unsolicited Control or Solicited Control. For example, this would allow the HBA FC-2 Hardware to route the Message Request/Response IU's directly to special VI data transfer assist hardware and route the Connection Request/Response IU's to a more general purpose processing path.

The following operations are mandatory in VI: Send, RDMA Write. RDMA Read is optional. The reliability modes that shall be mandatory for this specification are Unreliable Delivery and Reliable Delivery. Reliable Reception is optional. The Table 1 column 'Mand/Opt' defines which IUs are mandatory and optional. FCVI_READ_RQST is optional since RDMA Read is optional. All Message Responses are optional, since they are only required for RDMA Read or Reliable Reception. All connection IUs are mandatory. If a FC-VI implementation decides to support Reliable Reception, it shall also support streaming of VI Messages. Since VI requires that VI Messages are completed in order in Reliable Reception mode, then a Message Response IU shall only be sent after all of the expected VI Messages have been received (according to VI_MSG_ID field).

Table 1 – FC-VI Information Unit Summary

IU Category	FC-VI Information Unit Type	R_CTL	DF_CTL DH Size	FC-VI Opcode	Payload Content	Mand / Opt
Message Request	FCVI_SEND_RQST	0x01	16 Bytes	0x00	Message Data	M
Message Request	FCVI_WRITE_RQST	0x01	32 Bytes	0x01	Message Data	M
Message Request	FCVI_READ_RQST	0x06	32 Bytes	0x02	none	O
Message Response	FCVI_SEND_RESP	0x07	16 Bytes	0x08	none	O
Message Response	FCVI_WRITE_RESP	0x07	16 Bytes	0x09	none	O
Message Response	FCVI_READ_RESP	0x01	16 Bytes	0x0A	Message Data	O
Connection Request	FCVI_CONNECT_RQST	0x02	16 Bytes	0x10	Connect Info	M
Connection Request	FCVI_DISCONNECT_RQST	0x02	16 Bytes	0x12	none	M
Connection Response	FCVI_CONNECT_RESP1	0x03	16 Bytes	0x18	Connect Info	M
Connection Response	FCVI_CONNECT_RESP2	0x03	16 Bytes	0x19	none	M
Connection Response	FCVI_DISCONNECT_RESP	0x03	16 Bytes	0x1A	none	M

5.2 VI Message Exchange Operation

Each VI Message shall be transferred in a single Exchange, and each Exchange shall contain only one VI Message. A VI Message Transfer consists of a Request IU and Response IU, or a Request IU with no Response IU. Exchanges are dynamically created by the Fibre Channel transport mechanisms to move the VI Message from one VI Port to another. The Exchange shall be destroyed at the completion of the VI Message delivery. Utilizing the same Exchange for a Message Request IU and a Message Response IU (if needed) binds all Sequences and Frames associated with a VI Message.

NOTE – The FC-VI mapping of VI Connections to FC-PH Exchanges is dynamic to allow the number of VI Connections to be independent from the number of Exchanges. The maximum number of VI Connections is 2^{32} per port, limited by the VI_HANDLE field in the FC-VI Device_Header.

The following sub-sections will detail all of the different cases for VI Messages mapping into one FC Exchange. These sub-sections only define error free operation for FC-VI operation. Sequence and Exchange level error recovery mechanisms defined by FC-PH are used when needed.

5.2.1 FC-VI Send Message Operation

The FC-VI Send Message can be sent in Unreliable Delivery or Reliable Delivery mode, as shown in Figure 2, or sent in Reliable Reception mode, as shown in Figure 3. The Message Originator may send one or multiple FCVI_SEND_RQST IUs. No Response IU is allowed for Unreliable Delivery or Reliable Delivery modes. One FCVI_SEND_RESP IU is required for Reliable Reception mode. Each Figure below represents one FC Exchange, and each arrow represents one FC Sequence.

FC-VI Port A (Port A’s VI Handle = HA, Last Message ID Originated was MA)	**FC-VI Port B** (Port B’s VI Handle = HB, Last Message ID Originated was MB)
FCVI_SEND_RQST IU (Device Header details: VI_HANDLE=HB, VI_MSG_ID=MA+1)	→
FCVI_SEND_RQST IU Optional additional IU’s as needed (Device Header details: VI_HANDLE=HB, VI_MSG_ID=MA+1)	→

Figure 2 – FC-VI Send for Unreliable Delivery or Reliable Delivery

FC-VI Port A (Port A's VI Handle = HA, Last Message ID Originated was MA)	**FC-VI Port B** (Port B's VI Handle = HB, Last Message ID Originated was MB)
FCVI_SEND_RQST IU (Device Header details: VI_HANDLE=HB, VI_MSG_ID=MA+1)	→
FCVI_SEND_RQST IU Optional additional IU's as needed (Device Header details: VI_HANDLE=HB, VI_MSG_ID=MA+1)	→
←	**FCVI_SEND_RESP IU** (Device Header details: VI_HANDLE=HA, VI_MSG_ID=MA+1)

Figure 3 – FC-VI Send for Reliable Reception

5.2.2 FC-VI RDMA Write Message Operation

The FC-VI RDMA Write Message can be sent in Unreliable Delivery or Reliable Delivery mode, as shown in Figure 4, or sent in Reliable Reception mode, as shown in Figure 5. The Message Originator may send one or multiple FCVI_WRITE_RQST IUs. No Response IU is allowed for Unreliable Delivery or Reliable Delivery modes. One FCVI_WRITE_RESP IU is required for Reliable Reception mode. Each Figure below represents one FC Exchange, and each arrow represents one FC Sequence.

FC-VI Port A (Port A's VI Handle = HA, Last Message ID Originated was MA)	**FC-VI Port B** (Port B's VI Handle = HB, Last Message ID Originated was MB)
FCVI_WRITE_RQST IU (Device Header details: VI_HANDLE=HB, VI_MSG_ID=MA+1)	→
FCVI_WRITE_RQST IU Optional additional IU's as needed (Device Header details: VI_HANDLE=HB, VI_MSG_ID=MA+1)	→

Figure 4 – FC-VI RDMA Write for Unreliable Delivery or Reliable Delivery

FC-VI Port A (Port A's VI Handle = HA, Last Message ID Originated was MA)	FC-VI Port B (Port B's VI Handle = HB, Last Message ID Originated was MB)
FCVI_WRITE_RQST IU (Device Header details: VI_HANDLE=HB, VI_MSG_ID=MA+1)	⟶
FCVI_WRITE_RQST IU Optional additional IU's as needed (Device Header details: VI_HANDLE=HB, VI_MSG_ID=MA+1)	⟶
⟵	**FCVI_WRITE_RESP IU** (Device Header details: VI_HANDLE=HA, VI_MSG_ID=MA+1)

Figure 5 – FC-VI RDMA Write for Reliable Reception

5.2.3 FC-VI RDMA Read Message Operation

The FC-VI RDMA Read Message can be sent in Reliable Delivery, or Reliable Reception mode, as shown in Figure 4. The Message Originator can send one or multiple FCVI_READ_RESP IUs. A Response IU is required for all delivery modes. Each Figure below represents one FC Exchange, and each arrow represents one FC Sequence.

FC-VI Port A (Port A's VI Handle = HA, Last Message ID Originated was MA)	**FC-VI Port B** (Port B's VI Handle = HB, Last Message ID Originated was MB)
FCVI_READ_RQST IU (Device Header details: VI_HANDLE=HB, VI_MSG_ID=MA+1)	→
←	**FCVI_READ_RESP IU** (Device Header details: VI_HANDLE=HA, VI_MSG_ID=MA+1)
←	**FCVI_READ_RESP IU** Optional additional IU's as needed (Device Header details: VI_HANDLE=HA, VI_MSG_ID=MA+1)

Figure 6 – FC-VI RDMA Read for Reliable Reception and Reliable Delivery

5.3 VI Connection Establishment Operation

For any FCVI Connection Setup, which includes Client-Server and Peer-to-peer, connection IUs are sent over a single Exchange. For a Peer-to-peer Connection Setup, each Peer issues a connection request in a separate Exchange.

Each connection IU is a separate sequence. Sequence Initiative is always passed on the FCVI_CONNECT_RQST and FCVI_CONNECT_RESP1 IUs. The FCVI_CONNECT_RQST is the first Sequence of the Exchange, FCVI_CONNECT_RESP1 IU is the middle Sequence of the Exchange, and FCVI_CONNECT_RESP2 IU is the last Sequence of the Exchange.

The following sub-sections will detail all of the different cases for VI Connections being established or removed. These sub-sections only define error free operation for FC-VI operation. Sequence and Exchange level error recovery mechanisms defined by FC-PH are used when needed.

5.3.1 FC-VI Connection State Machine Diagram

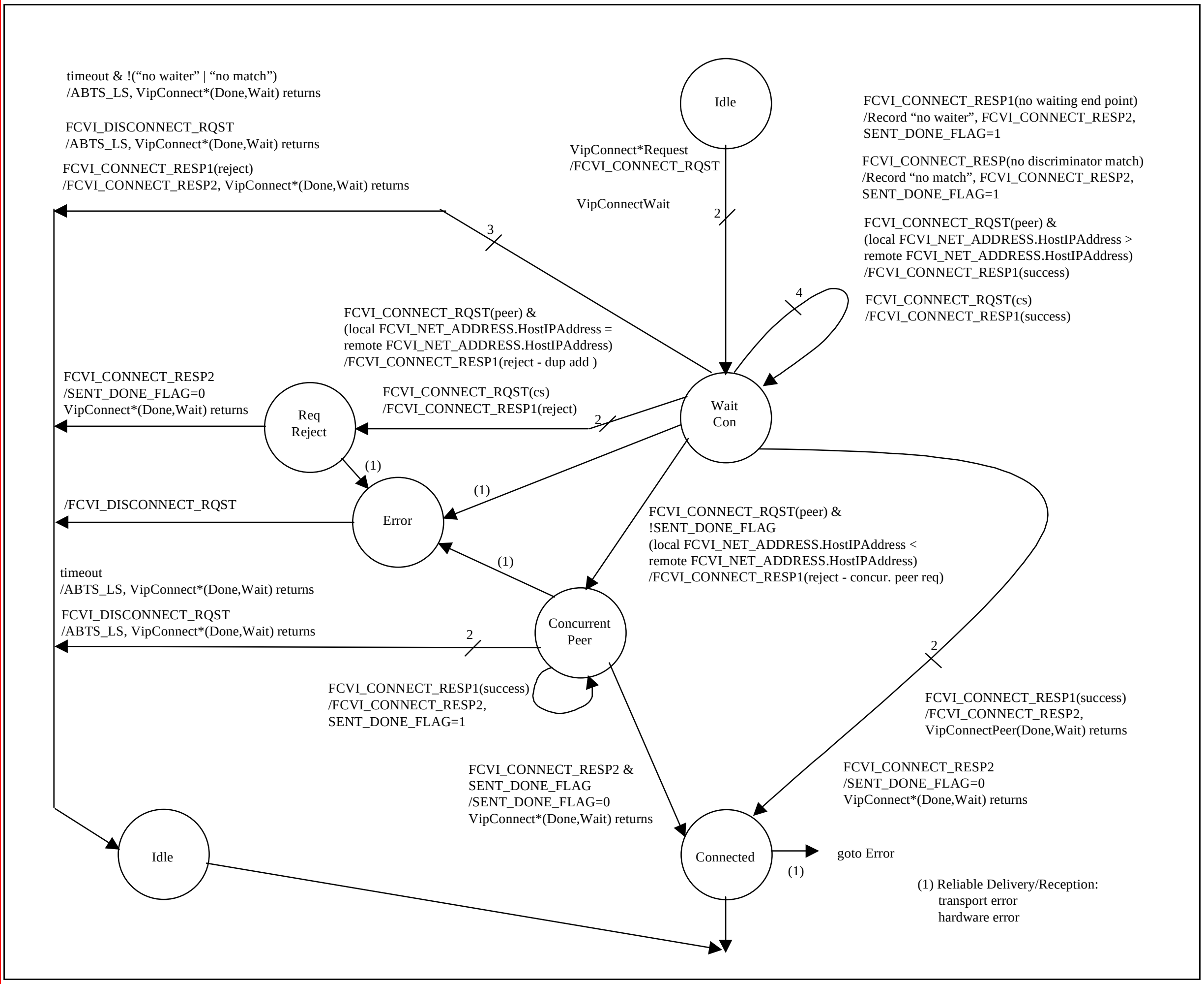

Figure 7 – FC_VI Connection State Machine

5.3.2 FC-VI Client-Server Connection Establishment

Figure 8 below describes the FC-VI Client-Server connection setup. The Client is Port A while the Server is Port B.

The VI_FLAGS = 0 in the FCVI_CONNECT_RESP1 IU indicate that the Server has accepted the Client's connection request. The VI_FLAGS = 0 in the FCVI_CONNECT_RESP2 indicate that the Client acknowledges the Server's connect accept.

FC-VI Port A (Client) Port A's VI Handle = HA	**FC-VI Port B (Server)** Port B's VI Handle = HB
FCVI_CONNECT_RQST IU First IU of Exchange (Device Header details: VI_HANDLE=0XFFFFFFFF, VI_RQST_HANDLE=HA, VI_MSG_ID=0)	→
←	**FCVI_CONNECT_RESP1 IU** Middle IU of Exchange (Device Header details: VI_HANDLE=HA, VI_ACPT_HANDLE=HB, VI_FLAGS=0, VI_MSG_ID=0)
FCVI_CONNECT_RESP2 IU Last IU of Exchange (Device Header details: VI_HANDLE=HB, VI_FLAGS=0, VI_MSG_ID=0)	→

Figure 8 – FC-VI Client-Server Connection Setup

5.3.3 FC-VI Peer-to-Peer Connection Establishment

Figure 9 below describes the FC-VI Peer-to-Peer connection protocol. Peer 1 is Port A, while Peer 2 is Port B. In the first case, the VI Application in Peer 2 has not issued a peer connection request when the FCVI_CONNECT_RQST IU is received from Peer 1. Peer 2 replies with a CONN_ERR and a

Reason Code of “No Waiting Remote Endpoint”. At some later point Peer 2 issues a matching FCVI_CONNECT_RQST IU and a connection is established between Port A and Port B.

FC-VI Port A (Peer 1) Port A’s VI Handle = HA	**FC-VI Port B (Peer 2)** Port B’s VI Handle = HB
FCVI_CONNECT_RQST IU First IU of Exchange 1 (Device Header details: VI_HANDLE=0xFFFFFFFF, VI_RQST_HANDLE=HA, VI_MSG_ID=0)	→
←	**FCVI_CONNECT_RESP1 IU** Middle IU of Exchange 1 (Device Header details: VI_HANDLE=HA, VI_ACPT_HANDLE=0xFFFFFF, VI_FLAGS=CONN_ERR, VI_PARAMETER.REASON_CODE= “No Waiting Remote Peer”, VI_MSG_ID=0)
FCVI_CONNECT_RESP2 IU Last IU of Exchange 1 (Device Header details: VI_HANDLE=0xFFFFFFFF, VI_FLAGS=0, VI_MSG_ID=0)	→
←	**FCVI_CONNECT_RQST IU** First IU of Exchange 2 (Device Header details: VI_HANDLE=0xFFFFFFFF, VI_RQST_HANDLE=HB, VI_FLAGS=0, VI_MSG_ID=0)
FCVI_CONNECT_RESP1 IU Middle IU of Exchange 2 (Device Header details: VI_HANDLE=HB, VI_ACPT_HANDLE=HA, VI_FLAGS=0, VI_MSG_ID=0)	→
←	**FCVI_CONNECT_RESP2 IU** Last IU of Exchange 2 (Device Header details: VI_HANDLE=HA, VI_FLAGS=0, VI_MSG_ID=0)

Figure 9 – Peer-to-Peer Connection Setup

5.3.4 FC_VI Concurrent Peer-to-Peer Connection Establishment

Figure 10 below describes the FC-VI Peer-to-Peer connection protocol when each peer issues a matching FCVI_CONNECT_RQST IU before a FCVI_CONNECT_RESP1 IU is received. Peer 1 is Port A, while Peer 2 is Port B. This situation is defined as concurrent peer-to-peer connection requests.

For concurrent peer-to-peer connection requests, the following arbitration protocol is used to ensure that only one peer will accept the connection:

1. If my Host Address > peer's Host Address, then accept connection request.
2. Else if my Host Address < peer's Host Adress, then reject connection request by setting CONN_ERR equal to one with a Reason Code of "Connect Reject - Concurrent Peer Requests".
3. Else my Host Address = peer's Host Address, then reject connection request by setting CONN_ERR equal to one with a Reason Code of "Connect Reject - Duplicate Addresses".

The comparison used in the arbitration protocol for concurrent Peer requests is a numerical comparison between two IP addresses.

In the case illustrated, Peer 1's Host Address is numerically greater than Peer 2's Host Address. Peer 1 accepts the connection request while Peer 2 rejects it. Thus, only one peer will complete the connection..

FC-VI Port A (Peer 1) Port A's VI Handle = HA, Port A's VI Host Address = 1.2.3.5	**FC-VI Port B (Peer 2)** Port B's VI Handle = HB, Port B's VI Host Address = 1.2.3.4
FCVI_CONNECT_RQST IU First IU of Exchange 1 (Device Header details: VI_HANDLE=0xFFFFFFFF, VI_RQST_HANDLE=HA, VI_MSG_ID=0) →	**FCVI_CONNECT_RQST IU** First IU of Exchange 2 (Device Header details: VI_HANDLE=0xFFFFFFFF, VI_RQST_HANDLE=HB, VI_FLAGS=0, VI_MSG_ID=0) ←
FCVI_CONNECT_RESP1 IU Middle IU of Exchange 1 (Device Header details: VI_HANDLE=HB, VI_ACPT_HANDLE=HA, VI_FLAGS=0, VI_MSG_ID=0) →	**FCVI_CONNECT_RESP1 IU** Middle IU of Exchange 1 (Device Header details: VI_HANDLE=HA, VI_ACPT_HANDLE=0xFFFFFF, VI_FLAGS=CONN_ERR, VI_PARAMETER.REASON_CODE= "Connect Reject - Concurrent Peer Requests", VI_MSG_ID=0) ←
FCVI_CONNECT_RESP2 IU Last IU of Exchange 1 (Device Header details: VI_HANDLE=0xFFFFFFFF, VI_FLAGS=0, VI_MSG_ID=0) →	**FCVI_CONNECT_RESP2 IU** Last IU of Exchange 2 (Device Header details: VI_HANDLE=HA, VI_FLAGS=0, VI_MSG_ID=0) ←

Figure 10 – Peer-to-Peer Connection Setup, Concurrent Peer Requests

5.3.5 FC-VI Disconnect Operation

Figure 11 below shows the method to disconnect a VI. A VI Application decides when it is appropriate to disconnect, according to the VI specification. A disconnect operation can be asynchronously initiated from either FC-VI Port by sending a Disconnect Request IU. The other FC-VI Port will respond with a Disconnect Response IU. The FC-VI Flags field indicates who is sending it and why.

FC-VI Port A (Port A's VI Handle = HA, Last Message ID Originated was MA)	**FC-VI Port B** (Port B's VI Handle = HB, Last Message ID Originated was MB)
FCVI_DISCONNECT_RQST IU (Device Header details: VI_HANDLE=HB, VI_MSG_ID=MA+1)	→
←	**FCVI_DISCONNECT_RESP IU** (Device Header details: VI_HANDLE=HA, VI_MSG_ID=MA+1)

Figure 11 – FC-VI Disconnect Operation

5.4 Sequence ID reuse

For FC-VI, all Information Units shall be represented by one FC Sequence. The Sequence ID can be reused by the Sequence initiator as soon as the Sequence is closed. In Class 3, the Sequence initiator shall consider a Sequence closed when the last frame is sent. In Class 2, the Sequence initiator shall consider a Sequence closed when the ACK from the last frame of the Sequence is received[2].

FC-VI frames shall use a continuously increasing sequence count for each Exchange to prevent out of order IU reception. Sequence Count shall be zero at the start of the Exchange. Sequence count shall not wrap within one IU. Figure 12 shows FC2 header operation during an FC-VI Exchange.

NOTE – Sequence streaming in FC-PH requires a different SEQ_ID in subsequent sequences.

2. The FC-VI Device_Header provides enough information to prevent synonyms in all cases.

FC-VI Port A	FC-VI Port B
Message Request Frame first frame of IU (FC2 Header details: SEQ_ID=S1, SEQ_CNT=0, OX_ID=OX1)	→
Message Request Frame last frame of IU (FC2 Header details: SEQ_ID=S1, SEQ_CNT=N, OX_ID=OX1)	→
Message Request Frame first frame of IU (FC2 Header details: SEQ_ID=S2, SEQ_CNT=N+1, OX_ID=OX1)	→
Message Request Frame last frame of IU (FC2 Header details: SEQ_ID=S2, SEQ_CNT=M, OX_ID=OX1)	→
←	**Message Response Frame** first and last frame of IU (FC2 Header details: SEQ_ID=S3, SEQ_CNT=0, OX_ID=OX1)

Figure 12 – FC2 Header for Send Operation

5.5 VI Message length

VI Messages can be a maximum length of 2^{32} -1 bytes in length as specified by VI.

5.6 FC-PH Header Usage for FC-VI

The format of the standard FC-PH header as used by FC-VI is defined in Table 2.

All fields use the standard FC-PH definitions. The following explanations of the fields provide information about the use of those fields to implement VI functionality. If a field is not defined below, then its useage is the same as defined by FC-PH.

5.6.1 R_CTL field

Together with the TYPE field, the R_CTL field identifies the frame as part of an FC-VI operation and identifies the information category. Refer to the IU definitions in Table 1 for specific R_CTL encodings.

Table 2 – FC-VI frame header

Bits Word	31– 24	23–16	15–08	07–00
0	R_CTL	D_ID		
1	CS_CTL	S_ID		
2	TYPE	F_CTL		
3	SEQ_ID	DF_CTL	SEQ_CNT	
4	OX_ID		RX_ID	
5	RLTV_OFF			

5.6.2 CS_CTL Field

FC-VI Providers that choose to make use of the PREF bit defined in the CS_CTL field of the FC-PH header shall conform to the following policy: Use of the Preference Function shall be on a per VI connection basis. If a particular VI connection is to make use of the preference function then each and every Fibre Channel frame sent on behalf of that VI Connection shall have the PREF bit set to 1 in the CS_CTL field. If a particular VI connection is not making use of the preference function then each and every Fibre Channel frame sent on behalf of that VI Connection shall have the PREF bit set to 0 in the CS_CTL field.

5.6.3 TYPE field

The TYPE field shall be 0x58 for all frames of FC-VI Sequences.

5.6.4 DF_CTL field

The DF_CTL field indicates any optional headers that may be present. FC-VI IUs require either a 16 or 32 byte Device_Header. The Device_Header is present in all frames of an FC-VI IU. The size of the Device_Header associated with each IU is defined in Table 1, in the DH Size column.

5.6.5 SEQ_CNT field

The SEQ_CNT field indicates the frame order within the sequence as defined by FC-PH. FC-VI shall require that SEQ_CNT shall be continuously increasing within any one Exchange for every frame sent by the requester or responder. The Sequence Initiator shall not transmit more than 65536 frames per sequence.

5.6.6 RLTV_OFF Field

The RLTV_OFF field shall indicate the relative displacement of the first byte of each frame's payload, referenced to a starting value of 0. For RDMA operations, the RLTV_OFF value of each frame is added to the VI_RMT_VA field in the Device Header to determine the starting address for the buffer read or write.

5.7 FC-VI Device_Header

Every FC-VI Frame will contain a Device_Header. For FCVI_READ_RQST or FCVI_WRITE_RQST IU's, a 32 byte Device_Header is used, as shown in Table 3. For all other IU's, as 16 byte Device_Header is used, as shown in Table 4.

Table 3 – 32-Byte FC-VI Device_Header

Word	Byte 0	Byte 1	Byte 2	Byte 3
0	VI_HANDLE			
1	VI_OPCODE	VI_FLAGS	VI_RSVD	
2	VI_MSG_ID			
3	VI_PARAMETER			
4	VI_RMT_VA			
5				
6	VI_RMT_VA_HANDLE			
7	VI_RMT_TOT_LEN			

Table 4 – 16-Byte FC-VI Device_Header

Word	Byte 0	Byte 1	Byte 2	Byte 3
0	VI_HANDLE			
1	VI_OPCODE	VI_FLAGS	VI_RSVD	
2	VI_MSG_ID			
3	VI_PARAMETER			

5.7.1 VI_OPCODE Field

The VI_OPCODE field completely specifies the FC-VI IU type and format. Table 1 indicates the VI_OPCODE assigned to the various FC-VI IU's. There are four different IU categories defined. The VI Connection Request/Response IU's are used for creating and destroying VI connections between VI Ports. The VI Message Request/Response IU's are used for Message transfer only after the VI connection is already established.

5.7.2 VI_FLAGS Field

The VI_FLAGS field contains control flags for the VI operation. Reserved bits shall always be zero. The bit positions and definitions for all Message Request IU's are shown in Table 5. The bit positions and definitions for all Message Response IU's are shown in Table 6. The bit positions and definitions for all Connection Request IU's are shown in Table 7. The bit positions and definitions for all Connection Response IU's are shown in Table 8.

Table 5 – VI_FLAGS Bit Definitions for Message Request IU's

Bit Position	Bit Name	Bit Definition
7	IMM_DATA	VI_PARAMETER field is Immediate Data when one.
6	Reserved	0
5	Reserved	0
4	Reserved	0
3	Reserved	0
2	Reserved	0
1	Reserved	0
0	Reserved	0

Table 6 – VI_FLAGS Bit Definitions for Message Response IU's

Bit Position	Bit Name	Bit Definition
7	Reserved	0
6	Reserved	0
5	Reserved	0
4	Reserved	0
3	TRANS_ERR	Transport Error when one.
2	PROT_ERR	DMA Protection Error when one.
1	DESC_ERR	Remote Descriptor Error when one.
0	RESP_ERR	Response IU successful when zero, errored when one.

The RESP_ERR bit is set when the responder wishes to indicate failure of the Request IU. The definitions of the TRANS_ERR, PROT_ERR, and DESC_ERR bits are in the VI Architecture Specification.

Table 7 – VI_FLAGS Bit Definitions for Connection Request IU's

Bit Position	Bit Name	Bit Definition
7	Reserved	0
6	Reserved	0
5	Reserved	0
4	Reserved	0
3	Reserved	0
2	MODE2	Connection Mode bit 2
1	MODE1	Connection Mode bit 1
0	MODE0	Connection Mode bit 0

The Connection Mode bits 2:0 shall define the connection mode requested. The currently defined values for connection setup are 01 for Client Server mode, and 02 for Peer to Peer. The currently defined values for disconnection are 04 for kernel initiated, and 05 for application initiated disconnect. All other values are reserved.

Table 8 – VI_FLAGS Bit Definitions for Connection Response IU’s

Bit Position	Bit Name	Bit Definition
7	Reserved	0
6	Reserved	0
5	Reserved	0
4	Reserved	0
3	Reserved	0
2	Reserved	0
1	Reserved	0
0	CONN_ERR	Connection Error when one.

The CONN_ERR bit is set to one when a Connection Error is detected. It may be set in the FCVI_CONNECT_RESP1 IU or the FCVI_CONNECT_RESP2 IU. If set, the VI_PARAMETER field will indicate the reason for the error.

5.7.3 VI_RSVD Field

All of the bits in the VI_RSVD field shall be set to zero by the frame sender. The receiver shall ignore all bit values in the VI_RSVD field.

5.7.4 VI_HANDLE Field

The VI_HANDLE field is used by the FC-VI Port to uniquely identify the VI Connection for the incoming FC-VI IU. The FC-VI Port sending the IU shall put the correct VI_HANDLE value into every IU when the VI Connection has been previously established. The VI_HANDLE value for each VI End Point of a VI Connection is established during VI Connection Setup. Each FC-VI Port chooses its own VI_HANDLE value for the other FC-VI Port to use. Values from 0x00000000 to 0xFFFFFFFE are valid VI Handles, and 0xFFFFFFFF is reserved for an undefined value that shall be used during VI Connection Setup.

5.7.5 VI_MSG_ID Field

The VI_MSG_ID field is used in conjunction with the VI_OPCODE to uniquely identify a VI Message to the VI ULP within a particular VI Connection. The VI_MSG_ID is also used to associate VI Response IU’s to VI Request IU’s. The VI_MSG_ID field also allows the responder to accurately detect lost messages. Each requestor maintains a VI_MSG_ID count for messages sent on each of its open VI Connections. The VI_MSG_ID shall be assigned by the requestor for all Request IU’s. The requestor shall use a value of 0x00000000 on the first message of the VI. Subsequent messages shall increment the VI_MSG_ID by one. When the VI_MSG_ID reaches 0xFFFFFFFF, the requestor shall wrap the VI_MSG_ID to 0x0000000 on the next Message. The responder shall return the same VI_MSG_ID value on all corresponding Response IU’s.

5.7.6 VI_PARAMETER Field

The format of the VI_PARAMETER field is specified by the VI_FLAGS field.

For VI Message Request IUs (except FCVI_READ_RQST), if the VI_FLAGS field has the IMM_DATA bit asserted, then the VI_PARAMETER field carries the VI Immediate Data from the VI descriptor. Otherwise, the VI_PARAMETER field shall be all zeroes for Message Request IUs.

For Message Response IUs, the Parameter field shall be set to zero.

For Connection Request IUs, the Parameter field shall be set to zero.

For VI Connection Response IUs, if the CONN_ERR bit is asserted in the VI_FLAGS field, then the VI_PARAMETER field contains additional information for the error as shown in Table 9. The Reason Codes are defined in Table 10.

Table 9 – VI_PARAMETER Field for VI Connection Response IUs

VI_PARAMETER Field	Byte Offset
Reserved	0
Reason Code	1
Reason Explanation	2
Vendor Unique	3

Table 10 – Reason Code for CONN_ERR

Code Point	Reason Code
0x01	Connect Reject
0x02	No Descriminator Match
0x03	Connect Timeout
0x04	No Waiting Remote Endpoint
0x05	Connect Reject - Concurrent Peer Requests
0x07	Connect Reject - Duplicate Addresses
0x07 to 0xFF	Reserved

5.7.7 VI_RMT_VA Field

The VI_RMT_VA field specifies the remote virtual address for VI_RDMA_READ_RQST or VI_RDMA_WRITE_RQST IU’s.

5.7.8 VI_RMT_VA_HANDLE Field

The VI_RMT_VA_HANDLE field specifies the remote virtual address handle for VI_RDMA_READ_RQST or VI_RDMA_WRITE_RQST IU’s.

5.7.9 VI_RMT_TOT_LEN Field

The VI_RMT_TOT_LEN field specifies the VI Message total data length for VI_RDMA_READ_RQST IU’s.

6 FC-VI Information Unit (IU) Formats

This clause describes the Device Header and Payload contents of the IU's transferred during FC-VI operations. The information units used by FC-VI Ports and their characteristics are shown in Table 1.

6.1 FCVI_SEND_RQST IU

The FCVI_SEND_RQST IU will send all or part of a VI Message to the remote VI End Point at the other end of the VI Connection. One or more FCVI_SEND_RQST IU shall be sent per Exchange. Refer to Figure 2 or Figure 3 to define the complete Exchange. The Payload of each frame in this IU carries VI application data.

Every frame of the IU shall contain the same values in the Device Header. The VI_OPCODE field shall be 0x00 for this IU. The VI_FLAGS field is defined by Table 5. Sending VI Immediate Data is optional with this IU. If Immediate Data is sent, then the IMM_DATA bit in the VI_FLAGS field shall be one, and the VI_PARAMETER field shall contain the Immediate Data value. Otherwise, all of these fields shall be zero. The MSG_ID field shall be incremented by one from the previously sent message on the connection.

6.2 FCVI_SEND_RESP IU

The FCVI_SEND_RESP IU will respond to a FCVI_SEND_RQST for Reliable Reception mode of operation. At most one FCVI_SEND_RESP IU shall be sent per VI Message. This IU is not allowed for Unreliable Delivery or Reliable Delivery modes of operation. Refer to Figure 3 to define the complete Exchange. The Payload of this IU shall be empty.

This IU shall contain only one frame. and Sequence Initiative must be passed by the VI Message origniator on the last frame of the last FCVI_SEND_RQST IU. The FCVI_SEND_RESP IU shall terminate the Exchange.

The VI_OPCODE field shall be 0x08 for this IU. The VI_FLAGS field is defined by Table 6. All flags can be set by the sender of this IU. The MSG_ID field shall be the same as received in the FCVI_SEND_RQST IU.

6.3 FCVI_WRITE_RQST IU

The FCVI_WRITE_RQST IU will write all or part of a VI Message to the remote VI End Point at the other end of the VI Connection to a specific remote virtual address. One or more FCVI_WRITE_RQST IUs shall be sent per Exchange. Refer to Figure 4 or Figure 5 to define the complete Exchange. The Payload of each frame in this IU carries VI application data.

Every frame of the IU shall contain the same values in the Device Header. The VI_OPCODE field shall be 0x01 for this IU. The VI_FLAGS field shall be defined by Table 5. Sending VI Immediate Data is optional with this IU. If Immediate Data is sent, then the IMM_DATA bit in the VI_FLAGS field shall be one, and the VI_PARAMETER field shall contain the Immediate Data value. Otherwise, all of these fields shall be zero. The MSG_ID field shall be incremented by one from the previously sent message on the connection.

6.4 FCVI_WRITE_RESP IU

The FCVI_WRITE_RESP IU will respond to a FCVI_WRITE_RQST for Reliable Reception mode of operation. At most one FCVI_WRITE_RESP IU shall be sent per VI Message. It is not allowed for Unreliable Delivery or Reliable Delivery modes of operation. Refer to Figure 5 to define the complete Exchange. The Payload of this IU shall be empty.

This IU shall contain only one frame, and Sequence Initiative must be passed by the VI Message origniator on the last frame of the last FCVI_WRITE_RQST IU.

The VI_OPCODE field shall be 0x09 for this IU. The VI_FLAGS field is defined by Table 6. The MSG_ID field shall be the same as received in the FCVI_WRITE_RQST IU.

6.5 FCVI_READ_RQST IU

The FCVI_READ_RQST IU will request the read of a VI Message to the remote VI End Point at the other end of the VI Connection, from a specific remote virtual address. Refer to Figure 6 to define the complete Exchange. The Payload of this IU shall be empty.

This IU shall contain only one frame, and Sequence Initiative must be passed at the completion of this IU. The VI_OPCODE field shall be 0x02 for this IU. Sending VI Immediate Data is not allowed with this IU. The IMM_DATA bit in the VI_FLAGS field shall be zero, and the VI_PARAMETER field shall be zero. The VI_TOT_RMT_LEN field shall not be used, and shall contain a value of zero. The MSG_ID field shall be incremented by one from the previously sent message on the connection.

6.6 FCVI_READ_RESP IU

The FCVI_READ_RESP IU will respond to a FCVI_READ_RQST for all reliability modes of operation, and will transfer all or part of a VI Message to the remote VI End Point at the other end of the VI Connection, to a specific remote virtual address. One or more FCVI_READ_RESP IU shall be sent per Exchange. Refer to Figure 6 to define the complete Exchange. The Payload of each frame in this IU carries VI Application data.

Every frame of the IU shall contain the same values in the Device Header. The VI_OPCODE field shall be 0x0A for this IU. The VI_FLAGS field is defined by Table 6. All flags can be set by the sender of this IU. The MSG_ID field shall be the same as received in the FCVI_READ_RQST IU.

6.7 FCVI_CONNECT_RQST IU

The FCVI_CONNECT_RQST IU will request a VI Connection to be established. Refer to Figure 8, Figure 9, or Figure 10 to define the complete Exchange. The Payload for the FCVI_CONNECT_RQST IU is shown in Table 11.

This IU shall contain only one frame, and Sequence Initiative must be passed at the completion of this IU. The VI_OPCODE field shall be 0x10 for this IU. The VI_FLAGS field shall be set to 01 for a Client-server connect request or to 02 for a Peer-to-peer connect request. The VI_PARAMETER field shall be zero. The VI_HANDLE field shall be set to 0xFFFFFFFF for this IU to indicate that this field is currently unassigned. The VI_MSG_ID field shall be set to zero for this IU.

Table 11 – FCVI_CONNECT_RQST Payload Information

Bytes	Field Name	Description	Size
0-3	VI_REVISION	Revision level of FC-VI Spec for Sender	4 bytes
4-7	VI_RQST_HANDLE	Sender's VI_Handle for messages on this VI	4 bytes
8-155	VI_LOC_ADDR	VI Local Address	148 bytes
156-303	VI_REM_ADDR	VI Remote Address	148 bytes
304-331	VI_LOC_ATTRS	VI Local Attributes	28 bytes

The VI_REVISION field shall be set to 0x00000001 to indicate the first revision of this specification.

The VI_RQST_HANDLE field shall be chosen by the sender of this IU. The VI Port on the other end of this VI Connection shall use this same VI_RQST_HANDLE in the VI_HANDLE field when sending VI Messages on this VI Connection.

VI_LOC_ADDR is the FCV_NET_ADDRESS for the local VI Connection Point. Table 13 defines the format of this field.

VI_REM_ADDR is the FCV_NET_ADDRESS for the remote VI Connection Point. Table 13 defines the format of this field.

VI_LOC_ATTRS are the connection attributes for the local VI End Point. Table 14 defines the format of this field.

6.8 FCVI_CONNECT_RESP1 IU

The FCVI_CONNECT_RESP1 IU will respond to a FCVI_CONNECT_RQST IU. Refer to Figure 8, Figure 9, or Figure 10 to define the complete Exchange. The Payload for the FCVI_CONNECT_RESP1 IU is shown in Table 12.

This IU shall contain only one frame, and Sequence Initiative must be passed at the completion of this IU. The VI_OPCODE field shall be 0x18 for this IU. The VI_FLAGS field shall be defined by Table 8. The MSG_ID field shall be the same as received in the FCVI_CONNECT_RQST IU.

Table 12 – FCVI_CONNECT_RESP1 Payload Information

Bytes	Field Name	Description	Size
0-3	VI_REVISION	Revision level of FC-VI Spec for Sender	4 bytes
4-7	VI_RQST_HANDLE	Sender's VI_Handle for messages on this VI	4 bytes
8-155	VI_LOC_ADDR	VI Local Address	148 bytes
156-303	VI_REM_ADDR	VI Remote Address	148 bytes
304-331	VI_LOC_ATTRS	VI Local Attributes	28 bytes

VI_LOC_ADDR is the FCVI_NET_ADDRESS for the local VI Connection Point. Table 13 defines the format of this field.

VI_REM_ADDR is the FCVI_NET_ADDRESS for the remote VI Connection Point. Table 13 defines the format of this field.

VI_LOC_ATTRS are the connection attributes for the local VI End Point. Table 14 defines the format of this field.

6.9 FCVI_CONNECT_RESP2 IU

The FCVI_CONNECT_RESP2 IU will respond to a FCVI_CONNECT_RESP1 IU. Refer to Figure 8, Figure 9, or Figure 10 to define the complete Exchange. The Payload for the FCVI_CONNECT_RESP2 IU shall be empty.

This IU shall contain only one frame, and the Exchange is terminated at the completion of this IU. The VI_OPCODE field shall be 0x19 for this IU. The VI_FLAGS field is defined by Table 8. The MSG_ID field shall be the same as received in the FCVI_CONNECT_RESP1 IU.

6.10 FCVI_DISCONNECT_RQST IU

The FCVI_DISCONNECT_RQST IU will request a VI Connection to be removed. Refer to Figure 11 to define the complete Exchange. The Payload for the FCVI_DISCONNECT_RQST IU shall be empty.

This IU shall contain only one frame. Sequence Initiative shall be passed at the completion of this IU. The VI_OPCODE field shall be 0x12 for this IU. The VI_FLAGS field shall be defined by Table 7. The MSG_ID field shall be incremented by one from the previously sent message on the connection.

6.11 FCVI_DISCONNECT_RESP IU

The FCVI_DISCONNECT_RESP IU will respond to a FCVI_DISCONNECT_RQST IU. Refer to Figure 11 to define the complete Exchange. The Payload for the FCVI_DISCONNECT_RESP IU shall be empty.

This IU shall contain only one frame, and the Exchange is terminated at the completion of this IU. The VI_OPCODE field shall be 0x1A for this IU. The VI_FLAGS field is defined by Table 8. The MSG_ID field shall be the same as received in the FCVI_DISCONNECT_RQST IU.

7 FC-VI Addressing and Naming

7.1 FC-VI Net Address Format

The VI Architecture specification does not include an addressing name space scheme. Instead it is left as implementation specific. The VI specification uses the following structure definition for operations requiring a VI Network Address:

```
typedef struct {
VIP_UINT16 HostAddressLen;
VIP_UINT16 DiscriminatorLen;
VIP_UINT8   HostAddress[1];
} VIP_NET_ADDRESS;
```

FC-VI uses IPv6 addresses for host addresses, where the numerical values are specified as a 32-bit or 128-bit unsigned integer. IPv4 addresses are represented as a 32 bit unsigned integer with 96 leading zeroes. FC-VI defines the Host Address Length to be 16 bytes.

The specific format of the Discriminator portion of the Host Address is outside the scope of this specification as it is intended to be application specific. However, The VI Architecture specifies that the Discriminator portion can be of variable width from 0 to MaxDiscriminatorLen bytes as indicated in the VI NIC Attributes. FC-VI defines a 128 byte field in the payload of the Connection IUs for the Discriminator. FC-VI NICs shall indicate a maximum MaxDiscriminatorLen of 128 bytes.

The actual space allocated within the IU payload for the VIP_NET_ADDRESS shell be 2 (HostAddressLen) + 2 (DiscriminatorLen) + 16 (IP Addr) + 128 (Discriminator) = 148 bytes and is defined as the FCVI_NET_ADDRESS.

Table 13 illustrates the actual format used within FC-VI Connection related IUs to convey FCVI_NET_ADDRESSES

Table 13 – FCVI Net Address Format

Bytes	Field Name	Description	Size
0-1	HOST_ADDR_LEN	Host Address Length, in Bytes	2 bytes
2-3	DISCRIM_LEN	Discriminator Length	2 bytes
4-19	HOST_IP_ADDR	Host IP Address	16 bytes
20-147	DISCRIM	Discriminator Value	128 bytes

7.2 FC-VI Attributes Format

The following table illustrates the format used within FC-VI Connection related IUs to convey FCVI_ATTRIBUTES.

Table 14 – FCVI Attributes Format

Bytes	Field Name	Description	Size
0	RELIABILITY_LVL	ReliabilityLevel	1 byte
1-3	Reserved	0x000000	3 bytes
4-7	MAX_TRANS_SIZE	Maximum Transfer Size, in Bytes	4 bytes
8-23	VIP_QOS	VI Quality of Service (defined in Annex B)	20 bytes
24	FLAGS	ViAttributeFlags field	1 byte
25-27	Reserved	0x000000	3 bytes

The ReliabilityLevel and MaxTransferSize are equal to the values as defined in the VIP_VI_ATTRIBUTES structure in Section 4.6 of the VI Architecture Developer's Guide, Revision 1.0.

The ViAttributeFlags field is defined in Table 15.

Table 15 – Format of ViAttributeFlags in FCVI_ATRIBUTES

Bit Position	Bit Name	Bit Definition
7	Reserved	0
6	Reserved	0
5	Reserved	0
4	Reserved	0
3	Reserved	0
2	Reserved	0
1	EnRdmaWr	VI enabled for RDMA Write when one.
0	EnRdmaRd	VI enabled for RDMA Read when one.

7.3 FC-VI Address Resolution

FC-VI Connection Setup operations require the Port Identifier or D_ID associated with the FC-VI Port to which the related IP Address refers to in order to send frames to that FC-VI Port. All compliant implementations of FC-VI Ports shall be capable of initiating FARP ELS requests to obtain IP to D_ID mappings. All compliant implementations of FC-VI Ports shall recognize and respond to FARP ELS requests with a FARP ELS reply.

If the FC-VI Port is attached to a Fabric that supports a Name Server, the Name Server shall be used in lieu of FARP to resolve IP Address to Port Identifer mappings.

7.4 FARP ELS

When issuing a FARP-REQ, a FC-VI Port shall

1) Set the Match Address Code Points equal to 0b00000100 (Match on IP Address of Responder).

2) Set the Responder Action equal to 0x02, requesting the Responder to originate a FARP-REPLY with the FARP-REQ originator if the Match is successful.

3) Set the Requesting N_Port Identifier equal to the Port Identifier of the FARP-REQ originator.

4) Set the Reserved bytes equal to zero.

5) Set the Requesting N_Port Port_Name equal to the WW_PN of the FARP-REQ originator.

6) Set the Responding N_Port Port_Name equal to zero.

7) Set the Requesting N_Port Node_Name equal to the WW_NN of the FARP-REQ originator.

8) Set the Responding N_Port Port_Name equal to zero.

9) Set the Requesting N_Port IP Address equal to the IP Address of the FARP-REQ originator.

10) Set the Responding N_Port IP Address equal to the IP Address of the Port the FARP-REQ originator wishes to establish a VI connection with.

When receiving a FARP-REQ, a FC-VI Port shall perform the following three tests:

1) Compare for equality its FC-VI Port IP address to the Responding N_Port IP Address in the FARP request payload

2) Test if the Match on IP Address of Responder code point, 0b00000100, is set in the FARP-REQ Match Address Code Points

3) Test if the Responder Action in the FARP-REQ to is set to 0x02, requesting the Responder to originate a FARP-REPLY with the Requesting N_Port Identifier if the Match is successful.

If any test fails, the FARP-REQ recipient shall take no action. If all three tests pass, the FARP-REQ recipient shall originate a FARP_REPLY to the FARP-REQ originator by formatting the FARP_REPLY payload as follows:

1) Set the Match Address Code Points equal to the Match Address Code Points in the FARP-REQ (0b00000100).

2) Set the Requesting N_Port Identifier equal to the Port Identifier of the FARP-REQ originator.

3) Set the Responder Action equal to the Responder Action in the FARP-REQ (0x02).

4) Set the Responding N_Port Identifier equal to its Port Identifier.

5) Set the Requesting N_Port Port_Name equal to the WW_PN of the FARP-REQ originator.

6) Set the Responding N_Port Node_Name equal to the WW_PN of the FARP_REPLY originator.

7) Set the Requesting N_Port Node_Name equal to the WW_NN of the FARP-REQ originator.

8) Set the Responding N_Port Node_Name equal to the WW_NN of the FARP_REPLY originator.

9) Set the Requesting N_Port IP Address equal to the IP Address of the FARP-REQ originator.

10) Set the Responding N_Port IP Address equal to the IP Address of the FARP_REPLY originator.

7.5 Name Server Queries

An FC-VI Port that is attached to a Fabric which supports the Name Server may map D_ID address to IP addresses by issuing a GID_IPP Request to the Name Server. The format of GID_IPP is defined in Section 6.6.23 of FC-GS-2, Rev 5.3.

8 FC-VI Error Handling

FC-VI is a request-response protocol. Except for Sends and RDMA Writes in Unreliable Delivery and Reliable Delivery, every FC-VI IU request IU shall expect a response IU.

In FC-VI Connection Setup, there are two interleaved request-response handshakes. A FCVI_CONNECT_RQST IU shall expect a FCVI_CONNECT_RESP1 IU. Similarly, a FCVI_CONNECT_RESP1 IU shall expect a FCVI_CONNECT_RESP2 IU.

For all FC-VI IUs that expect a response IU, the FC-VI IU originator shall wait for a response IU for FCVI_ULP_TIMEOUT. The value of FCVI_ULP_TIMEOUT is implementation dependent. If a response IU is not received within FCVI_ULP_TIMEOUT, the IU originator shall issue an ABTS_LS ELS to recover the Exchange.

Annex A

Proposed Additions and Changes to FC-FS

A.1 Introduction

This Annex contains several items which are, as of publication of this Standard, approved for inclusion in FC-FS (see reference [x]). In summary, these items are:

– Device Header. The Device Header definition is modified to allow a Device Header in every frame of a Sequence;

– Preference. A function is added to allow an N_Port to request that a frame be given Preferred delivery. This function is indicated in the CS_CTL field for Class 2 and Class 3. The availability of this function may be detected through Login;

– Fibre Channel Address Resolution Protocol (FARP). Two Extended Link Services are defined to implement the Internet Address Resolution Protocol on Fibre Channel.

The changes to FC-FS are presented on the following pages.

A.2 Device Header Change

These changes affect the Device Header definition. Clause numbers are FC-FS clause numbers.

This proposal expands the use of the Device_Header, which is one of the Optional Headers defined in FC-PH, and by extension, FC-FS.

Currently, the Device_Header may only appear in the first Data frame of a Sequence. The FC-VI project has identified a compelling usage for Device_Header that requires it to appear in every frame of the Sequence.

Old text:

19.6.4 Device_Header

Device_Header, if present, shall be present only in the first Data frame of a Sequence. A Device_Header may be used by a ULP type. For that ULP type, the Device_Header is required to be supported. The Device_Header may be ignored and skipped, if not needed. If a Device_Header is present for a ULP which does not require it, the related FC-4 may reject the frame with the reason code of TYPE not supported.

New Text (Only the first sentence is changed, but the whole paragraph is included for reference.):

19.6.4 Device_Header

<u>Device_Header, if present, shall be present either in the first Data frame or in all Data frames of a Sequence</u>. A Device_Header may be used by a ULP type. For that ULP type,

the Device_Header is required to be supported. The Device_Header may be ignored and skipped, if not needed. If a Device_Header is present for a ULP which does not require it, the related FC-4 may reject the frame with the reason code of TYPE not supported.

A.3 Preference Function

These changes add a new function called Preference. Clause numbers are FC-FS clause numbers.

18.4 Class Specific Control (CS_CTL)

...

18.4.2 Class 2

The CS_CTL field is defined in the following table for Class 2.

– PREF shall be meaningful in all Class 2 frames.

Table 40.1 – CS_CTL field - Class 2

Bit	Abbr.	Meaning
31	PREF	0 = Frame is delivered with no Preference 1 = Frame may be delivered with Preference
30-29		Reserved for additional Preference function
28-24		Reserved

18.4.3 Class 3

The CS_CTL field is defined in the following table for Class 3.

– PREF shall be meaningful in all Class 3 frames.

Table 40.2 – CS_CTL field - Class 3

Bit	Abbr.	Meaning
31	PREF	0 = Frame is delivered with no Preference 1 = Frame may be delivered with Preference
30-29		Reserved for additional Preference function
28-24		Reserved

...

22.2.2 Class 2 rules

...

p) The Class 2 PREF field is a Class of Service specific use of the CS_CTL field of the FC-PH-2 frame header. When PREF is zero, the Fabric shall deliver the frame normally. When PREF is one, the Fabric may deliver the frame to the destination N_Port prior to frames which have PREF set to zero. If the Fabric indicated through Login that it guarantees sequential delivery, the Fabric shall deliver frames with the same PREF value to a destination in the same order received from the source (see 23.7.4.2).

22.3.2 Class 3 rules

...

p) The Class 3 PREF field is a Class of Service specific use of the CS_CTL field of the FC-PH-2 frame header. When PREF is zero, the Fabric shall deliver the frame normally. When PREF is one, the Fabric may deliver the frame to the destination N_Port prior to frames which have PREF set to zero. If the Fabric indicated through Login that it guarantees sequential delivery, the Fabric shall deliver frames with the same PREF value to a destination in the same order received from the source (see 23.7.4.2).

23.6.8.2 Service options

....

Word 0, Bit 22 - CS_CTL Tolerance

0 = non-zero CS_CTL may be tolerated
1 = non-zero CS_CTL shall be tolerated

When an N_Port performs Login with another N_Port, it shall indicate tolerance for non-zero CS_CTL within the Class of Service by specifying bit 22 = 1. The other N_Port indicates tolerance for non-zero CS_CTL by specifying bit 22 = 1 in the ACC sent as a reply. An N_Port that tolerates a non-zero CS_CTL shall not reject or otherwise deprecate a frame solely because the CS_CTL field is non-zero.

Class 1

Bit 22 has no meaning in Class 1.

Class 2

In Class 2, if bit 22 = 1, the N_Port shall tolerate the PREF field in the CS_CTL field of the Frame Header. Tolerance for CS_CTL as a Sequence Initiator means that the PREF field may specify Preference to the Fabric. Tolerance for CS_CTL as a Sequence Recipient means that the PREF field shall be ignored by the N_Port. (see 22.2.2)

Class 3

In Class 3, if bit 22 = 1, the N_Port shall tolerate the PREF field in the CS_CTL field of the Frame Header. Tolerance for CS_CTL as a Sequence Initiator means that the PREF field may specify Preference to the Fabric. Tolerance for CS_CTL as a Sequence Recipient means that the PREF field shall be ignored by the N_Port. (see 22.3.2)

NOTE – Even if an N_Port never intends to set the PREF field to any value other than zero, the N_Port may still have reason to set bit 22 = 1. Setting the bit to one indicates to the other N_Port that the N_Port will accept frames with a non-zero CS_CTL field value.

Class 4

Bit 22 has no meaning in Class 4.

....

23.7.4.2 Service options

....

Word 0, Bit 22 - Preference

0 = Normal delivery
1 = Preferred delivery functional

If bit 22 is set to one by an N_Port, then it is requesting that all frames transmitted by the N_Port requesting this function be delivered according to the setting of the PREF field in the CS_CTL field of the Frame Header (see 22.2.2 and 22.3.2). If bit 22 is set to one by the F_Port, then the F_Port is indicating that it shall deliver Class 2 and 3 frames transmitted by the requesting N_Port according to the setting of the PREF field.

If the N_Port has requested this support and the Fabric responds with bit 22 = 1, then Preferred delivery shall be supported by the Fabric. If the N_Port has not requested this support and the Fabric responds with bit 22 = 1, then Preferred delivery shall be functional by the Fabric as default operation. If the Fabric responds with bit 22 = 0, then Preferred delivery may be functional by the Fabric as default operation.

NOTE – An F_Port that responds with bit 22 = 0 may not itself support Preferred delivery, but other Fabric Elements in the path to the destination may support it. An N_Port may attempt Preferred delivery even if the F_Port does not indicate support.

Class 1

Bit 22 has no meaning in Class 1.

Class 2

In Class 2, if bit 22 = 1, the Fabric shall deliver both Data and Link_Control frames according to the setting of the PREF field in the CS_CTL field of the Frame Header.

Class 3

In Class 3, if bit 22 = 1, the Fabric shall deliver Data frames according to the setting of the PREF field in the CS_CTL field of the Frame Header.

The following table summarizes the function of bit 22 for both Class 2 and Class 3.

N_Port Word 0, Bit 22	**F_Port Word 0, Bit 22**	
0	0	Preferred delivery by the Fabric may be functional
0	1	Preferred delivery by the Fabric shall be functional
1	0	N_Port support requested, Preferred delivery by the Fabric may be functional
1	1	N_Port requested, Fabric agrees, Preferred delivery by the Fabric shall be functional

The following table summarizes the relationship between Preferred delivery and sequential delivery for both Class 2 and Class 3.

Preference Functional	**Sequential delivery Functional**	
0	0	Frames may be delivered in any order
0	1	Frames shall be delivered to a destination in the same order received from the source, PREF is ignored
1	0	Frames may be delivered in any order, but frames with PREF set to one may be delivered prior to frames with PREF set to zero
1	1	Frames with PREF set to one shall be delivered to a destination in the same order received from the source relative to each other, and may be delivered prior to frames with PREF set to zero; frames with PREF set to zero shall also be delivered to a destination in the same order received from the source relative to each other

Class 4

Bit 22 has no meaning in Class 4.

A.4 Fibre Channel Address Resolution Protocol (FARP)

These changes add two new Extended Link Services to support the Internet Protocol over Fibre Channel. Clause numbers are FC-FS clause numbers.

21.4.x Fibre Channel Address Resolution Protocol Request (FARP-REQ)

The Fibre Channel Address Resolution Protocol (FARP) is a method using ELS commands to resolve WWN or IP address to D_ID mapping in environments without a Directory Server. That is, when the WWN or an IP address is known, but not the D_ID, and a suitable Directory Service does not exist. See Reference [{GS-2}] for the definition of Directory Services.

The FARP-REQ Extended Link Service request shall be used to resolve Port_IDs of communicating Fibre Channel devices. A FARP-REQ may be used to retrieve a specific N_Port's current Port_ID given a WW_PN, WW_NN or IP address. This is accomplished by requesting either a FARP-REPLY ELS Extended Link Service (see 21.4.y), and/or by indicating that the responding N_Port shall perform Login with the requesting N_Port.

A FARP-REQ Match shall be considered successful at the Recipient when the FARP-REQ request is successfully received and recognized by an N_Port, and the requested Match Address Code Point is successful.

Protocol:

FARP-REQ request Sequence
No Reply Sequence

Format: FT-1

Addressing: The S_ID designates the Requester N_Port requesting addressing information. The D_ID is the Broadcast Alias_ID, hex ‘FFFFFF’ (see 18.3).

Payload: The format of the FARP-REQ request Payload is shown in table x1.

Table x1 - FARP-REQ Request Payload

Item	Size Bytes
hex ‘54000000’	4
Match Address Code Points	1
Requesting N_Port Identifier	3
Responder Action	1
Reserved	3
Requesting N_Port Port_Name	8
Requesting N_Port Node_Name	8
Responding N_Port Port_Name	8
Responding N_Port Node_Name	8
Requesting N_Port IP Address	16
Responding N_Port IP Address	16

Match Address Code Points: The Match Address Code Points define what addresses to match based on these code points. The Match Address Code Points also indicate which Address fields shall be valid (see below). The values of these code points are defined in table x2.

Table x2 - FARP-REQ Match Address Code Points

Value (binary)	Definition
0000 0001	The Match is successful if all of the following are true: • the Port_Name of the N_Port which receives the FARP-REQ equals the Responding Port_Name; • ignore the Responding Node_Name; • ignore the Responding IP Address.
0000 0010	The Match is successful if all of the following are true: • ignore the Responding Port_Name; • the Node_Name of the N_Port which receives the FARP-REQ equals the Responding Node_Name; • ignore the Responding IP Address.
0000 0011	The Match is successful if all of the following are true: • the Port_Name of the N_Port which receives the FARP-REQ equals the Responding Port_Name; • the Node_Name of the N_Port which receives the FARP-REQ equals the Responding Node_Name; • ignore the Responding IP Address.
0000 0100	The Match is successful if all of the following are true: • ignore the Responding Port_Name; • ignore the Responding Node_Name; • the IP Address of the N_Port which receives the FARP-REQ equals the Responding IP Address.
0000 0101	The Match is successful if all of the following are true: • the Port_Name of the N_Port which receives the FARP-REQ equals the Responding Port_Name; • ignore the Responding Node_Name; • the IP Address of the N_Port which receives the FARP-REQ equals the Responding IP Address.

Table x2 - FARP-REQ Match Address Code Points

Value (binary)	Definition
0000 0110	The Match is successful if all of the following are true: • ignore the Responding Port_Name; • the Node_Name of the N_Port which receives the FARP-REQ equals the Responding Node_Name; • the IP Address of the N_Port which receives the FARP-REQ equals the Responding IP Address.
0000 0111	The Match is successful if all of the following are true: • the Port_Name of the N_Port which receives the FARP-REQ equals the Responding Port_Name; • the Node_Name of the N_Port which receives the FARP-REQ equals the Responding Node_Name; • the IP Address of the N_Port which receives the FARP-REQ equals the Responding IP Address.
all others	Reserved

Requesting N_Port Identifier: The Requesting N_Port Identifier contains the Address Identifier of the N_Port issuing the FARP_REQ.

Responder Action: The Responder Action defines the Responder action if the match requested by the Match Address Code Points is successful. These actions are defined in table x3.

Table x3 - Responder Action

Value (hex)	Definition
00	Perform no action
01	If the Match is successful, the receiving N_Port shall perform Login with the Requesting N_Port Identifier.
02	If the Match is successful, the receiving N_Port shall originate FARP-REPLY with the Requesting N_Port Identifier. The receiving N_Port shall not perform Login with the Requesting N_Port Identifier. The receiving N_Port shall not perform an explicit Logout with the Requesting N_Port Identifier.
03	If the Match is successful, the receiving N_Port shall perform Login with the Requesting N_Port Identifier; and, if the Login is successful, the receiving N_Port shall originate FARP-REPLY with the Requesting N_Port Identifier. {does Login need to be successful? - discuss}
all others	Reserved

Requesting N_Port Port_Name: The Requesting N_Port Port_Name contains the Port_Name of the N_Port issuing the FARP_REQ.

Requesting N_Port Node_Name: The Requesting N_Port Node_Name contains the Node_Name of the N_Port issuing the FARP_REQ.

Responding N_Port Port_Name: The Responding N_Port Port_Name contains the Port_Name which the receiving N_Port compares with its own Port_Name to determine a Match.

Responding N_Port Node_Name: The Responding N_Port Node_Name contains the Node_Name which the receiving N_Port compares with its own Node_Name to determine a Match.

Requesting N_Port IP Address: The Requesting N_Port IP Address contains the IP Address of the N_Port issuing the FARP_REQ. The format of the IP Address is IPv6 (see reference []).

Responding N_Port IP Address: The Responding N_Port IP Address contains the IP Address which the receiving N_Port compares with its own IP Address to determine a Match. The format of the IP Address is IPv6 (see reference []).

The Match Address Code Points indicate which of the above fields shall be specified, as defined in table x4. In this table, “correct” means the value of the field shall be correct for the N_Port; “ignored” means that the value shall be either zero, or any valid value, since the field is not examined for comparison.

Table x4 - FARP-REQ Address Fields

Code Point Value (binary)	Requesting N_Port Port_Name	Requesting N_Port Node_Name	Requesting N_Port IP Address	Responding N_Port Port_Name	Responding N_Port Node_Name	Responding N_Port IP Address
0000 0001	correct	ignored	ignored	correct	ignored	ignored
0000 0010	ignored	correct	ignored	ignored	correct	ignored
0000 0011	correct	correct	ignored	correct	correct	ignored
0000 0100	ignored	ignored	correct	ignored	ignored	correct
0000 0101	correct	ignored	correct	correct	ignored	correct
0000 0110	ignored	correct	correct	ignored	correct	correct
0000 0111	correct	correct	correct	correct	correct	correct
all others	Reserved					

Reply Link Service Sequence

none.

21.4.y Fibre Channel Address Resolution Protocol Reply (FARP-REPLY)

The FARP-REPLY Extended Link Service request shall be used to respond to a FARP-REQ which resulted in a successful Match, with a FARP-REQ Responder Action requesting a FARP-REPLY. Note that the Responder action indicates whether the Responder shall or shall not perform Login prior to the FARP-REPLY.

Protocol:

FARP-REPLY request Sequence
Accept (ACC) reply Sequence

Format: FT-1

Addressing: The S_ID designates the Responding N_Port providing addressing information. The D_ID designates the Requesting N_Port requesting addressing information.

Payload: The format of the FARP-REPLY request Payload is shown in table y1.

Table y1 - FARP-REPLY Request Payload

Item	Size Bytes
hex ‘55000000’	4
Match Address Code Points	1
Requesting N_Port Identifier	3
Responder Action	1
Responding N_Port Identifier	3
Requesting N_Port Port_Name	8
Requesting N_Port Node_Name	8
Responding N_Port Port_Name	8
Responding N_Port Node_Name	8
Requesting N_Port IP Address	16
Responding N_Port IP Address	16

Match Address Code Points: The Match Address Code Points define the address(es) that were matched. The values of these code points are defined in table x2. The value of this field shall equal the value sent in the corresponding FARP-REQ request.

Requesting N_Port Identifier: The Requesting N_Port Identifier contains the Address Identifier of the N_Port which issued the FARP_REQ.

Responding N_Port Identifier: The Responding N_Port Identifier contains the Address Identifier of the N_Port which is responding to the FARP_REQ.

Responder Action: The Responder Action field shall be set to the same value as in the corresponding FARP-REQ.

Requesting N_Port Port_Name: The Requesting N_Port Port_Name contains the Port_Name of the N_Port which issued the FARP_REQ.

Requesting N_Port Node_Name: The Requesting N_Port Node_Name contains the Node_Name of the N_Port which issued the FARP_REQ.

Responding N_Port Port_Name: The Responding N_Port Port_Name contains the Port_Name of the N_Port which is responding to the FARP_REQ.

Responding N_Port Node_Name: The Responding N_Port Node_Name contains the Node_Name of the N_Port which is responding to the FARP_REQ.

Requesting N_Port IP Address: The Requesting N_Port IP Address contains the IP Address of the N_Port which issued the FARP_REQ. The format of the IP Address is IPv6 (see reference []).

Responding N_Port IP Address: The Responding N_Port IP Address contains the IP Address of the N_Port which is responding to the FARP_REQ. The format of the IP Address is IPv6 (see reference []).

The Match Address Code Points indicate which of the above fields shall be specified, as defined in table y2. In this table, “correct” means the value of the field shall be correct for the N_Port; “ignored” means that the value shall be either the correct value, zero, or any valid value, since the field was not examined for comparison; “REQ” means that the value shall be equal to the value sent in the corresponding FARP-REQ request.

Table y2 - FARP-REPLY Address Fields

Code Point Value (binary)	Requesting N_Port Port_Name	Requesting N_Port Node_Name	Requesting N_Port IP Address	Responding N_Port Port_Name	Responding N_Port Node_Name	Responding N_Port IP Address
0000 0001	REQ	REQ	REQ	correct	ignored	ignored
0000 0010	REQ	REQ	REQ	ignored	correct	ignored
0000 0011	REQ	REQ	REQ	correct	correct	ignored
0000 0100	REQ	REQ	REQ	ignored	ignored	correct
0000 0101	REQ	REQ	REQ	correct	ignored	correct
0000 0110	REQ	REQ	REQ	ignored	correct	correct
0000 0111	REQ	REQ	REQ	correct	correct	correct
all others	Reserved					

Reply Link Service Sequence

Service Reject (LS_RJT)
signifies rejection of the FARP-REPLY command (see 21.5.2)

Accept (ACC)
signifies that the N_Port has transmitted the requested data.

The format of the Accept Payload is shown in table y3.

Table y3 - FARP-REPLY Accept Payload

Item	Size Bytes
hex ‘02000000’	4

Annex B

FC-VI Definition for VIP_QOS

B.1 Introduction

This section defines the structure values of the Quality of Service component of the VIP_VI_ATTRIBUTES. This definition is needed for two reasons. First, the FC-VI Connection IUs carry the components of the VIP_VI_ATTRIBUTES structure back and forth between communicating nodes during VI Connection Setup and there is currently no definition that defines what is meant by VIP_QOS. Second, there must be a method which extends all the way to the VIPL API that will allow an FC-VI implementor to take advantage of the new FC frame Precedence capabilities that are being proposed for FC-FS.

This section of the document defines a structure that can be used by a VI Application to specify a value for Precedence that can be used by FC-VI Providers to generate Fibre Channel frames that make use of the new Precedence capabilities that can be provided by Fibre Channel switched fabrics.

The new definition is defined in such a way as to avoid breaking binary compatibility with existing VIPL libraries and without forcing FC-VI Providers to support this functionality. Furthermore, this definiton will be proposed as an addendum to the "Virtual Interface Architecture Specification, Draft Revision 1.0" .

The "Virtual Interface Architecture Specification, Draft Revision 1.0" and "Intel Virtual Interface (VI) Developers Guide, Revision 1.0" do not offer a definition for one of the VI Attributes which is declared as VIP_QOS. "Intel Virtual Interface (VI) Developers Guide Revision 1.0" declares VIP_QOS as the following:

```
typedef VIP_PVOID VIP_QOS; /* details are not defined */
```

This would seem to indicate that the intention here is to have a structure containing desired Quality of Service (QOS) attributes and that this structure would be pointed to by VIP_QOS which would be passed into pertinent VIPL calls requiring the VIP_VI_ATTRIBUTES structure.

Below is the definition for VIP_QualityOfService that may be used by FC-VI Providers and shall be used within the FCVI_ATTRIBUTES carried in FC-VI Connection IUs.

```
typedef struct {
        VIP_UINT8       Precedence:3;
        VIP_UINT8       Reserved1:5;
        VIP_UINT8       Reserved2[3];
        VIP_UINT32      MaxBandwidth;           /* B/s */
        VIP_UINT32      MinBandwidth;           /* B/s */
        VIP_UINT32      MaxDelay;               /* uS */
} VIP_QualityOfService_t, *VIP_QualityOfService_p;

typedef VIP_QualityOfService_p VIP_QOS;
```

Where the following is true for this version of the FC-VI specification:

Table 1 – Values Possible for VIP_QualityOfService_t for FC-VI 1.0

Item	*Value*
Precedence	**000 = Routine**
	001 = Priority
	010 - 111 Reserved
Reserved1	**Must be 0**
Reserved2	**Must be 0**
MaxBandwidth	**Must be 0**
MinBandwidth	**Must be 0**
MaxDelay	**Must be 0**

Note: The fields and values chosen were designed to accomodate both “RFC: 791 INTERNET PROTOCOL DARPA INTERNET PROGRAM PROTOCOL SPECIFICATION September 1981” Type Of Service Field, Page 11 as well as “Fibre Channel PHYSICAL AND SIGNALLING INTERFACE - 2 (FC-PH-2) Rev 7.4” Pg 58, Clause 21.18.1 Quality of Service Request (QoSR), Table 141 QoSR Payload. Future revisions of the FC-VI specification may support more levels of precedence. Future revisions of the FC-VI specification may also support Class 4 Fibre Channel operation.

Table 2 – QoS Portion of VI_ATTRIBUTES payload in FC-VI Connection IUs

Item	*Offsets*
Precedence	**Bits 0-2 of Byte 0**
Reserved1	**Bits 3-7 of Byte 0**
Reserved2	**Bytes 1 - 3**
MaxBandwidth	**Bytes 4-7**
MinBandwidth	**Bytes 8-11**
MaxDelay	**Bytes12-15**