Copies of this document may be purchased from:
Global Engineering, 15 Inverness Way East,
Englewood, CO 80112-5704Project 2220-D
Phone: (800) 854-7179 or (303) 792-2181 Fax: (303) 792-2192

INCITS xxx-xxxx
T11/Project 2220-D/Rev 1.2

FIBRE CHANNEL

SWITCH FABRIC - 6

(FC-SW-6)

REV 1.2

INCITS working draft proposed
American National Standard
for Information Technology

February 1, 2013

Secretariat: Information Technology Industry Council

NOTE:

This is a working draft American National Standard of Accredited Standards Committee INCITS. 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.

Steven Wilson (T11 Chair)
Brocade
120 Holger Way
San Jose, Ca. 95134
Voice: 408-333-8128
Fax: 408-333-7930
swilson@brocade.com

Claudio Desanti (T11 Vice Chair)
Cisco Systems, Inc.
170 W. Tasman Dr.
San Jose, CA 95134
Voice: 408-853-9172
Fax: 408-853-9172
cds@cisco.com

Craig W. Carlson (T11.3 Chair)
QLogic Corporation
6321 Bury Dr.
Eden Prairie, MN 55346
Voice: 952-932-4064
Fax: 952-932-4037
craig.carlson@qlogic.com

Craig W. Carlson (FC-SW-6 Chair)
QLogic Corporation
6321 Bury Dr.
Eden Prairie, MN 55346
Voice: 952-932-4064
Fax: 952-932-4037
craig.carlson@qlogic.com

Steven Wilson (FC-SW-6 Editor)
Brocade
120 Holger Way
San Jose, Ca. 95134
Voice: 408-333-8128
Fax: 408-333-7930
swilson@brocade.com

Editor's Notes, Revision 1.0:

First Draft of FC-SW-6.

Revision 1.1:

Second Draft of FC-SW-6. Includes:

08-597v1 Definitions that were omitted from FC-SW-5;

11-313v2 Updates to RDI;

12-034v2 Distributed Switch Environment

11-412v2 Peer Zoning

Revision 1.2:

Third Draft of FC-SW-6. Includes:

12-036v2 Distributed Switch Payloads;

12-035v3 Distributed Switch Protocols;

12-476v0 Re-Login

BSR INCITS xxx-xxxx

American National Standard
for Information Technology

Fibre Channel — Switch Fabric - 6 (FC-SW-6)

Secretariat

Information Technology Industry Council

Approved (not yet approved)

American National Standards Institute, Inc.

Abstract

This standard describes the requirements for an interconnecting Fabric consisting of multiple Fabric Switch elements to support the ANSI/INCITS Fibre Channel - Framing and Signaling (FC-FS-3) and ANSI/INCITS Fibre Channel - Physical Interface (FC-PI-5) standards.

American National Standard

Approval of an American National Standard requires review 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 an interpretation of 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.

CAUTION: The developers of this standard have requested that holders 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 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 West 42nd Street, New York, NY 10036

Printed in the United States of America

Foreword (This Foreword is not part of American National Standard INCITS xxx-xxx.)

This standard describes the requirements for an interconnecting Fabric consisting of multiple Fabric Switch elements to support the ANSI/INCITS Fibre Channel - Framing and Signaling - Three (FC-FS-3) and ANSI/INCITS Fibre Channel - Physical Interface - Five (FC-PI-5) standards.

This standard was developed by Task Group T11 of Accredited Standards Committee INCITS during 2010-201x. The standards approval process started in 201x. This document includes annexes that are informative and are not considered part of the standard.

Requests for interpretation, suggestions for improvements or addenda, or defect reports are welcome. They should be sent to the INCITS Secretariat, Information Technology Industry Council, 1250 Eye Street, NW, Suite 200, Washington, DC 20005-3922.

This standard was processed and approved for submittal to ANSI by the International Committee for Information Technology Standards (INCITS). Committee approval of the standard does not necessarily imply that all committee members voted for approval.

At the time it approved this standard, INCITS had the following members:

(to be filled in by INCITS)

Technical Committee T11 on Fibre Channel Interfaces, which reviewed this standard, had the following members:

TBD, Chair
Claudio DeSanti, Vice-Chair
Bob Nixon, Secretary

Company **Name**

TBD.

Introduction

FC-SW-6 is one of the Fibre Channel family of standards. This family includes ANSI/INCITS FC-FS-3 and ANSI/INCITS FC-PI-5. ANSI/INCITS FC-GS-7, is a document related to Generic Fabric Services and is closely tied to FC-SW-6. ANSI/INCITS FC-BB-5 and FC-BB-6 describe how Fabrics are extended over transports complementary to Fibre Channel. ANSI/INCITS FC-MI-3 and ANSI/INCITS FC-DA-2 describe interoperability profiles that assists in the interoperability of Switches. INCITS 332: 1999, FC-AL-2, specifies the arbitrated loop topology. ANSI/INCITS FC-SP-2 describes the Security requirements and protocols associated with Fibre Channel networks. ANSI/INCITS FC-IFR describes the requirements and protocols associated with Inter-Fabric Routing.

FC-SW-6 describes how switches communicate and interact with one another to form a Fabric of switches. Included are Fabric initialization and configuration, routing, server communication, event distribution and repository exchanges (e.g., zoning information).

Acknowledgements

The technical editor would like to thank the following individuals for their special contributions to this standard:

Craig Carlson

John Crandall

Claudio Desanti

Roger Hawthorn

Howard Johnson

Bill Martin

David Peterson

Lou Ricci

Ralph Weber

Steve Wilson

Contents **Page**

Figure **Page**

Table **Page**

draft proposed American National Standard
for Information Technology—

Fibre Channel —
Switch Fabric - 6 (FC-SW-6)

1 Scope

This American National Standard for FC-SW-6 describes the operation and interaction of Fibre Channel Switches.

This standard includes:

a) E_Port Operation and Fabric Configuration;

b) Path selection (FSPF);

c) Bridge Port (B_Port) Operation;

d) Distributed server interaction and communication;

e) Exchange of information between Switches to support zoning;

f) Distribution of Event Notifications between Switches;

g) Virtual Fabrics Switch Support;

h) Enhanced Commit Service;

i) Virtual Channels;

j) Distributed Switch Model

2 Normative references

2.1 Overview

The following standards contain provisions that, through reference in the text, 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.

For electronic copies of ANSI and INCITS standards, visit ANSI's Electronic Standards Store (ESS) at http://www.ansi.org. For printed versions of most standards listed here, contact Global Engineering Documents, 15 Inverness Way East, Englewood, CO; 80112-5704, (800) 854-7179.

Orders for ISO Standards and ISO publications should normally be addressed to the ISO member in your country. If that is impractical, ISO Standards and ISO publications may be ordered from ISO Central Secretariat (ISO/CS):

Phone +41 22 749 01 11
Fax +41 22 749 09 47
E-mail sales@iso.org
Post ISO, 1, rue de Varembé, CH-1211
Geneva 20, Switzerland

In order to avoid delivery errors, it is important that you accurately quote the standard's reference number given in the ISO catalogue. For standards published in several parts, you should specify the number(s) of the required part(s). If not, all parts of the standard will be provided.

Copies of the following documents may be obtained from ANSI, an ISO member organization:

a) approved ANSI standards;
b) approved and draft international and regional standards (ISO and IEC); and
c) approved foreign standards (JIS and DIN).

For further information, contact the ANSI Customer Service Department:

Phone +1 212-642-4900
Fax: +1 212-302-1286
Web: http://www.ansi.org
E-mail: ansionline@ansi.org

or the InterNational Committee for Information Technology Standards (INCITS):

Phone 202-626-5738
Web: http://www.incits.org
E-mail: incits@itic.org

Additional availability contact information is provided below as needed.

2.2 Approved references

[1] ANSI/INCITS 332-1999, *Fibre Channel - Second Generation Arbitrated Loop - 2 (FC-AL-2)*

[2] INCITS TR-48-2012, *Fibre Channel - Methodologies for Interconnects - 3 (FC-MI-3)*

[3] ANSI/INCITS 479-2011, *Fibre Channel - Physical Interface - 5 (FC-PI-5)*

[4] ANSI/INCITS 470-2011, Fibre Channel - *Framing and Signaling - 3* (FC-FS-3)

[5] ANSI INCITS 463:2010, *Fibre Channel - Generic Services - 6 (FC-GS-6)*

[6] INCITS TR-49-2012, *Fibre Channel - Device Attach - 2 (FC-DA-2)*

[7] ANSI/INCITS 462-2010, *Fibre Channel - Backbone - 5 (FC-BB-5)*

[8] ANSI/INCITS 477-2011, *Fibre Channel - Link Services - 2 (FC-LS-2)*

[9] ANSI/INCITS 466-2011, *Fibre Channel - Single Byte Command Set- 4(FC-SB-4)*

[10] ANSI/INCITS 426-2007 *Fibre Channel - Security Protocols (FC-SP)*

[11] ANSI/INCITS 475-2011, *Fibre Channel - Inter-Fabric Routing* (FC-IFR)

2.3 References under development

At the time of publication, the following referenced Standards were still under development. For information on the current status of the document, or regarding availability, contact the relevant Standards body or other organization as indicated.

NOTE 1 – For more information on the current status of a document, contact the INCITS Secretariat at the address listed in the front matter. To obtain copies of this document, contact Global Engineering at the address listed in the front matter, or the INCITS Secretariat.

[12] ANSI/INCITS.487-201y, *Fibre Channel - Link Services - 3 (FC-LS-3),* T11/Project 2237D/Rev 2

[13] ANSI/INCITS.xxx-201y, *Fibre Channel - Generic Services - 7 (FC-GS-7)*, T11/Project 2204D/Rev 1.0

[14] ANSI/INCITS.xxx-201y, *Fibre Channel - Security Protocols - 2 (FC-SP-2)*, T11/Project 1835D/Rev 2.7

[15] ANSI/INCITS 488-201y, *Fibre Channel - Framing and Signaling - 4* (FC-FS-4), T11/Project 2238D/Rev 1.0

[16] ANSI/INCITS 485-201y, *Fibre Channel - Single Byte Commend Sets - 5* (FC-SB-5), T11/Project 2245D/Rev 1.0

[17] ANSI INCITS xxx-201y, *Fibre Channel - Backbone- 6* (FC-BB-6), T11/Project 2159D, Rev 1.04

[18] ANSI/INCITS xxx-201y, *Fibre Channel - Physical Interface - 6 (FC-PI-6),* T11 Project 2221D, Rev 1.0

2.4 IETF references

Copies of the following approved IETF standards may be obtained through the Internet Engineering Task Force (IETF) at www.ietf.org.

RFC 905, *ISO Transport Protocol Specification, ISO DP 8073,* April 1984.

RFC 1008, *Implementation Guide for the ISO Transport Protocol,* June 1987.

RFC 4936, *Fibre Channel Zone Server MIB,* August 2007.

3 Definitions and conventions

For FC-SW-6, the following definitions, conventions, abbreviations, acronyms, and symbols apply.

3.1 Definitions

3.1.1 Active Zone Set: The Active Zone Set is the Zone Set Definition currently in effect and enforced by the Fabric or other entity (e.g., the Name Server).

3.1.2 Address assignment: A process whereby addresses are dispensed to Switches and Switch Ports.

3.1.3 Address identifier: As defined in FC-FS-3**,** an unsigned 24-bit address value used to uniquely identify the source (S_ID) and destination (D_ID) of Fibre Channel frames.

3.1.4 Address Manager: A logical entity within a Switch that is responsible for address assignment.

3.1.5 Adjacent Switch: A remote Switch that does not require intermediate Switches in order to be reached.

3.1.6 Adjacency: A relationship between two Switches that have synchronized their topology databases.

3.1.7 Adjacent: Two Switches that have synchronized their topology databases are considered Adjacent.

3.1.8 AISL (Augmented ISL): an E_Port to E_Port link used by the redundancy protocol.

3.1.9 AISL Set: The set of AISLs that connect the two Controlling Switches that are part of a Distributed Switch.

3.1.10 A_Port (Adjacent Port): The combination of one PA_Port and one VA_Port operating together.

3.1.11 ASL (A_Port Switch Link): An A_Port to A_Port link.

3.1.12 Area: The second level in the three-level address partitioning system specified by this standard (see 4.8).

3.1.13 Area Identifier: Bits 15 through 8 of an address identifier.

3.1.14 Broadcast Address: An FFFFFFh value in the D_ID field shall specify that the frame be broadcast to all Nx_Ports.

3.1.15 Broadcast Zone: A zone with the Broadcast attribute specified.

3.1.16 Broadcast Zoning Enforcement: Zoning technique where the Fabric limits broadcast distribution among zone members using frame-by-frame filtering techniques.

3.1.17 B_Port: A Bridge Port is a Fabric inter-element port used to connect Bridge devices with E_Ports on a Switch. The B_Port provides a subset of the E_port functionality.

3.1.18 Class F service: A service that multiplexes frames at frame boundaries and is used for control and coordination of the internal behavior of the Fabric.

3.1.19 Class N service: Refers to any class of service other than Class F.

3.1.20 Controlling Switch: A Switch able to control a set of FCDFs in order to create a Distributed Switch.

3.1.21 Controlling Switch Set: The Switch_Names of the up to two Controlling Switches that are part of a Distributed Switch.

3.1.22 Core Switch: A set of entities with the same Core Switch_Name that may host multiple Virtual Switches. A Core Switch may be a set of ports in a physical chassis, or in multiple physical chassis.

3.1.23 Core Switch_Name: In a Virtual Fabric capable Switch, the Switch_Name identifying the Core Switch (see 12.2).

3.1.24 Distributed Switch: A set of FCDFs associated with at least one Controlling Switch, that controls the operations of the set of FCDFs.

3.1.25 Domain: The highest level in the three-level address partitioning system specified by this standard (see 4.8).

3.1.26 Domain Address Manager: A Switch that is responsible for address assignment to other Switches outside of its Domain.

3.1.27 Domain Identifier: Bits 23 through 16 of an address identifier.

3.1.28 Domain_ID_List: A list where each record contains a Domain_ID value and the Switch_Name of the Switch assigned the Domain_ID (see 6.1.5).

3.1.29 Downstream Principal ISL: From the point of view of the local Switch, the downstream Principal ISL is the Principal ISL to which frames may be sent from the Principal Switch to the destination Switch. All Principal ISLs on the Principal Switch are downstream Principal ISLs. A Switch that is not the Principal Switch may have zero or more downstream Principal ISLs.

3.1.30 Distributed Service: An implementation of a Generic Service operating at a well-known address (see FC-GS-7).

3.1.31 Distributed Services Time-Out Value (D_S_TOV): A value that indicates the maximum time that a distributed service requestor shall wait for a response.

3.1.32 Domain_ID Overlap: During Fabric configuration, a condition in which the Domain_ID List of a Switch and the Domain_ID List of a received EFP (see 7.3) are both non-null and have records that associate the same Domain_ID to different Switch_Names.

3.1.33 Entry Switch: A role that a Switch assumes with respect to a distributed service request. The Switch that is attached to an Nx_Port making a service request assumes the role of an Entry Switch with respect to that request.

3.1.34 E_Port: A Fabric “Expansion” Port that attaches to another Interconnect_Port to create an Inter-Switch Link. An E_Port is the combination of one PE_Port and one VE_Port operating together.

3.1.35 E_Port Index: An index value associated with an E_Port used by the Fabric Shortest Path First Protocol.

3.1.36 Error_Detect_Timeout value (E_D_TOV): A time constant defined in FC-FS-3.

3.1.37 F_Port: The combination of one PF_Port and one VF_Port operating together.

3.1.38 Fabric: As defined in FC-FS-3 an entity that interconnects various Nx_Ports attached to it, and is capable of routing frames using only the D_ID information in an FC-2 frame header.

3.1.39 Fabric Controller: The logical entity responsible for operation of the Fabric identified by the well-known address FFFFFDh.

3.1.40 Fabric Element: The smallest unit of a Fabric that meets the definition of a Fabric. From the point of view of an attached Nx_Port, a Fabric consisting of multiple Fabric Elements is indistinguishable from a Fabric consisting of a single Fabric Element.

3.1.41 FCDF (FC Data-Plane Forwarder): A simplified FC switching entity that forwards FC frames among VA_Ports and VF_Ports through a FCDF Switching Element (see 17.3).

3.1.42 FCDF Set: The Switch_Names of the FCDFs that are part of a Distributed Switch.

3.1.43 F_Port Controller: The entity at the well-known address FFFFFEh.

3.1.44 Flood: To cause information to be sent to all Switches within the Fabric.

3.1.45 FL_Port: An L_Port that is able to perform the function of an F_Port, attached via a link to one or more NL_Ports in an Arbitrated Loop topology (see FC-AL-2). The AL_PA of an FL_Port is 00h. In this Standard, an FL_Port is assumed to always refer to a port to which NL_Ports are attached to a Fabric, and does not include F_Ports.

3.1.46 Fx_Port: A Switch Port capable of operating as an F_Port or FL_Port.

3.1.47 Fabric_Stability_Timeout value (F_S_TOV): A time constant used to ensure that Fabric stability has been achieved during Fabric Configuration.

3.1.48 Fabric Shortest Path First (FSPF): The link state protocol used for Path Selection.

3.1.49 G_Port: A generic Fabric Port that may function either as an E_Port, or as an F_Port.

3.1.50 GL_Port: A generic Fabric Port that may function either as an E_Port, or as an Fx_Port.

3.1.51 Hard Zone: A zone with the Hard Zone attribute specified.

3.1.52 Hard Zoning Enforcement: Zoning technique in which the Fabric limits frame exchange by frame-by-frame filtering.

3.1.53 Interconnect_Port: A generic reference to an E_Port or a B_Port.

3.1.54 Inter-Switch Link (ISL): A Link directly connecting the E_Port of one Switch to the E_Port of another Switch.

3.1.55 Isolated: A condition in which it has been determined that no Class N traffic may be transmitted across an ISL (see 7.6).

3.1.56 L_Port: An FC_Port that contains Arbitrated Loop functions associated with the Arbitrated Loop topology.

3.1.57 Link Descriptor: Contains information about an Inter-Switch Link including link type, the Domain_ID of the remote Switch it is connected to, the local and remote Port IDs, and the cost of the link itself.

3.1.58 Link State Record: A collection of Link Descriptors that completely describes the connectivity of a Switch to all Switches to which it is directly attached.

3.1.59 Locally-Enabled VF_ID List: the configured list of VF_IDs that an FC_Port supporting Virtual Fabrics is able to enable on a link.

3.1.60 Link: As defined in FC-FS-3.

3.1.61 Loop Fabric Address (LFA): An address identifier used to address an FL_Port (see table 1) for the purpose of loop management (see FC-LS).

3.1.62 Multiplexer: An instance of the FC-2M sublevel, multiplexing and demultiplexing frames between physical and virtual ports based on the D_ID/S_ID and/or VF_ID (see FC-FS-3).

3.1.63 N_Port: As defined in FC-FS-3, an N_Port is assumed to always refer to an Nx_Port in a direct Fabric-attached PN_Port, and does not include NL_Ports.

3.1.64 N_Port Identifier: An address identifier assigned to an N_Port.

3.1.65 Name_Identifier: As defined in FC-FS-3, a 64-bit identifier.

3.1.66 NL_Port: An Nx_Port in a PN_Port that is operating a Loop Port State machine (see FC-AL_2). It may be attached via a link to one or more NL_Ports and zero or more FL_Ports in an Arbitrated Loop topology. In this Standard, an NL_Port is assumed to always refer to a loop-attached port, and does not include N_Ports.

3.1.67 Non-zero Domain_ID_List: A Domain_ID_List that contains at least one record (see 7.3).

3.1.68 Nx_Port: An end point for Fibre Channel frame communication (see FC-FS-3).

3.1.69 PA_Port (Physical A_Port): The LCF within the Fabric that attaches to another PA_Port through a link.

3.1.70 Path: A route through the Fabric between a source and a destination.

3.1.71 Path Selection: A process whereby paths are selected.

3.1.72 PE_Port (Physical E_Port): The LCF within the Fabric that attaches to another PE_Port or to a B_Port through a link.

3.1.73 PF_Port (Physical F_Port): the LCF within the Fabric that attaches to a PN_Port (see FC-FS-3) through a link.

3.1.74 PN_Port: An entity not within a topology that includes an LCF and one or more Nx_Ports (see FC-FS-3).

3.1.75 Port: 1. A generic reference to an N_Port, NL_Port, F_Port, FL_Port, B_Port, or E_Port. 2. The lowest level in The lowest level in the three-level address partitioning system specified by this standard (see 4.8).

3.1.76 Port VF_ID: A configurable VF_ID that is associated with any untagged frame received by a VF capable PE_Port or PF_Port.

3.1.77 Point-to-Point Link: A Fibre Channel link connecting two ports.

3.1.78 Port Identifier: Bits 7 through 0 of an address identifier.

3.1.79 Port Index: A three byte value used by FSPF to identify Switch ports.

3.1.80 Port Mode: A generic reference to E_Port, B_Port, F_Port or FL_Port operation.

3.1.81 Preferred Domain_ID: A Domain_ID previously granted to a Switch by the Domain Address Manager or through administrative means.

3.1.82 Principal ISL: An Inter-Switch Link that is used to communicate with the Principal Switch.

3.1.83 Principal Switch: A Switch that has been selected to perform certain Fabric Configuration duties.

3.1.84 Reliable Flood: Flooding where all Switches are guaranteed to receive the flooded message.

3.1.85 Remote Switch: A Switch that may be reached via one or more ISLs. A remote Switch may be adjacent to the local Switch, or may reached via one or more intermediate Switches.

3.1.86 Resource_Allocation_Timeout value (R_A_TOV): A time constant defined in FC-FS-3.

3.1.87 Router: An entity within a Switch responsible for the routing of connectionless frames.

3.1.88 Routing: A process whereby the appropriate Switch Port(s) to deliver a connectionless frame towards its destination is identified.

3.1.89 Soft Zoning Enforcement: Zoning technique in which the Fabric enforces membership through name server visibility.

3.1.90 Switch: 1. A Fabric Element conforming to this Standard. 2. A member of the Fabric collective.

3.1.91 Switch Construct: An entity within a Switch responsible for transporting frames between Switch Ports.

3.1.92 Switching Element: The set of functions performed by the Path Selector, The Router, The Switch Construct, the Address Manager and the Fabric Controller.

3.1.93 Switch_Name: A Name_Identifier that identifies a Switch or a Bridge device. The format of the name is specified in FC-FS-3. Each Switch and Bridge device shall provide a unique Switch_Name within the Fabric.

3.1.94 Switch Port: An E_Port, F_Port, or FL_Port.

3.1.95 Switch_Priority: A value used during Principal Switch selection to cause one Switch to be favored over another.

3.1.96 T10 Vendor ID: A character string that uniquely identifies a vendor.

3.1.97 Topology: The communication infrastructure that provides Fibre Channel communication among a set of PN_Ports (e.g., a Fabric, an Arbitrated Loop, or a combination of the two).

3.1.98 Upstream Principal ISL: From the point of view of the local Switch, the upstream Principal ISL is the Principal ISL to which frames may be sent from the local Switch to the Principal Switch. A Switch that is not the Principal Switch always has exactly one upstream Principal ISL. The Principal Switch does not have an upstream Principal ISL.

3.1.99 VA_Port (Virtual A_Port): An instance of the FC-2V sublevel of Fibre Channel that connects to another VA_Port.

3.1.100 Virtual Fabric: An interconnected set of Virtual Switches and/or Switches identified by a Virtual Fabric ID (VF_ID).

3.1.101 Virtual Fabric Tagging Header (VFT_Header): as defined in FC-FS-3.

3.1.102 VE_Port (Virtual E_Port): An instance of the FC-2V sublevel that connects to another VE_Port or to a B_Port to create an Inter-Switch Link. A VE_Port is addressable by the VE_Port or B_Port connected to it through the Fabric Controller well-known address identifier (i.e., FF FF FD).

3.1.103 VF_Port (Virtual F_Port): An instance of the FC-2V sublevel that connects to one or more VN_Ports (see FC-FS-3). A VF_Port is addressable by a VN_Port connected to it through the F_Port Controller well-known address identifier (i.e., FF FF FE).

3.1.104 Zero Domain_ID_List: A Domain_ID_List that is empty (see 7.3).

3.1.105 Zone: A group of Zone Members. Members of a Zone are made aware of each other, but not made aware of Zone Members outside the Zone.

3.1.106 Zone Definition: The parameters that define a Zone.

3.1.107 Zone Member: The specification of a device to be included in a Zone.

3.1.108 Zone Member Definition: The parameters that define a Zone Member including the Zone Member Type and Zone Member Information.

3.1.109 Zone Name: The name assigned to a Zone.

3.1.110 Zone Set: A set of Zones that are used in combination.

3.1.111 Zone Set Database: The database that contains the Zone Sets not enforced by the Fabric.

3.1.112 Zone Set Name: The name assigned to a Zone Set.

3.1.113 Zone Set State: The state of a Switch Zone Set (*Activated* or *Deactivated*).

3.1.114 Zoning Configuration: A set of Zoning data including the Zone Set state, and Zone definitions.

3.1.115 Zoning Database: A generic term used to indicate both the Active Zone Set and the Zone Set Database.

3.2 Editorial conventions

In FC-SW-6, a number of conditions, mechanisms, sequences, parameters, events, states, or similar terms are printed with the first letter of each word in uppercase and the rest lowercase (e.g., Exchange, Sequence). Any lowercase uses of these words have the normal technical English meanings.

Lists sequenced by letters (e.g., a-red, b-blue, c-green) show no ordering relationship between the listed items. Numbered lists (e.g., 1-red, 2-blue, 3-green) show an ordering relationship between the listed items.

In case of any conflict between figure, table, and text, the text, then tables, and finally figures take precedence. Exceptions to this convention are indicated in the appropriate clauses.

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. Exceptions to this convention are indicated in the appropriate clauses.

Data structures in this standard are displayed in Fibre Channel format (i.e., “big-endian”), while specifications originating in IEEE and IETF may display data structures in Ethernet format (i.e., “little-endian”).

When the value of the bit or field is not relevant, x or xx appears in place of a specific value. If a field or a control bit in a frame is specified as not meaningful, the entity that receives the frame shall not check that field or control bit.

Numbers that are not immediately followed by lower-case b or h are decimal values.

Numbers immediately followed by lower-case b (xxb) are binary values.

Numbers or upper case letters immediately followed by lower-case h (xxh) are hexadecimal values.

3.3 State Machine notation

State machines in this standard should use the style shown in figure 1.

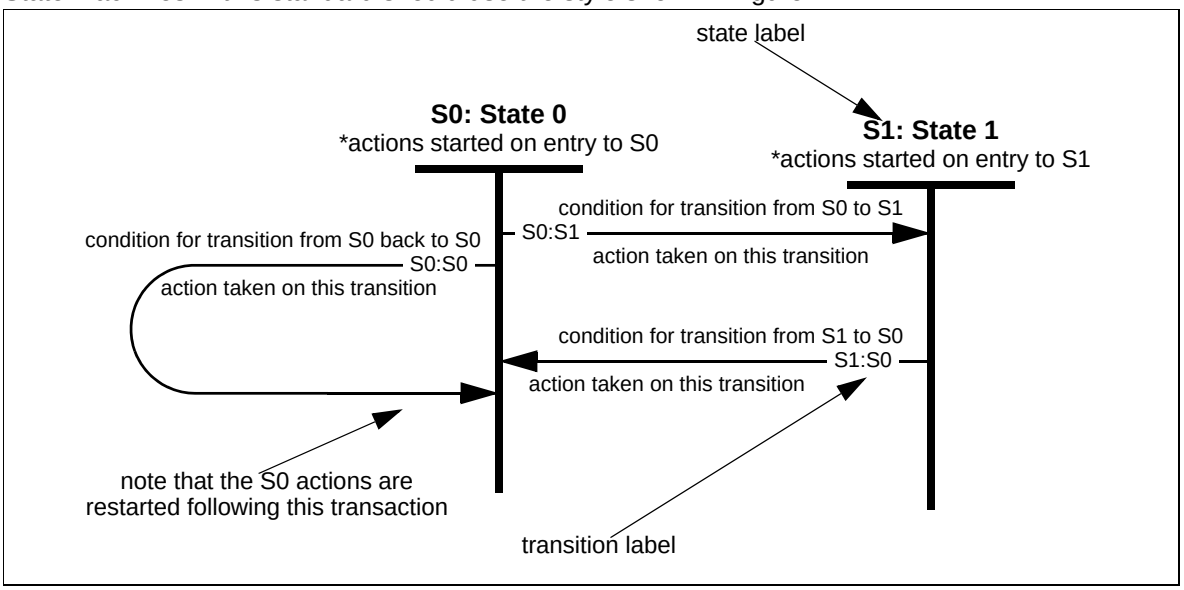

Figure 1 – Sample State Machine

These state machines make three assumptions:

a) Time elapses only within discrete states.

b) State transitions are logically instantaneous, so the only actions taken during a transition are setting flags and variables and sending signals. These actions complete before the next state is entered.

c) Every time a state is entered, the actions of that state are started. Note that this means that a transition that points back to the same state repeats the actions from the beginning. All the actions started upon entry complete before any tests are made to exit the state.

3.4 Abbreviations, acronyms, and symbols

Abbreviations and acronyms applicable to this International Standard are listed below. Definitions of several of these items are included in 3.1. Abbreviations used that are not listed below are defined in FC-FS-3.

Area_ID	Area Identifier
CT	Common Transport
Domain_ID	Domain Identifier
D_S_TOV	Distributed_Services_Timeout Value
E_D_TOV	Error_Detect_Timeout value
ELS	Extended Link Service
FAN	Fabric Address Notification Extended Link Service
FSM	Finite State Machine
FC-AL-2	Fibre Channel Arbitrated Loop-2, reference [1]
FC-BB-5	Fibre Channel Backbone-5, reference [7]
FC-BB-6	Fibre Channel Backbone-6, reference [17]

FC-DA-2	Fibre Channel- Device Attach-2, reference [6]
FC-FS-3	Fibre Channel - Framing and Signaling - 3, reference [4]
FC-FS-4	Fibre Channel - Framing and Signaling - 4, reference [15]
FC-GS-6	Fibre Channel - Generic Services-6, reference [5]
FC-GS-7	Fibre Channel - Generic Services-7, reference [13]
FC-IFR	Fibre Channel - Inter-Fabric Routing, reference [11]
FC-LS-2	Fibre Channel - Link Services-2, reference [8]
FC-LS-3	Fibre Channel - Link Services-3, reference [12]
FC-MI-3	Fibre Channel - Methodologies for Interconnects -3, reference [2]
FC-PI-5	Fibre Channel -Physical Interface - 5, reference [3]
FC-SB-4	Fibre Channel - Single Byte Command Sets - 4 reference [9]
FC-SB-5	Fibre Channel - Single Byte Command Sets - 5 reference [16]
FC-SP	Fibre Channel - Security Protocols reference [10]
FC-SP-2	Fibre Channel - Security Protocols - 2 reference [14]
F_S_TOV	Fabric_Stability_Timeout value
FDMI	Fabric Device Management Interface
FSPF	Fabric Shortest Path First
ISL	Inter-Switch Link
IU	Information Unit
LCF	Link Control Facility
LFA	Loop Fabric Address
LSR	Link State Record
R	Reserved
R_A_TOV	Resource_Allocation_Timeout value
RFC	Request For Comment
SM	State Machine
SW_ACC	Switch Fabric Link Service Accept
SW_ILS	Switch Internal Link Service
SWN	Switch Name
SWP	Switch Priority
SW_RJT	Switch Fabric Link Service Reject
VF_ID	Virtual Fabric Identifier
WKA	Well-Known Address
WWN	World Wide Name
1xAL_TIME	One times the AL_TIME
1xF_S_TOV	One times the F_S_TOV
2xAL_TIME	Two times the AL_TIME
2xF_S_TOV	Two times the F_S_TOV
3xAL_TIME	Three times the AL_TIME
=	Is equal to

3.5 Definition of Compliance Terms

The usual definitions of the following terms do not apply in this standard and therefore they are defined below:

Prohibited: If a feature or parameter value is Prohibited, it means that it shall not be used between compliant implementations.

Required: If a feature or parameter value is Required, it means that it shall be used between compliant implementations.

Allowed: If a feature or parameter value is Allowed, it means that it may be used between compliant implementations.

3.6 Keywords

3.6.1 ignored: A keyword used to describe an unused bit, byte, word, field or code value. The contents or value of an ignored bit, byte, word, field or code value shall not be examined by the receiving device and may be set to any value by the transmitting device.

3.6.2 invalid: A keyword used to describe an illegal or unsupported bit, byte, word, field or code value. Receipt of an invalid bit, byte, word, field or code value shall be reported as an error.

3.6.3 mandatory: A keyword indicating an item that is required to be implemented as defined in this standard.

3.6.4 may: A keyword that indicates flexibility of choice with no implied preference (equivalent to “may or may not”).

3.6.5 may not: A keyword that indicates flexibility of choice with no implied preference (equivalent to “may or may not”).

3.6.6 obsolete: A keyword indicating that an item was defined in prior Fibre Channel standards but has been removed from this standard.

3.6.7 optional: A keyword that describes features that are not required to be implemented by this standard. However, if any optional feature defined by this standards is implemented, then it shall be implemented as defined in this standard.

3.6.8 reserved: A keyword referring to bits, bytes, words, fields and code values that are set aside for future standardization. A reserved bit, byte, word or field shall be set to zero, or in accordance with a future extension to this standard. Recipients are not required to check reserved bits, bytes, words or fields for zero values. Receipt of reserved code values in defined fields shall be reported as error.

3.6.9 restricted: A keyword referring to bits, bytes, words, and fields that are set aside for use in other Fibre Channel standards. A restricted bit, byte, word, or field shall be treated as a reserved bit, byte, word or field for the purposes of the requirements defined in this standard.

3.6.10 shall: A keyword indicating a mandatory requirement. Designers are required to implement all such mandatory requirements to ensure interoperability with other products that conform to this standard.

3.6.11 should: A keyword indicating flexibility of choice with a strongly preferred alternative; equivalent to the phrase “it is strongly recommended.

3.7 T10 Vendor ID Fields

A T10 Vendor ID shall be a string of one to eight characters that is recorded in an informal list of Vendor IDs maintained by INCITS Technical Committee T10 (see http://www.t10.org).

A field described as containing a T10 Vendor ID shall contain the first character of the T10 Vendor ID in the highest order byte of the field, and successive characters of the T10 Vendor ID in successively lower order bytes of the field. Any bytes of the field not filled by characters of the T10 Vendor ID shall be filled with ASCII space characters (20h).

4 Structure and Concepts

4.1 Overview

This standard describes the operation and interaction of Fibre Channel Switches. This includes E_Port Operation and Fabric Operation.

4.2 E_Port Operation

E_Port operation specifies the tools and algorithms for interconnection and initialization of Fibre Channel Switches to create a multi-Switch Fabric. Fabric operation defines an E_Port (“Expansion Port”) that operates in a manner similar to an PN_Port and F_Port, as defined in FC-FS-3, with additional functionality provided for interconnecting Switches.

E_Port operation defines credit models and management between E_Ports for the various classes of service other than Class F. E_Ports conforming to this Standard support Class F, Class 2 and/or Class 3. Support for other classes of service are not defined by Fabric operation.

E_Port operation defines how ports that are capable of being an E_Port, F_Port, and/or FL_Port discover and self-configure for their appropriate operating mode. Once a port establishes that it is connected to another Switch and is operating as an E_Port, an address assignment algorithm is executed to allocate port addresses throughout the Fabric.

4.3 Fabric Operation

Fabric operation includes the following:

a) Fabric Configuration- Describes how a Principal Switch is selected and describes the address assignment algorithm.

b) Exchange Switch Capabilities - Allows Switches to exchange certain operational capabilities such as which path selection protocols are supported.

c) B_Port - A simplified E_Port that allows Bridge type devices to participate in Fabric operation.

d) Path selection - Path Selection is the process by which a Switch determines the best path from a source domain to a destination domain. These paths may then be used in any appropriate manner by the Switch to move frames to their destinations. This path selection process does not require nor preclude the use of static or dynamic load-balancing. The standard defines the Fabric Shortest Path First (FSPF) protocol.

e) Distributed Server communication - The Distributed Server model allows the Generic Services operating at well-known addresses (see FC-GS-7) to be distributed among Switches that comprise the Fabric. Both the distributed Name Server and the distributed Management Server are described. In addition, the Inter-Switch FDMI protocol has been defined.

f) Exchange of Zoning information - Defines how zoning information is communicated between Switches in the Fabric. Zoning information is exchanged between Switches when two Fabrics are merged, and when changes to zoning information are propagated between Switches.

g) Distributed Event Notification - Defines how Registered State Change Notifications (RSCNs) and Distribute Registered Link Incident Records (DRLIR) are distributed between Switches in the Fabric;

h) Virtual Fabrics Switch Support - Defines the operation of Virtual Fabrics from a frame tagging perspective;

i) Enhanced Commit Service - Defines a commit service that allows serialization based on application and advanced error recovery;

j) Virtual Channel Architecture - Defines virtual channels between E_Ports.

In addition to normal Fabric operation and topologies the Distributed Switch Environment is described in 17.

4.4 Fabric Definition

The Fabric serves as a transport that provides a switched interconnect between Nx_Ports.

4.5 Switch

A Switch is the smallest entity that may function as a Switch-based Fibre Channel Fabric. Figure 2 illustrates the conceptual model of a Basic Switch.

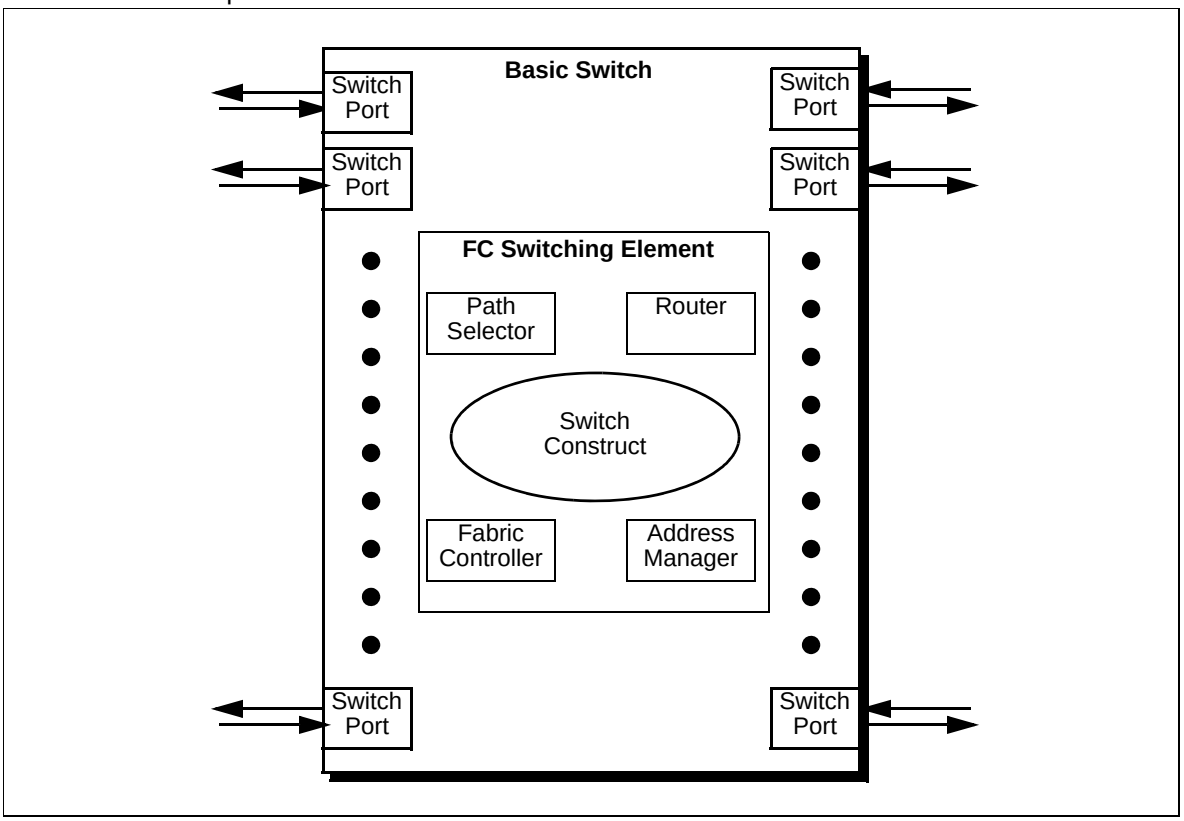

Figure 2 – Basic Switch Model

A Switch is composed of the following major components:

a) Two or more Switch Ports;

b) a Switch Construct, capable of either multiplexed frame switching or circuit switching, or both;

c) an Address Manager;

d) a Path Selector, which performs path selection;

e) a Router;

f) and a Fabric Controller.

The set of functions performed by the Path Selector, The Router, The Switch Construct, the Address Manager and the Fabric Controller is referred to as a FC Switching Element.

The FC Switching Element abstraction allows the definition of an Enhanced Switch functional model, based on the functional layering of Fibre Channel (see FC-FS-3). Figure 3 shows the functional model of an Enhanced Switch.

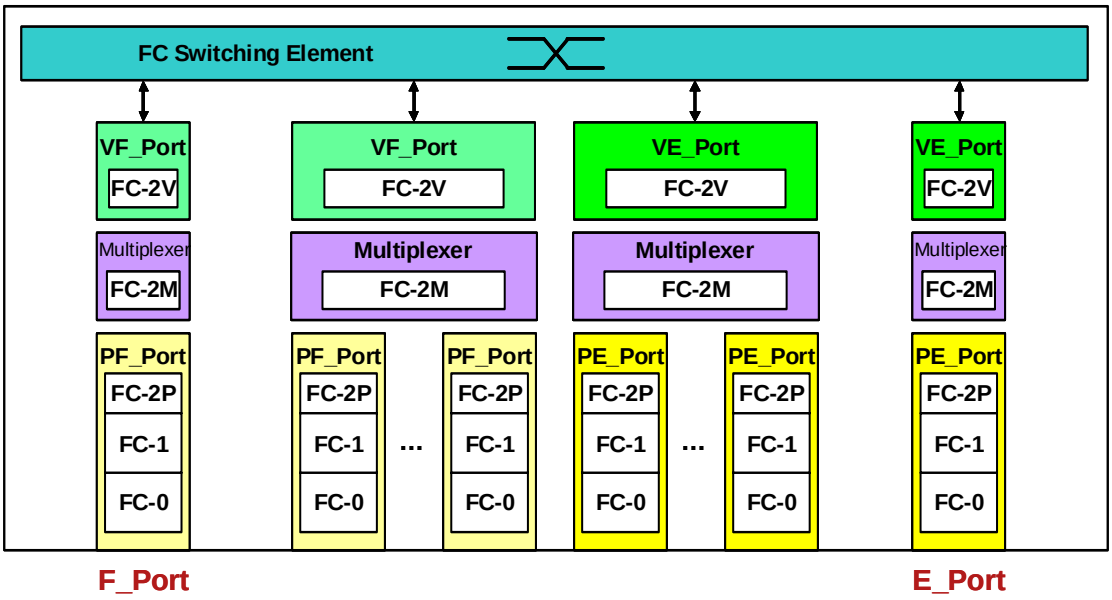

Figure 3 – Enhanced Switch Functional Model

An Enhanced Switch is able to aggregate its physical ports in sets that behave as virtual ports, providing higher bandwidth than the one available to a single physical port. A physical port is an LCF (see FC-FS-3), that may behave as a Physical F_Port (PF_Port) or as a Physical E_Port (PE_Port). A virtual port is an instance of the FC-2V sublevel of Fibre Channel (see FC-FS-3), that may behave as a Virtual F_Port (VF_Port) or as a Virtual E_Port (VE_Port).

As defined, a Switch Port may be either an E_Port, an F_Port, or an FL_Port. A Switch Port that is capable of assuming more than one of these roles is called a G_Port or GL_Port. Once a Switch Port assumes a role, via the Switch Port Initialization Procedure, it shall remain in that role until an event occurs that causes re-initialization.

The Link joining a pair of E_Ports is called an Inter-Switch Link (ISL). ISLs carry frames originating from Nx_Ports and those frames generated within the Fabric. The frames generated within the Fabric serve as control, management and support for the Fabric.

Switches may be joined freely or in a structured fashion to form a larger Fabric, as illustrated in figure 4

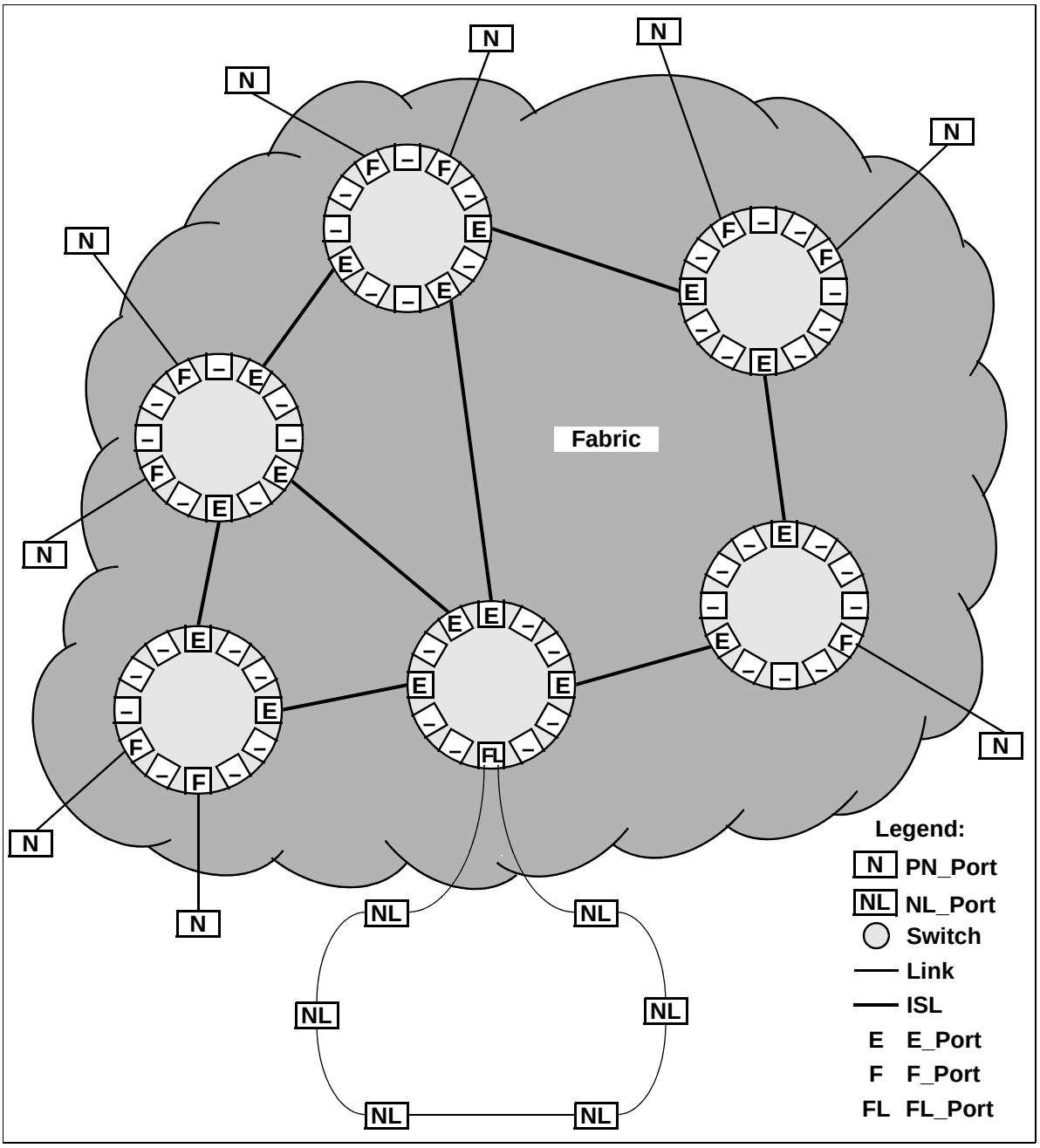

Figure 4 – Multiple Switch Fabric Example

The structure of the Switch Construct in the Switch, as seen in figure 2, is undefined and beyond the scope of this Standard. It may support either or both circuit switching and multiplexed frame switching. It may be non-blocking, allowing concurrent operation of all possible combinations or it may be blocking, restricting operations. The Switch Construct may also contain redundancy, as may be required for high availability configurations.

The Address Manager is responsible for the assignment of addresses within some portion of the Fabric. Within the Switch, the Address Manager is responsible for Domain_ID(s) for the Switch, and allocating Area_IDs and Port_IDs within the acquired Domain(s).

The Path Selector is a logical entity that establishes frame routing paths.

The Router is a logical entity that performs the routing of Class F, Class 2 and Class 3 frames to their final destination.

The Fabric Controller is a logical entity that performs the management of the Switch. The Fabric Controller has the characteristics of an N_Port, though it may or may not be attached to the Fabric via a Link.

4.6 Switching characteristics

4.6.1 Switching overview

Path, circuit switching, and frame routing within a Switch occurs synchronously to the current word alignment of the outbound fibre.

Synchronous switching guarantees retention of the established word alignment on the outbound fibre of the Switch Port.

A switching event occurs every time a connectionless frame is transmitted and when a connection based service is established, suspended or terminated.

Synchronous switching associated with connectionless frame routing and connection oriented Dedicated Connections or virtual connection Services shall guarantee the word alignment on the outbound fibre. Switches shall ensure that synchronous switching only occurs between frames.

4.7 Switch Ports and Bridge Ports

4.7.1 General Characteristics

A Switch shall have two or more Switch Ports. A Switch equipped only with F_Ports or FL_Ports forms a non-expandable Fabric. To be part of an expandable Fabric, a Switch shall incorporate at least one Switch Port capable of E_Port operation.

A Switch Port supports one or more of the following Port Modes: E_Port, B_Port, F_Port, or FL_Port. Switch Ports that assume either the E_Port or B_Port mode are generally referred to as Interconnect_Ports. A Switch Port that is capable of supporting more than one Port Mode attempts to configure itself first as an FL_Port (as defined in FC-AL-2), then as an E_Port or a B_Port (as defined in this Standard), and finally as an F_Port (as defined in FC-FS-3), depending on which of the four Port Modes are supported by the Switch Port. A Bridge Port shall only support B_Port operation.

NOTE 2 – The characteristics of a Bridge device are not described in this standard. However, one type of Bridge device is described in the FC-BB-3 standard.

The detailed procedure for Port Mode selection is described in 7.2.

4.7.2 F_Port

An F_Port is the point at which all frames originated by an N_Port enter the Fabric, and all frames destined for an N_Port exit the Fabric. An F_Port may also be the Fabric entry point for frames originated by an N_Port destined for an internal Fabric destination, such as the Fabric Controller. Similarly, an F_Port may also be the Fabric exit point for frames originated internal to the Fabric and destined for an N_Port. Frames shall not be communicated across a Link between an F_Port and anything other

than an N_Port. Functionally, an F_Port is the combination of one PF_Port and one VF_Port operating together.

F_Ports are described in detail in 5.3.

4.7.3 FL_Port

An FL_Port is the point at which all frames originated by an NL_Port enter the Fabric, and all frames destined for an NL_Port exit the Fabric. An FL_Port may also be the Fabric entry point for frames originated by an NL_Port destined for an internal Fabric destination, such as the Fabric Controller. Similarly, an FL_Port may also be the Fabric exit point for frames originated internal to the Fabric and destined for an NL_Port. Frames shall not be communicated across a Link between an FL_Port and anything other than an NL_Port.

FL_Ports are described in detail in 5.4.

4.7.4 E_Port

An E_Port is the point at which frames pass between the Switches within the Fabric. Frames with a destination other than the local Switch or any N_Port or NL_Port attached to the local Switch exit the local Switch through an E_Port. Frames that enter a Switch via an E_Port are forwarded to a local destination, or are forwarded towards their ultimate destination via another E_Port. Frames shall not be communicated across a Link between an E_Port and anything other than an E_Port or a B_Port. Functionally, an E_Port is the combination of one PE_Port and one VE_Port operating together.

E_Ports are described in detail in 5.5.

4.7.4 B_Port

A Bridge Port (B_Port) is a port on a Bridge device. It normally functions as a conduit between the Switch and the Bridge for frames destined for or through a Bridge device. A B_Port is also used to carry frames between a Switch and the Bridge device for purposes of configuring the Bridge device.

4.7.5 G_Ports and GL_Ports

A G_Port is a Switch Port that is capable of either operating as an E_Port or F_Port. A G_Port determines through Port Initialization whether it operates as an E_Port or as an F_Port. A GL_Port is a G_Port that is also capable of operating as an FL_Port.

4.7.6 PF_Port

A PF_Port is the LCF within the Fabric that attaches to a PN_Port (see FC-FS-3) through a link.

4.7.7 VF_Port

A VF_Port is an instance of the FC-2V sublevel that connects to one or more VN_Ports (see FC-FS-3). A VF_Port is addressable by a VN_Port connected to it through the F_Port Controller well-known address identifier (i.e., FF FF FEh).

4.7.8 PE_Port

A PE_Port is the LCF within the Fabric that attaches to another PE_Port or to a B_Port through a link.

4.7.9 VE_Port

A VE_Port is an instance of the FC-2V sublevel that connects to another VE_Port or to a B_Port to create an Inter-Switch Link. A VE_Port is addressable by the VE_Port or B_Port connected to it through the Fabric Controller well-known address identifier (i.e., FF FF FDh).

4.8 Fabric Addressing

Switches use the address partitioning model as described below. The 24-bit address identifier is divided into three fields: Domain, Area, and Port, as shown in figure 5.

23	**22**	**21**	**20**	**19**	**18**	**17**	**16**	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Domain_ID								Area_ID								Port_ID							
Address Identifier																							

Figure 5 – Domain, Area, and Port Address Partitioning

A Domain is one or more Switches that have the same Domain_ID for all N_Ports and NL_Ports within or attached to those Switches, except for Well-Known Addresses. If there is more than one Switch in the Domain, each Switch within the Domain shall be directly connected via an ISL to at least one other Switch in the same Domain.

An Area_ID shall apply to either of the following:

a) One or more N_Ports within and attached to a single Switch, except for Well-Known Addresses;

b) an Arbitrated Loop of NL_Ports attached to a single FL_Port.

A single Arbitrated Loop shall have exactly one Area_ID.

A Port_ID shall apply to either of the following:

a) a single N_Port within a Domain/Area, except for Well-Known Addresses;

b) the valid AL_PA of a single NL_Port or FL_Port on an Arbitrated Loop.

Address identifier values for this Standard are listed in table 1. Any value listed as Reserved is not meaningful within this Standard.

Table 1 – Address Identifier Values (Part 1 of 2)

Address Identifier (hex)			Description
Domain_ID	**Area_ID**	**Port_ID**	
00	00	00	Undefined[a]
00	00	AL_PA	E_Port: Reserved F_Port: Reserved FL_Port: Private Loop NL_Port[b and g]
00	00	non-AL_PA	Reserved
00	01 - FF	00 - FF	Reserved
01 - EF	00 - FF	00	F_Port: N_Port Identifier[h] FL_Port: Loop Fabric Address[c]
01 - EF	00 - FF	AL_PA	F_Port: N_Port Identifier[h] FL_Port: N_Port Identifier for Public Loop NL_Port[c]
01 - EF	00 - FF	non-AL_PA	F_Port: N_Port Identifier[h] FL_Port: Reserved
F0 - FE	00 - FF	00 - FF	Reserved
FF	00 - F9	00 - FF	Reserved
FF	FA	00-0F	Reserved for Internal Loopback Addresses
FF	FA	10-1F	Reserved for External Loopback Addresses
FF	FA	20-FF	Reserved
FF	FB	00 - FF	Obsolete in FC-SW-5
FF	FC	00	Reserved
FF	FC	01 - EF	N_Port Identifier for Domain Controller[d]
FF	FC	F0 - FF	Reserved
FF	FD - FE	00 - FF	Reserved
FF	FF	00 - EF	Reserved
FF	FF	F0 - FC	Well-Known Address[e]

Table 1 – Address Identifier Values (Part 2 of 2)

Address Identifier (hex)			Description
Domain_ID	Area_ID	Port_ID	
FF	FF	FD	N_Port Identifier for Fabric Controller[f]
FF	FF	FE	N_Port Identifier for F_Port Controller
FF	FF	FF	Broadcast Address

[a] This value is used by an N_Port requesting an address identifier during FLOGI.

[b] See FC-AL-2 for a definition of AL_PA and FC-DA for a definition of Private Loop and FL_Port operation with Private Loop devices.

[c] See FC-AL-2 for the definition and use of Loop Fabric Address, and for a definition of Public Loop.

[d] A Domain Controller Identifier may be used to address the Fabric Controller of a remote Switch that may or may not be connected via an ISL to the originating Switch. The Port_ID field is set to the Domain_ID of the remote Switch.

[e] The usage of Well-Known Addresses FFFFF0h through FFFFFCh, are not defined by this Standard. FC-FS-3 defines or reserves these values for Well-Known Addresses.

[f] This address identifier has special usage depending on the originator. If the originator is an attached external N_Port or NL_Port (attached via an F_Port or FL_Port) then the destination of a frame sent to FFFFFDh is the Fabric Controller of the local Switch. If the originator is the Fabric Controller of the local Switch, then the destination of a frame sent to FFFFFDh via an ISL is the Fabric Controller of the remote Switch at the other end of the ISL.

[g] This value is used by a public loop NL_Port requesting an address identifier during FLOGI.

[h] A Switch may use the same Domain_ID and Area_ID for the N_Port Identifier of N_Ports attached to different F_Ports.

4.9 Class F Service

Class F service is a connectionless service similar to Class 2 that is used for internal control of the Fabric. Class F service as used by this Standard is defined in 5.8.

5 Switch Ports and Bridge Ports

5.1 Overview

This clause defines the specific behaviors for all modes of a Switch Port and a Bridge port. Note that the models described below are defined for purposes of describing behavior. No implication is made as to whether the actual implementation of an element is in hardware or software. An element may be implemented on a per-Port basis, or may be a logical entity that is embodied in a single physical implementation shared by multiple ports.

A Switch Port may be able to operate in more than one mode, and configure itself to the appropriate mode during the initialization process (see 7.2). During initialization, the Switch Port may assume a mode for purposes of determining if that mode is appropriate. For example, a Switch Port operates in FL_Port mode to determine if it is attached to a loop of NL_Ports. If that is not successful, it then tries operating as an E_Port to see if another E_Port or B_Port is attached. The Switch Port continues until it finds a mode in which to operate.

A Bridge device may contain one or more Bridge Ports (B_Ports).

Ports that operate in the E_Port or B_Port mode are generically referred to as Interconnect_Ports. A single Inter-Switch Link (ISL) connects two Interconnect_Ports together. Valid combinations of Interconnect_Ports are described below:

a) E_Port to E_Port;

b) E_Port to B_Port.

B_Port to B_port ISLs are not allowed.

5.2 Model elements

5.2.1 FC Transports

The FC-FS-3 Transport includes all of the functionality described in FC-FS-3 to construct and deconstruct a frame, to encode and decode the words that make up the frame, and to transmit and receive the frame on the physical media. The FC-AL-2 Transport contains additional functionality to support the Arbitrated Loop protocols. See FC-BB-5 for additional Fibre Channel transports.

5.2.2 Switch Transport

The Switch Transport is an abstraction to show the “back end” of the Switch Port as it interacts with the Switch Construct and/or other Switch Ports within the Switch. The Switch Transport exists to move frames between the Switch Port and the rest of the Switch. No other implementation details are implied by this element.

5.2.3 Control Facilities

The Control Facilities are internal logical ports that receive and perform requests, and generate responses. Each Control Facility has associated with it an address identifier, and support for classes of service. The Control Facilities also manage the various Transport elements.

5.2.4 Link Services

The Link Services represent the various Link Services that are supported by the corresponding Control Facility.

5.3 F_Port Operation

An F_Port is the point at which an external PN_Port is attached to the Fabric. It normally functions as a conduit to the Fabric for frames transmitted by the PN_Port, and as a conduit from the Fabric for frames destined for the PN_Port.

An F_Port shall support one or more of the following classes of service: Class 2 service, Class 3 service. An F_Port shall not intentionally transmit Class F frames on its outbound fibre. An PN_Port that receives a Class F frame shall discard it, as required by FC-FS-3.

An F_Port shall not admit to the Fabric any Class F frames, any Primitive Sequences, or any Primitive Signals other than Idle, that the F_Port receives on its inbound fibre.

NOTE 3 – Primitive Signals and Primitive Sequences are prohibited from entering the Fabric by FC-FS-3. For example, if an R_RDY was admitted to a Fabric, it may propagate to another F_Port and be transmitted by that F_Port, disrupting credit on that Link.

The model of an F_Port on an FC-FS-3 Transport is shown in figure 6.

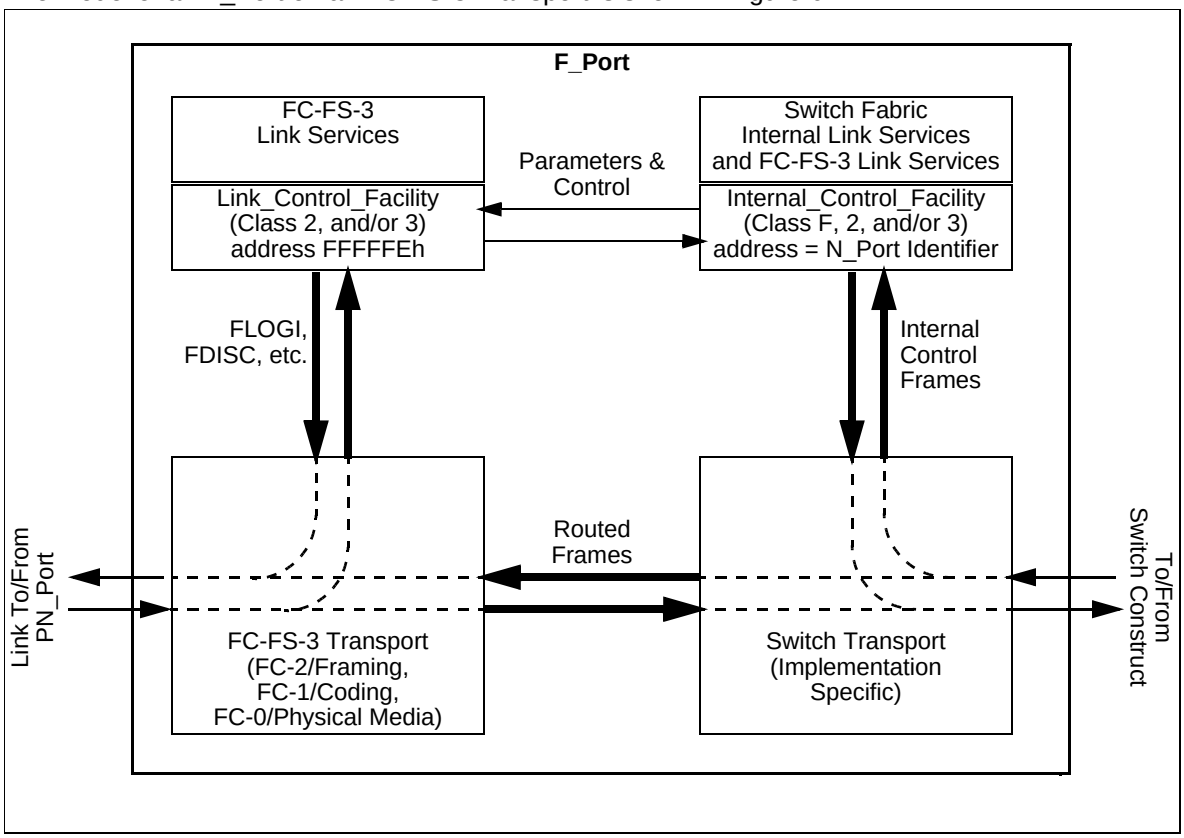

Figure 6 – F_Port Model

An F_Port contains an FC-FS-3 Transport element through which passes all frames and Primitives transferred across the Link to and from the PN_Port. Frames received from the PN_Port are either

directed to the Switch Construct via the Switch Transport element, or directed to the Link_Control_Facility.The Link_Control_Facility receives frames related to Link Services such as FLOGI, and transmits responses to those Link Service frames.

Frames received from the FC-FS-3 Transport element that are destined for other ports are directed by the Switch Transport to the Switch Construct for further routing. Frames received from the Switch Construct by the Switch Transport are directed either to the FC-FS-3 Transport for transmission to the PN_Port, or to the Internal_Control_Facility. The Internal_Control_Facility receives frames related to Switch Fabric Internal Link Services, and transmits responses to those Internal Link Services frames. Information is passed between the Internal_Control_Facility and the Link_Control_Facility to effect the control and configuration of the Transport elements.

The F_Port is used by Switches to transmit and receive frames with a single PN_Port. A Link to an F_Port always connects to exactly one PN_Port.

An F_Port Link follows the FC-0, FC-1, and FC-2 protocols defined for point-to-point Links as defined in FC-FS-3.

See FC-BB-5 for additional Fibre Channel transports.

5.4 FL_Port Operation

An FL_Port is the point at which one or more external NL_Ports are attached to the Fabric. It normally functions as a conduit to the Fabric for frames transmitted by the attached NL_Ports, and as a conduit from the Fabric for frames destined for the attached NL_Ports.

An FL_Port shall support one or more of the following classes of service: Class 2 service, Class 3 service. An FL_Port shall not intentionally transmit Class F frames on its outbound fibre. An FL_Port shall not admit to the Fabric any Class F frames, any Primitive Sequences, or any Primitive Signals other than Idle, that the FL_Port receives on its inbound fibre.

NOTE 4 – It is recommended that an FL_Port that conforms to this Standard also conform to the FL_Port requirements defined in FC-DA and FC-MI-2.

The model of an FL_Port is shown in figure 7.

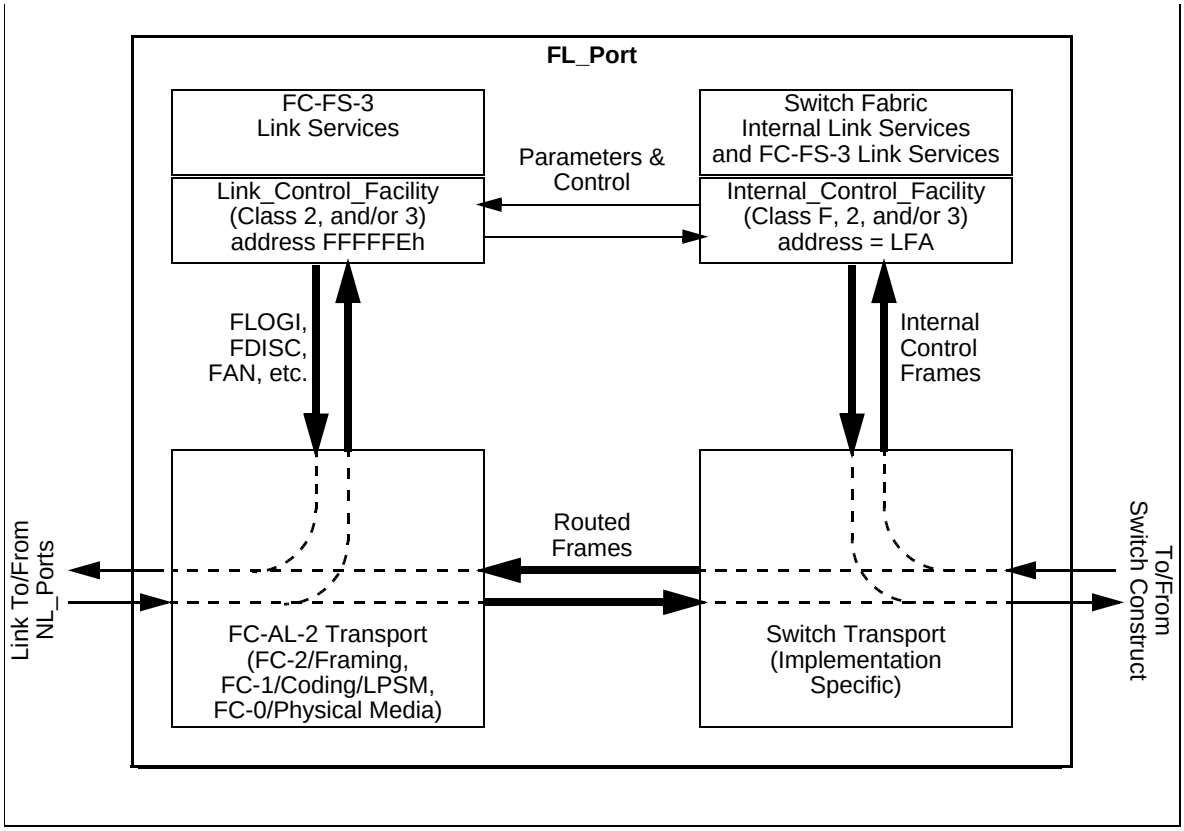

Figure 7 – FL_Port Model

An FL_Port contains an FC-AL-2 Transport element that passes all frames and Primitives transferred across the Link to and from the multiple NL_Ports. Frames received from the NL_Ports are either directed to the Switch Construct via the Switch Transport element, or directed to the Link_Control_Facility.The Link_Control_Facility receives frames related to Link Services such as FLOGI, and transmits responses to those Link Service frames. The Link_Control_Facility also transmits and receives Loop Initialization Sequences and transmits the FAN ELS.

Frames received from the FC-AL-2 Transport element that are destined for other ports are directed by the Switch Transport to the Switch Construct for further routing. Frames received from the Switch Construct by the Switch Transport are directed either to the FC-AL-2 Transport for transmission to the destination NL_Port, or to the Internal_Control_Facility. The Internal_Control_Facility receives frames related to Switch Fabric Internal Link Services and Loop management Extended Link Services (see FC-LS), and transmits responses to those Link Services frames. Information is passed between the Internal_Control_Facility and the Link_Control_Facility to effect the control and configuration of the Transport elements.

The FL_Port is used by Switches to transmit and receive frames with one or more attached NL_Ports. A Link to an FL_Port connects to one or more NL_Ports.

An FL_Port Link follows the FC-0, FC-1, and FC-2 protocols defined in FC-FS-3, with the additional Arbitrated Loop protocols defined in FC-AL-2.

5.5 E_Port Operation

An E_Port is the point at which a Switch is connected to another Switch to create a multi-Switch Fabric. Also, an E_Port is the point at which a Switch is connected to a Bridge device. It normally functions as a conduit between the Switches for frames destined for remote N_Ports and NL_Ports. An E_Port is also used to carry frames between Switches for purposes of configuring and maintaining the Fabric.

An E_Port shall support the Class F service. An E_Port shall also be capable of routing one or more of the following classes of service: Class 2 service, Class 3 service. An E_Port shall not admit to the Fabric any Primitive Sequences, or any Primitive Signals other than Idle, that the E_Port receives on its inbound fibre.

The model of an E_Port on an FC-FS-3 Transport is shown in figure 8.

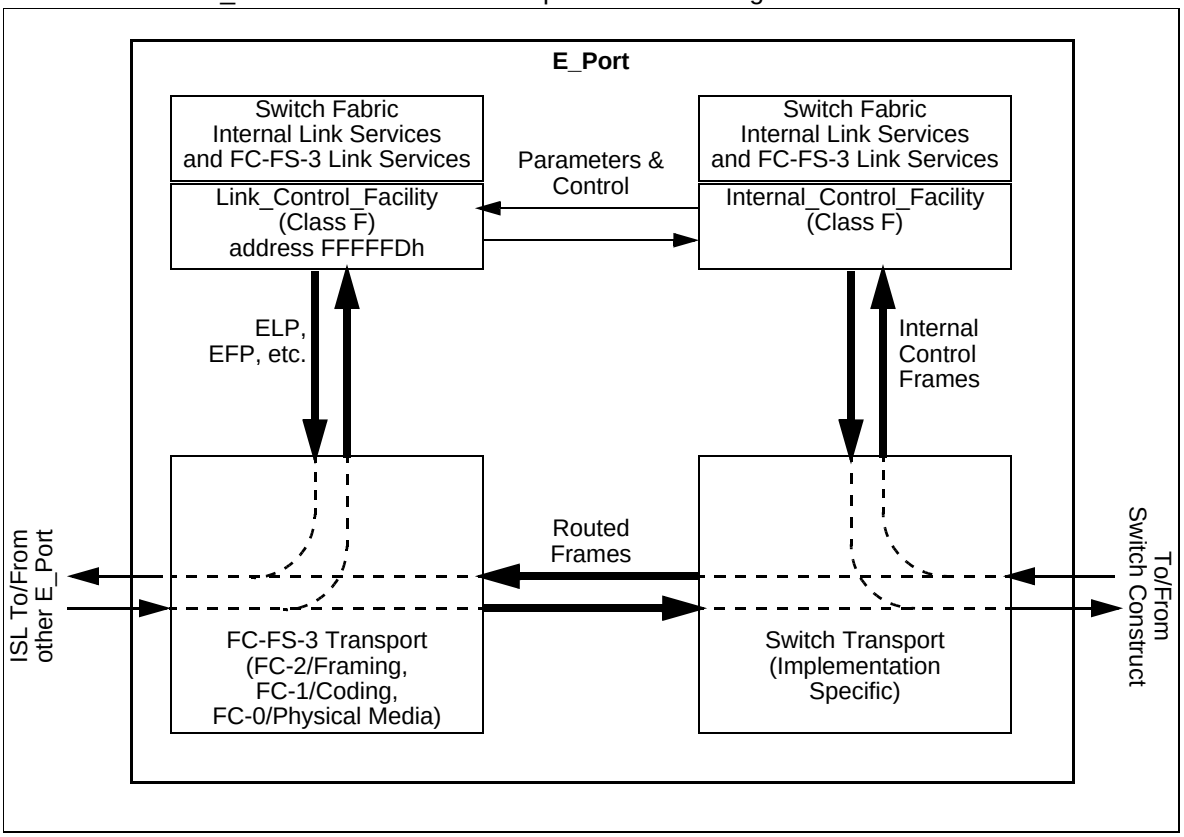

Figure 8 – E_Port Model

An E_Port contains an FC-FS-3 Transport element through which all frames are passed, and Primitives are transferred across the Link to and from the other E_Port. Frames received from the other E_Port are either directed to the Switch Construct via the Switch Transport element, or directed to the Link_Control_Facility. The Link_Control_Facility receives frames related to Switch Fabric Internal Link Services such as ELP, and transmits responses to those Link Service frames.

Frames received from the FC-FS-3 Transport element that are destined for other ports are directed by the Switch Transport to the Switch Construct for further routing. Frames received from the Switch Construct by the Switch Transport are directed either to the FC-FS-3 Transport for transmission to the other E_Port, or to the Internal_Control_Facility. The Internal_Control_Facility receives frames related to Switch Fabric Internal Link Services, and transmits responses to those Internal Link Services frames.

Information is passed between the Internal_Control_Facility and the Link_Control_Facility to effect the control and configuration of the Transport elements.

See FC-BB-5 for additional Fibre Channel transports.

5.6 B_Port Operation

A Bridge Port (B_Port) is a port that is used to connect a Switch to Bridge device. It normally functions as a conduit between the Switch and the Bridge for frames destined for or through a Bridge device. A B_Port is also used to carry frames between a Switch and the Bridge device for purposes of configuring the Bridge device.

A B_Port shall support Class F service. A B_Port shall also be capable of forwarding one or more of the following classes of service: Class 2 service, Class 3 service. A B_Port shall not admit to the Fabric any Primitive Sequences, or any Primitive Signals other than Idle, that the B_Port receives on its inbound fibre.

The model of a B_Port is shown in figure 9.

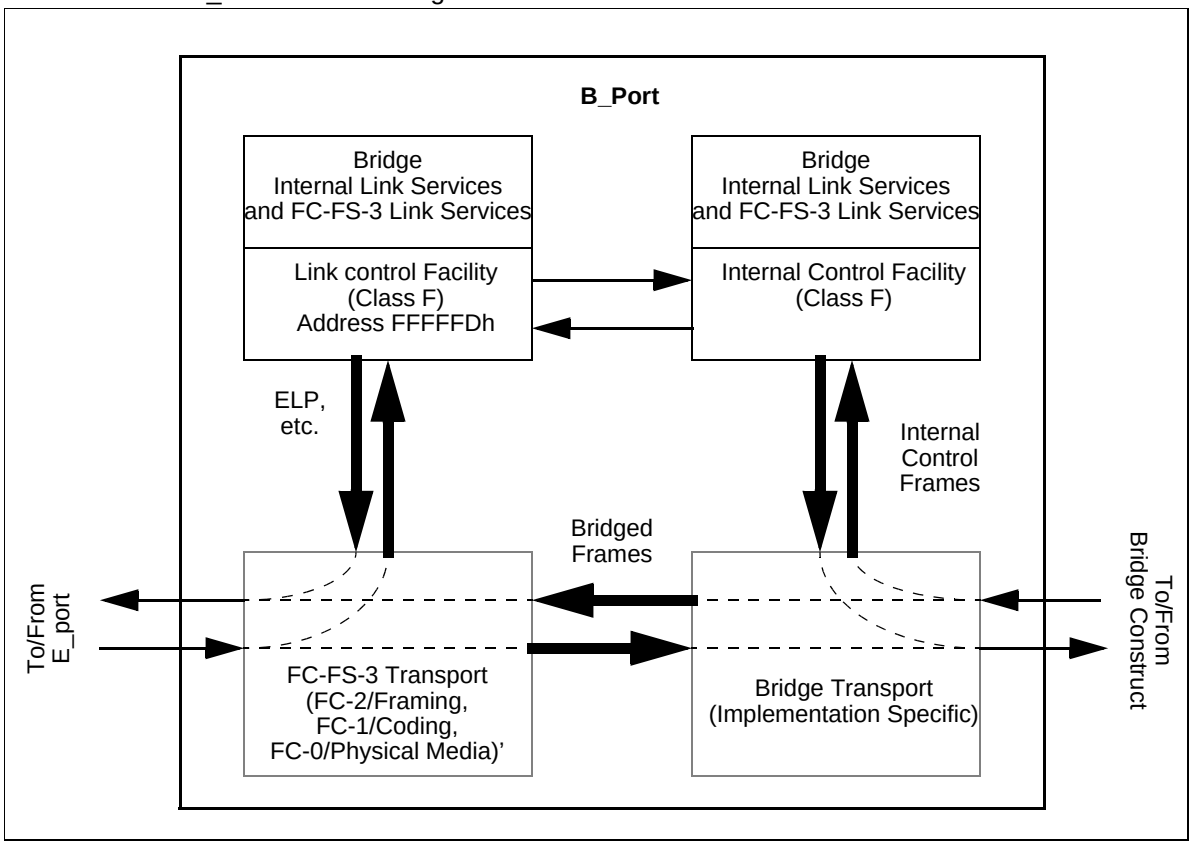

Figure 9 – B_Port Model

A B_Port contains an FC-FS-3 Transport element through which pass all frames and Primitives transferred across the Link to and from the E_Port. Frames received from the attached E_Port are either directed to the Bridge Construct via the Bridge Transport element, or directed to the Link_Control_Facility. The Link_Control_Facility receives frames related to Bridge Fabric Internal Link Services such as ELP, and transmits responses to those Link Service frames.

Frames received from the FC-FS-3 Transport element that are destined for other ports are directed by the Bridge Transport to the Bridge Construct for further forwarding. Frames received from the Bridge Construct by the Bridge Transport are directed either to the FC-FS-3 Transport for transmission to the

other E_Port, or to the Internal_Control_Facility. The Internal_Control_Facility receives frames related to Bridge Fabric Internal Link Services, and transmits responses to those Internal Link Services frames. Information is passed between the Internal_Control_Facility and the Link_Control_Facility to effect the control and configuration of the Transport elements.

The Bridge Port utilizes Class F service as a connectionless service exchanging frames between E_ports and B_Ports. The definition of Class F function and Class F rules apply to both E_ports and B_Ports as defined in clause 5.8.

5.7 Inter-Switch Link Behavior

Inter-Switch Links (ISLs) are used by Switches to transmit and receive frames with other Switches or Bridge devices. An ISL always connects exactly one E_Port on a Switch to exactly one E_Port on another Switch or exactly one B_Port on a Bridge device.

An ISL on an FC-FS-3 Transport follows the FC-0, FC-1, and FC-2 protocols defined for point-to-point Links as defined in FCFS-3, with the exception that Class F frames are allowed to transit the Link. R_RDY shall be used for the management of buffer-to-buffer flow control of Class F frames on the ISL prior to the completion of the exchange of Link parameters (see 6.1.4 and 7.2); an alternate method of buffer-to-buffer flow control may be defined in that process.

For purposes of defining and maintaining the Fabric Configuration, an ISL may be designated as a Principal ISL. The Principal ISL is a path that is used during configuration and address assignment to route Class F configuration frames, and is therefore a known path between two Switches. If a Principal ISL is lost, there may be no other available paths between the two affected Switches, so as a result the Fabric Configuration is possibly broken and shall be rebuilt (by issuing the BF SW_ILS, see 6.1.11). If a non-Principal ISL is lost, at least one other path is known to be available between the Switches (i.e., the Principal ISL), therefore the lost ISL may be resolved via a routing change.

A Switch discovers the Principal ISL(s) during the process of Principal Switch Selection (see 7.3) and Address Distribution (see 7.4). During this process, the Switch identifies two kinds of Principal ISLs. The Principal ISL that leads towards the Principal Switch is called the upstream Principal ISL. All frames from the Switch to the Principal Switch are sent via the upstream Principal ISL. The Principal Switch has no upstream Principal ISL; all other Switches have exactly one upstream Principal ISL.

A Principal ISL that leads away from the Principal Switch is called the downstream Principal ISL. Any frame sent by the Switch to another Switch as a result of a frame received on the upstream Principal ISL is sent via the downstream Principal ISL that leads towards that Switch. The Principal Switch may have one or more downstream Principal ISLs; all other Switches may have zero or more downstream Principal ISLs.

Principal ISLs are further illustrated in figure 10.

See FC-BB-5 for additional Fibre Channel transports.

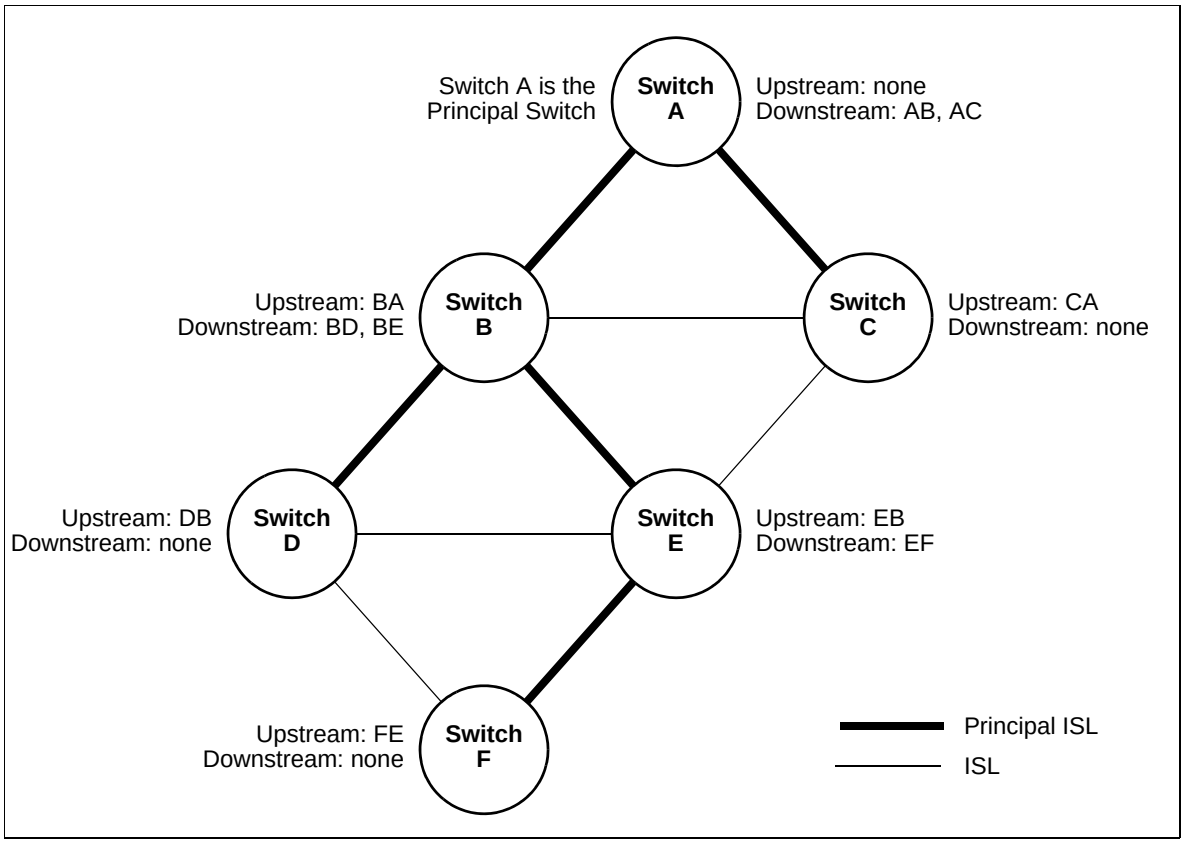

Figure 10 – Principal Inter-Switch Links

5.8 Class F Service

5.8.1 Class F Function

Class F Service is a connectionless service with notification of non-delivery between Interconnect_Ports. Class F service is used for control, coordination, and configuration of the Fabric. Class F Service is defined by this Standard for use by Switches communicating across Inter-Switch Links.

A Class F Service is requested by an Interconnect_Port on a frame by frame basis. The Fabric routes the frame to the destination Interconnect_Port. If an Interconnect_Port transmits consecutive frames to multiple destinations, the Fabric demultiplexes them to the requested destinations. Class F delimiters are used to indicate the requested service and to initiate and terminate one or more Sequences as described in FC-FS-3. Class F shall follow the rules for Class 2 except where otherwise stated in this standard.

5.8.2 Class F Rules

To provide Class F Service, the transmitting and receiving Interconnect_Ports and the Fabric shall obey the following rules:

a) Except for some Switch Fabric Internal Link Service protocols, an Interconnect_Port is required to have exchanged Link parameters (see 6.1.4 and 7.2) with the associated destination with which it intends to communicate (Login).

b) The Fabric routes the frames without establishing a Dedicated Connection between communicating Interconnect_Ports. To obtain Class F service, the Interconnect_Port shall use Class F delimiters as defined in 5.8.3. (Connectionless service)

c) An Interconnect_Port is allowed to send consecutive frames to one or more destinations. This enables an Interconnect_Port to demultiplex multiple Sequences to a single or multiple destinations concurrently (Demultiplexing).

d) A given Interconnect_Port may receive consecutive frames from different sources. Each source is allowed to send consecutive frames for one or more Sequences. (multiplexing)

e) An Interconnect_Port addressed by a Class F frame shall provide an acknowledgment to the source for each valid Data frame received. The destination Interconnect_Port shall use ACK_1 for the acknowledgment. If a Switch is unable to deliver the ACK_1 frame, the Switch shall return an or F_RJT. (Acknowledgment)

f) The Sequence Initiator shall increment the SEQ_CNT field of each successive frame transmitted within a Sequence. However, the Switches may not guarantee delivery to the destination in the same order of transmission. (Non-sequential delivery)

g) Since the SOFf delimiter does not indicate whether a frame is the first frame of a Sequence, the starting SEQ_CNT of every Sequence shall be zero. (Sequence reassembly)

h) An Interconnect_Port may originate multiple Exchanges and initiate multiple Sequences with one or more Interconnect_Ports. The Interconnect_Port originating an Exchange shall assign an X_ID unique to the Originator called OX_ ID and the Responder of the Exchange shall assign an X_ID unique to the responder called RX_ID. The value of OX_ ID or RX_ID is unique to a given Interconnect_Port. The Sequence Initiator shall assign a SEQ_ID, for each Sequence it initiates, that is unique to the Sequence Initiator and the respective Sequence Recipient pair while the Sequence is Open. (Concurrent Exchanges and Sequences)

i) Each Interconnect_Port exercises buffer-to-buffer flow control with the Interconnect_Port to which it is directly attached. End-to-end flow control is performed by communicating Interconnect_Ports. ACK_1 frames are used to perform end-to-end flow control and R_RDY is used for buffer-to-buffer flow control. However, some other agreed upon methods outside the scope of this standard may be used for buffer-to-buffer flow control (e.g., see FC-BB-5). (Dual flow control)

j) If a Switch is unable to deliver the frame to the destination Interconnect_Port, then the source is notified of each frame not delivered by an F_BSY or F_RJT frame with corresponding D_ ID, S_ID, OX_ID, RX_ID, SEQ_ID, and SEQ_CNT from the Switch. The source is also notified of valid frames busied or rejected by the destination Interconnect_Port by P_BSY or P_RJT. (Non-delivery)

k) A busy or reject may be issued by an intermediate Interconnect_Port or the destination Interconnect_Port with a valid reason code. (Busy/reject)

l) If a Class F Data frame is busied, the sender shall retransmit the busied frame up to the ability of the sender to retry, including zero. (Retransmit)

m) The Credit established during the ELP protocol by interchanging Link Parameters shall be honored. Class F may share Credit with other classes of service. (Credit)

n) Effective transfer rate between any given Interconnect_Port pair is dependent upon the number of Interconnect_Ports a given Interconnect_Port is multiplexing and demultiplexing. (Bandwidth)

o) Frames within a Sequence are tracked on a Sequence_Qualifier and SEQ_CNT basis. (Tracking)

p) An Interconnect_Port shall be able to recognize SOF delimiters for Class N service, whether or not all classes of service are supported by the Port. (Invalid Class)

q) An Interconnect_Port addressed by a Vendor Specific Class F frame, shall send an LS_RJT if it does not understand the frame. A Vendor Specific Class F frame is indicated by an R_CTL field value of F0h. (Vendor specific)

r) An Interconnect_Port shall use R_RDY and FC-FS-3 buffer-to-buffer flow control with the Interconnect_Port to which it is directly attached, until after the exchange of Link parameters (see 6.1.4 and 7.2). The BB_Credit prior to the exchange of Link parameters shall be 1. An Interconnnect Port may agree to use an alternate buffer-to-buffer credit model for Class F following the successful exchange of Link parameters. See FC-BB-5 for additional Fibre Channel transports. (Alternate credit models)

s) A Class F frame shall be forwarded to its destination without checking by an intermediate entity. A Class F frame not destined for the receiving E_Port (or E_Port's domain) shall always be forwarded regardless of whether or not the receiving E_Port recognizes the frame. (Frame forwarding)

5.8.3 Class F Frame Format

Class F frames shall use the Frame Content format defined in FC-FS-3. The Start_of_Frame Fabric (SOFf) delimiter shall precede the frame content of all Class F frames. The Data Field size of all Class F frames shall be less than or equal to 256 bytes prior to the successful completion of Exchange Link Parameters (see 6.1.4; Exchange Link Parameters establishes the maximum receive frame size for Class F frames). All Class F frames shall include the CRC defined in FC-FS-3. The End_of_Frame Normal (EOFn) delimiter shall immediately follow the CRC of all normally completed Class F Data frames and all normally completed Class F Link_Control frames except the last frame of a Sequence. The End_of_Frame Terminate (EOFt) delimiter shall immediately follow the CRC of all Class F Link_Control frames that indicate the last frame of a normally terminated Sequence.

An Interconnect_Port or Switch may invalidate or discard without notification any incorrectly formed Class F frame, or any Class F frame with a code violation or CRC error.

5.8.4 Class F Flow Control

Class F service uses end-to-end flow control in all Transports. ACK_1 frames are used to perform end-to-end flow control. ACK_1 frames shall be formatted as described in 5.8.3. The ACK_0 Link Control frame shall not be used for Class F service.

In the FC-FS-3 Transport, R_RDY is used for buffer-to-buffer flow control. R_RDY is transmitted by the Interconnect_Port at one end of the ISL, to the Interconnect_Port at the other end of the ISL, to indicate that a buffer is available for further frame reception by the first Interconnect_Port. This process operates in both directions on the ISL. After the successful exchange of Link Parameters, an alternate method of buffer-to-buffer flow control may be established on an ISL (see 7.2). This alternate method of buffer-to-buffer flow control remains in effect until a Link Offline or Link Failure occurs, or a new set of Link Parameters is successfully exchanged between the Interconnect_Ports.

See FC-BB-5 for additional Fibre Channel transports.

6 Internal Link Services

6.1 Switch Fabric Internal Link Services (SW_ILS)

6.1.1 SW_ILS Overview

This subclause describes Link Services that operate internal to the Fabric between Switches. In the case of Exchange Link Parameters (ELP), Link Services also operate internal to the Fabric between Switches and Bridge devices. All SW_ILS frames shall be transmitted using the Class F service. The following defines the header fields of all SW_ILS frames:

R_CTL: This field shall be set to 02h for all request frames, and to 03h for all reply frames.

CS_CTL: This field shall be set to 00h.

D_ID and S_ID: Set as indicated for the specific SW_ILS.

TYPE: This field shall be set to 22h, indicating Fibre Channel Fabric Switch Services.

All other fields shall be set as appropriate according to the rules defined in FC-FS-3. The first word in the payload specifies the Command Code. The Command Codes are summarized in table 2**.**

Table 2 – SW_ILS Command Codes (Part 1 of 3)

Encoded Value (hex)	Description	Abbr.
01000000	Switch Fabric Internal Link Service Reject	SW_RJT
02xxxxxx	Switch Fabric Internal Link Service Accept	SW_ACC
10000000	Exchange Link Parameters	ELP
11xxxxxx	Exchange Fabric Parameters	EFP
12000000	Domain Identifier Assigned	DIA
1300xxxx	Request Domain_ID	RDI
14000000	Hello	HLO
15000000	Link State Update	LSU
16000000	Link State Acknowledgement	LSA
17000000	Build Fabric	BF
18000000	Reconfigure Fabric	RCF
1B000000	Inter-Switch Registered State Change Notification	SW_RSCN
1E000000	Distribute Registered Link Incident Records	DRLIR
[a] ASF and EBP are currently defined in FC-BB-3.		

Table 2 – SW_ILS Command Codes (Part 2 of 3)

Encoded Value (hex)	Description	Abbr.
20000000	Obsoleted in FC-SW-5	DSCN
21000000	Obsoleted in FC-SW-3	LOOPD
22xxxxxx	Merge Request	MR
2300xxxx	Acquire Change Authorization	ACA
24000000	Release Change Authorization	RCA
25xxxxxx	Stage Fabric Configuration	SFC
26000000	Update Fabric Configuration	UFC
28xxxxxx	Reserved for FC-BB-3 Use[a]	
29000000	Check E_Port Connectivity	CEC
2A010000	Enhanced Acquire Change Authorization	EACA
2A020000	Enhanced Stage Fabric Configuration	ESFC
2A030000	Enhanced Update Fabric Configuration	EUFC
2A040000	Enhanced Release Change Authorization	EACA
2A050000	Transfer Commit Ownership	TCO
3000xxxx	Exchange Switch Capabilities	ESC
31000000	Exchange Switch Support	ESS
32000000	Reserved for Legacy Implementations (see FC-SP)	
33000000	Reserved for Legacy Implementations (see FC-SP)	
34000000	Merge Request Resource Allocation	MRRA
35010000	Switch Trace Route	STR
36000000	Exchange Virtual Fabrics Parameters	EVFP
40xxxxxx	Reserved for FC-SP Use	
41xxxxxx	Reserved for FC-SP Use	
42xxxxxx	Reserved for FC-SP Use	

[a] ASF and EBP are currently defined in FC-BB-3.

Table 2 – SW_ILS Command Codes (Part 3 of 3)

Encoded Value (hex)	Description	Abbr.
50000000	Fast Fabric Initialization for the Avionics Environment (See annex D)	FFI
70000000 to 7FFFFFFF	Vendor Specific	
90000000 to 9FFFFFFF	Vendor Specific	
others	Reserved	
[a] ASF and EBP are currently defined in FC-BB-3.		

Unless otherwise specified, the rules regarding the following aspects of Switch Fabric Internal Link Services are as defined for the Extended Link Services in FC-FS-3 (e.g., Sequence and Exchange Management, error detection and recovery). Time-out values for specific SW_ILS's and the actions following a time-out expiration are specified in 16.2.

6.1.2 Switch Fabric Internal Link Service Accept (SW_ACC)

The Switch Fabric Internal Link Service Accept reply Sequence shall notify the transmitter of an SW_ILS request that the SW_ILS request Sequence has been completed. The first word of the Payload shall contain 02 xx xx xxh. The remainder of the Payload is unique to the specific SW_ILS request.

Protocol: SW_ACC may be sent as a reply Sequence to an SW_ILS request. An SW_ACC shall not be sent for HLO, LSU, and LSA Request Sequences.

Addressing: The S_ID field shall be set to the value of the D_ID field in the SW_ILS request. The D_ID field shall be set to the value of the S_ID field in the SW_ILS request.

Payload: The Payload content following the first word is defined within individual SW_ILS requests.

6.1.3 Switch Fabric Internal Link Service Reject (SW_RJT)

The Switch Fabric Internal Link Service Reject shall notify the transmitter of an SW_ILS request that the SW_ILS request Sequence has been rejected. A four-byte reason code shall be contained in the Data_Field. SW_RJT may be transmitted for a variety of conditions that may be unique to a specific SW_ILS request.

Protocol: SW_RJT may be sent as a reply Sequence to an SW_ILS request. An SW_RJT shall not be sent for HLO, LSU, and LSA Request Sequences.

Addressing: The S_ID field shall be set to the value of the D_ID field in the SW_ILS request. The D_ID field shall be set to the value of the S_ID field in the SW_ILS request.

Payload: The format of the SW_RJT reply Payload is shown in table 3.

Table 3 – SW_RJT Payload

Item	Size Bytes
01 00 00 00h	4
Reserved	1
Reason Code	1
Reason Code Explanation	1
Vendor Specific	1

Reason Code: The Reason Codes are summarized in table 4.

Table 4 – SW_RJT Reason Codes

Encoded Value (hex)	Description
01	Invalid SW_ILS command code
02	Invalid revision level
03	Logical error
04	Invalid payload size
05	Logical busy
07	Protocol error
09	Unable to perform command request
0B	Command not supported
0C	Invalid Attachment
FF	Vendor Specific error
others	Reserved

Invalid SW_ILS command code: The Command Code is not recognized by the recipient.

Invalid revision level: The recipient does not support the specified revision level.

Logical error: The request identified by the Command Code and the Payload content is invalid or logically inconsistent for the conditions present.

Invalid payload size: The size of the Payload is inconsistent with the Command Code and/or any Length fields in the Payload.

Logical busy: The recipient is busy and is unable to process the request at this time.

Protocol error: An error has been detected that violates the protocol.

Unable to perform command request: The recipient is unable to perform the request.

Command not supported: The command code is not supported by the recipient.

Invalid Attachment: The recipient is in the Invalid Attachment state.

Vendor Specific Error: The Vendor Specific field indicates the error condition.

Reason Code Explanation: The Reason Code Explanation is summarized in table 5**.**

Table 5 – SW_RJT Reason Code Explanation (Part 1 of 3)

Encoded Value (hex)	Description
00	No additional explanation
01	Class F Service Parameter error
03	Class N Service Parameter error
04	Unknown Flow Control code
05	Invalid Flow Control Parameters
0D	Invalid Port_Name
0E	Invalid Switch_Name
0F	R_A_TOV or E_D_TOV mismatch
10	Invalid Domain_ID_List
19	Command already in progress
29	Insufficient resources available
2A	Domain_ID not available
2B	Invalid Domain_ID
2C	Request not supported
2D	Link Parameters not yet established

[a] The range of values 30h-3Fh are used to indicate Security reason code explanations.

[b] The range of values 40h-4Fh are used to indicate Zoning reason code explanations.

Table 5 – SW_RJT Reason Code Explanation (Part 2 of 3)

Encoded Value (hex)	Description
2E	Requested Domain_IDs not available
2F	E_Port is Isolated
31	Authorization Failed[a]
32	Authentication Failed
33	Incompatible Security Attribute
34	Security Checks in Progress
35	Policy Summary Not Equal
36	FC-SP Zoning Summary Not Equal
41	Invalid Data Length[b]
42	Unsupported Command
44	Not Authorized
45	Invalid Request
46	Fabric Changing
47	Update Not Staged
48	Invalid Zone Set Format
49	Invalid Data
4A	Unable to Merge
4B	Zone Set Size Not Supported
50	Unable to Verify Connection
58	Requested Application Not Supported

[a] The range of values 30h-3Fh are used to indicate Security reason code explanations.

[b] The range of values 40h-4Fh are used to indicate Zoning reason code explanations.

Table 5 – SW_RJT Reason Code Explanation (Part 3 of 3)

Encoded Value (hex)	Description
59	Transaction Specified by Request Does Not Exist
5A	Invalid Phase Transition in Transaction
5B	In Advanced Phase
5C	Switch Not Authorized
others	Reserved

[a] The range of values 30h-3Fh are used to indicate Security reason code explanations.

[b] The range of values 40h-4Fh are used to indicate Zoning reason code explanations.

Vendor Specific: This field is valid when the Reason Code indicates a Vendor Specific error, otherwise this field is reserved.

6.1.4 Exchange Link Parameters (ELP)

6.1.4.1 ELP Request

The Exchange Link Parameters Switch Fabric Internal Link Service requests the exchange of Link Parameters between two Interconnect_Ports connected via an ISL. The exchange of Link Parameters establishes the operating environment between the two Interconnect_Ports, and the capabilities of the Switches or Bridge devices that are connected by the Interconnect_Ports. When an ELP is received by an Interconnect_Port, any Active or Open Class F Sequences between the two Interconnect_Ports, and any Dedicated Connections, shall be abnormally terminated (prior to transmission of the SW_ACC reply Sequence.

Use of the ELP SW_ILS for Switch Port initialization is described in 7.2.

Protocol:

Exchange Link Parameters (ELP) Request Sequence
Accept (SW_ACC) Reply Sequence

Error Detection and Recovery: See table 207.

Addressing: For use in Switch Port initialization, the S_ID field shall be set to FFFFFDh, indicating the Fabric Controller of the originating Switch; the D_ID field shall be set to FFFFFDh, indicating the Fabric Controller of the destination Switch.

Payload: The format of the ELP request Payload is shown in table 6.

Table 6 – ELP Request Payload

Item	Size Bytes
10000000h	4
Revision	1
Flags	2
BB_SC_N	1
R_A_TOV	4
E_D_TOV	4
Requester Interconnect_Port_Name	8
Requester Switch_Name	8
Fabric Controller Class F Service Parameters	16
Obsolete in FC-SW-5	4
Class 2 Interconnect_Port Parameters	4
Class 3 Interconnect_Port Parameters	4
Reserved	20
ISL Flow Control Mode	2
Flow Control Parameter Length (N)	2
Flow Control Parameters	N

Revision: This field denotes the revision of the ELP payload format and the Switch Port Initialization protocol.

The Revision field value is increased only if there is a substantial change in the ELP payload format, or if there are procedural changes in the Switch Port Initialization protocol that are not compatible with the previous version. Examples of substantial changes in the ELP payload format are:

a) changes in the lengths of existing ELP payload fields; or

b) the addition of new fields in the ELP payload. The use of bits that were previously reserved does not constitute a substantial change to the ELP payload.

For this standard, the Revision value shall be 03h.

If a Fabric Controller receives an ELP Request containing a Revision field value that is higher than its supported value, the Fabric Controller shall respond with its highest supported Revision field value. The Fabric Controller is responsible for detecting and handling any incompatibility issues that may occur. If a Fabric Controller receives an ELP Request containing a Revision field value that is equal to or lower than its supported value, the Fabric Controller shall respond with the Revision field value received in the ELP Request.

Flags: This field contains flag bits that provide additional information about the ELP. The following flag bits are defined.

Bit 15, the Bridge Port bit, shall indicate whether the sending port is a B_Port. If bit 15 is zero, the sending port is an E_Port and not a B_Port. If bit 15 is one, the sending port is a B_port.

Bit 14, the Bridge Virtual Fabrics bit, is meaningful only for a B_Port and shall indicate whether the sending B_Port supports Virtual Fabric Tagging. If bit 14 is zero, the sending B_Port does not support the passing through of VFT tagged frames (see FC-FS-3). If bit 14 is one, the sending B_port supports the passing through of VFT tagged frames.

Bit 13, is the Controlling FCF/Switch bit. This bit set to one indicates that the originator of the ELP Request or SW_ACC is a Controlling Switch. This bit set to zero indicates that the originator of the ELP Request or SW_ACC is not a Controlling Switch.

Bit 12, is the FDF/FCDF bit. This bit set to one indicates that the originator of the ELP Request or SW_ACC is an FCDF. This bit set to zero indicates that the originator of the ELP Request or SW_ACC is not an FCDF.

Bits 11-0 shall be reserved.

BB_SC_N: This field indicates the Buffer-to-Buffer State Change number. The BB_SC_N field is valid only if the R_RDY_Flow Control mode is specified in the ISL Flow Control Mode field. A value between 0 and 15 indicates that the sender of the ELP frame is requesting a $2^{BB\_SC\_N}$ number of frames be sent between two consecutive BB_SCs Primitive Signals, and a $2^{BB\_SC\_N}$ number of R_RDY Primitive Signals be sent between two consecutive BB_SCr Primitive Signals. When the two ports exchanging link parameters specify different non-zero values of BB_SC_N, the larger value shall be used. If either port specifies a BB_SC_N value of zero, then the BB_Credit recovery process shall not be performed and no BB_SCx Primitive Signals shall be sent. If a port specifies a non-zero BB_SC_N value it shall support the BB_SCs and BB_SCr Primitive Signals. See FC-FS-3 for a description of the BB_Credit recovery process. The following BB_SC_N bits are defined:

Bits 7-4, Reserved

Bits 3-0, Buffer-to-buffer State Change Number (BB_SN_N).

If all frames or R_RDY Primitive Signals sent between two BB_SCx Primitive Signals are lost, then $2^{BB\_SC\_N}$ number of BB_Credits are lost, and are unable to be recovered by the scheme outlined in FC-FS-3. Therefore BB_SC_N should be chosen so that the probability of losing $2^{BB\_SC\_N}$ number of consecutive frames or R_RDY Primitive Signals is deemed negligible. Therefore the recommended value of BB_SC_N is 8.

R_A_TOV: This field shall be set to the value of R_A_TOV required by the Switch.

E_D_TOV: This field shall be set to the value of E_D_TOV required by the Switch.

NOTE 5 – The values of R_A_TOV and E_D_TOV may be established by a Profile(s) or other means.

Interconnect_Port_Name: The Interconnect_Port_Name is an eight-byte field that identifies an Interconnect_Port. The format of the name is specified in FC-FS-3. Each Interconnect_Port shall provide a unique Interconnect_Port_Name within the Fabric.

Switch_Name: The Switch_Name is an eight-byte field that identifies a Switch or Bridge device. The format of the name is specified in FC-FS-3. Each Switch_Name shall be unique within the Fabric.

Fabric Controller Class F Service Parameters: This field contains the E_Port Class F Service Parameters. The format of the Parameters is shown in table 7.

Table 7 – Fabric Controller Class F Service Parameters

Word	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	VAL	Reserved															Reserved															
1	R	XII	Reserved														Receive Data Field Size															
2	Concurrent Sequences																End-to-End Credit															
3	Open Sequences per Exchange																Reserved															

The Fabric Controller Class F Service Parameters are defined as follows:

a) VAL (Class Valid): This bit shall be set to one.

b) XII (X_ID Interlock): This bit when one indicates that the Fabric Controller supplying this parameter requires that an interlock be used during X_ID assignment in Class F. In X_ID assignment, the Sequence Initiator shall set the Recipient X_ID value to FFFFh in the first Data frame of a Sequence, and the Recipient shall supply 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.

c) Receive Data Field Size: This field shall specify the largest Data Field size in bytes for a frame that may be received by the Fabric Controller supplying the Parameters as a Sequence Recipient for a Class F frame. Values less than 256 or greater than 2112 are invalid. Values shall be a multiple of four bytes.

d) Concurrent Sequences: This field shall specify the number of Sequence Status Blocks provided by the Fabric Controller supplying the Parameters for tracking the progress of a Sequence as a Sequence Recipient. The maximum number of Concurrent Sequences that may be specified is 255. A value of zero in this field is reserved. In Class F, the value of SEQ_ID shall range from 0 to 255, independent of the value in this field. A Fabric Controller is allowed to respond with P_BSY to a frame initiating a new Sequence if Interconnect_Port resources are not available.

e) End-to-End Credit: End-to-end credit is the maximum number of Class F Data frames that may be transmitted by a Fabric Controller without receipt of accompanying ACK or Link_Response

frames. The minimum value of end-to-end credit is one. The end-to-end credit field specified is associated with the number of buffers available for holding the Data_Field of a Class F frame and processing the contents of that Data_Field by the Interconnect_Port supplying the Parameters. Bit 15 of this field shall be set to zero. A value of zero for this field is reserved.

f) Open Sequences per Exchange: The value of the Open Sequences per Exchange shall specify the maximum number of Sequences that may be Open at one time at the Recipient between a pair of Fabric Controllers for one Exchange. This value plus two shall specify the number of instances of Sequence Status that shall be maintained by the Recipient for a single Exchange in the Exchange Status Block. This value is used for Exchange and Sequence tracking. The value in this field limits the link facility resources required for error detection and recovery.

Interconnect_Port Parameters indicate that the Interconnect_Port is capable of transporting the indicated Class of Service, and the conditions under which it may transport the Class. One word of the ELP Payload is allocated for each Class.

Class 2 Interconnect_Port Parameters: This field contains the Class 2 Interconnect_Port Parameters. The format of the Parameters is shown in table 8.

Table 8 – Class 2 Interconnect_Port Parameters

Word	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	VAL	R R R			SEQ	Reserved											Receive Data Field Size																

The Class 2 Interconnect_Port Parameters are defined as follows:

a) VAL (Class Valid): This bit shall be set to one if the Interconnect_Port supports Class 2. If this bit is zero, all other Class 2 Interconnect_Port Parameters shall be invalid.

b) SEQ (Sequential Delivery): If this bit is set to one by an Interconnect_Port, it is indicating that the Switch is able to guarantee sequential delivery (as defined in FC-FS-3) of Class 2 frames. Sequential Delivery shall be functional only if both Interconnect_Ports indicate support for this feature.

c) Receive Data Field Size: This field shall specify the largest Data Field size in bytes for a frame that may be received by the Interconnect_Port supplying the Parameters for a Class 2 frame. Values less than 256 or greater than 2112 are invalid. Values shall be a multiple of four bytes.

Class 3 Interconnect_Port Parameters: This field contains the Class 3 Interconnect_Port Parameters. The format of the Parameters is shown in table 9.

Table 9 – Class 3 Interconnect_Port Parameters

Word	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	VAL	R R R			SEQ	Reserved											Receive Data Field Size																

The Class 3 Interconnect_Port Parameters are defined as follows:

a) VAL (Class Valid): This bit shall be set to one if the Interconnect_Port supports Class 3. If this bit is zero, all other Class 3 Interconnect_Port Parameters shall be invalid.

b) SEQ (Sequential Delivery): If this bit is set to one by an Interconnect_Port, it is indicating that the Switch is able to guarantee sequential delivery (as defined in FC-FS-3) of Class 3 frames. Sequential Delivery shall be functional only if both Interconnect_Ports indicate support for this feature.

c) Receive Data Field Size: This field shall specify the largest Data Field size in bytes for a frame that may be received by the Interconnect_Port supplying the Parameters for a Class 3 frame. Values less than 256 or greater than 2112 are invalid. Values shall be a multiple of four bytes.

ISL Flow Control Mode: This field contains a code that specifies the Flow Control method supported by the Interconnect_Port. Table 10 shows the allowed values for this field.

Table 10 – ISL Flow Control Mode Values

Value (hex)	Usage
0001	Vendor Specific
0002	R_RDY Flow Control
0003 - 1FFF	Vendor Specific
2000	VC_RDY Flow Control
AE02	Reserved for AE Use
Other Values	Reserved

Flow Control Parameter Length: This field specifies the length in bytes of the Flow Control Parameters that follow. Values shall be a multiple of four. A value of zero indicates no parameters follow.

Flow Control Parameters: These parameters contain information used to configure Flow Control for the ISL. Flow control parameters are specific to a given flow control mode.

6.1.4.2 R_RDY Flow Control

A value of 0002h in the ISL Flow Control Mode field indicates that R_RDY Flow Control (as defined in FC-FS-3) shall be used. When R_RDY Flow Control mode is used, the Flow Control Parameter Length field shall be set to 20h and the Flow Control Parameters field shall contain the fields as shown in table 11. Values other than 0002h for ISL Flow Control Mode are not required to follow the Flow Control Parameter Field format shown in table 11.

Table 11 – Flow Control Parameters

Item	Size
BB_Credit	4
Compatibility Parameters	16

Buffer-to-buffer Credit: The BB_Credit field specified shall be associated with the number of buffers available for holding Class 2, Class 3 or Class F frames received from the Interconnect_Port. The Buffer-to-buffer Credit shall be a single value that represents the total buffer-to-buffer Credit available for all Class 2 frames, and all Class 3 frames. The buffer-to-buffer credit value may also be applied to class F frames.

Compatibility Parameters: This field contains associated compatibility parameters to assist in assuring backward compatibility with existing implementations.

NOTE 6 – Recommended Compatibility Parameter values for the R_RDY Flow Control mode are described in FC-MI (reference [2]).

6.1.4.3 VC_RDY Flow Control

A value of 2000h in the ISL Flow Control Mode field indicates that VC_RDY Flow Control shall be used. When VC_RDY Flow Control mode is used, the Flow Control Parameter Length field shall be based on the number of VCs supported and the Flow Control Parameters field shall contain fields as shown in table 12.

Table 12 – VC_RDY Flow Control Parameters

Item	Size
BB_Credit	4
Assignment Scheme	2
VC Value	2
VC Credit 0	4
...	
VC Credit n-1	4

If VC_RDY flow control mode is specified, and the recipient of the ELP does not support the VC_RDY Flow Control mode, then the ELP is rejected with a reason code explanation of ”Invalid Flow Control Code”.

Buffer-to-buffer Credit: The BB_Credit field specified shall be associated with the number of buffers available for holding Class 2, Class 3 or Class F frames received from the Interconnect_Port independent of VC_Credit. The Buffer-to-buffer Credit shall be a single value that represents the total buffer-to-buffer Credit available for all Class 2 frames, and all Class 3 frames. The buffer-to-buffer credit value may also be applied to class F frames.

Assignment Scheme: The assignment scheme identifies how frames are assigned to virtual channels. The assignment schemes and their values are provided in table 13:

VC Value: This field contains an integer that maps to the number of VCs defined for a given assignment scheme. Valid VC values for the assignment schemes are shown in the tables below.

Valid VC values for the Simple assignment scheme are shown in table 14.

Table 13 – Assignment Schemes

Value	Scheme
0001h	Simple
0002h	Fixed
0003h	Variable
FF00h-FFFFh	Vendor Specific
other values	Reserved

Table 14 – VC Values - Simple

VC Value	Number of VCs (n)
0001h	2
other values	reserved

Valid VC values for the Fixed assignment scheme are shown in table 15.

Table 15 – VC Values - Fixed

VC Value	Number of VCs (n)
0001h	8
0002h	12
0003h	20
0004h	36
0005h	68
0006h	132
other values	Reserved

Valid VC values for the Variable assignment scheme are shown in table 16.

Table 16 – VC Values - Variable

VC Value	Number of VCs (n)
0002h	4
0003h	8
0004h	16
0005h	32
0006h	64
0007h	128
0008h	256
other values	Reserved

Clause 14 contains additional information regarding Virtual Channels and their associated assignment schemes.

6.1.4.4 ELP Reply

Reply Switch Fabric Internal Link Service Sequence:

Service Reject (SW_RJT)
Signifies the rejection of the ELP request
Accept (SW_ACC)
Signifies acceptance of the ELP request.
– Accept Payload

Payload: The format of the ELP Accept Payload is shown in table 17.

Table 17 – ELP Accept Payload

Item	Size Bytes
02000000h	4
Revision	1
Flags	2
BB_SC_N	1
R_A_TOV	4
E_D_TOV	4
Responder Interconnect_Port_Name	8
Responder Switch_Name	8
Fabric Controller Class F Service Parameters	16
Obsolete in FC-SW-5	4
Class 2 Interconnect_Port Parameters	4
Class 3 Interconnect_Port Parameters	4
Reserved	20
ISL Flow Control Mode	2
Flow Control Parameter Length (N)	2
Flow Control Parameters	N

The fields in table 17 are the same as defined in table 6.

6.1.5 Exchange Fabric Parameters (EFP)

The Exchange Fabric Parameters Switch Fabric Internal Link Service requests the exchange of Fabric Parameters between two E_Ports connected via an ISL. The exchange of Fabric Parameters is used to establish the address allocation within the Fabric. When an E_Port receives EFP from another E_Port, all Active or Open Class F Sequences and Dedicated Connections shall be unaffected.

Use of the EFP SW_ILS for Fabric Configuration is described in 7.3 and 7.4.

Protocol:

Exchange Fabric Parameters (EFP) request Sequence
Accept (SW_ACC) Reply Sequence

Addressing: For use in Fabric Configuration, the S_ID field shall be set to FFFFFDh, indicating the Fabric Controller of the originating Switch. The D_ID field shall be set to FFFFFDh, indicating the Fabric Controller of the destination Switch.

Payload: The format of the EFP request Payload is shown in table 18.

Table 18 – EFP Request Payload

Item	Size Bytes
Command code = 11h	1
Record length = 10h	1
Payload length	2
Reserved	3
Principal Switch_Priority	1
Principal Switch_Name	8
Domain_ID_List	N
Obsolete in FC-SW-5	N

Record Length: This field contains an 8-bit unsigned binary integer that specifies the total length of each record in the Payload (see below). The value shall be 10h.

Payload Length: This field contains a 16-bit unsigned binary integer that specifies the total length in bytes of the Payload. The value specified shall be greater than or equal to 16, and less than or equal to 65532.

Principal Switch_Priority: This field shall specify the priority level of the Switch that the transmitting Switch believes is the Principal Switch. Values for this field are summarized in table 19.

Table 19 – Switch_Priority Field Values

Value (hex)	Description
00	Reserved
01	Highest priority value[a]
02	The Switch was the Principal Switch prior to sending or receiving BF[b]
03 to FE	Higher to lower priority values[c]
FF	The Switch is not capable of acting as a Principal Switch

[a] This value allows the system administrator to establish which Switch becomes the Principal Switch.

[b] This allows the same Switch to become Principal Switch if it is still part of the Fabric after sending and/or receiving the Build Fabric SW_ILS.

[c] The Switch_Priority value for a given Switch is established by means not defined by this Standard.

Principal Switch_Name: This field shall specify the Switch_Name of the Switch that the transmitting Switch believes is the Principal Switch.

Domain_ID_List: This field shall contain a list of records that specify the Domain_ID and corresponding Switch_Name of the Switch that has been granted the Domain_ID by the Principal Switch. The Domain_ID_List shall contain a record for each value of Domain_ID that has been assigned. If no Switch has been assigned a Domain_ID, the Domain_ID_List shall contain no records. The format of a Domain_ID_List record is shown in table 20.

Table 20 – Domain_ID_List Record Format

Item	Size Bytes
Record_Type	1
Domain_ID	1
Reserved	2
Reserved	4
Switch_Name for Domain_ID	8

Record_Type: This field shall specify the type of record. Values for this field are summarized in table 21**.**

Table 21 – Record_Type Field Values

Value (hex)	Description
00	Reserved
01	Domain_ID_List record
02	Obsolete in FC-SW-5
all others	Reserved

Domain_ID: This field shall specify the Domain_ID assigned by the Principal Switch.

Switch_Name for Domain_ID: This field shall specify the Switch_Name of the Switch that has been assigned the Domain_ID by the Principal Switch.

Reply Switch Fabric Internal Link Service Sequence:

Service Reject (SW_RJT)
Signifies the rejection of the EFP request
Accept (SW_ACC)
Signifies acceptance of the EFP request.
– Accept Payload

Payload: The format of the EFP Accept Payload is shown in table 22**.**

Table 22 – EFP Accept Payload

Item	Size Bytes
Command code = 02h	1
Page length = 10h	1
Payload length	2
Reserved	3
Principal Switch_Priority	1
Principal Switch_Name	8
Domain_ID_List	N

The fields in table 22 are the same as defined for table 18 with the following exception. The Domain_ID_List in the EFP Request payload specifies the current Domain_ID_List of the originating Switch. The Domain_ID_List in the EFP Accept payload specifies the Domain_ID_List of the responding Switch prior to the merging of the received Domain_ID_List from the originating Switch with the

Domain_ID_List of the responding Switch. This ensures that both the sending Switch and the responding Switch each have the same Domain_ID_List following the EFP exchange.

6.1.6 Domain Identifier Assigned (DIA)

The Domain Identifier Assigned Switch Fabric Internal Link Service indicates that a Principal Switch has been selected, and that the upstream neighbor Switch has been assigned a Domain Identifier. This communication signals that the Recipient may request a Domain Identifier from the Originating E_Port.

Use of the DIA SW_ILS for Fabric Configuration is described in 7.4.

Protocol:

Domain Identifier Assigned (DIA) request Sequence
Accept (SW_ACC) Reply Sequence

Addressing: For use in Fabric Configuration, the S_ID field shall be set to FFFFFDh, indicating the Fabric Controller of the originating Switch. The D_ID field shall be set to FFFFFDh, indicating the Fabric Controller of the destination Switch.

Payload: The format of the DIA request Payload is shown in table 23**.**

Table 23 – DIA Request Payload

Item	Size Bytes
12000000h	4
Originating Switch_Name	8
Not Meaningful	4

Originating Switch_Name: This field shall contain the Switch_Name of the Switch that originated the DIA request.

Reply Switch Fabric Internal Link Service Sequence:

Service Reject (SW_RJT)
Signifies the rejection of the DIA request
Accept (SW_ACC)
Signifies acceptance of the DIA request.
– Accept Payload

Payload: The format of the DIA Accept Payload is shown in table 24.

Table 24 – DIA Accept Payload

Item	Size Bytes
02000000h	4
Responding Switch_Name	8
Not Meaningful	4

Responding Switch_Name: This field shall contain the Switch_Name of the Switch that responds to the DIA request.

6.1.7 Request Domain_ID (RDI)

The Request Domain_ID Switch Fabric Internal Link Service is sent by a Switch to request a Domain_ID from the Domain Address Manager. RDI shall not be sent by a Switch unless the Switch has received a DIA SW_ILS since the last reconfiguration event.

Use of the RDI SW_ILS for Fabric Configuration is described in 7.4.

Protocol:

Request Domain_ID (RDI) request Sequence
Accept (SW_ACC) reply Sequence

Addressing: For use in Fabric Configuration, the S_ID field shall be set to FFFFFDh, indicating the Fabric Controller of the originating Switch. The D_ID field shall be set to FFFFFDh, indicating the Fabric Controller of the destination Switch.

Payload: The format of the RDI request Payload is shown in table 25.

Table 25 – RDI request payload

Item	Size Bytes
13h	1
Reserved	1
Payload Length	2
Requesting Switch_Name	8
Reserved	3
Requested Domain_ID #1	1
Reserved	3
Requested Domain_ID #2	1
...	
Reserved	3
Requested Domain_ID #n	1

Payload Length: This field contains a 16-bit unsigned binary integer that specifies the total length in bytes of the payload. The value specified shall be greater than or equal to 16, and less than or equal to 964.

Requesting Switch_Name: This field specifies the Switch_Name of the Switch requesting a Domain_ID.

Requested Domain_ID: This field shall contain one or more requested Domain_IDs for the requesting Switch. If there is a Preferred Domain_ID, this field is set to the Preferred Domain_ID, otherwise it is set to zero. If more than one Domain_ID is requested then none of the requested Domain_IDs shall be zero.

Reply Switch Fabric Internal Link Service Sequence:

Service Reject (SW_RJT)
  Signifies the rejection of the RDI request
Accept (SW_ACC)
  Signifies acceptance of the RDI request.
– Accept Payload

Payload: The format of the RDI accept Payload is shown in table 26.

Table 26 – RDI accept payload

Item	Size Bytes
02h	1
Reserved	1
Payload Length	2
Requesting Switch_Name	8
Reserved	3
Granted Domain_ID #1	1
Reserved	3
Granted Domain_ID #2	1
...	
Reserved	3
Granted Domain_ID #n	1

Payload Length: This field contains a 16-bit unsigned binary integer that specifies the total length of the Payload. The least significant two bits shall be zero. The value specified shall be equal to the value specified in the request payload.

Requesting Switch_Name: This field specifies the Switch_Name of the Switch requesting a Domain_ID.

Granted Domain_ID: This field shall contain the Domain_ID granted by the Domain Address Manager to the requesting Switch. The Granted Domain_ID field is set to:

a) the Preferred Domain_ID specified in the request if it is available;

b) another Domain_ID if the Preferred Domain_ID specified in the request is not available; or

c) another Domain_ID if the Requested Domain_ID value specified in the request is zero.

If no Domain_ID is available then an SW_RJT shall be returned. An SW_RJT may be returned if the requested Domain_ID is not available, or for other reasons the Principal switch is unable to grant a Domain_ID to the requesting switch. If more than one Requested Domain_ID was specified in the request, the response shall contain a number of Granted Domain_IDs equal to the number requested. If the Domain Address Manager is unable to grant the full set of Domain_IDs, it shall reject the request.

NOTE 7 – The ability to grant more than one Domain_ID to a single Switch is intended to be used by Switches whose addressing scheme requires the use of more than one Domain_ID. The typical case, however, should be for one Switch to request exactly one Domain_ID.

6.1.8 Hello (HLO)

6.1.8.1 HLO Overview

The Hello Switch Fabric Internal Link Service is used to determine when two way communication is established with a neighbor Switch. The exchange of Domain_IDs is also used to determine the health of the ISL. When an E_Port receives HLO from another E_Port, all Active or Open Class F Sequences and Dedicated Connections shall be unaffected.

The HLO SW_ILS shall be sent as a unidirectional Exchange.

Use of the HLO SW_ILS for Path Selection is described in clause 8. Other uses of HLO are not defined by this Standard.

Protocol:

Hello (HLO) request Sequence

Addressing: For use in Path Selection, the S_ID field shall be set to FFFFFDh, indicating the Fabric Controller of the originating Switch. The D_ID field shall be set to FFFFFDh, indicating the Fabric Controller of the destination Switch.

Payload: The format of the HLO request Payload is shown in table 27**.**

Table 27 – HLO Request Payload

Item	Size Bytes
FSPF Header	20
Reserved	4
Hello_Interval	4
Dead_Interval	4
Recipient Domain_ID	4
Reserved	1
Originating Port Index	3

FSPF Header: The format of the FSPF Header is described in 6.1.8.2.

Hello_Interval: This field shall specify in seconds, the interval between two consecutive HLO messages generated by the Switch during the life of the Adjacency with the neighbor Switch (see table 206).

Dead_Interval: This field shall specify in seconds, the maximum interval the requesting Switch shall wait for reception of a Hello from its neighbor. If the interval expires and no Hello has been received, then the detecting Switch shall bring down the Adjacency (see table 206). Switches may also reset this timer on reception of an FSPF LSA or LSU.

NOTE 8 – The Hello_Interval and Dead_Interval values are configured separately for each port. It is imperative that two E_Ports connected by an ISL share the same two values.

Recipient Domain_ID: This field shall specify the Domain_ID of the neighbor Switch. If the neighbor Domain_ID is known, then the Recipient Domain_ID value shall be set to 000000hllDomain_ID'. If the neighbor Domain_ID is unknown, then the Recipient Domain_ID value shall be FFFFFFFFh.

Valid values for the Domain_ID are: 01h-EFh.

Originating Port Index: This field shall specify the source E_Port Index.

6.1.8.2 FSPF Header Format

The format of the FSPF Header is shown in table 28.

Table 28 – FSPF Header

Item	Size Bytes
Command	4
FSPF Version	1
Obsoleted in FC-SW-4	1
Authentication Type	1
Reserved	1
Originating Domain_ID	4
Authentication	8

Command: This field indicates the command code for the FSPF ILS. FSPF command code values are shown in table 29.

Table 29 – FSPF Command Codes

Value (hex)	Description
14000000	Hello
15000000	Link State Update
16000000	Link State Acknowledgement

FSPF Version: This field contains a code that indicates the FSPF protocol version. The value shall be 02h.

Authentication Type: This field shall specify the usage of the Authentication field. This value shall be set to 00h.

Originating Domain_ID: This field contains the Domain_ID of the Switch sending this request. The Domain_ID value shall be set to 000000hllDomain_ID. If multiple Domain_IDs are in use by the Switch, then the Switch shall use the lowest value Domain_ID as the Originating Domain_ID. Valid values for the Domain_ID are: 01h-EFh.

Authentication: This field shall specify the Authentication information appropriate for the specified Authentication Type. This field shall contain 0000000000000000h.

6.1.9 Link State Update (LSU)

6.1.9.1 LSU Overview

The Link State Update Switch Fabric Internal Link Service requests the transfer of one or more Link State Records from one Switch to another Switch. The transfer may be of updated Link State Records, or may be a transfer of an entire Link State Database. When an E_Port receives an LSU from another E_Port, all Active or Open Class F Sequences and Dedicated Connections shall be unaffected.

The LSU SW_ILS shall be sent as a unidirectional Exchange.

Use of the LSU SW_ILS for Path Selection is described in clause 8. Other uses of LSU are not defined by this Standard.

Protocol:

Link State Update (LSU) request Sequence

Addressing: For use in Path Selection, the S_ID field shall be set to FFFFFDh, indicating the Fabric Controller of the originating Switch. The D_ID field shall be set to FFFFFDh, indicating the Fabric Controller of the destination Switch.

Payload: The format of the LSU request Payload is shown in table 30**.**

Table 30 – LSU Request Payload

Item	Size Bytes
FSPF Header	20
Reserved	3
Flags	1
Number of Link State Records	4
Link State Records	n

FSPF Header: The format of the FSPF header is described in 6.1.8.2.

Flags: This field shall contain information used to synchronize the link state database. The bit map values are listed in table 31.

Table 31 – Flags Field Bit Map

Bit	Description
0	Data Base Exchange - Value b'1' - LSU is used for initial database synchronization Value b'0' - LSU is used for a topology update
1	Database Complete Value b'1' - Last sequence of data base synchronization. LSU contains no LSRs. Value b'0' - Not the last sequence of data base synchronization
2-7	Reserved

Number of Link State Records: This field shall specify the number of Link State Records that follow this field.

6.1.9.2 Link State Record (LSR) Format

Link State Record: There is one format for the LSR; the Link Descriptor Format. The Link Descriptor format is shown in table 32. One or more descriptors may be contained in a single LSR.

Table 32 – Link State Record - Link Descriptor Format

Item	Size Bytes
Link State Record Header (LSR Type 01h)	24
Reserved	2
Number of Links	2
Link Descriptor #1	16
...	16
...	16
Link Descriptor #n	16

Link State Header: The format of the link state header is described in 6.1.9.3.

Number of Links: This field specifies the number of link descriptors contained in the Link State Record.

NOTE 9 – A Switch keeps a list of all its ISLs, but only ISLs that are in the full state are advertised in the LSR.

Link Descriptor: The format of the Link Descriptor is described in 6.1.9.4.

6.1.9.3 Link State Header Format

The format of the Link State Header is described in table 33.

Table 33 – Link State Header Format

Item	Size Bytes
LSR Type	1
Reserved	1
LSR Age	2
Reserved	4
Link State Identifier	4
Advertising Domain_ID	4
Link State Incarnation Number	4
Checksum	2
LSR Length	2

LSR Type: The LSR types are depicted in table 34.

Table 34 – Link State Record Type Field Values

Value (hex)	Description
01	Switch Link Record
02	Obsolete in FC-SW-4
F0-FF	Vendor Specific
all others	Reserved

LSR Age: This field contains a value that indicates the time in seconds since the record has been generated.

NOTE 10 – LSR Age may be used to flush old records from the database.

Link State Identifier: This field contains the Domain_ID of the Switch that owns the LSR. The format of Link State Identifier shall be set to 000000hllDomain_ID'.

Advertising Domain_ID: This field contains the Domain_ID of the Switch that is advertising the LSR on behalf of the owning Switch.

Incarnation Number: This field contains the current incarnation of the LSR.

Checksum: This field contains the checksum value of the Link State Record. This value shall be calculated on all bytes of the LSR except for the LSR Age field. A complete description of how the checksum is calculated is given in 8.5.4.

NOTE 11 – Not calculating the checksum on the Age value allows the Age value to advance without requiring the recalculation of the checksum.

LSR Length: This field contains the length of the LSR in bytes. The LSR length includes the LSR Age field.

6.1.9.4 Link Descriptor Format

This field contains a descriptor that defines the state of the ISL, as defined in table 35.

Table 35 – Link Descriptor Format

Item	Size Bytes
Link ID	4
Reserved	1
Output Port Index	3
Reserved	1
Neighbor Port Index	3
Link Type	1
Reserved	1
Link Cost	2

Link Identifier: This field identifies the link and contains the Domain_ID of the neighbor Switch at the other end of the ISL, relative to the owning Switch.

Output Port Index: This field shall specify the source E_Port Index.

Neighbor Port Index: The field shall specify the destination E_Port Index.

Link Type: This field shall specify the type of ISL. Values are depicted in table 36.

Table 36 – Link Type Values

Value (hex)	Description
01	Point to Point Link
F0-FF	Vendor Specific
all others	Reserved

Link Cost: This field contains a value that describes the cost of transmitting a frame over the ISL. See 8.5.5 for a complete description of Link Cost calculation.

6.1.10 Link State Acknowledgement (LSA)

The Link State Acknowledgement Switch Fabric Internal Link Service is used to acknowledge the receipt of an LSR. When an E_Port receives LSA from another E_Port, all Active or Open Class F Sequences and Dedicated Connections shall be unaffected.

The LSA SW_ILS shall be sent as a unidirectional Exchange.

Use of the LSA SW_ILS for Path Selection is described in clause 8. Other uses of LSA are not defined by this Standard.

Protocol:

Link State Acknowledgement (LSA) request Sequence

Addressing: For use in Path Selection, the S_ID field shall be set to FFFFFDh, indicating the Fabric Controller of the originating Switch. The D_ID field shall be set to FFFFFDh, indicating the Fabric Controller of the destination Switch.

Payload: The format of the LSA request Payload is shown in table 37**.**

Table 37 – LSA Request Payload

Item	Size Bytes
FSPF Header	20
Reserved	3
Flags	1
Number of Link State Record Headers	4
Link State Record Headers	n

FSPF Header: The format of the FSPF Header is described in 6.1.8.2.

Flags: The bit settings shall match the bit settings specified in the Flags field of the corresponding LSU.

Number of Link State Record Headers: This field shall specify the number of Link State Record Headers that follow this field.

Link State Record Header: The format of the Link State Record header is described in 6.1.9.3.

6.1.11 Build Fabric (BF)

The Build Fabric Switch Fabric Internal Link Service requests a non-disruptive reconfiguration of the entire Fabric. Fabric Configuration is performed as described in clause 7.

NOTE 12 – The BF SW_ILS allows the Fabric to attempt reconfiguration without loss of or change of address. Examples of situations in which BF is appropriate include certain losses of a Principal ISL (Link Failure or Offline), or when two Fabrics are joined.

A BF shall cause the Domain_ID_List to be cleared.

The transmission or reception of BF shall not of itself cause the loss of Class N frames, or cause a busy response to any Class N frames. Active or Open Class F Sequences between the two E_Ports, and any Dedicated Connections, shall not be abnormally terminated.

Use of the BF SW_ILS for Fabric Configuration is described in 7.3 and 7.4.

Protocol:

Build Fabric (BF) request Sequence
Accept (SW_ACC) Reply Sequence

Addressing: For use in Fabric Configuration, the S_ID field shall be set to FFFFFDh, indicating the Fabric Controller of the originating Switch. The D_ID field shall be set to FFFFFDh, indicating the Fabric Controller of the destination Switch.

Payload: The format of the BF request Payload is shown in table 38**.**

Table 38 – BF Request Payload

Item	Size Bytes
17000000h	4

Reply Switch Fabric Internal Link Service Sequence:

Service Reject (SW_RJT)
Signifies the rejection of the BF command
Accept (SW_ACC)
Signifies acceptance of the BF request.
– Accept Payload

Payload: The format of the BF accept Payload is shown in table 39**.**

Table 39 – BF Accept Payload

Item	Size Bytes
02000000h	4

6.1.12 Reconfigure Fabric (RCF)

The Reconfigure Fabric Switch Fabric Internal Link Service requests a disruptive reconfiguration of the entire Fabric. Fabric Configuration is performed as described in clause 7.

NOTE 13 – Since the RCF causes a complete reconfiguration of the Fabric, and may cause addresses allocated to a Switch to change, this SW_ILS should be used with caution. The BF SW_ILS allows the Fabric to attempt reconfiguration without loss of or change of address and therefore should be attempted before an RCF.

Examples of situations in which RCF may be appropriate include resolution of overlapped Domains, or the failure of a Fabric Reconfiguration initiated by a BF.

An RCF shall cause the Domain_ID_List to be cleared.

When an RCF is transmitted by an E_Port, any Active or Open Class F Sequences between the two E_Ports, and any Dedicated Connections, shall be abnormally terminated. Also, all Class N frames shall be discarded, and all Dedicated Connections shall be abnormally terminated.

When an RCF is received and accepted by an E_Port, any Active or Open Class F Sequences between the two E_Ports, and any Dedicated Connections, shall be abnormally terminated prior to transmission of the SW_ACC reply Sequence. Also, all Class N frames shall be discarded, and all Dedicated Connections shall be abnormally terminated prior to transmission of the SW_ACC reply Sequence. If an E_Port rejects the RCF, the Switch to which it belongs shall not propagate the RCF over its other E_Ports, nor send an ELP over its Isolated Interconnect_Ports. The rejecting E_Port shall go in Isolated state and send an SW_RJT reply Sequence with reason code explanation “E_Port is Isolated”.

Use of the RCF SW_ILS for Fabric Configuration is described in 7.3 and 7.4.

Protocol:

Reconfigure Fabric (RCF) request Sequence
Accept (SW_ACC) Reply Sequence

Addressing: For use in Fabric Configuration, the S_ID field shall be set to FFFFFDh, indicating the Fabric Controller of the originating Switch. The D_ID field shall be set to FFFFFDh, indicating the Fabric Controller of the destination Switch.

Payload: The format of the RCF request Payload is shown in table 40**.**

Table 40 – RCF Request Payload

Item	Size Bytes
18000000h	4

Reply Switch Fabric Internal Link Service Sequence:

Service Reject (SW_RJT)
Signifies the rejection of the RCF command
Accept (SW_ACC)
Signifies acceptance of the RCF request.
– Accept Payload

Payload: The format of the RCF accept Payload is shown in table 41**.**

Table 41 – RCF Accept Payload

Item	Size Bytes
02000000h	4

6.1.13 Inter-Switch Registered State Change Notification (SW_RSCN)

The Fabric shall distribute RSCNs between Switches using the Inter-Switch RSCN payload.

The inter-Switch RSCN format is similar to the format used for Nx_Ports, except:

a) The affected N_Port field is as defined in the payload description in this subclause. The upper nibble is masked before delivery to an Nx_Port;

b) a "detection function" code is contained in the payload;

c) An SW_ILS is used as the transport.

d) If a Switch has any directly attached Nx_Ports registered to receive RSCNs, it shall convert a received SW_RSCN SW_ILS to an appropriate RSCN ELS.

Protocol:

Inter-Switch Registered State Change Notification (SW_RSCN) request Sequence
Accept (SW_ACC) Reply Sequence

Addressing: The S_ID shall be set to FFFCxxh designating the Domain Controller ID of the Switch that generates the SW_RSCN. The D_ID shall be set to FFFCyyh to designate the Domain Controller ID of the recipient Switch.

Payload: The format of the SW_RSCN request payload is shown in table 42.

.

Table 42 – SW_RSCN Request Payload

Item	Bytes
1B000000h	4
Affected N_Port	4
Detection Function	4
Number of Device Entries (m)	4
Device Entry 1	20
Device Entry 2	20
.....	
Device Entry m	20

Affected N_Port

This field specifies the address of the affected N_Port.

For Fabric events (see above). The first nibble in the high order byte shall be:

0xh = no additional info;

1xh = port is online;

2xh= port is offline;

where (x indicates a valid hexadecimal value).

The second nibble in the high order byte shall be:

x0h = port address format

x1h = area address format

x2h = domain address format

x3h = Fabric address format

where (x indicates a valid hexadecimal value).

The remaining three bytes contain the 24 bit address.

Detection function

The value used by SCR (see FC-FS-3) to describe the detector of the change:

00000001h = Fabric detected

00000002h= N_Port detected

Number of Device Entries

This field contains the number of device entries in the payload.

Device Entry

The format of the device entry is specified in table 43:

Table 43 – Device Entry Format

Item	Size (Bytes)
Port State	1
N_Port_ID	3
N_Port_Name	8
Node_Name	8

Port State

This byte may contain the Port State. The state values are the same values as defined in the Affected Port description.

N_Port_ID

This field contains the 24 bit Fibre Channel Address of the device.

N_Port_Name

This field contains the Name_Identifier of the Port associated with the device.

Node_Name

This field contains the Name_Identifier of the Node associated with the device

NOTE 14 – For an N_Port device the number of devices would be 1 and the N_Port_ID entry in the only device entry would be identical to the value in the N_Port_ID portion of the affected N_Port field in the payload. In case of a Loop port and where the SW_RSCN format is a AREA wide format, the number of devices would be the total number of devices in the loop port that is either coming online or going offline. Also note that if there are 126 devices in a loop port then the SW_RSCN itself may become a multi-frame sequence. An AREA format SW_RSCN should be converted to an RSCN ELS with only one Affected N_Port_ID page.

Reply Switch Fabric Internal Link Service Sequence:

Service Reject (SW_RJT)
Signifies the rejection of the SW_RSCN request
Accept (SW_ACC)
Signifies acceptance of the SW_RSCN request and its RSCN information.
– Accept Payload

Payload: The format of the SW_RSCN accept Payload is shown in table 44**.**

Table 44 – SW_RSCN Accept Payload

Item	Size Bytes
02000000h	4

6.1.14 Distribute Registered Link Incident Records (DRLIR)

Distribute Registered Link Incident Records (DRLIR) Switch Fabric Internal Link Service provides a method for a Fabric built RLIR to be distributed to every Switch in the Fabric. The normal response to a DRLIR SW_ILS sequence shall be an Accept (SW_ACC) reply sequence. If the recipient Switch does not support the DRLIR SW_ILS, the recipient Switch shall reply with an SW_RJT sequence with a reason code of “command not supported”. If the recipient Switch does not support the RLIR Format contained in the DRLIR, the recipient Switch shall reply with an SW_RJT sequence with a reason code of “unable to perform command request”.

When a Switch creates an RLIR, the Switch shall generate the corresponding DRLIRs. A DRLIR shall be created for every Established Registration List that the originating Switch supports, even if that Switch has no registrants in the Established Registration List. The Switch shall distribute the DRLIRs to every Switch in the Fabric via the Domain Controller Identifier.

When a Switch receives a DRLIR, the Switch shall extract the RLIR. The RLIR shall then be sent to the local registrants of the given RLIR format as if the RLIR was generated in the local Switch.

Protocol:

Distribute Registered Link Incident Record (DRLIR) request Sequence
Accept (SW_ACC) reply Sequence

Addressing: The S_ID shall be set to FFFCxxh designating the Domain Controller ID of the Switch that generates the DRLIR. The D_ID shall be set to FFFCyyh to designate the Domain Controller ID of the recipient Switch.

Payload: The format of the DRLIR request Payload is shown in table 45.

Table 45 – DRLIR Request Payload

Item	Size (Bytes)
1E000000h	4
Embedded RLIR	28-328

Embedded RLIR: The format of the embedded RLIR is defined in FC-LS-2.

Reply Switch Fabric Internal Link Service Sequence:

Service Reject (SW_RJT)
Signifies the rejection of the DRLIR request
Accept (SW_ACC)
Signifies acceptance of the DRLIR request and its Link Incident Record.
– Accept Payload

Payload: The format of the DRLIR accept Payload is shown in table 46.

Table 46 – DRLIR Accept Payload

Item	Size Bytes
02000000h	4

6.1.15 Merge Request (MR)

The Merge request Switch Fabric Internal Link Service requests that the recipient merge any zoning data with the zoning data supplied in the MR payload according to the rules specified in table 193. The Merge request provides a mechanism to distribute zoning information between adjacent Switches. Use of the Merge request is described in 10.2.

To distinguish between Enhanced and Basic Zoning, a Protocol Version field is used. In particular:

If Protocol Version is 00h:

a) the payload contains Basic Zoning structures

b) the Fabric is working in Basic Zoning mode

If Protocol Version is 01h:

a) the payload contains Enhanced Zoning structures

b) the Fabric is working in Enhanced Zoning mode

The Zone Merge may be successful only between Switches working in the same Zoning mode, i.e., both in Basic Zoning mode or both in Enhanced Zoning mode. This means that the value of the received Protocol Version field shall match the current Zoning operational mode of the Switch, other-

wise the link is Isolated. In particular, if a Switch working in Enhanced Zoning mode receives over a certain link a MR with Protocol Version = 0, then that link shall be Isolated.

Protocol:

Merge Request (MR) request Sequence
Accept (SW_ACC) reply Sequence

Addressing: The S_ID shall be set to FFFFFDh, indicating the Fabric Controller of the originating Switch. The D_ID shall be set to FFFFFDh, indicating the Fabric Controller of the Destination Switch.

Payload: The format of the MR request payload is described in 6.1.15.1.

6.1.15.1 Merge Request Payload

The format of the Merge Zone request payload is depicted in table 47.

Table 47 – Merge Request Payload

Item	Size
Merge Request - 22h	1
Protocol Version	1
Version Specific Payload	x

Protocol Version

The Protocol Version field contains a number that identifies the Zoning Operational mode of the Fabric (Basic or Enhanced) and the format of the Zoning structures conveyed in the payload. Table 48 depicts the defined values.

Table 48 – Protocol Version Values

Value	Meaning
00	Basic Zoning
01	Enhanced Zoning
others	Reserved

6.1.15.1.1 Merge Request Payload in Basic Zoning

The format of the Version Specific payload for Protocol Version = 00h is depicted in table 49.

Table 49 – Basic Zoning Payload

Item	Size
Active Zone Set Length	2
Active Zone Set Name	a
Active Zone Set Object List	x
Zone Set Database Object Length	4
Zone Set Database Object List	y

NOTE 15 – It is acceptable to ignore the Zone Set Database Object list and supply 0 for the Zone Set Database Object Length.

Active Zone Set Length

The Active Zone Set Length field contains the length of the Active Zone Set Name and Active Zone Set Object List, x+y (in bytes).

Active Zone Set Name

The Active Zone Set name field contains the name of the Active Zone Set. The format of the Active Zone Set name follows the structure and rules for the Name Entry described in 10.4.2.3.

Active Zone Set Object List

The Active Zone Set may only contain Zone Objects (type 2) in the Active Zone Set Object List.

In the Basic Zoning Framework each of the Zone Object members may be of member type N_Port_Name (type 1), Domain_ID and physical port (type 2), or N_Port_ID (type 3). All other zone member types are not allowed.

Zone Set Database Object List

The Zone Set Database Object list contains information regarding all zone configurations plus all objects that comprise the zone sets. The Active Zone Set, name and object list, shall not be included in the Zone Set Database Object list. Support of the Zone Set Database Object list is optional. A Zone Set Database Object length of 0 is required if the Zone Set Database is not supported.

In the Basic Zoning Framework the Zone Set Database does not use all Zoning Object types in the Zone Set Database Object List. Zone Set type objects shall have members that are only Zone Objects (type 2). Each of the Zone Object members may be of member type N_Port_Name (type 1), Domain_ID and physical port (type 2), N_Port_ID (type 3), or Alias Name (type 4). Each Zone Alias Object member may be of member type N_Port_Name (type 1), Domain_ID and physical port (type 2), or N_Port_ID (type 3). All other combinations are not allowed.

6.1.15.1.2 Merge Request Payload in Enhanced Zoning

The format of the Version Specific payload for Protocol Version = 01h is depicted in table 50.

Table 50 – Enhanced Zoning Payload

Item	Size
Reserved	2
Enhanced Zoning Flags	4
Active Zone Set Length	4
Active Zone Set Object List	x
Zone Set Database Object Length	4
Zone Set Database Object List	y

Enhanced Zoning Flags

The format of the Enhanced Zoning Flags field is as follows:

Bit 0- reserved.

Bit 1- reserved.

Bit 2- Indicates the Merge Control Setting. When this bit is one, this Switch is working in Restrict mode, so it may join a Fabric only if the Fabric's Zoning Database is equal to its Zoning Database. When this bit is zero, this Switch is working in Allow mode, so it may join a Fabric only if the Fabric's Zoning Database is mergeable with its Zoning Database.

Bit 3- Indicates the Default Zone Setting. When this bit is one this Switch denies traffic between members of the Default Zone. When this bit is zero this Switch permit traffic between members of the Default Zone.

Bit 4- Indicates that the Zone Set Database is supported. When this bit is one, the Zone Server on this Switch is able to maintain a Zone Set Database. When this bit is zero, the Zone Server on this Switch is not able to maintain a Zone Set Database.

Bit 5- Indicates that the Zone Set Database is enabled. When this bit is one, the Zone Server on this Switch is maintaining a Zone Set Database. When this bit is zero, the Zone Server on this Switch is not maintaining a Zone Set Database.

Bit 6-31 reserved.

Active Zone Set Length

The Active Zone Set Length field is extended to 4 bytes and contains the length of the Active Zone Set Object List, in bytes.

Active Zone Set Object List

The Active Zone Set may only contain Zone Objects (type ‘02’) in the Active Zone Set Object List.

Any Zone Member Identifier type may be used as Zone Member in the Active Zone Set’s Zone Objects, with the exception of the Alias Name identifier (type ‘04’).

Zone Set Database Object List

The Zone Set Database Object list contains information regarding all zone configurations plus all objects that comprise the zone sets. The Active Zone Set, name and object list, shall not be included in the Zone Set Database Object list. Support of the Zone Set Database Object list is optional. A Zone Set Database Object length of 0 is required if the Zone Set Database is not supported.

In the Enhanced Zoning Framework the Zone Set Database may use all Zoning Object types in the Zone Set Database Object List. Zone Set type objects shall have members that are only Zone Reference Objects (type ‘04’). Any Zone Member Identifier type may be used as Zone Member in the Zone Set Database’s Zone Objects. Any Alias Member Identifier type may be used as Zone Alias Member in the Zone Set Database’s Zone Alias Objects, with the exception of the Alias Name identifier (type ‘04’).

6.1.15.2 Merge Request Reply

Reply Switch Fabric Internal Link Service Sequence:

Service Reject (SW_RJT)
  Signifies the rejection of the MR request
Accept (SW_ACC)
  Signifies acceptance of the MR request.

Successful completion of the Merge Request is indicated by an SW_ACC. If the recipient is unable to complete the Merge Request, a SW_RJT with reason code “Unable to Complete Command Requested” and Reason Code Explanation indicating why the Merge Request was not completed shall be returned, and the E_Port shall enter the Isolated State.
The format of the Merge request Accept Payload is shown in table 51.

Table 51 – Merge Request Accept Payload

Item	Size (Bytes)
02000000h	4
Reserved	1
Obsolete	1
Obsolete	1
Obsolete	1

6.1.16 Acquire Change Authorization Request (ACA)

Acquire Change Authorization requests are SW_ILSs addressed from the Domain Controller of the Managing Switch to the Domain Controller of a Managed Switch. Acquire Change Authorization re-

quest messages are sent by a Managing Switch to Managed Switches to reserve local resources in each Switch.

The Acquire Change Authorization (ACA) request Switch Fabric Internal Link Service requests that the recipient reserve local resources for the purposes of changing Switch or Switch service resources. The Acquire Change Authorization request provides a mechanism to lock a Fabric to distribute information (e.g., Zoning) amongst Switches. Use of the Acquire Change Authorization is described in 10.6.2.

Protocol:

Acquire Change Authorization (ACA) request Sequence
Accept (SW_ACC) reply Sequence

Addressing: The S_ID shall be set to FFFCxxh, indicating the Domain Controller of the originating Switch. The D_ID shall be set to FFFCyyh, indicating the Domain Controller of the destination Switch.

Payload: The format of the ACA request payload is shown in table 52.

Table 52 – ACA Request Payload

Item	Size (Bytes)
23h	1
Reserved	1
Domain_ID List Length	2
Reserved	3
Domain_ID #1	1
Reserved	3
Domain_ID #2	1
...	
Reserved	3
Domain_ID #n	1

Domain_ID_List Length: This field specifies the length of the Domain_ID List in bytes.

Domain_ID List: The payload contains a list of Domain_ID's known to the Managing Switch. The Domain_ID List begins with the Reserved field immediately following the Domain_ID List Length field. The recipient checks the list of Domain_ID's against those it knows to be active within the Fabric. If the list differs from the Domain_ID's known to the Managed Switch, the request is rejected with an SW_RJT with a Reason Code "Unable to Perform Command Requested", and a Reject Reason Code Explanation of "Fabric Changing".

Reply Switch Fabric Internal Link Service Sequence:

Service Reject (SW_RJT)
Signifies the rejection of the ACA request
Accept (SW_ACC)
Signifies acceptance of the ACA request.

An SW_ACC indicates that the operation completed successfully.

If the Managed Switch is unable to accept the ACA due to another pending ACA, an SW_RJT with reason code “Logical Busy” shall be returned.

The format of the Acquire Change Authorization Accept Payload is shown in table 53.

Table 53 – Acquire Change Authorization Accept Payload

Item	Size (Bytes)
02000000h	4
Reserved	1
Obsolete	1
Obsolete	1
Obsolete	1

NOTE 16 – After an unsuccessful attempt to acquire change authorization, a Switch should release any acquired change authorization, and wait a random time before attempting ACA again.

6.1.17 Release Change Authorization (RCA) Request

Release Change Authorization requests are SW_ILSs addressed from the Domain Controller of the Managing Switch to the Domain Controller of a Managed Switch. Release Change Authorization request messages are sent by a Managing Switch to Managed Switches to release local resources in each Switch.

Protocol:

Release Change Authorization (RCA) request Sequence
Accept (SW_ACC) Reply Sequence

Addressing: The S_ID shall be set to FFFCxxh, indicating the Domain Controller of the originating Switch. The D_ID shall be set to FFFCyyh, indicating the Domain Controller of the Destination Switch.

Payload: The format of the RCA request payload is shown in table 54.

Table 54 – RCA Request Payload

Item	Size (Bytes)
24000000h’	4

Reply Switch Fabric Internal Link Service Sequence:

Service Reject (SW_RJT)
Signifies the rejection of the RCA request
Accept (SW_ACC)
Signifies acceptance of the RCA request.

The format of the Release Change Authorization Accept Payload is shown in table 55.

Table 55 – Release Change Authorization Accept Payload

Item	Size (Bytes)
02000000h	4
Reserved	1
Obsolete	1
Obsolete	1
Obsolete	1

6.1.18 Stage Fabric Configuration Update (SFC) Request

Stage Fabric Configuration Update requests are SW_ILSs addressed from the Domain Controller of the Managing Switch to the Domain Controller of a Managed Switch. Stage Fabric Configuration Update request messages are sent by a Managing Switch to Managed Switches to stage changes to local resources in each Switch.

The Stage Fabric Configuration Update request provides a mechanism to distribute information to other switches in the Fabric. Use of the Stage Fabric Configuration Update is described in 10.6.3.

Protocol:

Stage Fabric Configuration Update (SFC) request Sequence
Accept (SW_ACC) Reply Sequence

Addressing: The S_ID shall be set to FFFCxxh, indicating the Domain Controller of the originating Switch. The D_ID shall be set to FFFCyyh, indicating the Domain Controller of the Destination Switch.

Payload: The format of the SFC request payload is shown in table 56.

Table 56 – SFC Request Payload

Item	Size (Bytes)
25h	1
Operation Request (see table 57)	1
Operation Specific Payload	x

Operation Request: The operation request value further specifies the operation to be attempted by the recipient

Operation Specific Payload: The remaining part of the SFC payload is dependent on the operation requested. Table 57 depicts the currently defined Operation Request values.

.

Table 57 – Operation Request Value

Value (hex)	Description
00-02	Reserved
03	Activate Zone Set
04	Deactivate Zone Set
05-07	Reserved for FC-SP Use (See reference [14])
08	Activate Zone Set Enhanced
09	Deactivate Zone Set Enhanced
0A	Distribute Zone Set Database
0B	Activate Zone Set by Name
0C	Set Zoning Policies
0D-1F	Reserved
20-3F	Reserved for FC-SP Use (See reference [14])
40 thru DF	Reserved
E0 thru FF	Vendor Specific

Reply Switch Fabric Internal Link Service Sequence:

Service Reject (SW_RJT)
Signifies the rejection of the SFC request
Accept (SW_ACC)
Signifies acceptance of the SFC request.

Stage Fabric Configuration Update responses are Class F frames addressed from the Domain Controller of a Managed Switch to the Domain Controller of the Managing Switch. A Stage Fabric Configuration Update Accept is sent by a Managed Switch to a Managing Switch when a Stage Fabric Configuration Update request has been received.

The format of the Stage Fabric Configuration Accept Payload is shown in table 58.

Table 58 – Stage Fabric Configuration Update Accept Payload

Item	Size (Bytes)
02000000h	4
Reserved	1
Obsolete	1
Obsolete	1
Obsolete	1

6.1.18.1 SFC in Basic Zoning

Operation Requests values 03 and 04 are used in the context of Basic Zoning. Only the Basic Zoning Data structures defined in 10.4.2 shall be used with them. Enhanced Zoning Data Structures shall not be used with them. Table 59 depicts the payload structure for them.

Table 59 – Payload for Operation Request Values 03 and 04

Item	Size (Bytes)
Zone Set Length	2
Zone Set Name	a
Zoning Object List	x
Zone Set Database Object Length	4
Zone Set Database Object List	y

The Zone Set Length field specifies the length in bytes of the following.

a) Zone Set Name;

b) Zoning Object List.

The Zone Set Database Object Length field specifies the length in bytes of the Zone Set Database Object List. Refer to clause 6.1.15.1 for implementation notes.

If the request Value is 03h, then the remainder of the SFC payload contains the Zone Set configuration utilized by the recipient to determine if a Activate Zone set operation may be attempted.

If the request Value is 04h, the remainder of the SFC payload is ignored

6.1.18.2 SFC in Enhanced Zoning

The following Operation Requests are used in the context of Enhanced Zoning. Only the Enhanced Zoning Data structures defined in 10.4.4 shall be used with them. Basic Zoning Data Structures shall not be used with them.

6.1.18.2.1 Operation Request 'Activate Zone Set Enhanced'

Operation Request 'Activate Zone Set Enhanced' is used in Enhanced Zoning to activate a Zone Set distributing its definition across the Fabric. Together with the Zone Set to be activated, also the entire Zone Set Database may be distributed. Table 60 depicts the payload format.

Table 60 – Payload for Operation Request 'Activate Zone Set Enhanced'

Item	Size (Bytes)
Reserved	2
Active Zone Set Length	4
Active Zone Set Object List	x
Zone Set Database Object Length	4
Zone Set Database Object List	y

6.1.18.2.1.1 Length Fields

The Active Zone Set Length field contains the length of the Active Zone Set Object List, in bytes. The Active Zone Set Length field is extended to 4 bytes.

The Zone Set Database Object length field specifies the length of the Zone Set Database Object List, in bytes. If set to zero, then the Zone Set Database is not included in the payload.

6.1.18.2.1.2 Object Lists

The Object Lists shall use the appropriate Enhanced Zoning payloads for the Zone Set to be activated and the Zone Set Database, as described in 10.4.4.

6.1.18.2.2 Operation Request 'Deactivate Zone Set Enhanced'

Operation Request 'Deactivate Zone Set Enhanced' is used in Enhanced Zoning to deactivate the current Active Zone Set. Table 61 depicts the payload format.

Table 61 – Payload for Operation Request 'Deactivate Zone Set Enhanced'

Item	Size (Bytes)
Reserved	2

6.1.18.2.3 Operation Request 'Distribute Zone Set Database'

Operation Request 'Distribute Zone Set Database' applies to the Zone Set Database. Its purpose is to distribute in the Fabric a new definition of the Zone Set Database, without affecting the Active Zone Set. Table 62 defines its payload.

Table 62 – Payload for Operation Request 'Distribute Zone Set Database'

Item	Size (Bytes)
Reserved	2

Table 62 – Payload for Operation Request 'Distribute Zone Set Database'

Item	Size (Bytes)
Zone Set Database Object Length	4
Zone Set Database Object List	y

6.1.18.2.3.1 Zone Set Database Object Length

The Zone Set Database Object length field specifies the length of the Zone Set Database Object List. If the Zone Set Database Object Length is zero, the Zone Set Database Object List is not present, and this operation clears the entire Zone Set Database.

6.1.18.2.3.2 Zone Set Database Object Lists

The Object List shall use the appropriate Enhanced Zoning payloads for the Zone Set Database, as described in 10.4.4.

6.1.18.2.4 Operation Request 'Activate Zone Set by Name'

Operation Request 'Activate Zone Set by Name' applies to both Active Zone Set and Zone Set Database. Its purpose is to activate a Zone Set defined in the Zone Set Database without having to transmit over the Fabric its definition. Table 63 depicts the payload format.

Table 63 – Payload for Operation Request 'Activate Zone Set by Name'

Item	Size (Bytes)
Reserved	2
Zone Set Name	a

6.1.18.2.4.1 Zone Set Name

This field contains the Name of the Zone Set to be activated. It shall be defined in the Zone Set Database.

6.1.18.2.5 Operation Request 'Set Zoning Policies'

Operation Request 'Set Zoning Policies' is used in Enhanced Zoning to establish the Fabric Zoning Policies. Table 64 depicts the payload format.

Table 64 – Payload for Operation Request 'Set Zoning Policies'

Item	Size (Bytes)
Reserved	2
Enhanced Zoning Flags	4

6.1.18.2.5.1 Enhanced Zoning Flags

The format of the Enhanced Zoning Flags field is as follows:

Bit 0-1 reserved.

Bit 2- **Merge Control Setting.** When this bit is one the Fabric shall work in Restrict mode, so a Switch may join the Fabric only if its Zoning Database is equal to the Fabric's Zoning Database. When this bit is zero the Fabric shall work in Allow mode, so a Switch may join the Fabric only if its Zoning Database is mergeable with the Fabric's Zoning Database.

Bit 3- **Default Zone Setting.** When this bit is one the Fabric shall deny traffic between members of the Default Zone. When this bit is zero the Fabric shall permit traffic between members of the Default Zone.

Bit 4-31 reserved.

6.1.19 Update Fabric Configuration (UFC) Request

Update Fabric Configuration requests are SW_ILSs addressed from the Domain Controller of the Managing Switch to the Domain Controller of a Managed Switch. Update Fabric Configuration request messages are sent by a Managing Switch to Managed Switches to effect the changes to local resources in each Switch. There is no data included in this message.

Protocol:

Update Fabric Configuration (UFC) request Sequence
Accept (SW_ACC) Reply Sequence

Addressing: The S_ID shall be set to FFFCxxh, indicating the Domain Controller of the originating Switch. The D_ID shall be set to FFFCyyh, indicating the Domain Controller of the Destination Switch.

Payload: The format of the UFC request payload is shown in table 65.

Table 65 – Update Fabric Configuration Request Payload

Item	Size (Bytes)
26000000h	4

Reply Switch Fabric Internal Link Service Sequence:

Service Reject (SW_RJT)
  Signifies the rejection of the UFC request
Accept (SW_ACC)
  Signifies acceptance of the UFC request.

The format of the Update Fabric Configuration Accept Payload is shown in table 66.

Table 66 – Update Fabric Configuration Accept Payload

Item	Size (Bytes)
02000000h	4
Reserved	1
Obsolete	1
Obsolete	1
Obsolete	1

6.1.20 Check E_Port Connectivity (CEC)

The Check E_Port Connectivity (CEC) SW_ILS requests the exchange of Link Parameters between two E_Ports connected through B_Ports. The exchange of Link Parameters establishes the operating environment between the two E_Ports, and the capabilities of the Switches that are connected by the E_Ports. The CEC SW_ILS is transparent to B_Ports. Use of the CEC SW_ILS for Switch Port initialization is described in 7.2.

Protocol:

Check E_Port Connectivity (CEC) Request Sequence
Accept (SW_ACC) Reply Sequence

Addressing: For use in Switch Port initialization, the S_ID field shall be set to FFFFFDh, indicating the Fabric Controller of the originating Switch; the D_ID field shall be set to FFFFFDh, indicating the Fabric Controller of the destination Switch.

Payload: The format of the CEC request Payload is shown in table 67.

Table 67 – CEC Request Payload

Item	Size (Bytes)
2900 0000h	4
Revision	1
Flags	2
Reserved	1
R_A_TOV	4
E_D_TOV	4
Requester E_Port_Name	8
Requester Switch_Name	8
Fabric Controller Class F Service Parameters	16
Obsolete in FC-SW-5	4
Class 2 E_Port Parameters	4
Class 3 E_Port Parameters	4
Reserved	20

The descriptions of the fields in the CEC Request payload are as defined in the ELP Request payload (see 6.1.4). In this case the Interconnect_Port is functioning as an E_Port.

Reply Switch Fabric Internal Link Service Sequence:

Service Reject (SW_RJT)
Signifies the rejection of the CEC request.
Accept (SW_ACC)
Signifies acceptance of the CEC request.
– Accept Payload

Payload: The format of the CEC Accept Payload is shown in table 68.

Table 68 – CEC Accept Payload

Item	Size (Bytes)
0200 0000h	4
Revision	1
Flags	2
Reserved	1
R_A_TOV	4
E_D_TOV	4
Responder E_Port_Name	8
Responder Switch_Name	8
Fabric Controller Class F Service Parameters	16
Obsolete in FC-SW-5	4
Class 2 E_Port Parameters	4
Class 3 E_Port Parameters	4
Reserved	20

The descriptions of the fields in the CEC Accept payload are as defined in the ELP Accept payload (see 6.1.4). In this case the Interconnect_Port is functioning as an E_Port.

6.1.21 Exchange Switch Capabilities

The Exchange Switch Capabilities SW_ILS defines a mechanism for two Switches to exchange vendor and protocol information.

A Switch is not required to support the ESC SW_ILS. If the receiving Switch does not support the ESC SW_ILS, it shall respond with an SW_RJT with a reason code of “Command Not Supported”.

Protocol:

Exchange Switch Capabilities (ESC) request Sequence
Accept (SW_ACC) Reply Sequence

Addressing: For use in Fabric Configuration, the S_ID field shall be set to FFFFFDh, indicating the Fabric Controller of the originating Switch. The D_ID field shall be set to FFFFFDh, indicating the Fabric Controller of the destination Switch.

Payload: The format of the ESC request Payload is shown in table 69.

Table 69 – ESC Request Payload

Item	Size Bytes
Command Code = 30h	1
Flags	1
Payload Length	2
Vendor ID String	8
Protocol Descriptor #1	12
...	
Protocol Descriptor #n	12

Flags: This field contains flag bits (7:0) that provide additional information about the ESC request. The following flag bits are defined.

Bit 0, Multiple Protocol Descriptors. This bit shall indicate whether the ESC Accept may contain multiple Accepted Protocol Descriptors. When the bit is set to one, the ESC Accept may contain multiple Accepted Protocol Descriptors. When the bit is set to zero, the ESC Accept shall contain only one Accepted Protocol Descriptor.

Bits 1-7 shall be reserved.

Payload Length: This field contains a 16-bit unsigned binary integer that specifies the total length in bytes of the Payload. The least significant two bits shall be zero. The value specified shall be greater than or equal to 24, and less than or equal to 65532.

Vendor ID String: This field shall contain a T10 Vendor ID of either the manufacturer of the requesting Switch, or an OEM of the requesting Switch.

Protocol Descriptor: This field allows the requesting Switch to identify which Switch-to-Switch protocols it supports. There may be more than one Protocol Descriptors specified in the ESC SW_ILS frame. This list of Protocol Descriptors allows a single port on the requesting Switch to specify that it supports more than one Switch-to-Switch protocol. The format of the Protocol Descriptor is shown in table 70.

Table 70 – Protocol Descriptor Format

Item	Size Bytes
Vendor ID String	8
Reserved	2
Protocol ID	2

Vendor ID String: For non-vendor specific protocols, this field shall be zero filled. For vendor specific protocols, this field shall contain the T10 Vendor ID associated with the Protocol ID field. It is the intention that this field contain the T10 Vendor ID of the original Switch manufacturer that designed the protocol being described.

Protocol ID: This field shall contain a value identifying the protocol. If the value of this field is in the range 8000h to FFFFh, then this field combined with the Vendor ID String field specifies a vendor specific protocol. If the value of this field is in the range 0000h - 7FFFh then a non-vendor specific protocol is specified. Values for this field are summarized in table 71.

NOTE 17 – The term “Protocol” refers to all messages and methods from the completion of the ELP process through to the completion of path selection as well as any other messages and methods required to create a fully functional Fabric.

Table 71 – Protocol ID Values

Value	Use
0000h	Reserved
0001h	Obsoleted in FC-SW-4
0002h	FSPF Protocol
0003h	Virtual Fabrics Supported
0004h - 7FFFh	Reserved
8000h - FFFFh	Vendor Specific (see note)
Note: Vendor Specific values are only meaningful when combined with the Vendor ID String field.	

Reply Switch Fabric Internal Link Service Sequence:

Service Reject (SW_RJT)
  Signifies the rejection of the ESC request
Accept (SW_ACC)
  Signifies acceptance of the ESC request.
– Accept Payload

Payload: The format of the ESC accept Payload is shown in table 72.

Table 72 – ESC Accept Payload(Part 1 of 2)

Item	Size Bytes
Command Code = 02h	1
Reserved	1
Payload Length	2

Table 72 – ESC Accept Payload(Part 2 of 2)

Item	Size Bytes
Vendor ID String	8
Accepted Protocol Descriptor #1	12
...	
Accepted Protocol Descriptor #n	12

Payload Length: If the ESC Accept contains only one Accepted Protocol Descriptor, this field shall be set to zero. If the Multiple Protocol Descriptors bit is set to one in the ESC Request and the ESC Accept contains two or more Accepted Protocol Descriptors, then this field specifies the total length in bytes of the Payload. The least significant two bits shall be zero. The value specified shall be greater than or equal to 36, and less than or equal to 65532. If the Multiple Protocol Descriptors bit is set to zero in the ESC Request, the ESC Accept shall contain only one Accepted Protocol Descriptor.

Vendor ID String: This field shall contain a T10 Vendor ID of the responding Switch. This field shall contain either an identifier for the Switch manufacturer, or an OEM identifier.

Accepted Protocol Descriptor: This field shall contain a Protocol Descriptor chosen by the responding Switch. This Protocol Descriptor shall be chosen from the list presented in the ESC Request. The format of this field is as shown in table 70.

6.1.22 Exchange Switch Support (ESS)

The Exchange Switch Support (ESS) SW_ILS defines a mechanism for two switches to exchange vendor and support information relative to various supported features within the Fabric services and switch link services payloads.

Exchange Switch Support requests are addressed from the Domain Controller of a requesting Switch to the Domain Controller of a responding Switch.

Path selection shall complete before an ESS request may be issued to a destination Switch.

Protocol:

Exchange Switch Support (ESS) request Sequence
Accept (SW_ACC) Reply Sequence

Addressing: For use in determining switch support of Fabric services and SW_ILS support, the S_ID shall be set to FFFCxxh, indicating the Domain Controller of the originating Switch. The D_ID shall be set to FFFCyyh indicating the Domain Controller of the Destination Switch.

6.1.22.1 ESS Request Payload

The format of the ESS request payload is shown in table 73.

Table 73 – ESS Request Payload

Item	Size (Bytes)
31000000h	4
Revision	4
Payload Length	4
Interconnect Element Information Object	256
Number of Capability Objects	2
Reserved	2
Capability Object #1	Length of Object
Capability Object #2	Length of Object
...	
Capability Object #n	Length of Object

Revision: The revision field shall contain a value of 01h.

Payload Length: The length shall specify the number of bytes in the ESS Request payload. This value does not include the request code, revision and payload length bytes. This value shall be a multiple of 4.

Number of Capability Objects: Number of capability objects contained in the payload.

6.1.22.2 Interconnect Element Information Object

The Interconnect Element Information object contains vendor name, model name and number, release code and vendor specific information related to the Switch. The Interconnect Element object shall be supported.

The format of the Interconnect Element Information is described in FC-GS-7.

6.1.22.3 Capability Object

The capability object is used to convey levels of support for FC-SW-6 and FC-GS-7 functionality. The format of the Capability Object is shown in table 74.

Table 74 – Capability Object Format

Item	Size (Bytes)
Well-Known Address Type	1
Well-Known Address Subtype	1
reserved	1
Number of Capability Entries	1
Capability Entry #1	8
Capability Entry #2	8
...	
Capability Entry #n	8

Well-Known Address Type: The Well Known Address (WKA) type represents the type of service that the capability object represents. Allowed values are specified in FC-GS-7.

Well-Known Address Subtype: This field specifies a sub-service type for the specific service.

Number of Capability Entries: Number of capability entries within this capability object.

Capability Entry: Each Capability Entry shall be eight bytes in length and contain information specific to a particular service.

6.1.22.4 Service Specific Capability Formats

6.1.22.4.1 Directory Server Capability

The Well-known Address Type shall be set to FCh and the Well-known Address Subtype shall be set to 02h

Table 75 defines the bit definition for identifying specific support for the Name server subtype of the Directory server.

Table 75 – Name Server Capability Flags

Item	Size (Bytes)
Name Server Support Flags	4
NS Vendor Specific Support Flags	4

The format of the Name Server Support Flags field is as follows:

Bit 0- Name Server Entry Object 00h Support - When set indicates that the name server instance may accept large name server objects.

Bit 1- Name Server Entry Object 01h Support - When set indicates that the name server instance may accept small name server objects.

Bit 2- Name Server Entry Object 02h Support - When set indicates that the name server instance may accept Large + FC-4 Features name server objects.

Bit 3- Name Server Entry Object 03h Support - When set indicates that the name server instance may accept Small + FC-4 Features name server objects (see 9.3.3).

Bit 4 - GE_PT Zero Length Accept **-** When set indicates that the name server may support receipt of a 0 length ACCept payload from an interswitch GE_PT (or other GE_*) query.

Bits 5-31 reserved.

The format of the Name Server Vendor Specific Support Flags field is vendor specific and dependent on the Vendor Name.

6.1.22.4.2 Fabric Controller Capability

The Well-known Address Type shall be set to FDh and the Well-known Address Subtype shall be set to 00h.

Table 76 defines the bit definition for identifying specific support for the Fabric Controller.

Table 76 – Fabric Controller Capability Flags

Item	Size (Bytes)
Fabric Controller Support Flags	4
Fabric Controller Vendor Specific Support Flags	4

The format of the Fabric Controller Support Flags field is as follows:

Bit 0- SW_RSCN Support - When set indicates that the transmitting Fabric Controller supports receiving the SW_RSCN Request.

Bits 1-31 Reserved.

The format of the Fabric Controller Vendor Specific Support Flags field is vendor specific and dependent on the Vendor Name.

6.1.22.4.3 ESS Fabric Configuration Server Capability Object

The WKA Type shall be set to FAh and the WKA Subtype shall be set to 01h.

Table 77 depicts the bit definitions for identifying specific support for the Fabric Configuration server subtype of the Management server.

Table 77 – Fabric Configuration Server Capability flags

Item	Size (Bytes)
Fabric Configuration Server support flags	4
Reserved	4

The format of the Fabric Configuration Server support flags field is as follows:

Bit 0- Basic Configuration Services - When this bit is one, the Switch supports commands that are members of the Basic Configuration Service class (see table 150). When this bit is zero, the Switch does not support commands that are members of the Basic Configuration Service class.

Bit 1- Platform Configuration Services - When this bit is one, the Switch supports commands that are members of the Platform Configuration Service class (see table 150). When this bit is zero, the Switch does not support commands that are members of the Platform Configuration Service class.

Bit 2- Topology Discovery Configuration Services - When this bit is one, the Switch supports commands that are members of the Topology Discovery Configuration Service class (see table 150). When this bit is zero, the Switch does not support commands that are members of the Topology Discovery Configuration Service class.

Bit 3- Enhanced Configuration Services - When this bit is one, the Switch supports commands that are members of the Enhanced Configuration Service class (see table 150). When this bit is zero, the Switch does not support commands that are members of the Enhanced Configuration Service class.

Bits 4-31 reserved.

6.1.22.4.4 ESS Enhanced Zone Server Capability Object

The WKA Type shall be set to FAh and the WKA Subtype shall be set to 03h.

Table 78 depicts the bit definitions for identifying specific support for the Zone server subtype of the Management server.

Table 78 – Enhanced Zone Server Capability flags

Item	Size (Bytes)
Switch Enhanced Zoning support flags	4
Reserved	4

The format of the Switch Enhanced Zoning Server Support Flags field is as follows:

Bit 0- Enhanced Zoning supported - When this bit is one, the Switch is able to work in Enhanced Zoning mode. When this bit is zero, the Switch is not able to work in Enhanced Zoning mode.

Bit 1- Enhanced Zoning enabled - When this bit is one, the Switch is working in Enhanced Zoning mode. When this bit is zero, the Switch is working in Basic Zoning mode.

Bit 2- Merge Control Setting - When this bit is one, this Switch is working in Restrict mode, so it may join a Fabric only if the Fabric's Zoning Database is equal to its Zoning Database. When this bit is zero, this Switch is working in Allow mode, so it may join a Fabric only if the Fabric's Zoning Database is mergeable with its Zoning Database.

Bit 3- Default Zone Setting - When this bit is one this Switch denies traffic between members of the Default Zone. When this bit is zero this Switch permit traffic between members of the Default Zone.

Bit 4- Zone Set Database supported - When this bit is one, the Zone Server on this Switch is able to maintain a Zone Set Database. When this bit is zero, the Zone Server on this Switch is not able to maintain a Zone Set Database.

Bit 5- Zone Set Database enabled - When this bit is one, the Zone Server on this Switch is maintaining a Zone Set Database. When this bit is zero, the Zone Server on this Switch is not maintaining a Zone Set Database.

Bit 6- Activate Direct command supported **-** When this bit is one, this Switch supports the Activate Direct command. When this bit is zero, this Switch does not support the Activate Direct command.

Bit 7- Hard Zoning supported - When this bit is one, this Switch supports Hard Zoning. When this bit is zero, this Switch does not support Hard Zoning.

Bit 8- FC-SP Zoning supported (see FC-SP).

Bit 9- FC-SP Zoning enabled (see FC-SP).

Bits 10-31 reserved.

6.1.22.4.5 Security Policy Server Capability Object

The Security Policy Server Capability Object allows a Switch to discover the level of support for individual Policy Object types provided by other Switches of a Fabric. See FC-SP for the definition of the Security Policy Server Capability Object.

6.1.22.4.6 ESS-Vendor Specific Capability Object

The general format of the Vendor Specific Capability object closely follows the format of the Capability object currently defined for ESS. The format of the Vendor Specific Capability object is depicted below:

Table 79 – Vendor Specific Capability Object

Item	Size Bytes
Well-Known Address Type	1
Well-Known Address Subtype	1
Reserved	1
Length (n)	1
T10 Vendor ID	8
Vendor Specific Information	n*8

Well-Known Address Type: The Well-Known Address Type field shall be set to E0h to indicate that the Capability Object is a Vendor Specific Type Capability object.

Well-Known Address Subtype: The Well-Known Address Subtype field shall contain a value specified by the vendor.

Length: The Length field shall contain a value between 01h and FFh to indicate the number of doublewords of vendor specific information contained in the capability object. The T10 Vendor ID field is included in the doubleword count.

T10 Vendor ID: The T10 Vendor ID field shall contain the vendor's eight byte T10 Vendor ID.

Vendor Specific Information: The Vendor Specific Information field contains the vendor's information. The format of the information is defined by the vendor and not by this standard. When the vendor specific information does not align on a doubleword boundary the information is padded with nulls (00h) to the right to complete the final doubleword.

6.1.22.4.7 Domain Controller Capability Object

The Well-known Address Type shall be set to ECh and the Well-known Address Subtype shall be set to 00h. The Domain Controller Capability Object is specified in table 80.

Table 80 – Domain Controller Capability Object

Item	Size (Bytes)
Receive Data Field Size	2
End-to-End Credit	2
Concurrent Sequences	2
Open Sequences per Exchange	2

Receive Data Field Size: This field shall specify the largest Data Field size in bytes for a frame that may be received by the Domain Controller supplying the Parameters as a Sequence Recipient for a Class F frame. Values less than 256 or greater than 2112 are invalid. Values shall be a multiple of four bytes.

End-to-End Credit: End-to-end credit is the maximum number of Class F Data frames that may be transmitted by a Domain Controller without receipt of accompanying ACK or Link_Response frames. The minimum value of end-to-end credit is one. The End-to-End Credit field specified is associated with the number of buffers available for holding the Data_Field of a Class F frame and processing the contents of that Data_Field by the Domain Controller supplying the Parameters. Bit 15 of this field shall be set to zero. A value of zero for this field is reserved.

Concurrent Sequences: This field shall specify the number of Sequence Status Blocks provided by the Domain Controller supplying the Parameters for tracking the progress of a Sequence as a Sequence Recipient. The maximum number of Concurrent Sequences that may be specified is 255. A value of zero in this field is reserved. In Class F, the value of SEQ_ID shall range from 0 to 255, independent of the value in this field. A Domain Controller is allowed to respond with P_BSY to a frame initiating a new Sequence if Domain Controller resources are not available.

Open Sequences per Exchange: The value of the Open Sequences per Exchange shall specify the maximum number of Sequences that may be Open at one time at the Recipient between a pair of Domain Controllers for one Exchange. This value plus two shall specify the number of instances of Sequence Status that shall be maintained by the Recipient for a single Exchange in the Exchange Status Block. This value is used for Exchange and Sequence tracking. The value in this field limits the link facility resources required for error detection and recovery.

6.1.22.4.8 Event Server Capability

The Well-known Address Type shall be set to F4h and the Well-known Address Subtype shall be set to 01h. The bit definitions for identifying specific support for the Event Server are defined in table 81.

Table 81 – Event Server Capability Flags

Item	Size (Bytes)
Event Server Flags	4
Event Server Vendor Specific Support Flags	4

The format of the Event Server Support Flags is as follows:

Bit 0- When set indicates that the Event Server instance is supported.

The format of the Event Server Vendor Specific Flags field is vendor specific and dependent on the Vendor Name.

6.1.22.4.9 Switch Support Capability Object

The Well-known Address Type shall be set to 20h and the Well-known Address Subtype shall be set to 01h. The Switch Support Capability Object is specified in table 82.

Table 82 – Switch Support Capability Object

Item	Size (Bytes)
Switch Support Flags	4
Reserved	4

The format of the Switch Support flags fields is as follows:

Bit 0- Encapsulated F_RJT/F_BSY - When this bit is set to one, the Switch supports the encapsulated Class 2/F F_RJT and F_BSY frame format (see 15).

Bits 1-31 - reserved.

6.1.22.5 ESS Accept Payload

The format of the accept payload is shown in table 83.

Table 83 – ESS Accept Payload

Item	Size (Bytes)
02000000h	4
Revision	4
Payload Length	4
Interconnect Element Information Object	256
Number of Capability Objects	2
Reserved	2
Capability Object #1	Length of Object
Capability Object #2	Length of Object
...	
Capability Object #n	Length of Object

6.1.23 Merge Request Resource Allocation (MRRA)

The Merge Request Resource Allocation (MRRA) SW_ILS defines a mechanism for switches to request resources to be allocated for the transfer of a Merge Request SW_ILS. MRRA enables buffer management in the Fabric Controller.

MRRA SW_ILSs are addressed from the Fabric Controller of a requesting Switch to the Fabric Controller of a responding Switch.

Protocol:

Merge Request Resource Allocation (MRRA) request Sequence
Accept (SW_ACC) Reply Sequence

Addressing: For use in determining resource availability, the S_ID shall be set to FFFFFDh, indicating the Fabric Controller of the originating Switch. The D_ID shall be set to FFFFFDh, indicating the Fabric Controller of the Destination Switch.

The format of the MRRA request payload is shown in table 84.

Table 84 – MRRA Request Payload

Item	Size (Bytes)
34000000h	4
Revision	4
Merge Request Size	4
Vendor Specific	16

Revision: Shall be set to 00000001h.

Merge Request Size: The Merge Request Size is the number of words in the entire Merge Request SW_ILS that is subsequently sent to the adjacent Switch.

Vendor Specific: The format of the Vendor Specific field is depicted in table 85.

Table 85 – Vendor Specific Field

Item	Size (Bytes)
Vendor ID	8
Vendor Specific Information	8

Vendor ID: This field contains the T10 Vendor ID of the vendor that defines the content of the Vendor Specific Information field.

Vendor Specific Information: The format of this field is specific to the vendor.

Reply Merge Request Resource Allocation Sequence:

Service Reject (SW_RJT)
Signifies the rejection of the MRRA request
Accept (SW_ACC)
Signifies the acceptance of the MRRA request
- Accept Payload

Payload: The format of the MRRA Accept Payload is shown in table 86.

Table 86 – MRRA Response Payload

Item	Size (Bytes)
02000000h	4
Vendor ID	8
MRRA Response	4
Maximum Resources Available	4
Waiting Period	4

Vendor ID: This field contains the T10 Vendor ID of the vendor that defines the content of the vendor specific fields in the Merge Response field.

MRRA Response: This field shall specify the response to the MRRA request. The values are defined in table 87.

Table 87 – MRRA Response Values

Value (Hex)	Description
0	Reserved
1	Requested resources available
2	Requested resources are not available
E0-FF	Vendor Specific
Others	Reserved

Maximum Resources Available: This field specifies the maximum size in words in the entire Merge Request SW_ILS that the Fabric Controller is able to accept.

Waiting Period: This field specifies the time in seconds that the requesting port should wait before retrying the MRRA request. The Waiting Period should be less than R_A_TOV. This value is only meaningful when the Merge Response Value is set to 2.

6.1.24 Switch Trace Route (STR)

6.1.24.1 Basic Function

The STR Request initiates the STR operation to find the route between two Nx_Ports (e.g., Source Port and Destination Port) in a common Zone. After receiving a FC Trace Route (FTR) Request (see FC-GS-7), the Managing Switch shall send a STR Request to the Domain Controller of the Switch to which the Source Port is attached.

The Source Port's Switch shall append its Path Information to the request, increment the number or Path Information Entries, and forward the STR request to the next switch in the Fabric ingress path (i.e., From Source Port to Destination Port). Each Switch in the Fabric ingress path shall append its

Path Information to the STR request, increment the Number of Path Information Entries, and forward the STR request to the next switch in the path.

The Destination Port's Switch shall append the Path Information for the Fabric ingress path and the Fabric egress path (i.e., from the Destination Port to the Source Port) and change the Fabric Egress Path bit in the Flags field. The Fabric Egress Path bit shall inform switches as to the direction that the STR request is traveling. When the Source Port's Switch receives the STR request for the second time, it shall append its Path Information to Source Port, increment the Number of Path Information Entries, and forward the frame to the Fabric Configuration Server of the Entry Switch.

If the two N_Ports are not in a common Zone, the STR Request shall be rejected with the SW_RJT Reason Code of "Unable to Perform Command Request" and a SW_RJT Reason Code Explanation of "Invalid Operation".

If a Switch in the path rejects or does not Accept the STR Request, the last Switch to append its information shall set the STR Reject Reason Codes in the payload and send the STR Request to the Domain Controller ID of the Managing Switch. The Switch shall make a best effort to return the STR Request to the Managing Switch.

Protocol:

Switch Trace Route (STR) request Sequence
Accept (SW_ACC) Reply Sequence

Addressing: The S_ID shall be set to FFFCxxh which designates the Domain Controller ID of the Switch that generates the STR. The D_ID shall be set to FFFCyyh to designate the Domain Controller ID of the recipient Switch. The Switch shall forward the STR Request on the egress port that was entered in the Path Information Entry to ensure that the data path is operational.

Payload: The format of the STR Request is shown in table 88

Table 88 – STR Request Payload (Part 1 of 2)

Item	Size (Bytes)
35010000h	4
Revision	4
Source Port Tag	2
Source Port Length	2
Source Port Value	n
Destination Port Tag	2
Destination Port Length	2
Destination Port Value	n
Token	4
T10 Vendor ID	8
Vendor Specific Information	8
Flags	4
Remaining Hop Count	4

Table 88 – STR Request Payload (Part 2 of 2)

Item	Size (Bytes)
STR Reject Reason Code	4
Managing Switch's Domain Controller Address	1
Requesting Port's N_Port_ID	3
Number or Path Information Entries	1
Reserved	3
Source Port's Fabric Ingress Path Information	36
Intermediate Switch's Path Information	36
...	36
Destination Port's Fabric Ingress Path Information	36
Destination Port's Fabric Egress Path Information	36
...	36
Source Port's Fabric Egress Path Information	36

Revision: The revision shall contain a value of 01h.

Source Port Tag: The tag used to identify Source Port as shown in table 89.

Table 89 – Nx_Port Tags

Tag (hex)	Item
01	N_Port_ID
02	Nx_Port Name_Identifier

Source Port Length: The length of the Source Port Value in bytes. The length shall be a multiple of four.

Source Port Value: The value of Source Port. Fill bytes are added as necessary to the end of the actual value in order to ensure that the length of the value field is a multiple of four. Fill bytes shall be nulls (00h). The number of fill bytes (f) is zero, one, two or three depending on the length of the actual value (m). The total length of the value field is (n= f + m).

Destination Port Tag: The tag used to identify Destination Port as shown in Table 89.

Destination Port Length: The length of the Destination Port Value in bytes. The length shall be a multiple of four.

Destination Port Value: The value of Destination Port. Fill bytes are added as necessary to the end of the actual value in order to ensure that the length of the value field is a multiple of four. Fill bytes shall be nulls (00h). The number of fill bytes (f) is zero, one, two or three depending on the length of the actual value (m). The total length of the value field is (n= f + m).

Token: An identifier for the FTR Request that is provided by the requesting Nx_Port.

T10 Vendor ID: Contains the T10 Vendor ID of the vendor that defines the content of the Vendor Specific Information field.

Vendor Specific Information: The Vendor Specific Information field shall contain the vendor's information. The format of the information is defined by the vendor and not by this standard.

Flags: The Flags Field has 32 bits that are defined in table 90.

Table 90 – Flags Field Values

Bit	Description
0	Fabric Egress Path - Value b'0' - The STR Request is in the Fabric Ingress Path. Value b'1' - The STR Request is in the Fabric Egress Path.
1	Trace Complete Bit Value b'0' - The STR Request is still tracing the route. Value b'1' - The STR Request has completed tracing the route.
28-32	Vendor Specific Information
Others	Reserved

Remaining Hop Count: The Remaining Hop Count is the Maximum Hop Count specified by the user minus the number of hops that the STR Request has traveled between the Source and Destination Port and back to the Source Port. Starting with the Source Port's Switch, each Switch in the path decrements the Remaining Hop Count by 1. If the Remaining Hop Count is decremented to 0, then the Switch shall fill in its path routing information and forward the STR Request to the Entry Switch's Domain Controller and include the STR Reject Reason Code of "Reached Maximum Hop Count".

STR Reject Reason Code: If the STR Request is completed successfully, the STR Reject Reason Code is set to 00h to signify a successful completion. If the STR request is not acceptable by the next Switch in the path, then the Switch with the STR Payload shall send the STR Request to the Entry Switch's Domain Controller. The STR Payload shall set one of the following STR Reject codes.

Table 91 – STR Reject Reason Code Values

Value (hex)	Item
00	Command Completed Successfully
01	Command Not Supported in Next Switch
02	No Response from Next Switch
03	Maximum Hop Count Reached
04	Source Port not in Fabric
05	Destination Port not in Fabric
06	Devices not in Common Zone
07	No Route Between Designated Ports
08	No Additional Explanation
09	Fabric Busy
0A	Fabric Build in Progess

Table 91 – STR Reject Reason Code Values

Value (hex)	Item
F0-FF	Vendor Specific Error Codes
Others	Reserved

Managing Switch's Domain Controller Address: The Domain ID for the Domain Controller of the Managing Switch that received the FTR Request. This address may be the N_Port_ID for an Nx_Port that is the Fabric Configuration Server of the Fabric.

Requesting Port's N_Port_ID: The N_Port_ID of the requesting Port is needed so that the Token will not be mistaken for another N_Port_ID's duplicate token.

Number of Path Information Entries: The number or Path Information Entries in the Request. As each switch appends its path routing information onto the STR payload, it shall increment the number of path information entries by one. The Destination Port's switch shall increment the count by two since it shall append two entries.

Source Port's Fabric Ingress Path Information: The Path Information for Source Port's Switch in the Fabric ingress path. The format for the Path Information is shown in table 92

Table 92 – Path Information

Item	Size (Bytes)
Switch Name	8
Domain_ID	4
Ingress Port_Name	8
Ingress Physical Port Number	4
Egress Port_Name	8
Egress Physical Port Number	4

Switch Name: The Switch Name of the Switch in the path that is appending the path information.

Domain_ID: The Domain_ID of the Switch reporting the Path Information. The format of Domain_ID shall be set to 000000h||Domain_ID'

Ingress Port_Name: The Port_Name of the F_Port or E_Port on the Switch that the frame enters.

Ingress Physical_Port_Number: The Physical_Port_Number of the F_Port or E_Port on the Switch that the frame enters.

Egress Port_Name: The Port_Name of the F_Port or E_Port on the Switch that the frame exits.

Egress Physical_Port_Number: The Physical_Port_Number of the F_Port or E_Port on the Switch that the frame exits.

Intermediate Switch's Path Information: The Path Information for the second Switch in the Fabric ingress path (if any).

Destination Port's Fabric Ingress Route Information: The Path Information for Destination Port's Switch in the Fabric ingress path.

Destination Port's Fabric Egress Route Information: The Path Information for Destination Port's Switch in the Fabric egress path.

Source Port's Fabric Egress Route Information: The Path Information for Source Port's Switch in the Fabric egress path.

Reply Switch Fabric Internal Link Service Sequence:

Service Reject (SW_RJT)
Signifies rejection of the STR request
Accept (SW_ACC)
Signifies acceptance of the STR request
-Accept Payload

Payload: The format of the STR Accept payload is depicted in table 93.

Table 93 – STR Accept Payload

Item	Size (Bytes)
02000000h	4

6.1.25 Exchange Virtual Fabrics Parameters (EVFP)

6.1.25.1 Basic Function

The Exchange Virtual Fabrics Parameters (EVFP) SW_ILS provides support for Virtual Fabrics (see clause 12).

Protocol: Exchange Virtual Fabrics Parameters (EVFP) Request Sequence

Addressing: The S_ID field shall be set to FFFFFDh, indicating the Fabric Controller of the originating Switch. The D_ID field shall be set to FFFFFDh, indicating the Fabric Controller of the destination Switch.

Payload: Two types of EVFP messages are defined. All EVFP Request messages share the same message structure, shown in table 94.

Table 94 – EVFP Request Payload

Item	Size (Bytes)
3600 0000h	4
Protocol Version	1
EVFP Message Code	1
Transaction Identifier	2
Core Switch_Name	8
Reserved	2
Message Payload Length	2
Message Payload	variable

Protocol Version: This field denotes the version of the EVFP protocol. A value of 01h shall be used to indicate the version specified in this standard. If a Fabric Controller receives an EVFP Request containing a Version field value that is higher than its supported value, the Fabric Controller shall respond with its highest supported Revision field value. The Fabric Controller is responsible for detecting and handling any incompatibility issues that may occur. If a Fabric Controller receives an EVFP Request containing a Version field value that is equal to or lower than its supported value, the Fabric Controller shall respond with the Version field value received in the EVFP Request.

EVFP Message Code: Specifies the EVFP message that is to be transmitted from the source to the destination. The defined EVFP message codes are shown in table 95.

Table 95 – EVFP Message Codes

Value	Description	Reference
01h	EVFP_SYNC	6.1.25.2
02h	EVFP_COMMIT	6.1.25.3
all others	Reserved	

Transaction Identifier: Uniquely identifies an EVFP transaction between two entities. The Transaction Identifier shall be set by the EVFP Initiator, and each subsequent EVFP message shall contain the same value, until the EVFP transaction is completed.

NOTE 18 – The usage of the Transaction Identifier is very similar to the usage of an OX_ID when an Exchange Originator is enforcing uniqueness via the OX_ID mechanism (see FC-FS-3), but it is not related in any way to the OX_ID present in the Fibre Channel frames carrying the EVFP messages.

Core Switch_Name: Core Switch_Name of the originating Switch.

Payload Length: Shall be set to the total length in bytes of the EVFP Payload (i.e., 20 + the Message Payload length).

Reply Switch Fabric Internal Link Service Sequence:

Service Reject (SW_RJT)
Signifies rejection of the EVFP request
Accept (SW_ACC)
Signifies acceptance of the EVFP request
-Accept Payload

Accept Payload: All EVFP Accept messages share the same message structure, shown in table 96. With the exception of the first four bytes, the fields in table 96 are the same as the fields in table 94.

Table 96 – EVFP Accept Payload

Item	Size (Bytes)
0200 0000h	4
Protocol Version	1
EVFP Message Code	1
Transaction_ID	2
Core Switch_Name	8
Reserved	2
Message Payload Length	2
Message Payload	variable

Table 97 shows the use of Reason Codes and Reason Code Explanations under some error conditions.

Table 97 – SW_RJT Reason Codes

Error Condition	Reason Code	Reason Code Explanation
EVFP SW_ILS not supported	Command Not Supported	No Additional Explanation
EVFP collision	Unable to perform command request	Command Already in Progress
EVFP Protocol Version not supported	Invalid Revision Level	No Additional Explanation
EVFP_COMMIT before EVFP_SYNC	Logical Error	No Additional Explanation
Insufficient Resources	Unable to Perform Command Request	Insufficient Resources Available
Invalid Payload Message	Protocol Error	No Additional Explanation

6.1.25.2 EVFP_SYNC Message Payload

6.1.25.2.1 Overview

The EVFP_SYNC Message Payload carries a list of descriptors. The format of the EVFP_SYNC Message Payload is shown in table 98. This Message Payload is used in both EVFP_SYNC Request and EVFP_SYNC Accept.

Table 98 – EVFP_SYNC Message Payload

Item	Size (Bytes)	Reference
Descriptor #1 = Tagging Administrative Status	8	6.1.25.2.2
Descriptor #2 = Port VF_ID	8	6.1.25.2.3
Descriptor #3 = Locally-Enabled VF_ID List	516	6.1.25.2.4
...		
Descriptor #m	variable	

All descriptors share the same format, as shown in table 99.

Table 99 – Descriptor Format

Item	Size (Bytes)
Descriptor Control	1
Descriptor Type	1
Descriptor Length	2
Descriptor Value	variable

Descriptor Control: Specifies the behavior of the receiving entity if the descriptor is unsupported. The defined codes are shown in table 100.

Table 100 – Descriptor Control Codes

Value	Description
01h	Critical. Abort the EVFP transaction if the descriptor is unsupported.[a]
02h	Non critical. Skip the descriptor if unsupported and continue the EVFP transaction.[a]
all others	Reserved

[a] The Descriptor Control provides extensibility to the protocol. An implementation supporting a subset of the descriptors is able to process the unknown ones as specified by the Descriptor Control value.

Descriptor Type: Specifies the type of the descriptor. The defined descriptors are summarized in table 101.

Table 101 – Descriptor Types

Value	Description	Reference
01h	Tagging Administrative Status Descriptor	6.1.25.2.2
02h	Port VF_ID Descriptor	6.1.25.2.3
03h	Allowed VF_ID List Descriptor	6.1.25.2.4
F0h .. FEh	Vendor Specific Descriptor	6.1.25.2.5
all others	Reserved	

Descriptor Length: Specifies the length in bytes of the Descriptor Value.

6.1.25.2.2 Tagging Administrative Status Descriptor

The format of the Tagging Administrative Status descriptor is shown in table 102.

Table 102 – Tagging Administrative Status Descriptor

Item	Size (Bytes)
Descriptor Control = 01h	1
Descriptor Type = 01h	1
Descriptor Length = 0004h	2
Administrative Tagging Mode	4

The defined Administrative Tagging Modes are shown in table 103.

Table 103 – Administrative Tagging Modes

Value	Notation	Description
0000 0001h	OFF	The Interconnect_Port shall not perform VFT Tagging
0000 0002h	ON	The Interconnect_Port may perform VFT Tagging if the peer does not prohibit it
0000 0003h	AUTO	The Interconnect_Port may perform VFT Tagging if the peer request it

In absence of any explicit configuration, the default Administrative Tagging Mode of a Switch Port of a VF capable Switch should be AUTO.

Table 104 shows how VFT tagging is negotiated between peer Interconnect_Ports.

Table 104 – Tagging Mode Negotiation

		Peer Tagging Mode		
		OFF	**ON**	**AUTO**
Local Tagging Mode	**OFF**	Non Tagging	Non Tagging	Non Tagging
	ON	Non Tagging	Tagging	Tagging
	AUTO	Non Tagging	Tagging	Non Tagging

6.1.25.2.3 Port VF_ID Descriptor

The format of the Port VF_ID descriptor is shown in table 105.

Table 105 – Port VF_ID Descriptor

Item	Size (Bytes)
Descriptor Control = 01h	1
Descriptor Type = 02h	1
Descriptor Length = 0004h	2
Port Flags	2
Port VF_ID	2

Port Flags: Reserved. Shall be set to zero.

Port VF_ID: The 12 least significant bit of this field shall be set to the Port VF_ID. The four most significant bits shall be set to zero. In absence of any explicit configuration, the value 001h should be used as Port VF_ID.

6.1.25.2.4 Locally-Enabled VF_ID List Descriptor

The format of the Locally-Enabled VF_ID List descriptor is shown in table 106.

Table 106 – Locally-Enabled VF_ID List Descriptor

Item	Size (Bytes)
Descriptor Control = 01h	1
Descriptor Type = 03h	1
Descriptor Length = 0200h	2
VF_ID Bitmap	512

VF_ID Bitmap: Each Virtual Fabric is identified by a bit in the VF_ID Bitmap. The high-order bit represents VF_ID zero, each successive bit represents the successive VF_ID, and the low-order bit represents VF_ID 4095. Virtual Fabric K is allowed on the Interconnect_Port if the Kth bit of the VF_ID Bitmap is set to one; is disallowed if the Kth bit of the VF_ID Bitmap is set to zero. The bit representing the Control VF_ID (see FC-FS-3) shall be set to zero.

The list of Virtual Fabrics operational over a link is computed by performing a bit-wise 'AND' between the received VF_ID Bitmap and the locally configured VF_ID Bitmap.

6.1.25.2.5 Vendor Specific Descriptor

The format of the Vendor Specific descriptor is shown in table 107.

Table 107 – Vendor Specific Descriptor

Bits Word	31 .. 24	23 .. 16	15 .. 08	07 .. 00
0	Descriptor Control	Descriptor Type	Descriptor Length	
1	T10 Vendor ID			
2				
3	Vendor Specific			
..				
N				

T10 Vendor ID: Shall be set to the Vendor's T10 Vendor ID.

6.1.25.3 EVFP_COMMIT Message Payload

Both EVFP_COMMIT Request and EVFP_COMMIT Accept have no Message Payload.

6.1.26 Enhanced Acquire Change Authorization Request (EACA)

The EACA request is sent from the Domain Controller of the Managing Switch to the Domain Controller of each Managed Switch contained in the ECS Switch List (see 6.1.26.1.4). The EACA request instructs the Managed Switch to reserve local resources associated with the designated application for the purpose of ensuring the consistency of the application's data.

The use if this ILS is detailed in clause 13.

Protocol:
Enhanced Acquire Change Authorization (EACA) request Sequence
Accept (SW_ACC) reply Sequence

Addressing: The S_ID shall be set to hex'FFFCxx', indicating the Domain Controller of the originating Switch. The D_ID shall be set to hex'FFFCyy', indicating the Domain Controller of the destination Switch.

Payload: The format of the EACA request payload is shown in table 108.

Table 108 – EACA Request Payload

Item	Size (Bytes)
2A01h	2
Reserved	2
Commit Exchange Preamble Length	4
Commit Exchange Preamble	n

Commit Exchange Preamble Length: This field specifies the length of the Commit Exchange Preamble in bytes.

6.1.26.1 Commit Exchange Preamble

The format of the Commit Exchange Preamble is described in table 109 below.

Table 109 – Commit Exchange Preamble

Item	Size (Bytes)
Transaction_Identifier	12
Number of Switch Identifiers (m)	1
Flags	1
Reserved	2
ECS Switch List	m*12

6.1.26.1.1 Transaction Identifier

The Transaction Identifier is used to uniquely identify a transaction in the Fabric. The scope of a Transaction Identifier is from the EACA request that begins the transaction to the ERCA request that ends

the transaction. This allows multiple applications to use the Enhanced Commit Service simultaneously. The format of the Transaction Identifier is depicted in table 110 below.

Table 110 – Transaction Identifier

Item	Size (Bytes)
Generation Count	4
Application_ID	1
Domain_ID	1
Reserved	2
Timestamp	4

Generation Count: The generation count contains a monotonically increasing value that is used to distinguish transactions related to the same application. When a Switch originates an EACA, it shall increment the previous generation count by 1. The generation count shall be 32-bit unsigned integers that shall wrap to zero on exceeding FFFF FFFFh.

Application_ID: The Application_ID contains a value that represents an application, service, or function in the Fabric. The Application ID values are shown in table 111 below:

Table 111 – Application ID Value

Value (hex)	Description
00	Vendor Specific
01	Fabric Policies (see FC-SP)
E0-EF	Vendor Specific
FF	Reserved for RFC 4936
other values	Reserved

Domain_ID: This field contains the Domain_ID of the originating Switch.

Timestamp: This field contains a timestamp that specifies the time that the request was sent from the originating Switch. The timestamp indicates in milli-seconds the elapsed time since the last system boot on the originating Switch.

Together the Generation Count, Application_ID, Domain_ID, and the Timestamp provide the means for a Fabric to distinguish all ECS requests. All fields of the Transaction Identifier are established by the Managing Switch when a transaction is initiated with an EACA. All ESFC, EUFC, and TCO requests within the transaction and the ERCA ending the transaction are identified by the Transaction Identifier established by the EACA request.

6.1.26.1.2 Number of Switch Identifiers

Specifies the number of Switch Identifiers in the ECS Switch List including the Managing Switch.

6.1.26.1.3 Flags

This field contains flag bits (7:0) that provide additional information about the specified transaction. The following flag bits are defined.

Bit 7, the Assisted/Autonomous Mode bit, shall indicate whether ECS is operating in assisted mode or in autonomous mode for the specified transaction. When the bit is set to one, ECS is operating in assisted mode for the specified transaction (see 13.2). When the bit is set to zero, ECS is operating in autonomous mode for the specified transaction (see 13.3).

Bits 6-0 shall be reserved.

6.1.26.1.4 ECS Switch List

The ECS Switch List contains a list of all Switches participating in the specified transaction. This list represents a subset of all Switches in the Fabric. The ECS Switch List contains authorized Switches and non-authorized Switches. Only the authorized Switches shall participate in the ECS error recovery processing. It is the responsibility of the initial Managing Switch to construct the ECS Switch List according to the requirements of the specified application.

The Managing Switch shall send ECS requests to Managed Switches in the order indicated in the ECS Switch list. The first Switch in the list shall be the initial Managing Switch, followed by Switches that are authorized, and then by Switches that are not authorized. The format of the ECS Switch List is shown in table 112 below:

Table 112 – ECS Switch List

Item	Size (Bytes)
Managing Switch Identifier	12
Switch Identifier 1	12
Switch Identifier 2	12
Switch Identifier...	12
Switch Identifier...	12
Switch Identifier m-1	12

Switch Identifier: The Switch Identifier contains the Domain_ID and the Switch_Name for a Switch in the Fabric. The format of the Switch Identifier is depicted in table 113 below.

Table 113 – Switch Identifier

Item	Size (Bytes)
Domain_ID	1
Flags	1
Reserved	2
Switch_Name	8

Flags: This field contains flag bits (7:0) that provide additional information about the designated Switch.

Bit 7, the Switch Authorized bit, shall indicate whether the Switch shall participate in ECS recovery processing if the ECS protocol is operating in autonomous mode. When the bit is set to one, the Switch is authorized and shall participate in ECS recovery processing. When the bit is set to zero, the Switch is not authorized and shall not participate in ECS recovery processing.

Bits 6-0 shall be reserved.

Reply Switch Fabric Internal Link Service Sequence:

Service Reject (SW_RJT)
  Signifies the rejection of the EACA request
Accept (SW_ACC)
  Signifies acceptance of the EACA request.

An SW_ACC indicates that the operation completed successfully.
SW_RJT shall be returned as a reply to signify the rejection of the EACA_ILS Request Sequence.

6.1.27 Enhanced Stage Fabric Configuration (ESFC) Request

An ESFC request is sent from the Domain Controller of the Managing Switch to the Domain Controller of each Managed Switch contained in the ECS Switch List (see 6.1.26.1.4). The ESFC request allows application specific data to be transported to all Managed Switches and signals each Switch to validate and stage the data locally. During the staging process the Managed Switch performs the necessary consistency checks to ensure that the operation will complete successfully with respect to that Switch.

The use if this ILS is detailed in clause 13.

Protocol:

Enhanced Stage Fabric Configuration (ESFC) request Sequence
Accept (SW_ACC) reply Sequence

Addressing: The S_ID shall be set to hex'FFFCxx', indicating the Domain Controller of the originating Switch. The D_ID shall be set to hex'FFFCyy', indicating the Domain Controller of the destination Switch.

Payload: The format of the ESFC request payload is shown in table 114.

Table 114 – ESFC Request Payload

Item	Size (Bytes)
2A02h	2
Reserved	2
Commit Exchange Preamble Length	4
Commit Exchange Preamble (see 6.1.26.1)	n
Application Data Length	4
Application Data	m

Commit Exchange Preamble Length: This field specifies the length of the Commit Exchange Preamble in bytes.

Application Data Length: The field contains the length of the application specific data in bytes.

Application Data: This field contains any application specific operations and the actual application data that is to be operated upon. An application shall define the format of its Application Data.

Reply Switch Fabric Internal Link Service Sequence:

Service Reject (SW_RJT)
Signifies the rejection of the ESFC request
Accept (SW_ACC)
Signifies acceptance of the ESFC request.

An SW_ACC indicates that the operation completed successfully.
An SW_RJT shall be returned as a reply to signify the rejection of the ESFC_ILS Request Sequence.

6.1.28 Enhanced Update Fabric Configuration (EUFC) Request

An EUFC request is sent from the Domain Controller of the Managing Switch to the Domain Controller of each Managed Switch contained in the ECS Switch List (see 6.1.26.1.4). The EUFC request is sent to each Managed Switch to request that the Managed Switch commit the changes specified by the application data contained in the proceeding ESFC request.

The use if this ILS is detailed in clause 13.

Protocol:

Enhanced Stage Fabric Configuration (EUFC) request Sequence
Accept (SW_ACC) reply Sequence

Addressing: The S_ID shall be set to hex'FFFCxx', indicating the Domain Controller of the originating Switch. The D_ID shall be set to hex'FFFCyy', indicating the Domain Controller of the destination Switch.

Payload: The format of the EUFC request payload is shown in table 115.

Table 115 – EUFC Request Payload

Item	Size (Bytes)
2A03h	2
Reserved	2
Commit Exchange Preamble Length	4
Commit Exchange Preamble (see 6.1.26.1)	n

Commit Exchange Preamble Length: This field specifies the length of the Commit Exchange Preamble in bytes.

Reply Switch Fabric Internal Link Service Sequence:

Service Reject (SW_RJT)
Signifies the rejection of the EUFC request
Accept (SW_ACC)
Signifies acceptance of the EUFC request.

An SW_ACC indicates that the operation completed successfully.
An SW_RJT shall be returned as a reply to signify the rejection of the EUFC_ILS Request Sequence.

6.1.29 Enhanced Release Change Authorization (ERCA) Request

The ERCA request is sent from the Domain Controller of a Switch in the ECS Switch List (see 6.1.26.1.4) to all other Switches contained in the ECS Switch List. Typically, ERCA requests are sent by the Managing Switch to all Managed Switches. ERCA requests are sent to Managed Switches in order to free the resources reserved by the previous EACA request and completes the specified transaction with respect to each Managed Switch.

NOTE 19 – To facilitate recovery and minimize the window for denial of service attacks, an ERCA may be sent by any Switch that is a member of the ECS Switch List.

The use if this ILS is detailed in clause 13.

Protocol:

Enhanced Release Change Authorization (ERCA) request Sequence
Accept (SW_ACC) reply Sequence

Addressing: The S_ID shall be set to hex'FFFCxx', indicating the Domain Controller of the originating Switch. The D_ID shall be set to hex'FFFCyy', indicating the Domain Controller of the destination Switch.

Payload: The format of the ERCA request payload is shown in table 116.

Table 116 – ERCA Request Payload

Item	Size (Bytes)
2A04h	2
Reserved	2
Commit Exchange Preamble Length	4
Commit Exchange Preamble (see 6.1.26.1)	n

Commit Exchange Preamble Length: This field specifies the length of the Commit Exchange Preamble in bytes.

Reply Switch Fabric Internal Link Service Sequence:

Service Reject (SW_RJT)
Signifies the rejection of the ERCA request
Accept (SW_ACC)
Signifies acceptance of the ERCA request.

An SW_ACC indicates that the operation completed successfully.
SW_RJT shall be returned as a reply to signify the rejection of the ERCA_ILS Request Sequence.

6.1.30 Transfer Commit Ownership (TCO) Request

The TCO request is sent from the Domain Controller of one Managing Switch to the Domain Controller of another Managing Switch. The TCO request transfers the role of Managing Switch to another Switch in the ECS Switch List (see 6.1.26.1.4). This provides a mechanism to ensure that only one Managing Switch is chosen to complete the transaction.

The use if this ILS is further detailed in clause 13.

Protocol:

Transfer Commit Ownership (TCO) request Sequence
Accept (SW_ACC) reply Sequence

Addressing: The S_ID shall be set to hex'FFFCxx', indicating the Domain Controller of the originating Switch. The D_ID shall be set to hex'FFFCyy', indicating the Domain Controller of the destination Switch.

Payload: The format of the TCO request payload is shown in table 117.

Table 117 – TCO Request Payload

Item	Size (Bytes)
2A05h	2
Reserved	2
Commit Exchange Preamble Length	4
Commit Exchange Preamble (see 6.1.26.1)	n

Commit Exchange Preamble Length: This field specifies the length of the Commit Exchange Preamble in bytes.

Reply Switch Fabric Internal Link Service Sequence:

Service Reject (SW_RJT)
Signifies the rejection of the TCO request
Accept (SW_ACC)
Signifies acceptance of the TCO request.

An SW_ACC indicates that the operation completed successfully.
SW_RJT shall be returned as a reply to signify the rejection of the TCO_ILS Request Sequence.

7 Fabric Configuration

7.1 Fabric Configuration Summary

The Fabric Configuration process enables a Switch Port to determine its operating mode, exchange operating parameters, and provides for distribution of addresses. This process is summarized in table 118.

Table 118 – Fabric Configuration Summary

Operation	Starting Condition	Process	Ending Condition
Establish Link Parameters and Switch Port operating mode	Switch Port has achieved word synchronization.	The Switch Port attempts to discover whether it is an FL_Port, an E_Port or an F_Port.	Switch Port mode is known. If a Port is an E_Port, Link Parameters have been exchanged and Credit has been initialized.
Select Principal Switch	BF or RCF SW_ILS transmitted or received.	Switch_Names are exchanged over all ISLs to select a Principal Switch, the Principal Switch becomes the Domain Address Manager.	The Principal Switch is selected.
Domain_ID Acquisition	Domain Address Manager has been selected.	Switch requests a Domain_ID from the Domain Address Manager.	Switch has a Domain_ID.
Zoning Merge	Switch has a Domain_ID.	Zoning data are exchanged over the E_Ports, following the merge protocol defined in clause 10.	The Zoning definitions are consistent across the E_Ports.
Path Selection	Switch has a Domain_ID.	Path Selection (FSPF) is defined in clause 8.	Switch is operational with routes established.

Domain_IDs may be assigned statically or dynamically. When Domain_IDs are assigned statically, the administrator shall configure a Domain_ID and a Fabric_Name on each Switch of the Fabric, and the operations described in 7.3 and 7.4 shall not be performed by a Switch. A configured Fabric_Name shall conform to the rules regarding Name_Identifiers specified in FC-FS-3. When Domain_IDs are assigned dynamically, the operations described in 7.3 and 7.4 shall be performed by a Switch.

NOTE 20 – An erroneous condition in which two Switches have been assigned the same Domain_ID may be detected when FSPF begins its operations.

Once path selection has completed, routes for Class N Frames are established and Class N Frames shall traverse the Fabric using established routes. Class N Frames shall continue to traverse the Fabric until an RCF clears the routes or a previously determined route is invalidated (e.g., Max_Age expires for an LSR, physical link is removed).

7.2 Switch Port Initialization

7.2.1 Basic Operation

Switch Ports shall initialize as described below. Figure shows the state machine of the process. If the state machine is different than the text, the state machine shall apply. A Switch Port that is running this state machine shall be capable of at least E_Port operation; either E/F/FL_Port, E/F_Port, E/FL_Port, or E_Port. Initialization of Switch Ports that are F/FL_Port, FL_Port, or F_Port is defined in FC-FS-3 and FC-AL-2. This state machine is also applicable to B_Port operation.

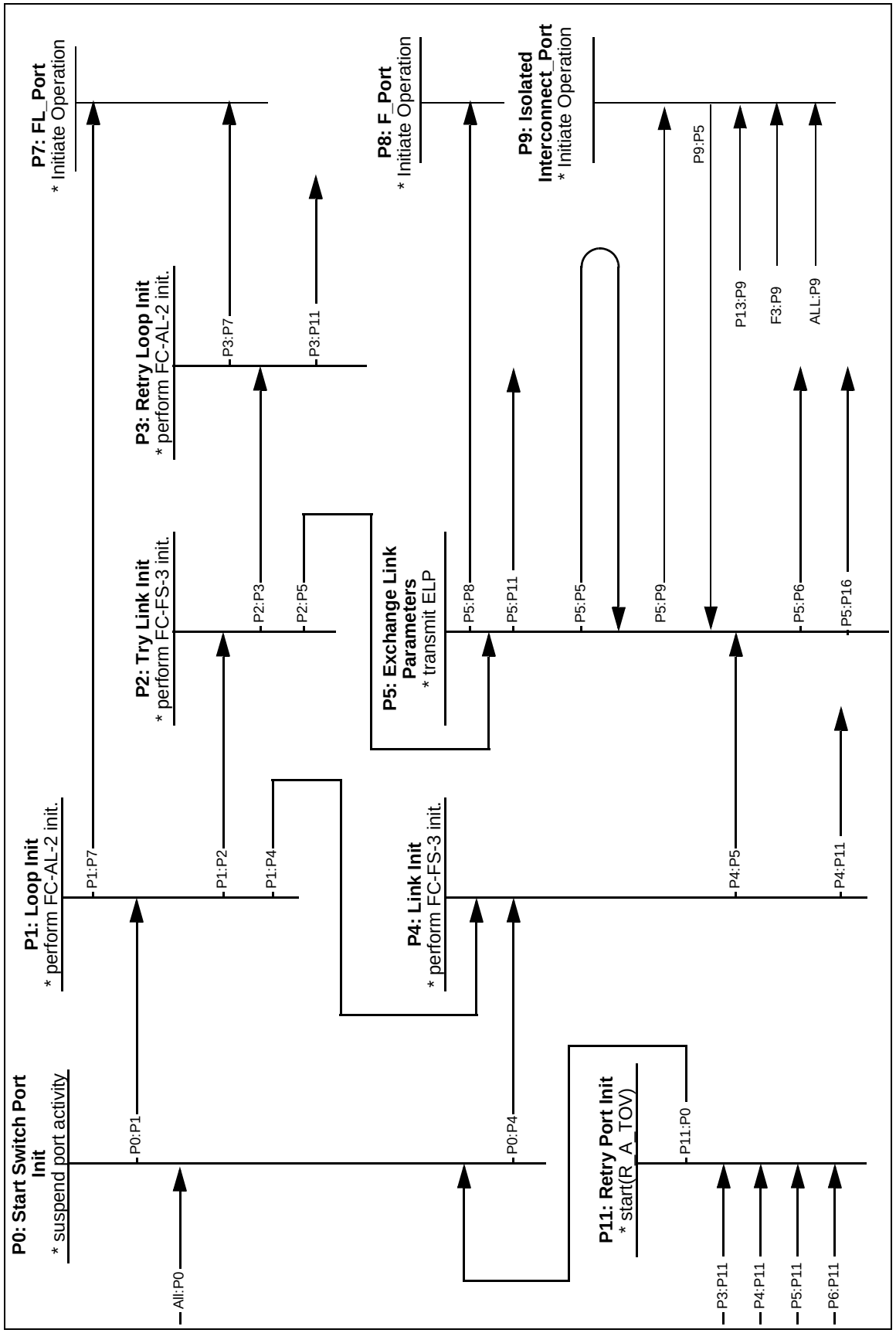

Figure 11 – Switch Port Mode Initialization State Machine

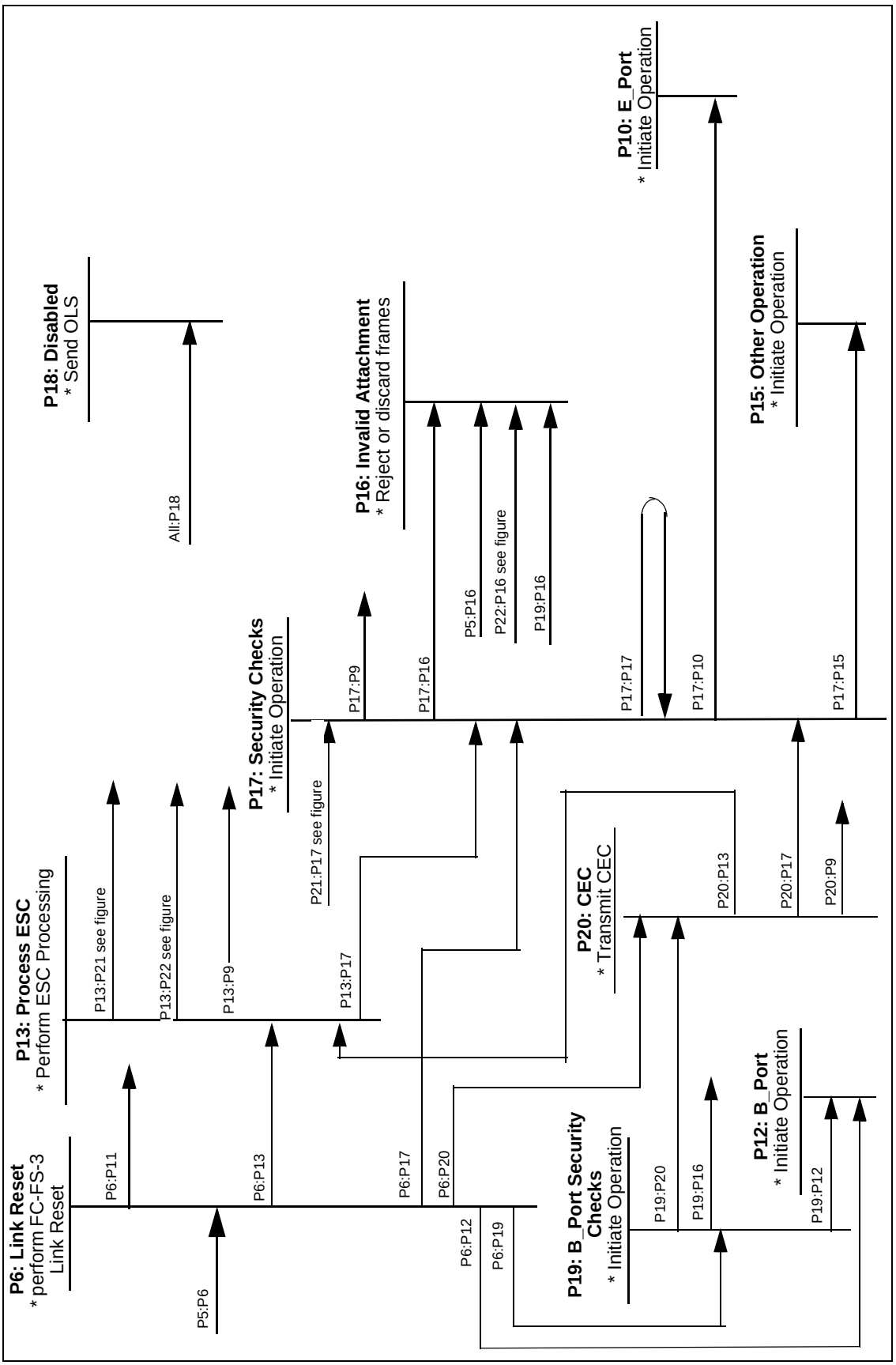

Figure 12 – Switch Port Mode Initialization State Machine - Continued

Transition All:P0. This transition occurs whenever an initialization event occurs in a state where it is not already handled. An Initialization Event may be:

a) a power-on reset condition; or,

b) receiving an initialization Primitive Sequence, such as OLS, NOS, LIP; or,

c) outside intervention requesting an initialization; or,

d) a transition to Link Offline, as defined in FC-FS-3; or,

e) a loss of word synchronization for greater than R_T_TOV; or,

f) a failure to successfully complete a prior initialization attempt, and the time-out period has expired.

NOTE 21 – LR is not considered an Initialization Event, but shall operate as specified in FC-FS-3.

State P0: Start Switch Port Initialization. This state marks the beginning of Switch Port initialization. All activity on the Switch Port is suspended until the Initialization is complete.

Transition P0:P1. The Switch Port is capable of becoming an FL_Port. Attempt Loop Initialization first (as defined in FC-AL-2).

Transition P0:P4. The Switch Port is not capable of becoming an FL_Port. Attempt Link Initialization.

State P1: Loop Initialization. An FL_Port-capable Switch Port attempts Loop Initialization (as defined in FC-AL-2).

Transition P1:P7. This transition occurs if the FL_Port transitions from the OPEN_INIT state to the MONITORING state, is in participating mode, and the resulting AL_PA bitmap generated during the LISA Loop Initialization Sequence indicates that one or more L_Port (other than the Switch Port) is attached. This transition also occurs if Switch Port is in non-participating mode.

Transition P1:P2. This transition occurs if the FL_Port transitions from the OPEN_INIT state to the MONITORING state, is in participating mode, and the resulting AL_PA bitmap generated during the LISA Loop Initialization Sequence indicates zero or one L_Port (other than the Switch Port) is attached; or, if the Loop Initialization procedure did not complete and OLS or NOS is received (see annex A).

Transition P1:P4. This transition occurs if the Loop Initialization does not complete successfully. This may occur if the Switch Port is attached to a non-L_Port capable port, so the next thing to try is a Link Initialization.

State P2: Try Link Initialization. The Switch Port is FL_Port-capable, is in participating mode, and has detected zero attached NL_Ports, then there is a possibility that the Switch Port is point-to-point attached to another FL_Port-capable Switch Port. In this case the Switch Port shall attempt Link Initialization by transmitting LIPs and, when receiving LIPs, OLSs for up to 2xAL_TIME until receiving NOS or LR (see annex A), and then complete Link Initialization as defined in FC-FS-3. Otherwise the Switch Port shall complete the Link Initialization protocol initiated by the other Switch Port.

Transition P2:P3. This transition occurs if the Link Initialization does not complete successfully.

Transition P2:P5. This transition occurs if the Link Initialization completes successfully.

State P3: Retry Loop Initialization. The Switch Port had detected that it may be able to operate point-to-point with another loop device, but the attempt to do so failed. In this case, the Switch Port shall then attempt to go back to loop operation by retrying Loop Initialization (as defined in FC-AL-2).

Transition P3:P7. This transition occurs if the Loop Initialization succeeds (the FL_Port transitions from the OPEN_INIT state to the MONITORING state and participating).

Transition P3:P11. This transition occurs if the Loop Initialization fails following a re-attempt of Loop Initialization.

State P4: Link Initialization. The Switch Port shall attempt Link Initialization as defined in FC-FS-3.

Transition P4:P5. This transition occurs if the Link Initialization procedure succeeds.

Transition P4:P11. This transition occurs if the Link Initialization procedure fails.

State P5: Exchange Link Parameters. The Switch Port shall originate an ELP SW_ILS request Sequence (see 6.1.4). Table 119 below defines the responses and actions to an ELP request for the originating Interconnect_Port**.**

Table 119 – Responses to ELP Request for Originating Interconnect_Port (Part 1 of 2)

Response to ELP	Indication	Originating Interconnect_Port Action
1. R_RDY	Request received at destination	Wait E_D_TOV+4 for response frame
2. ACK_1	Request received at destination	Wait E_D_TOV+4 for response frame
3. SW_ACC	Destination Interconnect_Port received and processed request	Send ACK_1, Transition (P5:P6)
4. F_BSY or P_BSY	Destination is busy	Retry[a], Transition (P5:P11)
5. F_RJT or P_RJT	The frame is not acceptable	Respond accordingly[c], Transition (P5:P11) if appropriate
6. ELP (rcvd Switch_Name > own Switch_Name)	Both Interconnect_Ports sent ELP at the same time	Send SW_ACC or SW_RJT based on the values of the ELP parameters, Transition (P5:P6) (see figure 13 for an example of this response)

[a] The retry is performed following a time-out period, as defined in P11 below.

[b] The Reason Code shall be “Unable to perform command request” with an Reason Explanation of “Command already in progress”.

[c] Response is defined in FC-FS-3.

[d] An SW_ACC is sent for the other ELP Exchange in progress, as described in Response #6, if the received ELP parameters are acceptable. If the received ELP parameters are not acceptable, then an SW_RJT is sent. See figure 13.

Table 119 – Responses to ELP Request for Originating Interconnect_Port (Part 2 of 2)

Response to ELP	Indication	Originating Interconnect_Port Action
7. ELP (rcvd Switch_Name < own Switch_Name)	Both Interconnect_Ports sent ELP at the same time	Send SW_RJT[b], (see figure 13 for an example of this response)
8. ELP (rcvd Switch_Name = own Switch_Name)	Interconnect_Port output is looped back to input	Remove loopback condition, Transition (P5:P9)
9. SW_RJT	Reason code/explanation: - Command already in progress[d] - Logical busy - other	(see figure 13 for an example of this response) - retry transition to P11[a], or P5 - respond accordingly, and transition to P11 if appropriate
10. FLOGI	Destination is a PN_Port	Respond accordingly[c], transition to P8
11. any other frame	Indeterminate	Discard frame and retry[a], transition to P11
12. E_D_TOV+4 expires	Destination is busy; or, ELP, SW_ACC, ACK_1 frame lost; or, destination is not an Interconnect_Port	Retry[a], transition to P11

[a] The retry is performed following a time-out period, as defined in P11 below.

[b] The Reason Code shall be “Unable to perform command request” with an Reason Explanation of “Command already in progress”.

[c] Response is defined in FC-FS-3.

[d] An SW_ACC is sent for the other ELP Exchange in progress, as described in Response #6, if the received ELP parameters are acceptable. If the received ELP parameters are not acceptable, then an SW_RJT is sent. See figure 13.

The originating Interconnect_Port shall consider the exchange of Link Parameters complete (but not necessarily successful) when it has received the SW_ACC or SW_RJT and has transmitted the ACK_1 for the SW_ACC or SW_RJT reply Sequence.

The responding Interconnect_Port shall consider the exchange of Link Parameters complete when it has received the ACK_1 for the SW_ACC or SW_RJT.

The exchange of Link Parameters shall be considered successful when the exchange of Link Parameters is complete, and the reply to the ELP is an SW_ACC, and both Interconnect_Ports agree that the parameters exchanged are acceptable.

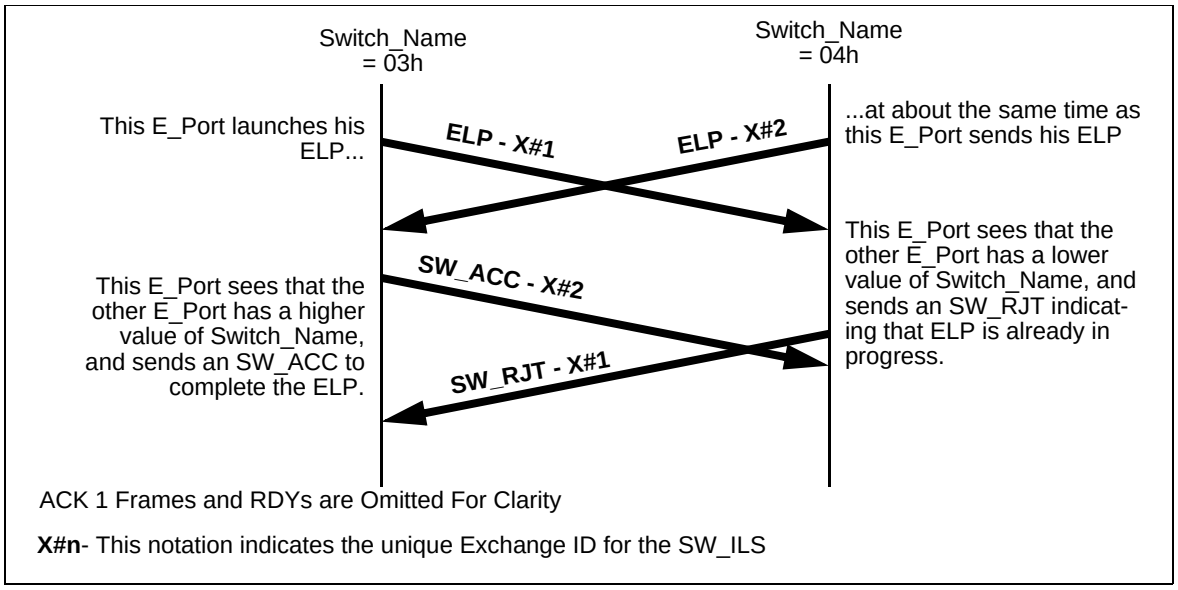

Figure 13 – Simultaneous ELP Processing- Parameters Acceptable to Both Switches

Transition P5:P5. This transition occurs if the originating Interconnect_Port does not agree that the parameters in the SW_ACC are acceptable, or it receives an SW_RJT indicating the parameters in the ELP request were not acceptable to the responding Interconnect_Port, and it is able to originate a new ELP request Sequence with modified parameters. This transition may also occur if an SW_RJT is received indicating a logical busy.

Transition P5:P6. This transition is taken by the originator of the ELP if the exchange of link parameters are complete.

Transition P5:P8. This transition occurs if the exchange of Link Parameters is unable to be completed, and FLOGI is received.

Transition P5:P9. This transition occurs if the originating Interconnect_Port does not agree that the parameters in the SW_ACC are acceptable, or it receives an SW_RJT indicating the parameters in the ELP request were not acceptable to the responding Interconnect_Port, and it is not able to originate a new ELP request Sequence with modified parameters (see 7.6).

Transition P5:P11. This transition occurs if the ELP is rejected with “unable to perform command request”, and no FLOGI is received. The Switch Port performs the Link Offline protocol as defined in FC-FS-3 during the transition.

Transition P5:P16. This transition is taken when authorization checks that are based on data from the ELP fail.

State P6: Link Reset. When the exchange of link parameters has completed successfully, the value of buffer-to-buffer and end-to-end Class F Credit are initialized. In order to initialize the Flow Control parameters, the Switch Port that originated the successful ELP SW_ILS shall attempt the Link Reset protocol as defined in FC-FS-3.

NOTE 22 – The re-initialization of Link credit is necessary since the Flow Control parameters in the ELP Payload are intended to communicate Link credit parameters for a specific credit model. The Link Reset is the common method defined by FC-FS-3 for establishing a known credit state.

Transition P6:P11. This transition occurs if the Link Reset fails.

Transition P6:P12. This transition occurs if the Link Reset is successful, the Port is a B_Port and no security checks are required.

Transition P6:P13. This transition occurs if the Link Reset is successful and ESC is supported.

Transition P6:P17. This transition occurs if the Link Reset is successful and ESC is not supported.

Transition P6:P19. Occurs for an E_port when the E_Port detected, during the ELP processing in state P5, it is connected to a B_Port, and the E_Port requires B_Port authentication. This transition occurs also for a B_Port when the B_Port supports authentication.

Transition P6:P20. Occurs for an E_port when the E_Port detected, during the ELP processing in state P5, it is connected to a B_Port, and the E_Port does not require B_Port authentication.

State P7: Operate as an FL_Port. The Switch Port has detected a functional Arbitrated Loop. The Switch Port shall continue to operate as an FL_Port until the next Initialization Event. If a Switch Port enters the state in the non-participating mode, it shall remain in the non-participating mode until the next initialization event.

State P8: Operate as an F_Port. The Switch Port has detected an attached PN_Port. The Switch Port shall continue to operate as an F_Port until the next Initialization Event.

Transition All:P9. This transition occurs whenever an Interconnect_Port receives an SW_RJT with a reason code explanation of “E_Port is Isolated”.

State P9: Operate as an Isolated Interconnect_Port. The Interconnect_Port shall become Isolated and not continue with Fabric Configuration as specified in 7.6. The Switch Port shall continue to operate as an isolated Interconnect_Port until the next Initialization Event.

Transition P9:P5. This transition occurs when an ELP is received by an Isolated Interconnect_Port (see 7.6).

State P10: Initialize as an E_Port. The Switch Port has completed the exchange of Link Parameters with another E_Port. If the Link Parameters exchanged were acceptable, then the E_Port shall participate in the next phase of Fabric Configuration, described in 7.3. The Switch Port shall continue to operate as an E_Port until the next Initialization Event.

State P11: Retry Switch Port Initialization. The Switch Port shall wait for R_A_TOV before retrying Switch Port Initialization. If the Switch Port detects an Initialization Event during the time-out period, it shall not wait for the time-out period to expire.

State P12: Operate as a B_Port. ELPs have been exchanged, the link reset is successful, security checks have been performed, and the port is operating as a B_Port. Any further normal fabric configuration or routing operations are transparent to this port.

State P13: Send ESC. The link reset has been successful and ESC is supported. Information exchanged using ESC shall be carried through P17. The port shall perform ESC processing as described in 7.2.2.

Transition P13:P9. This transition occurs because of an ESC reject with reason code: “unable to perform command request”.

Transition P11:P0. This transition occurs if the R_A_TOV time-out period has expired.

State P15: Other Operation. The Port operates in a mode other than FSPF.

State P16: Invalid Attachment. The port operates in Invalid Attachment mode and SW_ILSs shall be rejected with an Invalid Attachment SW_RJT Reason Code with the following exceptions. FSPF SW_ILSs (e.g., HLO, LSU and LSA) shall be discarded and ACKs shall be sent upon receipt. Distributed Service CT_IUs shall be rejected with an F_RJT with a Reason Code of “Invalid Attachment”. Class N service frames shall be discarded and rejects shall be sent as appropriate to each Class of Service (See FC-FS-3). To leave this port state, the port shall receive OLS.

State P17: Security Checks. The port initiates and responds to all required security checks, if any, while in this state. If the port receives an EFP before security checks are complete, then the port shall respond with an SW_RJT with a Logical busy SW_RJT Reason Code and a SW_RJT Reason Code Explanation of Security Checks in Progress. The order and protocol of the security checks is defined in Fibre Channel Security Protocols (FC-SP). Switch_Name usage shall abide by the rules defined in FC-SP.

Transition P17:P9. This transition occurs when a required Policy or FC-SP Zoning check (see FC-SP) fails or is rejected.

Transition P17:P10. This transition occurs when all required security checks are successful and the port is to operate as an E_Port. The port is to operate as an E_Port if FSPF is the agreed upon path selection mechanism per the prior ESC exchange, or the prior ESC command was rejected with the reason code “command not supported”, or the port does not support ESC.

Transition P17:P15. This transition occurs if a routing protocol other than FSPF is agreed to in the prior ESC exchange.

Transition P17:P16 This transition occurs when a required authentication or authorization check (see FC-SP) fails or is rejected.

Transition P17:P17. This transition occurs each time a required security check is successful.

State P18: Disabled. While in this state, the port transmits the offline sequence until either, a power-on reset condition occurs or outside intervention requests an initialization of the port.

Transition All:P18. The transition to this state occurs when the Switch determines that a model dependent threshold has been exceeded.

State P19: B_Port Security Checks. While in this state an E_Port shall authenticate the B_Port by initiating a B_AUTH_ILS Authentication transaction (see FC-SP). While in this state a B_Port shall respond to the B_AUTH_ILS Authentication transaction.

State P20: CEC. While in this state an E_Port shall originate a CEC SW_ILS request Sequence. The processing is as specified for the ELP SW_ILS, except for configuration of flow control that is performed in state P5. The E_Port that sent the CEC message with the numerically higher Switch_Name shall become the CEC Initiator, while the E_Port that sent the CEC message with the numerically lower Switch_Name shall become the CEC Responder. The CEC SW_ILS is propagated by B_Ports.

Transition P19:P20. Occurs for an E_Port when the Authentication transaction performed in state P19 completes successfully.

Transition P19:P12. Occurs for a B_Port when the Authentication transaction performed in state P19 completes successfully.

Transition P19:P16. Occurs when the Authentication transaction performed in state P19 fails.

Transition P20:P13. Occurs when the CEC Exchange performed in state P20 is successful and ESC is supported. This transition may occur also when the remote Switch Port does not support CEC.

Transition P20:P17. Occurs when the CEC Exchange performed in state P20 is successful and ESC is not supported. This transition may occur also when the remote Switch Port does not support CEC.

Transition P20:P9. Occurs when the CEC Exchange performed in state P20 is not successful. This transition may occur also when the remote Switch Port does not support CEC.

When an Inter-Switch Link is established the Switch shall request EFP and enter state F2.

7.2.2 Switch_Name Usage

The Switch_Name presented during ELP Processing in state P5 shall be identical to that used when the Switch Port initializes as an E_Port in state P10 and for any subsequent operation or protocol.

7.2.3 Exchange Switch Capabilities Processing

Figure 14 shows a typical exchange involving the ESC SW_ILS. In this case, the ELP Initiator (the Switch that receives the ELP SW_ACC) initiates the ESC SW_ILS. Contained within the payload of the ESC is a list of supported Switch-to-Switch protocols. The receiver of the ESC determines the protocol it shall use from the list presented, and responds with that protocol in the payload of the ESC SW_ACC.

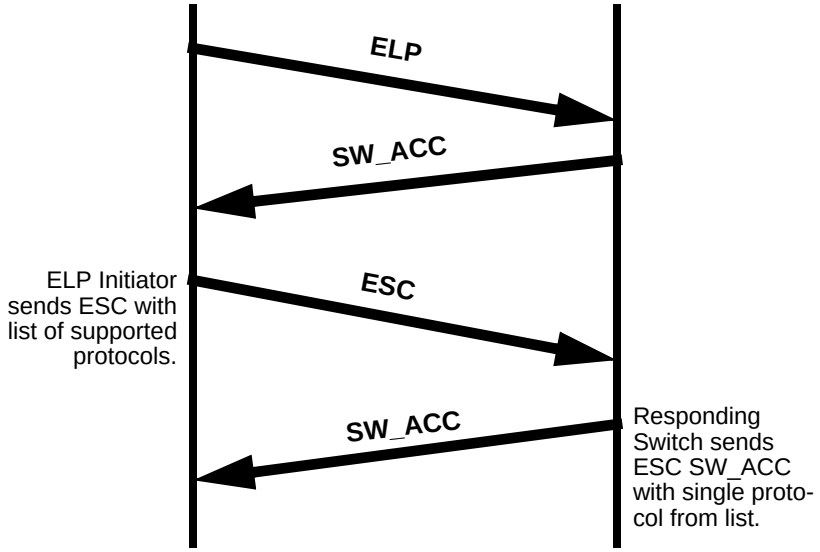

Figure 14 – ESC Processing

More formally, the process of exchanging Switch-to-Switch protocol capabilities shall progress as follows:

a) The ELP initiator originates the ESC SW_ILS. In the case of simultaneous ELPs from both Switches, the Switch that receives the ELP SW_ACC shall be considered the ELP initiator. The payload of the ESC contains a list of protocols supported by the sending Switch.

b) The responding Switch shall wait for a maximum of R_A_TOV to receive the ESC SW_ILS request. After this time, it shall proceed in the Port initialization process as if the ELP initiator does not support ESC. The responding Switch shall also proceed in the Port initialization process if it receives other messages from the ELP initiator, and shall reply accordingly.

c) If the receiving Switch does not support the ESC SW_ILS, it responds with a SW_RJT and a reason code of “Command not supported”. If the receiving Switch does support the ESC SW_ILS, continue to the next step.

d) If the receiving Switch does not support any of the protocols listed in the ESC SW_ILS, it responds with a SW_RJT and a reason code of “Unable to perform command request”. If the receiving Switch does support one of the protocols listed, continue to the next step.

e) The receiving Switch chooses a single protocol from the list presented in the ESC SW_ILS and responds with this protocol in the payload of the ESC SW_ACC.

7.2.4 B_Port Impact on ESC Processing

When no B_Ports exist between two E_Ports, ESC processing is initiated by the ELP Initiator. This works when two E_Ports are directly connected, but does not work when there are B_Ports between two E_Ports (e.g., the ELP Initiator could be a B_Port). To accommodate the presence of B_Ports between two E_Ports, the ESC processing shall occur as follows:

a) if two E_Ports are directly connected, the ESC processing shall be initiated by the ELP Initiator; or

b) if two E_Ports are connected through B_Ports, the ESC processing shall be initiated by the CEC Initiator.

7.2.5 Extensions to Support Virtual Fabrics

The Switch Port Mode Initialization State Machine is extended to support Virtual Fabrics on the Switch. These extensions are described in clause 12.

7.3 Principal Switch Selection

If Domain_IDs are assigned dynamically, a Principal Switch shall be selected whenever at least one Inter-Switch Link is established. The selection process chooses a Principal Switch, that is then desig-

nated as the Domain Address Manager. Figure 15 shows the state machine of the process. The recommended uses of BF and RCF are summarized in table 120.

Table 120 – Recommended BF and RCF Usage Summary

Event	BF or RCF
A Principal ISL experiences Link Failure or a transition to Offline or Isolated State	BF[a]
A configured Fabric is joined to another configured Fabric, and their Domain_ IDs do not overlap	BF
An unconfigured Switch or Fabric is joined to a configured Fabric	neither (see figure 17)
A configured Fabric is joined to another configured Fabric, and an overlap in Domain_ID is detected	Isolate or RCF Originated by Management
Reconfiguration caused by BF fails for any reason	Isolate or RCF Originated by Management

[a] In lieu of BF, a Switch may attempt Principal ISL Recovery as described in 7.5

Non-disruptive reconfiguration of Fabrics (BF) requires that Domain_IDs do not overlap. To ensure that the switches being joined do not have a Domain_ID overlap, an EFP shall be exchanged prior to either Switch issuing a Build Fabric request.

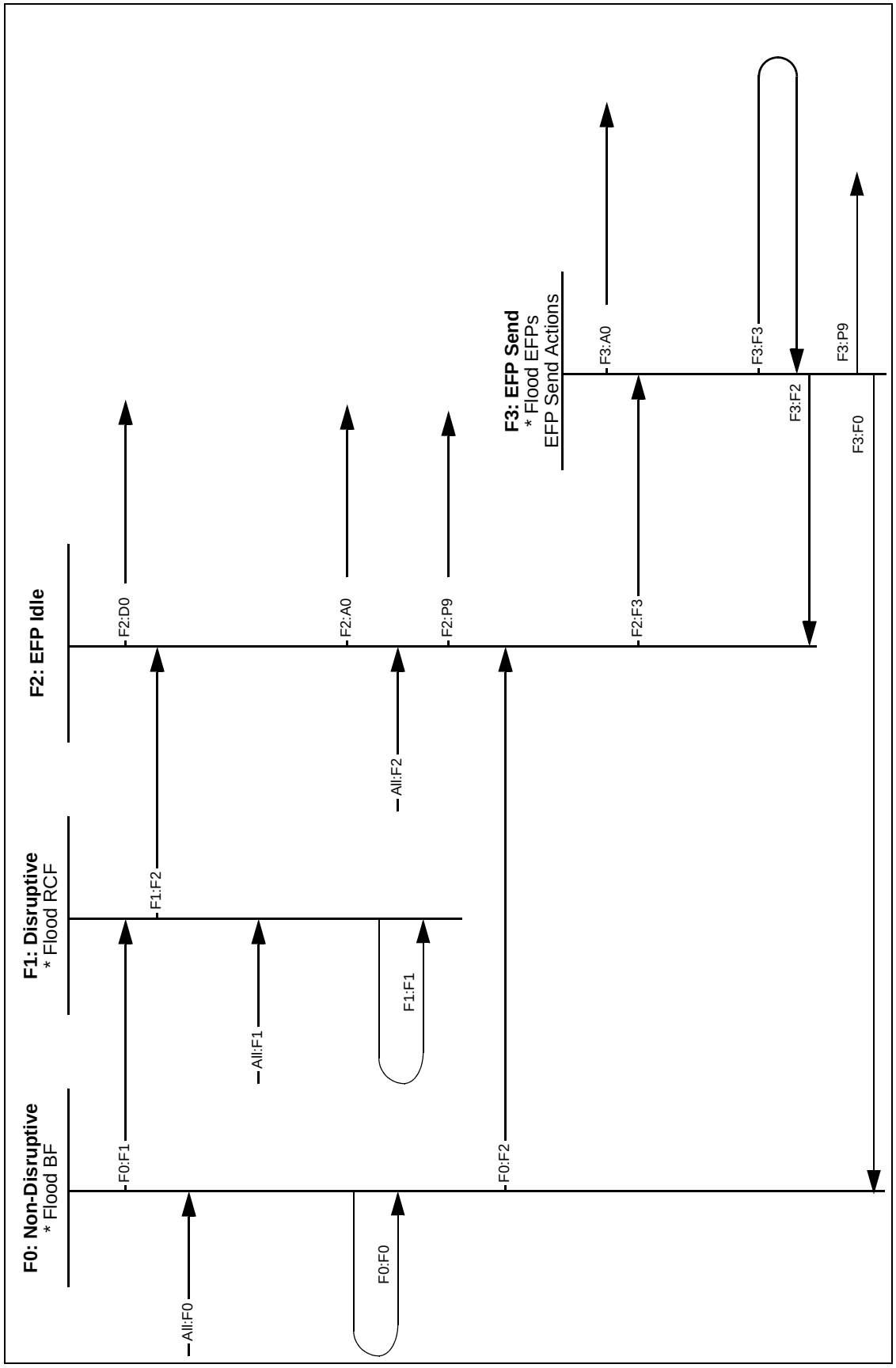

Figure 15 – Principal Switch Selection State Machine

State F0: Non-disruptive. A Switch may request a Fabric Reconfiguration by transmitting a BF on all E_Ports that have completed Switch Port Initialization. Unless warranted by current conditions, a Switch shall always first attempt a non-disruptive Fabric Reconfiguration by sending a BF. If the Switch is attempting a non-disruptive Fabric Reconfiguration, the Switch shall transmit a BF to all neighbor Switches on an E_Port that has completed Switch Port initialization, and from which the Switch has not yet received a BF request. The switch may transmit a BF on all E_Ports that have completed Switch Port initialization, and from which the Switch has not yet received a BF request.

While in this state:

a) the Switch shall accept any BF received on any E_Port, and shall not transmit a BF on any E_Port from which a BF has been received;

b) if an E_Port from a previously unconnected neighbor completes Switch Port Initialization, the Switch shall transmit a BF on that E_Port unless it has already received a BF on that E_Port since Switch Port Initialization completed;

c) any received EFP, DIA, RDI SW_ILSs shall result in the origination of an SW_RJT response with a Reason Code of “Logical busy”.

Figure 16 provides an example flow for BF requests.

Transition All:F0. This transition enters the state machine performing a non-disruptive Fabric Reconfiguration. This transition occurs when the Switch originates a BF, or when it receives a BF, or when the Switch receives a EFP where the received Domain_ID_List is non-zero, the retained Domain_ID_list is non-zero, the Domain_IDs do not overlap, and the received Switch_Priority||Switch_Name and the retained Switch_Priority||Switch_Name are not the same. In addition, F_S_TOV shall be started when the first BF is received or when the Switch initiates non-disruptive Fabric Configuration.

Transition F0:F0. Occurs when a EFP, DIA, or RDI SW_ILS is received. An SW_RJT specifying a reason code of “logical busy” is originated.

Transition F0:F1. If a Switch receives and accepts an RCF request Sequence while it is in the process of attempting a non-disruptive Fabric Reconfiguration, it shall stop the non-disruptive Fabric Reconfiguration and begin processing RCF requests. Any Active or Open BF Sequences shall be abnormally terminated. In addition, F_S_TOV shall be started when the first RCF is accepted or when the Switch initiates disruptive Fabric Configuration.

Transition F0:F2. The Switch shall wait for F_S_TOV following the reception or origination of the first BF before originating or responding to an EFP request Sequence. At the start of a non-disruptive Fabric Reconfiguration (BF), the Domain_ID_List shall be empty (“zero Domain_ID_List”). During Fabric reconfiguration, the Switch shall retain a Switch_Priority||Switch_Name value that it believes is the lowest in the Fabric. This value shall be initialized at the start of Fabric Reconfiguration (caused by BF or RCF) to the Switch’s value of Switch_Priority||Switch_Name. After the Switch is configured, it shall retain as the lowest value the Switch_Priority||Switch_Name of the Principal Switch.

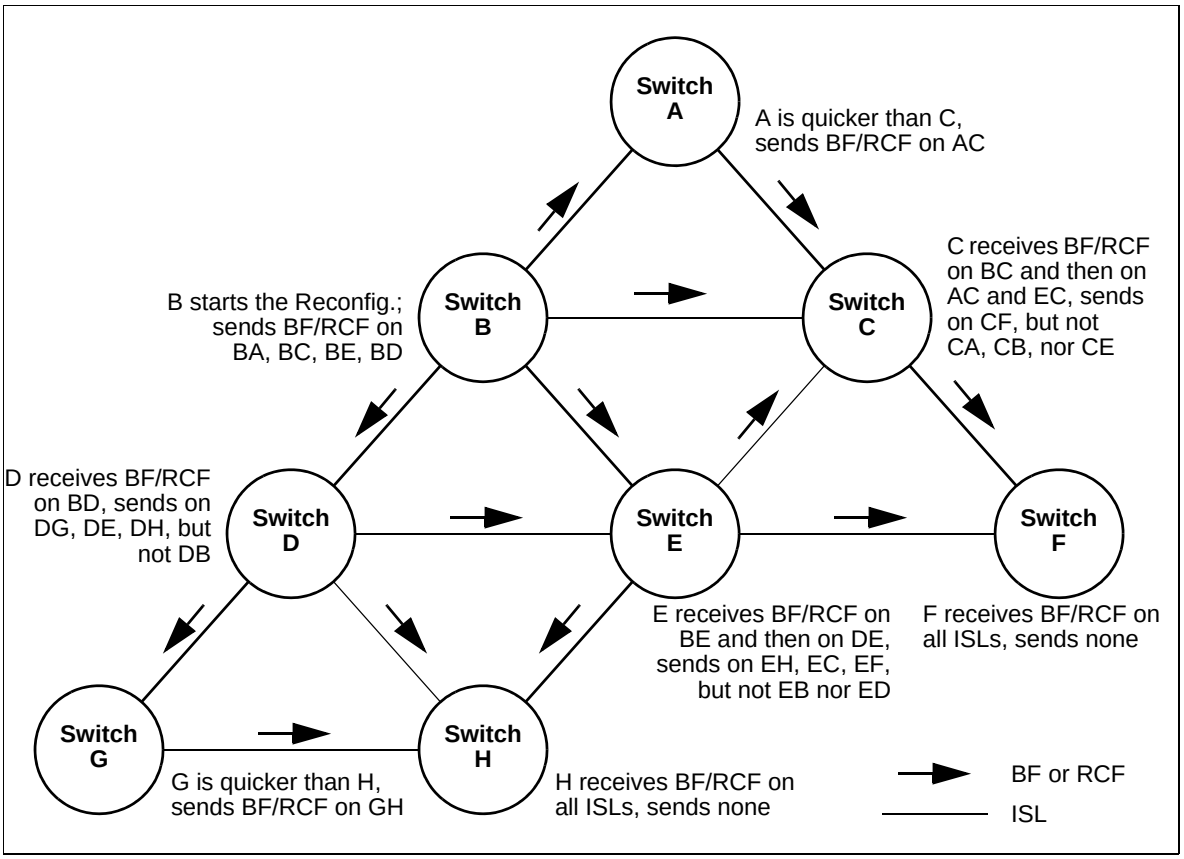

Figure 16 – Example Propagation of BF and RCF SW_ILS requests

State F1: Disruptive. The Switch is attempting a disruptive Fabric Reconfiguration, either originating or accepting an RCF.

Entering in this state:

a) the Switch shall transmit an ELP on all Isolated Interconnect_Ports and an RCF to all neighbor Switches on an E_Port that has completed Switch Port Initialization, and from which the Switch has not yet received an RCF. The switch may transmit a RCF on all E_Ports that have completed Switch Port initialization, and from which the Switch has not yet received a RCF request;

b) any lock due to an ACA request shall be released;

c) the FSPF Link State database and the associated initial message number counter shall be cleared.

NOTE 23 – The RCF processing may make out-of-date the local copies of some databases related to Distributed Services, that later it may be necessary to clear.

While in this state:

a) the Switch shall respond to any RCF received on any E_Port, and shall not transmit an RCF on any E_Port from which an RCF has been received;

b) if an E_Port completes Switch Port Initialization, the Switch shall transmit a RCF on that E_Port unless it has already received a RCF on that E_Port since Switch Port Initialization completed;

c) any received SW_ILS shall result in the origination of an SW_RJT response with a Reason Code of “Logical busy” except the SW_ACC, SW_RJT, ELP, ESC, RCF, HLO, LSU, and LSA SW_ILSs;

d) SW_ILSs shall not be sent except the SW_ACC, SW_RJT, and RCF SW_ILSs;

e) the HLO, LSU and LSA SW_ILSs shall be ignored on reception and shall not be sent;

f) any received Class F CT frame related to Distributed Services (Type = 20h) shall result in the origination of an F_RJT response with a reason code of “Nx_Port not available, temporary”;

g) Class F CT frames related to Distributed Services (Type = 20h) shall not be sent.

Upon exiting from this state until a new domain ID is granted to the switch (states D0 or A1):

a) any received SW_ILS shall result in the origination of an SW_RJT response with a Reason Code of “Logical busy” except the SW_ACC, SW_RJT, ELP, ESC, RCF, EFP, DIA, RDI, HLO, LSU, and LSA SW_ILSs;

b) SW_ILSs shall not be sent except the SW_ACC, SW_RJT, EFP, DIA, RDI, and RCF SW_ILSs;

c) the HLO, LSU and LSA SW_ILSs shall be ignored on reception and shall not be sent;

d) any received Class F CT frame related to Distributed Services (Type = 20h) shall result in the origination of an F_RJT response with a reason code of “Nx_Port not available, temporary”;

e) Class F CT frames related to Distributed Services (Type = 20h) shall not be sent.

Figure 16 shows an example diagram of the process to illustrate the flow of the RCF requests.

Transition All:F1. This transition enters the state machine performing a disruptive Fabric Reconfiguration. In this case, “All” refers to all Fx states other than F0. This transition occurs when the Switch originates an RCF, or when it receives and accepts an RCF request Sequence. In addition, F_S_TOV shall be started when the first RCF is received or when the Switch initiates disruptive Fabric Configuration.

Transition F1:F1.This transition occurs when any SW_ILS and any Class F CT frame related to Distributed Services (Type = 20h) is received, except the SW_ACC, SW_RJT, ELP, ESC, RCF, HLO, LSU, and LSA SW_ILSs. An SW_RJT specifying a Reason Code of “logical busy” is originated.

Transition F1:F2. The Switch shall wait for F_S_TOV following the acceptance or origination of the first RCF before originating or responding to an EFP request Sequence. At the start of a disruptive Fabric Reconfiguration (RCF), the Domain_ID_List shall be empty (“zero Domain_ID_List”). The Switch shall retain a Switch_Priority||Switch_Name value that it believes is the lowest in the Fabric. This value shall be initialized at the start of Fabric Reconfiguration (caused by RCF) to the Switch’s value of Switch_Priority||Switch_Name. After the Switch is configured, it shall retain as the lowest value the Switch_Priority||Switch_Name of the Principal Switch.

State F2: EFP Idle. The Switch shall remain in this state until it receives an EFP or DIA frame, or the 2xF_S_TOV timer expires and one of the following is true:

a) The retained Switch_Priority||Switch_Name equals the Switch_Priority||Switch_Name of the Switch;

b) The retained Switch_Priority is FFh.

In this state the Switch processes and generates EFP requests as required by the rules defined in state F3:EFP Send.

Transition All:F2. A Switch that is not yet configured (for example, after initial power-on and exchange of ELPs) shall transmit an EFP SW_ILS to all initialized E_Ports to determine if the Switch is attached to a configured Fabric (note that the Switch shall transition to the appropriate state and process any received BF or RCF requests as described above, as required by All:F0 and All:F1). When the first ISL to an adjacent switch becomes operational the switch shall transmit an EFP on that Link to determine the configuration of the Fabric that it is joining. On other ISLs the switch may transmit an EFP. “All” in this case does not include F1:F2.

Transition F2:F3. When the Switch receives an EFP, or if it has not yet sent an EFP, or responded to an EFP since the reconfiguration started, it shall transition.

Transition F2:D0. If the retained value of Switch_Priority||Switch_Name does not change for twice F_S_TOV, and if the retained value of the Switch_Priority||Switch_Name is equal to the value of the Switch, then the Switch has become the Principal Switch.

Transition F2:A0. If the Switch receives a DIA request Sequence from the upstream switch, then a Principal Switch has been selected. The Switch shall request a Domain_ID as described in 7.4.

Transition F2:P9. If the retained value of Switch_Priority||Switch_Name does not change for twice F_S_TOV, and if the retained value of Switch_Priority is equal to FFh, then there is no Switch capable of becoming a Principal Switch. The Switch shall cause all E_Ports to become Isolated, as described in 7.6.

State F3: EFP Send. The Switch shall process all EFP Payloads based on the information available at the time of processing. A Switch may receive an EFP Payload either by receiving an EFP request Sequence at an E_Port, or by receiving at an E_Port an SW_ACC reply Sequence in response to an EFP request Sequence. EFP Send actions shall be as described below.

a) The Switch shall communicate its retained Switch_Priority||Switch_Name to neighbor Switches that it has not yet communicated that value. The Switch shall accomplish this either by originating a new EFP request Sequence, or by an SW_ACC reply Sequence to a received EFP request.

b) If the Switch receives in an EFP Payload a non-zero Domain_ID_List (the list contains one or more records) and the Switch has a zero Domain_ID_List, then the Switch shall retain the received Switch_Priority||Switch_Name as the new value, and the received Domain_ID_List. The Switch shall also note from which neighbor Switch it received the new value, for potential use as the upstream Principal ISL during address distribution.

c) If the Switch receives in an EFP Payload a zero Domain_ID_List and the Switch has a non-zero Domain_ID_List, the Switch shall retain its current lowest Switch_Priority||Switch_Name value as the lowest value, without comparing with the received value. If the Switch has received a Domain_ID, the Switch shall send a DIA to the Switch from which it received the zero Domain_ID_List as described in 7.4.2.

d) If the Switch receives in an EFP Payload a zero Domain_ID_List and the Switch has a zero Domain_ID_List, and the received Switch_Priority||Switch_Name is lower than its current retained value, it shall discard the old value and retain the new value. The Switch shall also note from which neighbor Switch it received the new value, for potential use as the upstream Principal ISL during address distribution.

e) If the Switch receives a new lower value of Switch_Priority||Switch_Name before it has had a chance to communicate a prior lower value to all other E_Ports, it shall not attempt to communicate the prior value, and shall instead attempt to communicate the new value. The Switch shall not abort or otherwise abnormally terminate an existing EFP Exchange originated by the Switch for the sole reason of the value of Switch_Priority||Switch_Name being adjusted lower prior to the completion of the Exchange.

f) The Switch shall always return the lowest known value of Switch_Priority||Switch_Name in a SW_ACC reply Sequence to an EFP request Sequence.

g) The Switch shall retain a merged Domain_ID list after sending or receiving the SW_ACC to the EFP.

Transition F3:F0. This transition is made if the received Domain_ID List is non-zero, the retained Domain_ID List is non-zero, and the received Switch_Priority||Switch_Name and the retained Switch_Priority||Switch_Name are not the same.

Transition F3:F2. This transition is made if the received Domain_ID_List is zero or the retained Domain_ID_List is zero. In this transition, the Switch_Priority||Switch_Name of the Switch does not change.

Transition F3:F3. When the Switch is in the process of sending and receiving EFP requests and responses for the most recently received EFP, and receives a new EFP that causes the retained values to change, as described in state F3, it shall re-enter state F3 and start the process over.

Transition F3:A0. If the Switch receives a DIA request Sequence, then a Principal Switch has been selected. The Switch shall request a Domain_ID as described in 7.4.

Transition F3:P9. If the Domain_ID_List of the Switch is non-zero, and the Domain_ID_List in a received EFP Payload is non-zero, and if corresponding records in the Domain_ID_Lists are set to the same Domain_ID value (Domain_ID overlap), then the E_Port shall not continue with Fabric Configuration, and shall become Isolated, as described in 7.6.

At the completion of the Principal Switch selection process, all Switches other than the Principal Switch shall retain knowledge of the E_Port through which was received the lowest value of Switch_Priority||Switch_Name. This E_Port is the start of the first ISL in the path to the Principal Switch for the Switch; this ISL is called the upstream Principal ISL. The Switch_Name of the Principal Switch shall be used as the Fabric_Name.

7.4 Address Distribution

7.4.1 Address Distribution Overview

If Domain_IDs are assigned dynamically, once a Principal Switch (Domain Address Manager) has been selected, Switches that are not a principal Switch may request a Domain_ID. The Principal Switch shall assign all Domain_IDs. All other non-isolated Switches shall request Domain_IDs from the Principal Switch. Figure 17 shows the state machines of each process.

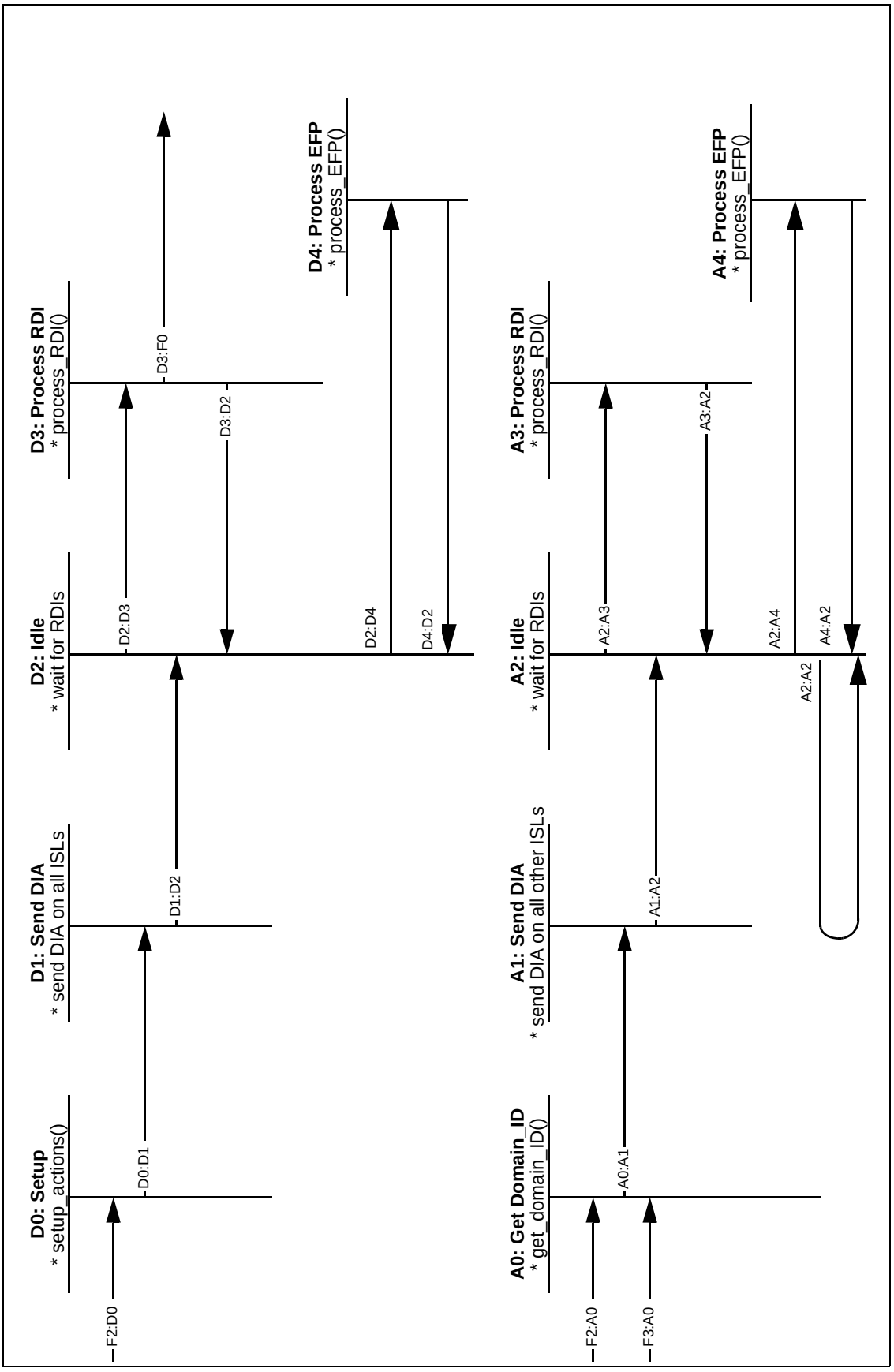

Figure 17 – Address Distribution State Machines

7.4.2 Domain_ID Distribution by the Principal Switch

The Principal Switch shall conduct Domain_ID distribution as indicated in figure 17 and as described below.

State D0: Setup. At the completion of Principal Switch Selection, the Principal Switch shall assume the role of Domain Address Manager, and perform the following setup actions:

a) The Principal Switch shall set its Switch_Priority value to 02h, if the current value of its Switch_Priority is greater than or equal to 02h. This setup action shall not cause an EFP request to be generated.

b) The Principal Switch shall empty its Domain_ID_List. This setup action shall not cause an EFP request to be generated.

c) The Principal Switch shall then grant itself one (or more) Domain_ID from the pool of available Domain_IDs. This pool is maintained by the Principal Switch. If the Principal Switch had a specific Domain_ID prior to the Reconfiguration Event, it shall grant itself that Domain_ID. This action shall cause an EFP request to be generated as described in the

State D3: Process RDI description below.

Transition F2:D0. As defined in 7.3.

Transition D0:D1. This transition occurs when the setup actions described above are completed and an EFP request Sequence is sent.

State D1: Send DIA. The Principal Switch shall then transmit a DIA SW_ILS request Sequence on all E_Ports. After receiving the SW_ACC reply, the Principal Switch may receive one or more RDI SW_ILS request Sequences via one or more of the E_Ports.

Transition D1:D2. This transition occurs when the send DIA actions described above are completed.

State D2: Idle. The Principal Switch shall remain in this state until it receives an RDI SW_ILS request Sequence. Reception of RDIs and or EFPs shall be queued in this state.

Transition D2:D3. This transition occurs when the Principal Switch receives an RDI SW_ILS request Sequence via one of its E_Ports.

State D3: Process RDI. The Principal Switch shall perform the following RDI processing actions:

a) When the Principal Switch receives an RDI SW_ILS request Sequence with a non-zero requested Domain_ID, in the absence of any error condition preventing it, it shall allocate the requested Domain_ID(s) to the requesting Switch, if available. If the requested Domain_ID is zero, it shall grant an available Domain_ID to the requesting Switch. If the requested Domain_ID is not available, it shall either grant an available Domain_ID to the requesting Switch or return an SW_RJT with reason code “Domain ID not available”. The Domain_ID is communicated to the requesting Switch by transmitting the SW_ACC reply Sequence via the E_Port on which the corresponding RDI request Sequence was received.

b) The Principal Switch shall not grant the same Domain_ID to more than one requesting Switch.

c) If the Principal Switch receives an RDI request for a Domain_ID of zero, or the same requested Domain_ID as it granted to that Switch in a previous RDI request received after Principal

Switch Selection, it shall not be considered an error and the Principal Switch shall grant the Domain_ID to the Switch using the SW_ACC reply sequence.

d) If a Switch that has already been granted a Domain_ID transmits a request to the Principal Switch for a different Domain_ID, the Principal Switch shall perform a Fabric Reconfiguration (see 7.3).

e) If the Principal Switch receives an RDI request and no appropriate Domain_IDs are available, the Principal Switch shall return SW_RJT with a reason/explanation of: “Unable to perform command request”, “Domain_ID not available“.

f) All Principal ISLs via which the Principal Switch receives RDI requests shall be downstream Principal ISLs.

g) Each time the Principal Switch grants a Domain_ID to a Switch (including itself), it shall transmit an EFP SW_ILS request Sequence via all Principal ISLs, with each record in the Domain_ID_List corresponding to a granted Domain_ID set to the Switch_Name granted the Domain_ID. An example of this process is illustrated in figure 18.

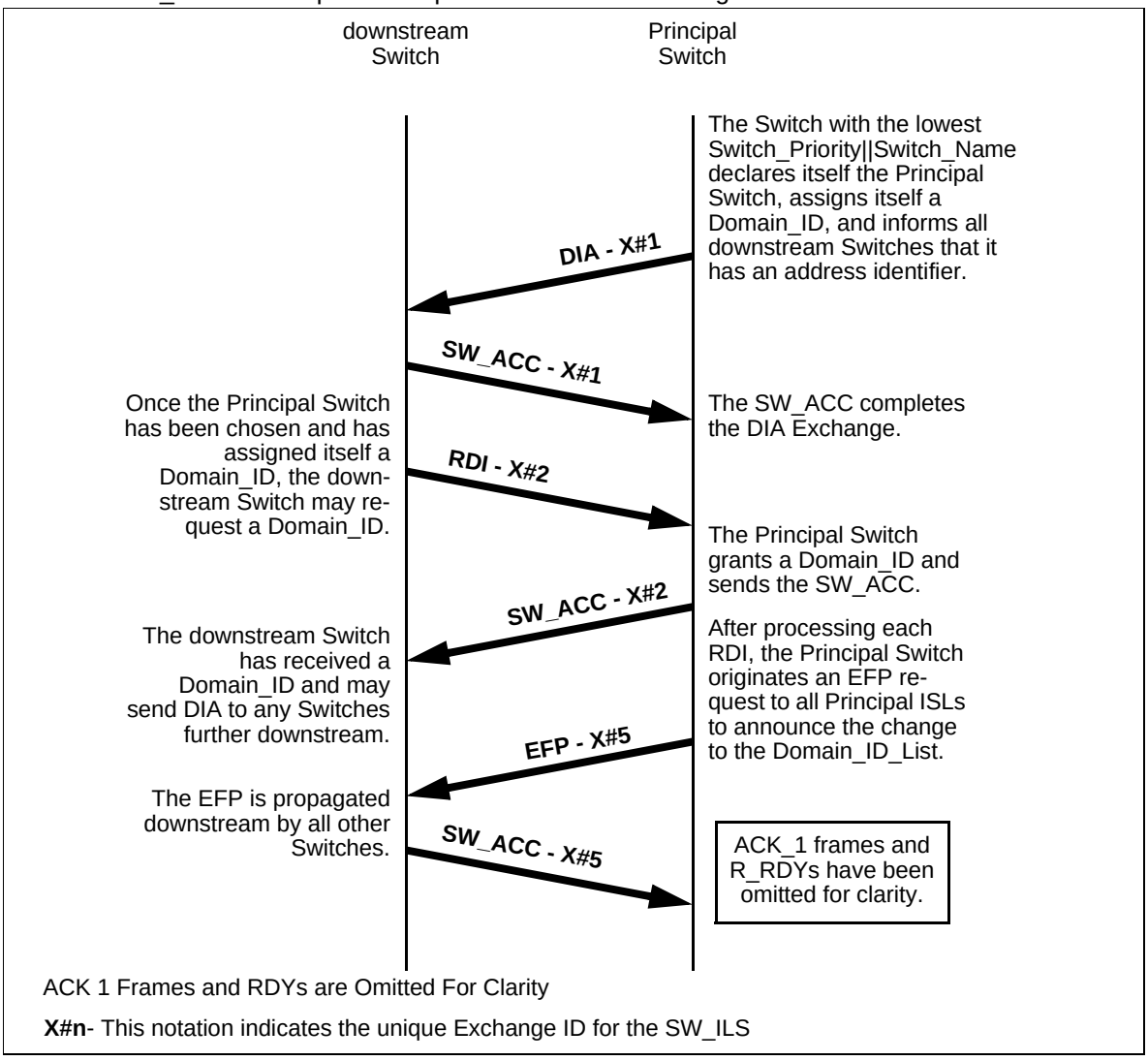

Figure 18 – RDI Request Processing by Principal Switch

Transition D3:D2. This transition occurs when the process RDI actions described above are completed.

Transition D3:F0. This transition occurs when a Switch that has already been granted a Domain_ID transmits a request to the Principal Switch for a different Domain_ID, and the Principal Switch elects to perform a non-disruptive Fabric Reconfiguration (see 7.3).

State D4: Process EFP. A configured Principal Switch enters this state following the reception of an EFP request Sequence.

Transition D2:D4. This transition occurs when the Principal Switch receives an EFP request from an unconfigured Switch.

Transition D4:D2. This transition occurs when the Principal Switch transmits an EFP response and a DIA to an unconfigured Switch.

7.4.3 Domain_ID Requests by the Switches

The Switches shall request a Domain_ID as indicated in figure 17, and as described below.

Transition F2:A0. As defined in 7.3.

Transition F3:A0. As defined in 7.3.

State A0: Get Domain_ID. At the completion of Principal Switch Selection, the Switch receives the DIA SW_ILS request Sequence via the upstream Principal ISL. The Switch shall reply to the request with the appropriate SW_ACC or other response, and perform the following actions to request a Domain_ID:

a) The Switch shall set its Switch_Priority value to a value greater than 02h.

b) The Switch shall empty its Domain_ID_List.

c) A DIA request Sequence received on any other ISL shall be replied to with the appropriate SW_ACC or other response, but shall otherwise be ignored. The DIA request received via the upstream Principal ISL is the indication that the Principal Switch has assigned a Domain_ID to all Switches between the Principal Switch and the Switch receiving the DIA request.

d) After transmitting an SW_ACC reply to the DIA request, the Switch shall transmit an RDI request Sequence via the upstream Principal ISL. If the Switch receives the reply SW_ACC to the RDI request, it shall assign address identifiers to all Ports within its Domain as appropriate. If the Switch receives an SW_RJT to the RDI, it shall originate a new RDI with a different payload, or go to state P9 and become isolated.

e) If as a result of the RDI processing a Switch has to change its Domain_ID, it shall perform a Link Initialization on each F_Port and a Loop Initialization with the L bit set on the LISA Sequence on each FL_Port. Additionally, it shall transmit an ELP on all Isolated Interconnect_Ports, release any lock due to an ACA request, flood a LSR with the old Domain_ID and the age field set to Max_Age.

NOTE 24 – The change of Domain_ID may make out-of-date the local copies of some databases related to Distributed Services, or the FSPF Link State Database, that later it may be necessary to clear. Additionally, if an implementation keeps track of why a Switch Port is in Isolated state, it may avoid sending an ELP over the Interconnect_Ports isolated for incompatible link parameters.

Transition A0:A1. This transition occurs when the setup actions described above are completed.

State A1: Send DIA. After the Switch is granted a Domain_ID, it shall then transmit a DIA SW_ILS request Sequence via all ISLs other than the Principal ISL. After receiving the SW_ACC reply, the Switch may receive one or more RDI SW_ILS request Sequences from one or more of the E_Ports.

Transition A1:A2. This transition occurs when the send DIA actions described above are completed.

State A2: Idle. The Switch shall remain in this state until it receives an RDI SW_ILS request Sequence. Reception of RDIs and or EFPs shall be queued in this state.

Transition A2:A3. This transition occurs when the Switch receives an RDI SW_ILS request Sequence via one of its E_Ports.

Transition A2:A2. This transition occurs when the Switch receives an EFP request from an upstream ISL and an EFP response is sent, and the EFP request is forwarded on all other ISLs.

State A3: Process RDI. The Switch shall perform the following RDI processing actions:

a) All Principal ISLs via which the Switch receives valid RDI requests shall be downstream Principal ISLs. If the Switch receives an RDI request on its upstream Principal ISL, it shall return SW_RJT with a reason/explanation of logical error, request not supported.

b) When the Switch receives a valid RDI request Sequence from one of its E_Ports via a downstream Principal ISL, it shall originate an RDI request Sequence with the same Payload via its upstream Principal ISL. When the reply SW_ACC is received via the upstream Principal ISL, it shall transmit an SW_ACC reply Sequence via the downstream Principal ISL on which the initial request was received. An example of this process is illustrated in figure 19.

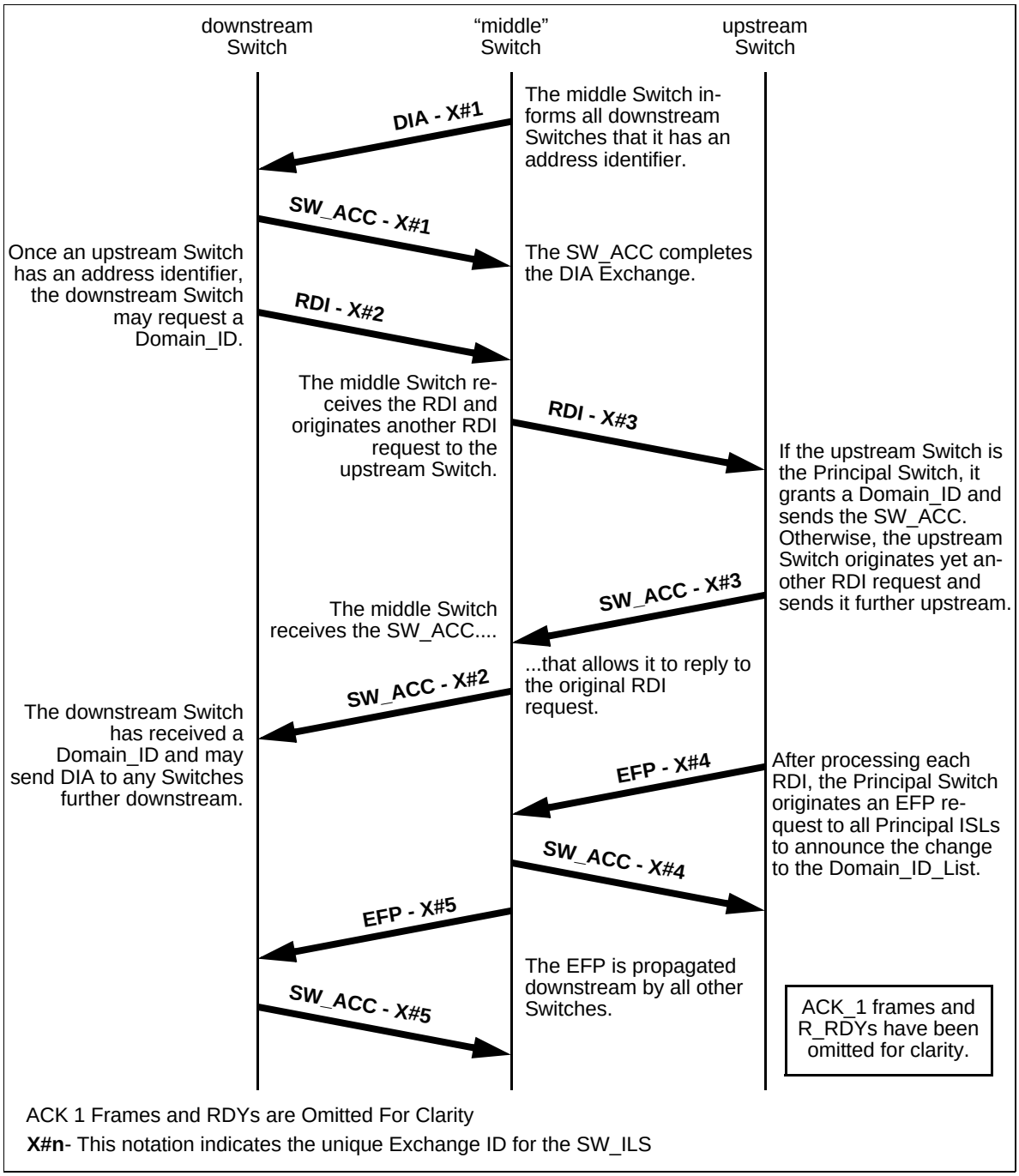

Figure 19 – RDI Request Processing by non-Principal Switch

Transition A3:A2. This transition occurs when the process RDI actions described above are completed.

State A4: Process EFP. A configured non-principal Switch enters this state following the reception of an EFP request Sequence. The Switch shall transmit EFP on all downstream principal ISLs.

Transition A2:A4. This transition occurs when a Switch that has already been configured receives an EFP request from a downstream unconfigured Switch.

Transition A4:A2. This transition occurs when a Switch that has already been configured transmits an EFP response and a DIA to a downstream unconfigured Switch.

7.5 Principal ISL Recovery

7.5.1 Overview

Failure of a Principal ISL disrupts control communication between the Principal Switch and downstream Switches. If other paths to the Principal Switch exist in the Fabric, recovery from this failure is possible with a Build Fabric operation. However, a Build Fabric operation creates a great deal of Fabric control traffic. In the case that additional ISLs exist between the two Switches that encountered the Principal ISL failure, it is possible for the two Switches to select a new Principal ISL without any impact to the remainder of the Fabric. This subclause describes this recovery process. Implementation of this process is optional.

7.5.2 Downstream Principal ISL Discovery

If a Switch implements Principal ISL Recovery and detects a Downstream Principal ISL failure for which it has additional ISLs connecting to the same downstream Switch, the Switch shall select one of the additional ISLs to become the new Downstream Principal ISL and send an EFP on that link within F_S_TOV. Then the Switch shall:

a) Use the selected link as Downstream Principal ISL upon receipt of an EFP SW_ACC; or

b) Proceed with a Build Fabric (see 7.3) upon receipt of an EFP SW_RJT or if no response is received within F_S_TOV.

7.5.3 Upstream Principal ISL Recovery

If a Switch implements Principal ISL Recovery and has ISLs in addition to the Upstream Principal ISL to the same upstream Switch, the Switch shall:

a) Upon receiving an EFP from the upstream Switch on one of the additional ISLs:

   1) Respond with an SW_ACC; and

   2) Use the ISL on which the EFP was received as the new Upstream Principal ISL;

b) Upon detection of a failure of the Upstream Principal ISL without receiving an EFP within 2xF_S_TOV on any of the additional ISLs, initiate a Build Fabric (see 7.3).

7.6 E_Port and Fabric Isolation

An E_Port connected via an Inter-Switch Link to another E_Port may determine that it is unable to communicate with the other E_Port for one of the reasons listed below.

a) The two E_Ports have incompatible Link Parameter requirements. For example, if one Switch has an E_D_TOV setting different than another, Class 2 frames sent by an N_Port on one Switch may not receive timely F_BSY responses from the other Switch.

b) Similarly, the two E_Ports have incompatible Fabric Parameter requirements. For example, if an E_Port receives an EFP that contains records it does not support, it shall Isolate.

c) When the E_Port receives an EFP Payload and the received Domain_ID_List is non-zero, the retained Domain_ID_List is non-zero, the Domain_IDs overlap, and the received Principal Switch_Name is not equal to the retained Principal Switch_Name.

d) The two E_Ports are a Link between Switches that are not capable of performing the Domain Address Manager function, and are each also not attached via an ISL to any other Switch capable of performing the Domain Address Manager function. Since no Switch may allocate Domain_IDs, no Class N frames may be sent between the Switches.

e) The two E_Ports have exchanged zoning information via the Merge Request in an attempt to resolve a zoning configuration. As a result of the Merge processing the zoning configuration may not be merged (see 10.5).

f) An SW_RJT is received in response to an RDI request and the Switch chooses to not send a new RDI with a different payload.

g) The E_Port rejects an RCF SW_ILS;

h) An SW_RJT with reason code explanation of “E_Port is Isolated” is received;

i) A Switch configured to assign Domain_IDs statically, on receiving an EFP, BF, RCF, DIA or RDI SW_ILS shall reply with an SW_RJT having Reason Code Explanation ‘E_Port is Isolated’ and shall isolate the receiving E_Port.

When any of the above conditions occurs, the E_Port shall Isolate itself from the other E_Port. The following is a list of appropriate Class F frames that may be communicated between Isolated E_Ports.

a) An ELP SW_ILS request may be sent by an Isolated E_Port in an attempt to establish a working set of Link Parameters. This ELP SW_ILS request may be used to support a negotiation process as outlined in annex B;

b) An SW_ACC response may be sent in response to any of the above SW_ILS requests;

c) An SW_RJT response may be sent in response to any of the above SW_ILS requests, if necessary, and shall be sent as the appropriate response to any other SW_ILS request not listed above. The SW_RJT response shall indicate the following SW_RJT reason/explanation code: Unable to perform command request/ E_Port is isolated.

The buffer-to-buffer credit between the Isolated E_Ports shall be a value of one; no alternate credit shall be in effect. No routing of Class N frames shall occur across the ISL.

A Switch may override the Isolated condition by originating an ELP, or any of the events that cause the transition ALL:P0.

7.7 B_Port Operation

7.7.1 Differences Between E_Ports and B_Ports

A B_Port supports a subset of the E_port Internal Link Services (ILS) and a B_Port has the same facilities as described in this standard for an E_port. The underlying differences between B_Port and

E_port initialization are that B_Ports perform ELP and are transparent to most other SW_ILSs (see 5.6).

7.7.2 B_Port Internal Link Services

The B_Port shall generate a subset of the Internal Link Services defined in this standard. Table 121 details the ILS support as either being propagated or generated by the B_Port.

Table 121 – B_Port - ILS Support (Part 1 of 2)

FC-SW-6 Internal Link Service (ILS)	Generated By B_Port	B_Port Response	Propagated by B_Port
Exchange Link Parameter (ELP)	Allowed	SW_ACC or SW_RJT	Prohibited
Exchange Fabric Parameters (EFP)	Prohibited	Propagate	Allowed
Domain Identifier Assigned (DIA)	Prohibited	Propagate	Allowed
Request Domain_ID (RDI)	Prohibited	Propagate	Allowed
Hello (HLO)	Prohibited	Propagate	Allowed
Link State Update (LSU)	Prohibited	Propagate	Allowed
Link State Acknowledgment (LSA)	Prohibited	Propagate	Allowed
Build Fabric (BF)	Prohibited	Propagate	Allowed
Reconfigure Fabric (RCF)	Prohibited	Propagate	Allowed
Exchange Switch Capabilities (ESC)	Prohibited	Propagate	Allowed
Acquire Change Authorization (ACA)	Prohibited	Propagate	Allowed
Release Change Authorization (RCA)	Prohibited	Propagate	Allowed
Stage Fabric Configuration Update (SFC)	Prohibited	Propagate	Allowed
Update Fabric Configuration (UFC)	Prohibited	Propagate	Allowed
Registered State Change Notification (SW_RSCN)	Prohibited	Propagate	Allowed
Distribute Registered Link Incident Report (DRLIR)	Prohibited	Propagate	Allowed
Check E_Port Connectivity (CEC)	Prohibited	Propagate	Allowed
Exchange Switch Support (ESS)	Prohibited	Propagate	Allowed
Merge Request Resource Allocation (MRRA)	Prohibited	Propagate	Allowed
Switch Trace Route (STR)	Prohibited	Propagate	Allowed

Table 121 – B_Port - ILS Support (Part 2 of 2)

Exchange Virtual Fabrics Parameters (EVFP)	Prohibited	Propagate	Allowed
Enhanced Acquire Change Authorization (EACA)	Prohibited	Propagate	Allowed
Enhanced Stage Fabric Configuration Update (ESFC)	Prohibited	Propagate	Allowed
Enhanced Update Fabric Configuration (EUFC)	Prohibited	Propagate	Allowed
Enhanced Release Change Authorization (ERCA)	Prohibited	Propagate	Allowed
Transfer Commit Ownership (TCO)	Prohibited	Propagate	Allowed

7.7.3 B_Port Initialization

The Fabric Configuration process enables a Switch to determine its operating mode, exchange operating parameters, and provides for distribution of addresses. Changes to support Bridge devices and the B_Port in this process are summarized in table 122

Table 122 – Bridge Port Initialization Summary

Operation	Starting Condition	Process	Ending Condition
Establish Link Parameters and Switch Port operating mode	Bridge Port has achieved word synch.	ELPs are exchanged and the B_Port determines that it is attached to an E_Port.	Link Parameters have been exchanged and Credit has been initialized, and it is known if the attached port is an E_Port.

7.7.4 Example B_Port Configuration

The following diagram shows an example Fabric configuration utilizing B_Ports. In this instance, two bridge devices enable the existence of a virtual ISL between two Switches. With the exception of ELP, the B_Port is transparent to Fabric E_Port operation.

NOTE 25 – The routing protocol does not run on the bridge device.

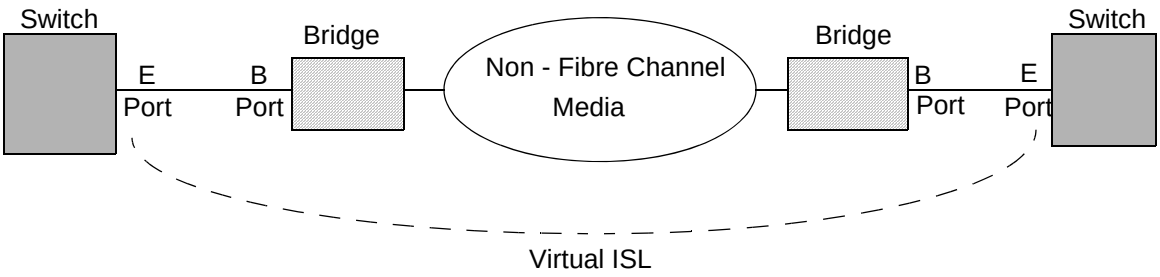

Figure 20 – Example B_Port Configuration - Virtual ISL

8 Fabric Shortest Path First (FSPF)

8.1 Overview

8.1.1 Basic Components

FSPF is a link state path selection protocol. FSPF keeps track of the state of the links on all Switches in the Fabric and associates a cost with each link. The protocol computes paths from a Switch to all the other Switches in the Fabric by adding the cost of all the links traversed by the path, and choosing the path that minimizes the cost. The collection of link states (including cost) of all the Switches in a Fabric constitutes the Link State Database.

FSPF has four major components:

a) A Hello protocol, used to establish connectivity with a neighbor Switch, to establish the identity of the neighbor Switch, and to exchange FSPF parameters and capabilities;

b) A replicated Link State Database, with the protocols and mechanisms to keep the databases synchronized across the Fabric;

c) A path computation algorithm;

d) A routing table update.

The Link State Database synchronization in turn consists of two major components: an initial database synchronization, and an update mechanism. The initial database synchronization is used when a Switch is initialized, or when an Inter-Switch Link (ISL) comes up. The update mechanism is used in two circumstances:

a) When there is a link state change, for example an ISL going down or coming up;

b) On a periodic basis, to prevent Switches from deleting topology information from the database.

The path computation algorithm shall be any algorithm that yields a minimum cost path, as defined above.

NOTE 26 – One possible candidate is Dijkstra's algorithm, but this is not a requirement for interoperability.

The routing table update is not covered in this specification, since it is strictly implementation dependent. Different Switches may have different internal routing mechanisms, and still interoperate.

8.1.2 Fabric connectivity

All the connections between Fibre Channel Switches shall be point-to-point. There are no direct connections to broadcast media, where multiple routing-capable Switches may co-exist.

8.1.3 Addressing

A path selection protocol requires an addressing scheme to uniquely identify the final destination of a frame. FSPF supports the addressing scheme described in 4.8. If multiple Domain_IDs are used by a Switch, the Switch shall use the lowest value Domain_ID as the Originating Domain_ID in all FSPF headers. It shall also send an LSR for each Domain_ID that it has been assigned.

8.1.4 Path Selection and Routing

In this standard, the term “path selection” indicates the discovery of the best path from source to destination, and the term “routing” indicates the actual forwarding of frames to a specific destination. FSPF performs hop-by-hop routing meaning that a Switch only needs to know the next hop on the best path to the destination. The replicated Link State Database insures that every Switch in the Fabric has the same view of the Fabric itself, allowing consistent routing decisions to be made by all Switches. The replicated data base is essential to avoid routing loops.

Typically a Switch needs to know, for each destination domain in the Fabric, which path should be used to route a frame to that domain. A routing table entry minimally consists of a destination Domain_ID, and an E_Port to which frames are forwarded to the destination Switch.

8.1.5 FSPF Path Selection Summary

Table 123 summarizes path selection via FSPF.

Table 123 – Path Selection (FSPF) Operation Summary

Operation	Starting Condition	Process	Ending Condition
1. Perform Initial HELLO Exchange	The Switch originating the HELLO has a valid Domain_ID.	HLO SW_ISL frames are exchanged on the link until each Switch has received a HELLO with a valid neighbor Domain field.	Two way communication has been established
2. Perform Initial Database Exchange	Two way communication has been established.	LSU SW_ISL frames are exchanged containing the Initial database.	Link State Databases have been exchanged.
3. Running State	Initial Database Exchange completed.	Routes are calculated and set up within each Switch. Links are maintained by sending HELLOs every Hello_Interval. Link databases are maintained by flooding link updates as appropriate.	FSPF routes are fully functional.

8.2 FSPF Message Processing

8.2.1 Message transmission

FSPF information is transported using FSPF SW_ILS messages. Details of these three FSPF SW_ILS messages are described in 6.1.

Before sending a message, a Switch shall set the values in the header fields as follows:

a) Command Code: The value that identifies the type of message, Hello (14000000h), Link State Update (15000000h) or Link State Acknowledgement (16000000h);

b) Version: The version number of the protocol as documented in this standard (02h);

c) Authentication Type: No authentication is specified at this time. This field shall be set to 00h;

d) Originating Domain_ID: The Domain_ID of the Switch that is transmitting this message. The Originating Domain_ID shall be a valid value as specified in 6.1.8.2;

e) Authentication: No authentication is specified at this time. This field is 8 bytes long and shall be set to 0000000000000000h.

8.2.2 Message Reception and Tests

When an FSPF message is received, the following tests shall be performed on its content. These tests are described below.

a) The Version number shall be 02h;

b) The Authentication Type shall be 00h;

c) The Originating Domain_ID shall be checked for a valid value as specified in 6.1.8.2;

d) The Authentication field shall be 0000000000000000h.

If any of these tests fails, the message shall be discarded. If all the tests succeed, the message shall be passed to the relevant protocol for further processing.

8.3 Hello Protocol

8.3.1 Basic Functions

The Hello protocol is used to establish two-way communication with a neighbor Switch, and determine when this communication is interrupted. An Inter-Switch Link (ISL) may be used for routing user traffic through the Fabric only if there is two-way communication between the Switches. The Hello protocol also provides some information about remote connectivity, and in particular, it allows the association of the local E_Port with the remote E_Port.

8.3.2 Hello Message Transmission

A Switch is required to know that a port is connected to another Switch through an ISL, before that port may be used to route data in a multi-Switch Fabric. Prior to a Hello being sent, the following shall be true:

a) The port shall be an E_Port;

b) The Switch where the E_Port resides shall have a Domain_ID assigned;

c) The Switches on the two sides of an ISL shall have agreement on a common set of Link Parameters and Fabric Parameters.

After a Switch determines that a port is an E_Port and the Switch has acquired a Domain_ID, the Switch starts sending Hello messages to the neighbor Switch. Hello messages contain the FSPF header and the parameters specific in the Hello protocol.

8.3.3 Hello Message Parameters

The Hello_Interval is defined to be the interval in seconds between two consecutive transmissions of a Hello message by the local Switch.

The Dead_Interval is the interval in seconds after which the local Switch shall bring down the Adjacency, if it has not received a valid Hello message from the remote Switch. Switches may also reset this timer on reception of an FSPF LSA or LSU.

The Hello_Interval and the Dead_Interval are values that may be configured separately on each port. It is absolutely necessary that the two ports that are connected by an ISL on two Switches have the same value for these two variables. Default values that are appropriate for most circumstances are provided in table 206.

The Recipient Domain_ID is the Domain_ID of the Switch on the other side of the ISL. It is set to FFFFFFFFh in the first transmitted Hello, to indicate that the Switch has not received an Hello message from the neighbor Switch. Once an Hello message is received from the neighbor Switch, the local Switch stores the Domain_ID of the remote Switch on that port, and from then on it sets the Recipient Domain_ID to that value in all future Hello messages. The Recipient Domain_ID is set back to FFFFFFFFh when the two-way communication between Adjacent Switches is disrupted. This typically happens either because the E_Port goes offline, or because the Dead_Interval timer expires.

The Originating Port Index shall be set to the index of the port that transmits the Hello message.

If the Domain_ID of a Switch changes, then the Switch shall perform a one-way Hello with FFFFFFFFh set in its Recipient Domain_ID field.

8.3.4 Hello Message Reception

When a Hello message is received, the message header shall be checked according to the rules described in 8.2.2. In addition, the following checks are performed:

a) The Hello_Interval value shall match the value configured for the port that originated the message. If it does not, the Hello message is discarded.

b) The Dead_Interval value shall match the value configured for the port that originated the message. If it does not, the Hello message is discarded.

c) The Recipient Domain_ID shall be either FFFFFFFFh, or the Domain_ID of the local Switch. Any other value in this field causes the Hello message to be discarded.

When the local Domain_ID is recognized in the incoming Hello message, a two-way communication has been established with the remote Switch, and the Neighbor FSM may proceed to the next transition. If the value FFFFFFFFh is detected in an incoming Hello message at any time after the two-way communication has been established, the neighbor shall fall back to a one-way state and the FSM transitions to that state.

The Originating Port Index does not need to be checked. Its value shall be stored in the neighbor data structure, together with the Domain_ID of the sending Switch, the Hello_Interval and the Dead_Interval.

8.4 The Link State Database

The Link State Database is central to the operation of FSPF. It is a replicated database where all Switches in the Fabric have the same exact copy of database at all times.

The database consists of a collection of Link State Records (LSRs). Link State Records may be of different types and have different formats and contents. This standard describes one type of LSR:

a) Switch Link Record.

A Switch Link LSR completely describes the connectivity of a Switch to all Switches to which it is directly attached. The information contained in a LSR is a list of all the individual ISLs that the Switch may use to forward user data to a remote Switch. Each ISL is associated with a link type, the Domain_ID of the remote Switch it is connected to, the local and remote Port ID, and the cost of the link itself.

Every Switch in the Fabric is responsible for issuing and maintaining its own LSR. An LSR is identified by a Link-State ID. For a Switch link LSR the Link-State ID is the Domain_ID of the Switch that issues the LSR. A Switch shall not issue an Switch Link LSR with a Domain_ID different from its own Domain_ID. A Switch shall not generate new instances of an LSR, unless it generated the original LSR. However, a Switch shall forward LSRs that it has not generated as part of the flooding process.

Multiple instances of an LSR are issued over time. Sometimes the content of the new instance is the same as the previous instance, sometimes it is different. Every Switch is responsible for maintaining the most recent copy of its own LSR throughout the Fabric.

Multiple instances of an LSR may be temporarily present in a Fabric at the same time. Ultimately, only the most recent instance shall survive, and all Switches shall keep that instance in their Link State Database. The process of purging old instances of an LSR within the Fabric should be as fast as possible because it impacts the ability to properly route Class N frames through the Fabric.

Several fields in a LSR are used to identify the LSR and to determine which instance is the newest. The Link State Database is used by a Switch to compute the least cost path to all other Switches in the Fabric. This is why it is essential that all Switches have the same Link State Database, or different Switches may build inconsistent paths. An ISL shall be considered in the path computation only if both LSRs of the two connected Switches list this ISL (two way communication between the Switches).

The path computation is local, and the results of the computation are not distributed to other Switches, only topology information is distributed. This is a characteristic of link-state path selection protocols.

8.5 Usage of LSR Fields

8.5.1 LSR Age

The LSR age field indicates how long a particular instance of an LSR has been in the database. The LSR age field is based in seconds and is a 16-bit unsigned integer.

The LSR age is initialized to 0000h by the advertising Switch when it is first issued. The LSR Age is incremented by one every second by every Switch in the Fabric as long as it stays in that Switch's database. It is also incremented by 1 every time it is transmitted during the flooding procedure.

NOTE 27 – This somewhat arbitrary increment represents the transmission time on the ISL and insures that a flooded LSR does not loop forever.

This field is also used to help determine which of two instances of an LSR is more recent, when other fields are equal.

A new instance of an LSR shall be issued when the LSR age field of the LSR in the database reaches the value LS_Refresh_Time. Only the Switch that originated the LSR shall refresh it with the issue of a new instance.

The age of an LSR shall never exceed Max_Age (3600, 1 hour). If an LSR reaches the age of 3600, it shall be flushed from the Fabric. This operation is accomplished by flooding the LSR with the LSR age field set to Max_Age. Upon receiving this instance of an LSR, other Switches shall remove the LSR from the database. In order to be completely flushed from the Fabric, an aged LSR shall be removed from the database in all Switches.

Any Switch in the Fabric may flush an LSR that has reached Max_Age from the Fabric.

8.5.2 LSR Incarnation Number

This field is a progressive number that identifies the incarnation of the LSR. It is used to determine which one of two incarnations of an LSR is more recent, in particular, the one with the larger incarnation number is the most recent.

The incarnation number is a 32-bit signed integer and is incremented in two's complement form. The lowest possible negative number is 80000000h, and it is not used. The lowest incarnation number is 80000001h. The first instance of an LSR shall have an incarnation number of 80000001h. Each new instance shall have its incarnation number incremented by one. A new instance may be issued for several reasons, but it shall always have its incarnation number increased by one, even if the content of the LSR is identical to the previous instance.

NOTE 28 – This causes the new instance to have a different checksum.

The maximum incarnation number is 7FFFFFFFh. When an LSR reaches this value as an incarnation number, the originating Switch shall flood the LSR through the Fabric with an LSR Age = Max_Age. After the LSR is acknowledged by an LSA on all ISLs, then the originating Switch shall issue a new instance of the LSR with an incarnation number of 80000001h.

This process causes a brief interruption of service because paths to the Switch that is rolling over its incarnation number are not available until the LSR with the smallest incarnation number is installed. However, this event should be extremely rare since most of the time a new instance of an LSR is issued every 30 minutes.

8.5.3 LSR Instance Rules

Two LSR instances shall be considered identical when both of the following conditions are met:

a) The Link State ID fields are the same;

b) The Link State Incarnation values are the same.

For two instances of the same LSR, the LSR incarnation number, LSR Age, and LSR checksum fields shall be used to determine which instance is more recent:

a) The LSR instance with the highest incarnation value shall be considered more recent. If both instances have the same incarnation value, then;

b) If the LSR age fields of only one of the two instances is equal to MaxAge, it shall be considered more recent;

c) Else, If the two instances have different LSR checksums, then the instance having the larger LSR checksum (when considered as a 16-bit unsigned integer) shall be considered more recent;

d) Else, if the LSR age fields of the two instances differ by a value less than or equal to Max_Age_Diff, the instance having the smaller (younger) LSR Age shall be considered more recent;

e) Else, the two instances shall be considered to be identical.

8.5.4 LSR Checksum

The checksum field is used to detect data corruption in an LSR, both when it is received and when it is stored in Switch memory. When an LSR is received with a bad checksum, the LSR shall be ignored.

The integrity of the Link State Database shall be checked by calculating checksums for all the LSRs. If any of the LSRs fail this checksum, this may be an indication of a memory corruption problem, and the Switch should be reinitialized.

NOTE 29 – A reliable notification of this event may not be possible since the device may not be operating correctly.

The LSR checksum covers the whole LSR, except the LSR Age field. The checksum algorithm is known as the Fletcher Checksum, and shall be computed byte by byte, by accumulating the sum of the payload one byte at the time.

NOTE 30 – The Fletcher algorithm is documented in RFC 905. The Nakassis algorithm, for an optimized computation of the checksum, is given in RFC 1008.

The checksum shall be computed as depicted in table 124:

Table 124 – Checksum Byte Order Calculation

Word	Bits 31 to 24	Bits 23 to 16	Bits 15 to 8	Bits 7 to 0
0	LSR Type A1	Reserved A0	LSR Age	LSR Age
1	Reserved B3	Reserved B2	Reserved B1	Reserved B0
2	Link State Identifier B7	Link State Identifier B6	Link State Identifier B5	Link State Identifier B4
3	Advertising Domain_ID B11	Advertising Domain_ID B10	Advertising Domain_ID B9	Advertising Domain_ID B8
4	Link State Incarnation Number B15	Link State Incarnation Number B14	Link State Incarnation Number B13	Link State Incarnation Number B12
5	Checksum B19	Checksum B18	LSR Length B17	LSR Length B16
6	Reserved C3	Reserved C2	Number of Links C1	Number of Links C0
7	Link ID 0,D3	Link ID 0,D2	Link ID 0,D1	Link ID 0,D0
8	Reserved 0,D7	Output Port Index 0,D6	Output Port Index 0,D5	Output Port Index 0,D4
9	Reserved 0,D11	Neighbor Port Index 0,D10	Neighbor Port Index 0,D9	Neighbor Port Index 0,D8
10	Link Type 0,D15	Reserved 0,D14	Link Cost 0,D13	Link Cost 0,D12

The two Checksum bytes are initialized to 00h for the checksum calculation. The checksum calculation is performed in the following order:

A0, A1, B0, B1, B2, B3, B4, B5, B6 B7, B8, B9, B10, B11, B12, B13, B14, B15, B16, B17, B18, B19, C0, C1, C2, C3;

(0, D0), (0, D1), (0, D2), (0, D3), (0, D4), (0, D5), (0, D6), (0, D7), (0, D8), (0, D9), (0, D10), (0, D11), (0, D12), (0, D13), (0, D14), (0, D15);

(1, D0), (1, D1), (1, D2), (1, D3), (1, D4), (1, D5), (1, D6), (1, D7), (1, D8), (1, D9), (1, D10), (1, D11), (1, D12), (1, D13), (1, D14), (1, D15)...
...
...(n-1, D0), (n-1, D1), (n-1, D2), (n-1, D3), (n-1, D4), (n-1, D5), (n-1, D6), (n-1, D7), (n-1, D8), (n-1, D9), (n-1, D10), (n-1, D11), (n-1, D12), (n-1, D13), (n-1, D14), (n-1, D15)

(n, D0), (n, D1), (n, D2), (n, D3), (n, D4), (n, D5), (n, D6), (n, D7), (n, D8), (n, D9), (n, D10), (n, D11), (n, D12), (n, D13), (n, D14), (n, D15)

In this nomenclature:

(x, Dy) refers to Link x up to n starting at 0;

y= 0 through 15 representing the fields in each Link Descriptor that is a part of the LSR.

8.5.5 Link Cost

The Link Cost for each link is calculated based on the baud rate of the link, plus an administratively set factor. The link cost calculation is:

Link Cost = S * (1.0625e12 / Signalling Rate)

Where *S* is an administratively defined factor. By default *S* is set to 1.0.

NOTE 31 – This value allows an administrator to adjust link cost based on a particular environment.

This calculation shall be performed on a link by link basis. Each link in a Fabric may be advertised with a different cost. These costs shall be used by the path selection algorithm to determine the most efficient paths. If more than one least-cost path exists, use of the multiple least-cost paths is implementation specific.

NOTE 32 – For example, when the Link Cost is calculated for a 1.0625 GBit/s Fibre Channel Link, this calculation yields (with S set to 1.0): 1.0 * (1.0625e12 / 1.0625e9) = 1000.

8.6 Link State Database Synchronization

8.6.1 Synchronization Overview

The Link State Database shall be periodically synchronized across all Switches in the Fabric. This synchronization is required for the following reasons:

a) LSRs in the database may change because an ISL comes up or goes down;

b) Switches are added or removed from the Fabric;

c) LSRs may be added or removed;

d) Periodic issuance of new LSR instances.

Every time a new instance of an LSR or a new LSR is issued, the whole Fabric shall be informed. FSPF achieves this through reliable flooding of LSRs. A new instance of an LSR is transmitted to the directly attached Switches in a reliable fashion. The attached Switches in turn reliably forward the LSR to their attached Switches, and the process continues until all Switches in the Fabric have received the new instance of the LSR.

In addition, when an ISL between two Switches becomes operational and the Switches have successfully established two-way communication using the Hello protocol, the two Switches shall synchronize their database. Each Switch sends its full database to the other Switch, and receives the database from the other Switch. Each Switch updates its database with any received LSR that is either absent from its database, or is a newer instance of that LSR.

Generally, the initial synchronization and the ongoing database updates are implemented in the same fashion. One difference is that the initial synchronization involves the transmission of the whole database in two directions, whereas the ongoing updates typically involve only one or a few LSRs, and occur in one direction only. This characteristic of FSPF minimizes the number of different messages required by the protocol, and improves code reusability. Another difference is that the initial synchronization occurs on one ISL only, whereas the ongoing synchronization occurs on one or more ISLs.

8.6.2 Neighborhood and Adjacency

Two Switches connected via an ISL shall be referred to as neighbors. Two Switches that are simply neighbors shall not use their common ISL to carry Class N frames until their Link State Databases have been synchronized. Once the Link State Databases have been synchronized, the two Switches are referred to as Adjacent on that ISL. If two Switches are connected by multiple ISLs, they may be neighbors on some ISLs and Adjacent on others at any given time.

After a Switch detects that one of its ports is connected to another Switch, it starts exchanging Hello messages with its neighbor. Initially, a Switch knows only its own Domain_ID, and the Port Index of the port that connects the Switch to its neighbor. The Switch provides this information in the Hello message that it transmits to the neighbor. At this time the Switch does not know the Recipient Domain_ID on the neighbor Switch. Therefore, the Switch shall set the Domain_ID to FFFFFFFFh in the transmitted Hello message.

When a Hello message is received, the Switch stores the Domain_ID and Port Index of the neighbor Switch and shall send the Domain_ID in subsequent Hello messages on that ISL. When a Switch sees its own Domain_ID as the Recipient Domain_ID in a received Hello message, two-way communication is established on that ISL and Link State Database exchange shall be initiated. Upon detection of two-way communication, the Switch should send a Hello message immediately, rather than waiting for expiration of the Hello_Interval time.

The next step is to synchronize the Link State Databases on the two Switches. The original databases may be already identical if the two Switches are already connected by an ISL, and the new ISL is just an additional one. The two databases may be totally different if the ISL is used to connect two previously disjointed Fabrics. In some cases it is possible for some portions of the Data Base to be identical while other portions are not.

During the process of synchronizing databases, an algorithm determines the most recent of two instances of a database entry (i.e., LSR). Both Switches shall keep in their database only the most re-

cent version of an LSR. This algorithm is used both for the initial database synchronization process, and for any subsequent database update.

In the database synchronization phase, each Switch sends its complete Link State Database to the neighbor Switch. This topology information is transported as LSRs contained in one or more LSUs. Both Switches shall examine every LSR in the LSU, determine whether each is more recent than the associated instance in the database, and shall update the database accordingly. If the received instance of the LSR is more recent than the one contained in the database, or if the database did not contain the LSR in its database, then the received version of the LSR is stored in the database. LSRs shall be acknowledged by an LSA if they are newer or identical to the local copy, or no local copy exists. Otherwise an LSR is acknowledged by a newer LSR instance.

The receiving Switch shall acknowledge each LSR within an LSU separately. An acknowledgement for an LSR shall consist of the LSR header that uniquely identifies the instance of an LSR. Acknowledgements shall be sent in Link State Acknowledgement messages and an LSA may contain zero, one, or more acknowledgements. An LSA containing no LSR headers shall be used to acknowledge reception of the Database Complete flag from the neighbor, and shall confirm the end of the initial Link State Database synchronization process.

Unacknowledged LSRs shall be retransmitted by the sender after the Rxmt_Interval interval expires until they are acknowledged by the neighbor.

At the end of the process, both Switches have exchanged their topology data bases and they are considered Adjacent on that ISL. Any subsequent changes shall be communicated via LSU's. The ISL itself may then be used to carry user data. Both Switches shall issue a new LSR that includes the newly Adjacent ISL. This LSR shall be flooded reliably (see 8.6.4) on the new ISL, and on all other ISLs, together with any updated LSR.

After the new LSR has been transmitted and acknowledged on all of the Switch's ISLs, the Switch shall recompute the paths to all other Switches in the Fabric, and update the routing table accordingly. All the other Switches in the Fabric shall do the same, having received the new LSR.

The process of transmitting, receiving, processing, and acknowledging LSRs is identical for the initial database synchronization process, and for ongoing or periodic updates. The same messages and the same algorithm shall be used in both cases. This characteristic of FSPF simplifies the implementation, by reducing the number of different messages, and by improving code reusability.

The Adjacency bring-up process is detailed in this standard as a Finite State Machine (FSM) called the Neighbor FSM (see 8.7).

8.6.3 Continuous Link State Database Synchronization

After initial database synchronization with its neighbors, a Switch shall maintain a synchronized database through a continuous database synchronization process. Continuous database synchronization is achieved via reliable flooding of the LSRs. This assures that the databases reflect the current topology of the Fabric.

The current topology of the Fabric may change as a result of the following:

a) Inter-Switch Links changing state;

b) A Switch problem that causes it not to respond to Hello messages;

c) An ISL becoming operational and the neighbor FSM going to the Full state.

When a Switch detects a local Fabric topology change, it shall flood the Fabric with a new LSR.

A Switch shall issue a new LSR at the LS_Refresh_Time to ensure that the Switch shall delete entries in its database after the Max_Age interval expires if they are not refreshed. This allows for Switches that are permanently disconnected from the Fabric to be removed from the database. The periodic LSR update is independent of any other updates.

8.6.4 Reliable Flooding

8.6.4.1 Basic Operation

Reliable flooding is the mechanism by which topology changes are propagated throughout the Fabric. Reliable flooding shall be used whenever any change of a Switch link state occurs. Reliable flooding shall not be used for the initial database synchronization when an ISL between two Switches initializes. Normally, flooding occurs on all ISLs that are in the Full state at the same time, and not just between two Switches. Further, reliable flooding carries the new LSR(s) hop by hop to all Switches in the Fabric, whereas the initial database synchronization involves only two Switches.

Reliable flooding and initial database synchronization shall use LSU and LSA message structures for the updates.

8.6.4.2 The Flooding Procedure

The flooding procedure starts when a Switch issues a new instance of its LSR. The new Switch Link Record LSR for a Domain_ID shall only be issued by a Switch with that same Domain_ID.

The originating Switch shall package the LSR in an LSU and shall transmit it on all ISLs in the Full state. If there are other LSRs that are waiting to be acknowledged on an ISL, and the timer Rxmt_Interval for that ISL has expired, all the LSRs that have been waiting longer than Rxmt_Interval may be included in the LSU.

A receiving Switch shall acknowledge the LSR if appropriate. If it is a more recent instance than the one in its Link State Database, the Switch replaces the instance in the database with the new one. The Switch shall send the new LSR on all ISLs in the Full state, except the one from which the LSR was received. This step insures that the LSR is actually flooded throughout the Fabric.

If there are physical loops in the Fabric, a Switch may receive multiple instances of the same LSR update, even an instance that was originated by the Switch itself. A potential forwarding loop is prevented by forwarding only LSRs that are newer that the ones currently in the database. If a Switch receives an older instance of an LSR, or an LSR of the same instance as the one contained in the database, it shall acknowledge the LSR, and shall not forward the LSR to other Switches.

An LSR shall be acknowledged by sending the LSR header packaged in an LSA back to the sender. One or more LSRs may be acknowledged in the same LSA. The sender shall stop transmitting an LSR after it receives the acknowledgement.

8.6.4.3 Generating a New LSR

When a Switch generates an LSR, it shall set the LSR Age field to 0000h and increment the incarnation number by one. Typically different instances of an LSR have a different incarnation number that indicates a more recent instance. Under some circumstances this information is not sufficient, and other fields in the LSR shall be taken into account. The complete algorithm to determine the most recent incarnation between two LSRs is described below.

When a Switch first initializes, its Link State Database is empty because it has not recognized any neighbors yet. Link State Database information shall not be stored in non-volatile memory or be retrieved after a re-initialization. The Switch shall build a new database at every initialization or re-initialization.

A Switch generates its first LSR when the first ISL enters the Full state. The first LSR shall have an incarnation number of 80000001h. As other ISLs enter the Full state, the Switch shall generate a new incarnation of its LSR and shall increase the incarnation number by one every time. The LSR Age of a newly generated LSR shall always be 0000h.

After generating a new instance of an LSR, the Switch stores it into its Link State Database, and floods it to the rest of the Fabric.

8.6.4.4 Transmitting an LSR

An LSR shall be transmitted to a neighbor embedded in an LSU. Before transmission, the age of the LSR shall be incremented by 1. This value represents a nominal delay incurred by the LSR when it is transmitted.

NOTE 33 – The purpose of this increment is to prevent an LSR from being retransmitted forever (e.g., because of a software error).

The LSR shall be acknowledged by the receiving Switch. If the acknowledgement is not received within Rxmt_Interval the LSR shall be retransmitted. The LSR shall be retransmitted until an acknowledgment is received, or until the neighbor exits the Full state.

There shall be no distinction between the first transmission and subsequent retransmissions of an LSR except for the LSR Age field. The LSRs shall be identical with the exception of the LSR Age field. If a newer instance of the LSR being retransmitted is received, the newer instance shall replace the older instance in the Link State Database.

NOTE 34 – Since more than one LSR may be queued waiting for an acknowledgement, all of them may be transmitted in a single LSU for efficiency.

An LSR update shall not be sent more frequently than the Min_LS_Interval.

8.6.4.5 Receiving an LSR

An LSR is received in an LSU. After the processing of the LSU header, each LSR shall be processed separately and acknowledgements shall be provided separately for each LSR contained in the LSU. There is no specific acknowledgement to an LSU. Acknowledgements to multiple LSRs may be contained in a single LSA message.

Upon receipt, the following checks are performed in order:

a) The checksum is verified. If the checksum is incorrect, the LSR shall be ignored and no acknowledgement shall be returned;

b) The LSR Type is checked. If the type is not recognized, the LSR is ignored and no acknowledgement is returned;

c) If the LSR has an age equal to Max_Age, the LSR shall be stored in the local database long enough to flood it and receive acknowledgements if already present in the database or if the neighboring Switches are in the initial database synchronization process (i.e., the states Init,

Database Exchange, Database ACK Wait, Database Wait). The LSR is then deleted from the database. This ensures that when an LSR's age reaches Max_Age in any Switch, it is removed from the databases of all Switches simultaneously.

d) If Min_LS_Arrival has not expired, then the LSR is ignored and no acknowledgement is returned;

e) If there is no such LSR in the Link State Database, or the received LSR is a more recent incarnation than what is stored in the database, then the new LSR is installed in the database. If an LSR existed in the database, then the older incarnation is removed from the database, and an acknowledgement is returned.

f) If a Switch receives an LSR containing its own Domain_ID in the Link State identifier field, but with an incarnation number greater than its current incarnation number, the Switch shall set the incarnation number of its current LSR to the value in the received LSR plus one, and flood the resulting LSR on all links.

8.7 Neighbor Finite State Machine (FSM)

The Neighbor FSM initializes to Down state. In this state, the FSM is waiting for the notification that the port is connected to another Switch. This notification is issued internally to the Switch when a port reaches the E_Port status.

After the FSM receives the E_Port input, it transitions to Init state. In this state, attempts are made to determine if the other Switch supports FSPF. If it does, these attempts may result in the establishment of two-way communication between the two Switches, which is essential for the operation of the protocol. The Link State Databases on the two Switches are unable to be reliably synchronized in the absence of two-way communication. Successful two-way communication depends on configurable parameters matching between the two Switches.

After the two-way communication is established, a Switch knows the Domain_ID and the Port Index of the neighbor Switch on the opposite side of the ISL, and the FSM transitions to Database Exchange state.

In Database Exchange state, the two neighbor Switches share their view of the Fabric topology by exchanging their complete Link State Database. Each entry in the database is called a Link State Record (LSR). A Switch compares every LSR it receives from the neighbor to the same LSR in its database. If the received LSR is newer than the one present in the database, or if there is no LSR exists for that Domain in the database, then the received LSR is entered in the database. In the case where an LSR already exists in the database, the new LSR supersedes that LSR in the database. At the end of the process, both Switches shall have an identical Link State Database that consists of the most recent LSRs.

From Database Exchange state, the FSM may transition to two different states. If the next event is the reception of the end of database message from the neighbor, it transitions to Database Ack Wait state. If the next event is the reception of an ack to the end of database message, it transitions to Database Wait state.

From Database Ack Wait state, the FSM transitions to Full state when it receives an ack to the end of database message.

From Database Wait state, the FSM transitions to Full state when it receives the end of database message from the neighbor.

Once in Full state, the neighbor becomes an Adjacency, and the ISL that joins the two Adjacent Switches may be used to forward user data. Both Switches shall issue a new instance of their LSR to inform the rest of the Fabric about this fact.

The following aspects of the Neighbor FSM are described in detail below:

a) state by state;

b) the current state;

c) an input;

d) the new state;

e) actions taken in response to that input.

States are listed first, and for each state all the legal inputs are described. For ease of documentation, states are ordered. Each state in the list is considered greater than the previous one. The following states are defined, from lower to higher:

a) Down;

b) Init;

c) Database Exchange;

d) Database Ack Wait;

e) Database Wait;

f) Full.

An instance of the FSM shall run on each E_Port of the Switch.

Table 125 – Neighbor Finite State Machine (Part 1 of 3)

State	Input	Next State	Actions
Down	E_Port	Init	This input indicates that a port on the Switch is connected to another Switch. Send a Hello message to the neighbor. Start the Hello_Interval timer. The expiration of this periodic timer triggers the transmission of a Hello message to the neighbor.
Init	One-Way Received	Init	This input indicates that a Hello message that did not carry the correct Domain_ID of the local Switch has been received from the neighbor Switch. Start the Dead_Interval Timer. The expiration of the Dead_Interval Timer causes the Neighbor FSM to transition to Init State.

Table 125 – Neighbor Finite State Machine (Part 2 of 3)

State	Input	Next State	Actions
Init	Two-Way Received	Database Exchange	This input indicates that a Hello message carrying both the remote and the local Domain_ID has been received from the neighbor Switch. Send the Link State Database to the neighbor. The database may be sent in multiple frames, and even in multiple Fibre Channel Sequences. Restart the Dead_Interval Timer.
Database Exchange	Database Received	Database Ack Wait	This input indicates that an LSU with the Database Complete flag set has been received from the neighbor Switch. No action necessary, just a state transition.
Database Exchange	Database Sent	Database Exchange	This input indicates that all the Link State Records that describe the Link State Database have been sent to the neighbor. Send an LSU to the neighbor with no LSRs and the Database Complete flag set.
Database Exchange	Database Acked	Database Wait	This input indicates that an LSA with the Database Complete flag set has been received from the neighbor Switch. No action necessary, just a state transition.
Database Ack Wait	Database Sent	Database Ack Wait	This input indicates that all the Link State Records that describe the Link State Database have been sent to the neighbor. Send an LSU to the neighbor with no LSRs and the Database Complete flag set.
Database Ack Wait	Database Acked	Full	This input indicates that an LSA with the Database Complete flag set has been received from the neighbor Switch. Issue a new instance of the LSR, to reflect the new Adjacency. Compute the paths to all the Switches and program the routing tables.
Database Wait	Database Received	Full	This input indicates that an LSU with the Database Complete flag set has been received from the neighbor Switch. Issue a new instance of the LSR, to reflect the new Adjacency. Compute the paths to all the Switches and program the routing tables.
Any state except Down and Init	Two-Way Received	Same state	This input indicates that a Hello message carrying both the remote and the local Domain_ID has been received from the neighbor Switch. This is a normal periodic Hello message received from the neighbor. Restart the Dead_Interval Timer.

Table 125 – Neighbor Finite State Machine (Part 3 of 3)

State	Input	Next State	Actions
Any state except Down and Init	One-Way Received	Init	This input indicates that a Hello message that did not carry the correct Domain_ID of the local Switch has been received from the neighbor Switch. The retransmission lists shall be emptied, and all the timers associated with the retransmission lists shall be stopped.
Any state	Port Offline	Down	This input indicates that a port went offline. All the data structures related to the neighbor shall be removed. The retransmission lists shall be emptied, and all the timers associated with the neighbor shall be stopped. These include the retransmission timers, the Hello_Interval Timer and the Dead_Interval Timer. The same port may come back connected to a different Switch, or even as an F_Port, in which case the Neighbor FSM does not run. Issue a new instance of the LSR, that excludes this neighbor, to reflect the removal of an Adjacency.
Any state except Down	Hello_Interval	Same state	This input indicates that the Hello_Interval Timer has expired. Send a Hello message to the neighbor. In Down state Hello messages shall not be sent.
Any state except Down	Hello_Dead_ Interval	Init	This input indicates that the Dead_Interval Timer has expired and the port is still in E_Port state. Re-initialize the data structures associated with the neighbor. Stop the Dead_Interval Timer. Issue a new instance of the LSR, that excludes this neighbor, to reflect the removal of an Adjacency.
Full	Initial Database Received	Init	This input indicates that an LSU with the Initial Database Exchange flag set has been received from the neighbor Switch. Go into Init state and send One-Way Hellos.
Full	LSU/LSA Received	Full	Indicates that a LSU or LSA was received. The switch may reset the Dead_Interval Timer.

Figure 21 is a pictorial representation of the FSM where only the major state transitions are represented.

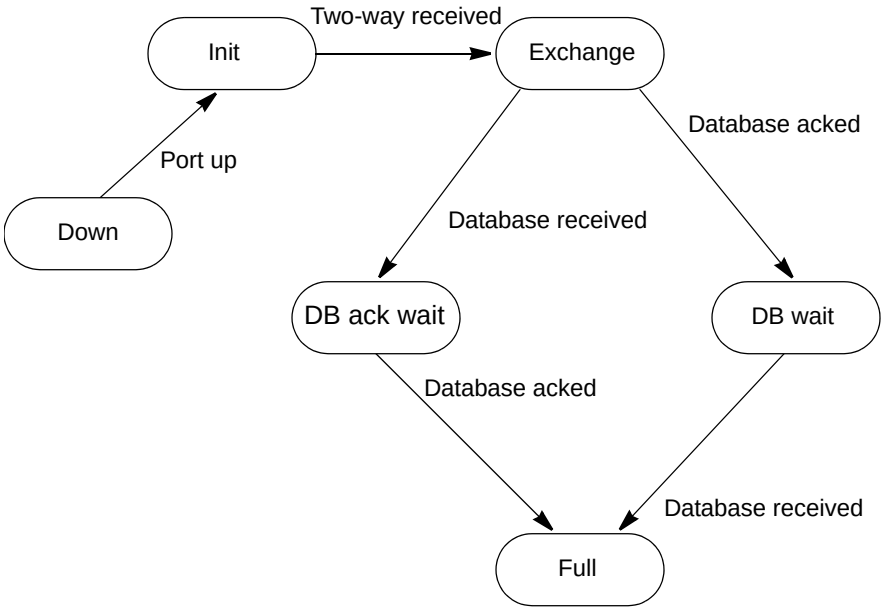

Figure 21 – Neighbor Finite State Machine

9 Distributed Services

9.1 Basic Model

A distributed services model is used to allow a Fabric to provide consistent services to all attached N_Ports. This Standard defines a common framework by which all Distributed Services communicate. Specific mappings onto this framework are also specified for the distributed Name Server and the distributed Management Server. Please note that in the following discussion it is convenient to say that a server is “contained” within a Switch. In this case the term “contain” does not imply that an entity is physically inside the Switch; it may be physically outside the Switch, and still operate as described below.

9.2 Distributed Services Framework

9.2.1 Goals and Characteristics of the Distributed Services Framework

All Distributed Services are mapped onto a common framework. The goal of this framework is threefold:

a) Define a consistent method for distributing services across Switches in a Fabric;

b) Define a distribution method that is topology independent;

c) Define a method that preserves processing facilities for existing frame formats.

In order to accomplish these goals this standard defines the following Distributed Server characteristics:

a) Transport;

b) Common Characteristics;

c) Work categories;

d) Frame formats.

9.2.2 Distributed Service Transport

9.2.2.1 Required FC-2 Parameters

Generic Service requests and responses are transported between Distributed Servers using the Common Transport (FC-CT) defined by FC-GS-7.

All CT frames shall be transmitted using the Class F service. The following defines the FC-2 header fields of all Distributed Services frames:

R_CTL: This field shall be set to 02h for all request frames, and to 03h for all reply frames.

CS_CTL: This field shall be set to 00h.

D_ID: This field shall be set to the Domain Controller Identifier of the destination Switch.

S_ID: This field shall be set to the Domain Controller Identifier of the source Switch.

TYPE: This field shall be set to 20h, indicating Fibre Channel Fabric Generic Services.

Each request shall be the first Sequence of an Exchange and the associated response shall be the last Sequence of the same Exchange. All other fields shall be set as appropriate according to the rules defined in FC-FS-3.

9.2.2.2 FC-CT Header Usage

The following values shall be set in the FC-CT Header for all Distributed Services requests and responses:

FC-CT revision: Obsolete in this standard, set to 01h.

IN_ID: The value of IN_ID in a Switch related request shall be preserved in all responses to that request. This only applies to an IN_ID value set by the Entry Switch.

Options: The X_Bit shall be set to 0 to indicate a single bidirectional exchange per request/response. The Partial Response bit shall be set to zero in Switch-to-Switch requests.

NOTE 35 – Multiple requests/responses may be active using multiple bidirectional exchanges between any pair of Switches.

9.2.2.3 Frame Distribution

It is important to note that for a Distributed Services request, a remote Switch shall never send a response directly to the requesting Nx_Port. All responses shall be sent to the Entry Switch. It is the responsibility of the Entry Switch to send the appropriate response to the requesting Nx_Port.

Furthermore, an Nx_Port shall always communicate to a Distributed Service via the Well-known address of the Distributed Service and Nx_Ports shall not send Distributed Service requests to Domain Controllers. In addition, Distributed Services request and responses are transported only between Switches and not between a Switch and an Nx_Port.

9.2.2.4 Domain Controller Service Parameters

The default values for Domain Controller communication should be as specified in table 126.

Table 126 – Default Domain Controller Service Parameters values

Item	Value
Receive Data Field Size	2112
End-to-End Credit	1
Concurrent Sequences	1
Open Sequences per Exchange	1

Optionally, the Domain Controller Service Parameters may be established using the ESS SW_ILS Domain Controller Capability Object (see 1.1). When a Switch supporting Domain Controller Service Parameter establishment joins a Fabric, it shall use the ESS SW_ILS (see 6.1.23.4.4) to determine the Domain Controller Service Parameters of the other Switches.

9.2.3 Common Characteristics

Each Distributed Service shares a set of common characteristics. These characteristics shall be defined as follows:

a) Timeouts: For requests between Switches, the time-out value shall be D_S_TOV.

b) Local Data Copies: Local data copies may be optionally allowed by a Distributed Service. If a Distributed Service allows local data copies it shall also specify the method by which the integrity of the local copied data is maintained.

c) Exchange Management: Each request between Switches shall be mapped to a unique exchange. Multiple outstanding requests are allowed between a pair of Switches up to the end-to-end credit resources specified by the receiving Switch.

d) Responses: Each request sent shall receive a response. If the receiving Switch is unable to perform a requested operation, then it shall respond with a Reject CT_IU specifying an appropriate Reason Code and Reason Code Explanation. If a response is not received from all Switches to which a request was sent within the time-out period, then the request shall be considered partial and a response shall be sent back to the Nx_Port as appropriate for the Service.

e) Partial Response: For many requests even a partial response to the requesting Nx_Port is useful. A partial response may occur for a number of reasons: 1) one of the Switches a request is directed to is busy and unable to respond within the time-out period, 2) one of the Switches a request is directed to does not support the service requested. A service may allow partial responses for a subset of its requests. If the response to a request is partial, the service shall set the partial response bit in the CT Header of the response back to the Nx_Port. This notifies the Nx_Port that the data in the response may not be complete.

NOTE 36 – There are legacy implementations that return an LS_RJT instead of a Reject CT_IU when a receiving Switch is unable to perform a requested operation.

f) Data Merge: Describes how data from multiple responses is consolidated.

g) Error Recovery: If an error on a Distributed Services frame is detected (e.g., No ACK, P_BSY), the frame may be retransmitted for a time interval up to D_S_TOV.

9.2.4 Zoning Considerations

If Zoning is present in a Fabric, Distributed Services may be affected. The following rules shall apply for Zoning with regard to Distributed Services:

a) Switch-to-Switch communications shall not be zoned. This only applies to the Class F CT Header based Distributed Services frames;

b) Zoning is applied by the Entry Switch. If a particular Distributed Service is affected by Zoning, it is the responsibility of the Entry Switch to make sure that a requesting Nx_Port does not receive data for that Distributed Service that is outside of the Nx_Port's Zone.

9.2.5 Work Categories

Work categories are definitions that allow consistent mapping of services to Distributed Services. These categories define how each Distributed Service maps its commands given the Distribution characteristic:

The work categories are defined below:

Local

Local requests are those that may be handled entirely by the Entry Switch. A request is local for the following reasons:

a) The data being requested is owned entirely by the Entry Switch. This situation would be dependant on the type of request;

b) The Entry Switch is maintaining a local copy of the data being requested. This situation may occur for any request depending on the local data copy rules of the Distributed Service to which the request belongs.

Any request that is determined to be local shall be processed as appropriate for the service as defined in FC-GS-7.

1-to-1

A 1-to-1 request is a request that is unable to be handled entirely by the Entry Switch, but for which the Entry Switch has identified a single remote Switch that may handle the request. The local Switch sends the request frame directly to the Domain Controller of remote Switch.

1-to-Many

A 1-to-Many request is a request that is unable to be handled entirely by the Entry Switch, but for which the Entry Switch has identified multiple remote Switches that may handle the request. The local Switch sends request frames directly to the Domain Controller of all remote Switches that it has identified to contain requested data.

1-to-All

A 1-to-All request is a request that is unable to be handled entirely by the Entry Switch, but for which the Entry Switch is unable to identify the set of remote Switches to query. The Entry Switch sends request frames directly to the Domain Controller of all Switches in the Fabric.

9.2.6 Frame Formats

Where possible Distributed Services should use the same frame formats for Switch-to-Switch communications as are used for Nx_Port requests.

9.2.7 FC-CT Command Restrictions

To avoid overlap of command codes associated with FC-CT commands originated external to the Fabric with FC-CT commands originated internal to the Fabric, the following FC-CT command codes shall not be used by any well-known server for the FC-GS-7 client/server interface.

Command codes 0400h-04FFh and E000h-EFFFh: Fabric Internal FC-CT Commands;

Command codes F000h-FFFFh: Vendor specific FC-CT Commands.

9.3 Distributed Name Server

9.3.1 General Behavior

The distributed Name Service is provided as follows:

a) Each Switch contains its own resident Name Server, called a distributed Name Server (dNS);

b) Each dNS within a Switch is responsible for the name entries associated with the Domain(s) assigned to the Switch;

c) Each dNS within a Switch shall only return information associated with the Domain(s) for which the Switch is responsible;

d) A client Nx_Port communicates its Name Service request (as defined in FC-GS-7) to the Entry Switch via the well-known address;

e) The dNS within the local Switch services the request by making any needed requests of other dNS's contained by the other Switches, if the required information is not available locally;

f) A dNS may maintain local data copies. Integrity of locally copied data is maintained via SW_RSCN notification. This implies that all Switches shall distribute SW_RSCN notification throughout the Fabric whenever a change takes place in their local dNS database;

g) The communication between dNS's to acquire the requested information is transparent to the original requesting client;

h) Partial responses to dNS queries are allowed. If an Entry Switch sends a partial response back to an Nx_Port it shall set the partial response bit in the CT Header.

9.3.2 FC-CT for Distributed Name Servers

9.3.2.1 dNS Command Codes

The Command Codes for FC-CT requests defined for dNS use are summarized in table 127. Codes 0100h - 0300h shall be as defined in FC-GS-7. All other requests are defined below. The format of the Entry field used in the following commands follow the Name Server Entry format described in 9.3.3.

Table 127 – FC-CT Command Codes for dNS (Part 1 of 4)

Encoded Value (hex)	Description	Mnemonic	Work Category	Payload Defined in	Notes
0100	Get all next	GA_NXT	1-to-All	FC-GS-7	
0101	Get Identifiers According to Scope	GID_A	1-to-1 or 1-to-Many	FC-GS-7	a
0112	Get Port_Name, based on Port Identifier	GPN_ID	1-to-1	FC-GS-7	
0113	Get Node_Name, based on Port Identifier	GNN_ID	1-to-1	FC-GS-7	
0114	Get Class of Service, based on Port Identifier	GCS_ID	1-to-1	FC-GS-7	
0117	Get FC-4 TYPEs, based on Port Identifier	GFT_ID	1-to-1	FC-GS-7	
0118	Get Symbolic Port_Name, based on Port Identifier	GSPN_ID	1-to-1	FC-GS-7	
011A	Get Port Type, based on Port Identifier	GPT_ID	1-to-1	FC-GS-7	
011B	Obsolete in FC-SW-5				
011C	Get Fabric Port_Name, based on Port Identifier	GFPN_ID	1-to-1	FC-GS-7	
011D	Get Hard Address, based on Port Identifier	GHA_ID	1-to-1	FC-GS-7	
011E	Obsolete				
011F	Get FC-4 Features - Port Identifier	GFF_ID	1-to-1	FC-GS-7	

a The GID_A request may either be 1-to-1 or 1-to-Many depending on the format of the request.

b Registration requests may be distributed if performed by a third party. Otherwise, they are local and outside the scope of this document.

c De-registration requests may be distributed if performed by a third party. Otherwise, they are local and outside the scope of this document.

d Work Categories for Name Server Entry Object requests are at the discretion of the originating Switch.

Table 127 – FC-CT Command Codes for dNS (Part 2 of 4)

Encoded Value (hex)	Description	Mnemonic	Work Category	Payload Defined in	Notes
0121	Get Port Identifier, based on Port_Name	GID_PN	1-to-All	FC-GS-7	
012B	Obsolete in FC-SW-5				
0131	Get Port Identifier, based on Node_Name	GID_NN	1-to-All	FC-GS-7	
0132	Get Port_Names based on Node_Name	GPN_NN	1-to-All	FC-GS-7	
0135	Obsolete in FC-SW-5				
0136	Get Initial Process Associator, based on Node_Name	GIPA_NN	1-to-All	FC-GS-7	
0139	Get Symbolic Node_Name, based on Node_Name	GSNN_NN	1-to-All	FC-GS-7	
0153	Obsolete in FC-SW-5				
0156	Obsolete in FC-SW-5				
0171	Get Port Identifiers, based on FC-4 TYPE	GID_FT	1-to-All	FC-GS-7	
0172	Get Port_Names, based on FC-4 TYPE	GPN_FT	1-to-All	FC-GS-7	
0173	Get Node_Names, based on FC-4 TYPE	GNN_FT	1-to-All	FC-GS-7	
01A1	Get Port Identifiers, based on Port Type	GID_PT	1-to-All	FC-GS-7	
01B1	Obsolete in FC-SW-5				
01B2	Obsolete in FC-SW-5				
01C1	Get Port Identifiers, based on Fabric Port_Name	GID_FPN	1-to-All	FC-GS-7	

[a] The GID_A request may either be 1-to-1 or 1-to-Many depending on the format of the request.

[b] Registration requests may be distributed if performed by a third party. Otherwise, they are local and outside the scope of this document.

[c] De-registration requests may be distributed if performed by a third party. Otherwise, they are local and outside the scope of this document.

[d] Work Categories for Name Server Entry Object requests are at the discretion of the originating Switch.

Table 127 – FC-CT Command Codes for dNS (Part 3 of 4)

Encoded Value (hex)	Description	Mnemonic	Work Category	Payload Defined in	Notes
01D1	Get Permanent Port_Name	GPPN_ID	1-to-All	FC-GS-7	
01F1	Get Port Identfiers, based on FC-4 Features	GID_FF	1-to-All	FC-GS-7	
01F2	Get Port_Identifer	GID_DP	1-to-All	FC-GS-7	
0212	Obsolete				
0213	Register Node_Name	RNN_ID	1-to-1	FC-GS-7	b
0214	Register Class of Service	RCS_ID	1-to-1	FC-GS-7	b
0217	Register FC-4 TYPEs	RFT_ID	1-to-1	FC-GS-7	b
0218	Register Symbolic Port_Name	RSPN_ID	1-to-1	FC-GS-7	b
021A	Obsolete				
021B	Obsolete in FC-SW-5				
021D	Register Hard Address - Port Identifier	RHA_ID	1-to-1	FC-GS-7	b
021E	Obsolete				
021F	Register FC-4 Features - Port Identifier	RFF_ID	1-to-1	FC-GS-7	b
0235	Register IP Address (Node)	RIP_NN	1-to-All	FC-GS-7	b
0236	Register Initial Process Associator	RIPA_NN	1-to-All	FC-GS-7	b
0239	Register Symbolic Node_Name	RSNN_NN	1-to-All	FC-GS-7	b
0300	De-register all	DA_ID	1-to-1	FC-GS-7	c
0410	Get Entry, based on Port Identifier	GE_ID	Any	FC-SW-6	d
0420	Get Entry, based on Port_Name	GE_PN	Any	FC-SW-6	d

a The GID_A request may either be 1-to-1 or 1-to-Many depending on the format of the request.

b Registration requests may be distributed if performed by a third party. Otherwise, they are local and outside the scope of this document.

c De-registration requests may be distributed if performed by a third party. Otherwise, they are local and outside the scope of this document.

d Work Categories for Name Server Entry Object requests are at the discretion of the originating Switch.

Table 127 – FC-CT Command Codes for dNS (Part 4 of 4)

Encoded Value (hex)	Description	Mnemonic	Work Category	Payload Defined in	Notes
0430	Get Entries, based on Node_Name	GE_NN	Any	FC-SW-6	d
0450	Obsolete in FC-SW-5				
0470	Get Entries, based on FC-4 TYPE	GE_FT	Any	FC-SW-6	d
04A0	Get Entries, based on Port Type	GE_PT	Any	FC-SW-6	d
04B0	Get Entries, based on Zone Member	GE_ZM	Any	FC-SW-6	
04C0	Get Entries, Based on Zone Name	GE_ZN	Any	FC-SW-6	
04D0	Obsolete in FC-SW-5				
04E0	Get Entries, Based on FC-4 Features	GE_FF	Any	FC-SW-6	
04F0	Get Entries, Based on Fabric Port_Name	GE_FPN	Any	FC-SW-6	
8001	Reject CT_IU	CT_RJT	1-to-1	FC-GS-7	
8002	Accept CT_IU	CT_ACC	1-to-1	FC-GS-7	

[a] The GID_A request may either be 1-to-1 or 1-to-Many depending on the format of the request.

[b] Registration requests may be distributed if performed by a third party. Otherwise, they are local and outside the scope of this document.

[c] De-registration requests may be distributed if performed by a third party. Otherwise, they are local and outside the scope of this document.

[d] Work Categories for Name Server Entry Object requests are at the discretion of the originating Switch.

9.3.2.2 FC-CT Header usage for dNS

The following FC-CT Header parameters, beyond those defined in 9.2.2.2, shall be used for dNS frames:

GS_Type: FCh (Directory Service application);

GS_Subtype: 02h (Name Service);

Command Code: see table 127.

9.3.3 Name Server Objects

The Name Server Objects communicated between distributed Name Servers using FC-CT are as defined in FC-GS-7 with no modification, but with several additions. The format of a Name Server Entry Object is as shown in table 128.

Table 128 – Name Server Entry Object

Mandatory	Item	Size Bytes	Comments
Yes	Entry Object Format Indicator	1	
Yes	Owner Identifier	3	
Yes	Port Type	1	
Yes	N_Port_ID	3	
Yes	N_Port_Name	8	
No	Port Symbolic Name	256	a
Yes	Node_Name	8	
No	Node Symbolic Name	256	a
Yes	Initial Process Associator	8	
Yes	Obsolete in FC-SW-5	16	
Yes	Class of Service	4	
Yes	FC-4 TYPEs	32	
Yes	Obsolete in FC-SW-5	16	
Yes	F_Port_Name	8	
Yes	Reserved	1	
Yes	Hard Address	3	
No	FC-4 Features	128	b

a This field is not present in the Small Name Server Entry Object.

b This field is not present in the Large or Small Name Server Entry Objects.

All fields shall be fixed length as indicated in the table 128. The Owner Identifier shall be the Domain Controller Identifier for the Switch that owns this Entry. All other fields shall be formatted as defined in FC-GS-7.

The Entry Object Format Indicator is depicted in table 129.

Table 129 – Entry Object Format Indicator

Bit	Description (Bit Value=1)
0	The Port Symbolic Name and Node Symbolic Name are not included in the Entry Object
1	The FC-4 Features are Included in the Entry Object
2-7	Reserved

The sizes of the Name Server Entry Object is depicted in table 130.

Table 130 – Name Server Entry Object Description

Value (Hex)	Length (Bytes)	Description
00	624	Large Name Server Entry Object
01	112	Small Name Server Entry Object
02	1012	Large Name Server Entry Object + FC-4 Features
03	500	Small Name Server Entry Object + FC-4 Features

The normal response to Get Entry requests in a distributed Name Server model returns one or more Name Server Entry Objects.

When a response to a request contains either a Port Symbolic Name or Node Symbolic Name that is greater than zero in length, and does not contain FC-4 Features, the Name Server Entry Object with an Entry Object Format Indicator of 00h shall be used by the responder.

The responder may return the Name Server Entry Object with an Entry Object Format Indicator of 01h if neither a Port Symbolic Name, or Node Symbolic Name is registered for the port and would result in those Name Server Objects being of length zero, and FC-4 Features have not been registered for the port.

When a response to a request contains either a Port Symbolic Name or Node Symbolic Name that is greater than zero in length, and contains FC-4 Features, the Name Server Entry Object with an Entry Object Format Indicator of 02h shall be used by the responder.

The responder shall return the Name Server Entry Object with an Entry Object Format Indicator of 03h if it would contain FC-4 Features, and does not contain a Port Symbolic Name or Node Symbolic Name.

9.3.4 FC-CT requests for dNS

9.3.4.1 Get Entry based on Port Identifier

The dNS shall, when it receives a GE_ID request, return the Entry object for the specified Port Identifier. The format of the GE_ID request is shown in table 131.

Table 131 – GE_ID request payload

Item	Size Bytes
FC-CT Header	16
Reserved	1
Port Identifier	3

The Port Identifier format shall be as defined in FC-GS-7. The dNS may reject a GE_ID request for reasons not specified in this document.

The format of the reply payload to a GE_ID request is shown in table 132.

Table 132 – GE_ID Accept payload

Item	Size Bytes
FC-CT Header	16
Number of Entries	4
Entry	n

Since this request returns only one Entry, the Number of Entries field shall always be set to one for this reply. The Entry field shall contain the Entry for the requested Port Identifier.

9.3.4.2 Get Entry based on Port_Name

The dNS shall, when it receives a GE_PN request, return the Entry object for the specified Port_Name. The format of the GE_PN request is shown in table 133.

Table 133 – GE_PN request payload

Item	Size Bytes
FC-CT Header	16
Port_Name	8

The Port_Name format shall be as defined in FC-GS-7. The dNS may reject a GE_PN request for reasons not specified in this document.

The format of the reply payload to a GE_PN request is shown in table 134.

Table 134 – GE_PN Accept payload

Item	Size Bytes
FC-CT Header	16
Number of Entries	4
Entry	n

Since this request returns only one Entry, the Number of Entries field shall always be set to one for this reply. The Entry field shall contain the Entry for the requested Port_Name.

9.3.4.3 Get Entries based on Node_Name

The dNS shall, when it receives a GE_NN request, return the Entry object for the specified Node_Name. The format of the GE_NN request is shown in table 135.

Table 135 – GE_NN request payload

Item	Size Bytes
FC-CT Header	16
Node_Name	8

The Node_Name format shall be as defined in FC-GS-7. The dNS may reject a GE_NN request for reasons not specified in this document.

The format of the reply payload to a GE_NN request is shown in table 136.

Table 136 – GE_NN Accept payload

Item	Size Bytes
FC-CT Header	16
Number of Entries	4
Entry 1	n
...	
Entry N	n

The Number of Entries field shall be set to the number of Entries returned. Each Entry field shall contain an Entry for the requested Node_Name.

9.3.4.4 Get Entries based on FC-4 TYPEs

The dNS shall, when it receives a GE_FT request, return the Entry object for the specified FC-4 TYPEs; note that more than one FC-4 TYPE may be specified. The format of the GE_FT request is shown in table 137.

Table 137 – GE_FT request payload

Item	Size Bytes
FC-CT Header	16
FC-4 TYPEs	32

The FC-4 TYPE format shall be as defined in FC-GS-7. The dNS may reject a GE_FT request for reasons not specified in this document.

The format of the reply payload to a GE_FT request is shown in table 138.

Table 138 – GE_FT Accept payload

Item	Size Bytes
FC-CT Header	16
Number of Entries	4
Entry 1	n
...	
Entry N	n

The Number of Entries field shall be set to the number of Entries returned. Each Entry field shall contain an Entry for the requested FC-4 TYPEs.

9.3.4.5 Get Entries based on Port Type

The dNS shall, when it receives a GE_PT request, return the Entry object for the specified Port Type. The format of the GE_PT request is shown in table 139.

Table 139 – GE_PT request payload

Item	Size Bytes
FC-CT Header	16
Reserved	3
Port Type	1

The Port Type format shall be as defined in FC-GS-7. The dNS may reject a GE_PT request for reasons not specified in this document.

The format of the reply payload to a GE_PT request is shown in table 140.

Table 140 – GE_PT Accept payload

Item	Size Bytes
FC-CT Header	16
Number of Entries	4
Entry 1	n
...	
Entry N	n

The Number of Entries field shall be set to the number of Entries returned. Each Entry field shall contain an Entry for the requested Port Type.

9.3.4.6 Get Entries based on Zone Member

The dNS shall, when it receives a GE_ZM request, return the Entry objects that are in the same zone as the Zone Member specified in the GE_ZM request. The format of the GE_ZM request is shown in table 141.

Table 141 – GE_ZM request payload

Item	Size Bytes
FC-CT Header	16
Zone Member	n

The Zone Member format shall be as defined in 10.4.4.6.1.

The format of the reply payload to a GE_ZM request is shown in table 142.

Table 142 – GE_ZM Accept payload

Item	Size Bytes
FC-CT Header	16
Number of Entries	4
Entry 1	n
...	
Entry N	n

The Number of Entries field shall be set to the number of Entries returned. Each Entry field shall contain an Entry for a Port that is in the same zone as the Zone Member specified in the GE_ZM request. Each Entry shall be only for ports local to the Switch to which the request was sent.

9.3.4.7 Get Entries based on Zone Name

The dNS shall, when it receives a GE_ZN request, return the Entry objects that are in the same zone as the Zone Name indicates in the GE_ZN request. The format of the GE_ZN request is shown in table 143.

Table 143 – GE_ZN request payload

Item	Size Bytes
FC-CT Header	16
Zone Name	n

The Zone Name format shall be as defined in 10.4.2.3.

The format of the reply payload to a GE_ZN request is shown in table 144.

Table 144 – GE_ZN Accept payload

Item	Size Bytes
FC-CT Header	16
Number of Entries	4
Entry 1	n
...	
Entry N	n

The Number of Entries field shall be set to the number of Entries returned. Each Entry field shall contain an Entry for a Port that is in the same zone as the Zone Name specified in the GE_ZN request. Each Entry shall be only for ports local to the Switch to which the request was sent.

9.3.4.8 Get Entries based on FC-4 Features

The dNS shall, when it receives a GE_FF request, return the Entry objects for the specified FC-4 features code specified in the GE_FF request. The format of the GE_FF request is shown in table 145.

Table 145 – GE_FF request payload

Item	Size Bytes
FC-CT Header	16
FC-4 Features	128

The format of the FC-4 Features value is as specified in FC-GS-7. The dNS may reject a GE_FF for reasons not specified in this standard.

The format of the reply payload to a GE_FF request is shown in table 146.

Table 146 – GE_FF Accept payload

Item	Size Bytes
FC-CT Header	16
Number of Entries	4
Entry 1	n
...	
Entry N	n

The Number of Entries field shall be set to the number of entries returned. Each Entry field shall contain an Entry for the requested FC-4 features code.

9.3.4.9 Get Entries based on Fabric Port_Name

The dNS shall, when it receives a GE_FPN request, return the Entry objects for the specified Fabric Port_Name specified in the GE_FPN request. The format of the GE_FPN request is shown in table 147.

Table 147 – GE_FPN request payload

Item	Size Bytes
FC-CT Header	16
Fabric Port_Name	8

The Fabric Port_Name format shall be as defined in FC-GS-7. The dNS may reject a GE_FPN request for reasons not specified in this document. The format of the reply payload to a GE_FPN request is shown in table 148.

Table 148 – GE_FPN Accept payload

Item	Size Bytes
FC-CT Header	16
Number of Entries	4
Entry 1	n
...	
Entry N	n

The Number of Entries field shall be set to the number of entries returned. Each Entry field shall contain an Entry for the requested Fabric Port_Name. There is one entry for an F_Port and more than 1 entry for an FL_Port.

9.4 Distributed Management Servers

9.4.1 General Behavior

Servers that comprise the Management Service are provided as follows:

a) For each Management Server of the Management Service, each Switch contains its own instance of the Server. Generically, each server instance is called a distributed Management Server (dMS);

b) Each dMS within a Switch is responsible for the entries associated with the Domain(s) assigned to the Switch;

c) A client Nx_Port communicates its Management Server request (as defined in FC-GS-7) to the Entry Switch via the well-known address and appropriate sub-type;

d) A dMS within the Entry Switch services the request by making any needed requests of other dMS instances contained by the other Switches, if the required information is not available locally;

e) A dMS may maintain local data copies, and a dMS shall notify other dMS that they should remove local data copies;

f) The communication between each dMS to acquire the requested information is transparent to the original requesting client;

g) Partial responses for some dMS requests are allowed. Partial response support is specified in the following clauses on a per dMS basis;

h) The responses returned to a client for some dMS servers are not subject to zoning as indicated in table 149.

Table 149 – Zoning effect on Servers of the distributed Management Service

Server	Subject to zoning
Fabric Configuration Server	No
Unzoned Name Server	No
Fabric Zone Server	No
Fabric Device Management Interface Server	No
Security Information Server[a]	
[a] The impact of Zoning on the Security Information Server is separately specified for each Security Information Request (see FC-GS-7).	

9.4.2 FC-CT Header

9.4.2.1 FC-CT Header Parameters

The following FC-CT Header parameters, beyond those defined in 9.2.2.2, shall be used for dMS frames:

CT_Type: FAh (Management Service);

CT_Subtype:

00h - Non-Server Specific;
01h - Fabric Configuration Server;
02h - Unzoned Name Server;
07h - Security Information Server;
10h - Fabric Device Management Server
others - Reserved.

Command Code: see tables 127, 149, 150, 154, 159, and 160.

9.4.2.2 FC-CT Header Rule for Fabric Internal Requests

For non-server specific requests, the GS_Subtype value shall be set to 00h.

9.4.3 Fabric Configuration Service

The FC-CT Command Codes defined for use by Fabric Configuration Service requests of the distributed Management Server are summarized in table 150.

Table 150 – Fabric Configuration Service Command Codes for dMS (Part 1 of 4)

Encoded Value (hex)	Description	Mnemonic	Work Category	Payload Defined in	Capability See 6.1.22.4.3
0100	Get Topology Information	GTIN	1-to-1	FC-GS-7	Topology
0101	Get Interconnect Element (IE) List[a]	GIEL	Local	FC-GS-7	Basic
0111	Get Interconnect Element Type	GIET	Local or 1-to-1	FC-GS-7	Basic
0112	Get Domain Identifier[a]	GDID	Local	FC-GS-7	Basic
0113	Get Management Identifier	GMID	Local or 1-1	FC-GS-7	Basic
0114	Get Fabric Name[a]	GFN	Local	FC-GS-7	Basic
0115	Get Interconnect Element Logical Name	GIELN	Local or 1-to-1	FC-GS-7	Basic
0116	Get Interconnect Element Management Address List	GMAL	Local or 1-to-1	FC-GS-7	Basic
0117	Get Interconnect Element Information List	GIEIL	Local or 1-to-1	FC-GS-7	Basic
0118	Get Port List	GPL	Local or 1-to-1	FC-GS-7	Basic
0121	Get Port Type	GPT	Local or 1-to-All	FC-GS-7	Basic

[a] These requests are handled by the Entry Switch with no assistance from other Switches.

[b] These requests function differently than the other requests (see 6.1.24 and FC-GS-7).

Table 150 – Fabric Configuration Service Command Codes for dMS (Part 2 of 4)

Encoded Value (hex)	Description	Mnemonic	Work Category	Payload Defined in	Capability See 6.1.22.4.3
0122	Get Physical Port Number	GPPN	Local or 1-to-All	FC-GS-7	Basic
0124	Get Attached Port_Name List	GAPNL	Local or 1-to-All	FC-GS-7	Basic
0126	Get Port State	GPS	Local or 1-to-All	FC-GS-7	Basic
0127	Get Port Speed Capabilities	GPSC	Local or 1-to-All	FC-GS-7	Basic
0128	Get Attached Topology Information	GATIN	Local or 1-to-All	FC-GS-7	Topology
0130	Get Switch Enforcement Status	GSES	Local or 1-to1	FC-GS-7	Enhanced
0140	Get Interconnect Element Attribute Group	GIEAG	Local or 1-to1	FC-GS-7	Enhanced
0141	Get Port Attribute Group	GPAG	Local or 1-to1	FC-GS-7	Enhanced
0191	Get Platform Node_Name List	GPLNL	Local or 1-to-All	FC-GS-7	Platform
0192	Get Platform Type	GPLT	Local or 1-to-All	FC-GS-7	Platform
0193	Get Platform Management Address List	GPLML	Local or 1-to-All	FC-GS-7	Platform

[a] These requests are handled by the Entry Switch with no assistance from other Switches.

[b] These requests function differently than the other requests (see 6.1.24 and FC-GS-7).

Table 150 – Fabric Configuration Service Command Codes for dMS (Part 3 of 4)

Encoded Value (hex)	Description	Mnemonic	Work Category	Payload Defined in	Capability See 6.1.22.4.3
0197	Get Platform Attribute Block	GPAB	Local or 1-to-All	FC-GS-7	Platform
01A1	Get Platform Name - Node_Name	GNPL	Local or 1-to-All	FC-GS-7	Platform
01A2	Get Platform List	GPNL	Local or 1-to-All	FC-GS-7	Platform
01A4	Get Platform FCP Type	GPFCP	Local or 1-to-All	FC-GS-7	Platform
01A5	Get Platform OS LUN Mappings	GPLI	Local or 1-to-All	FC-GS-7	Platform
01B1	Get Node Identification Data - Node_Name	GNID	Local or 1-to-All	FC-GS-7	Platform
215	Register Interconnect Element Logical Name	RIELN	Local or 1-to-1	FC-GS-7	Basic
0280	Register Platform	RPL	Local or 1-to-All	FC-GS-7	Platform
0291	Register Platform Node_Name	RPLN	Local or 1-to-All	FC-GS-7	Platform
0292	Register Platform Type	RPLT	Local or 1-to-All	FC-GS-7	Platform
0293	Register Platform Management Address	RPLM	Local or 1-to-All	FC-GS-7	Platform
0298	Register Platform Attribute Block	RPAB	Local or 1-to-All	FC-GS-7	Platform
029A	Register Platform FCP Type	RPFCP	Local or 1-to-All	FC-GS-7	Platform
029B	Register Platform OS LUN Mappings	RPLI	Local or 1-to-All	FC-GS-7	Platform

[a] These requests are handled by the Entry Switch with no assistance from other Switches.

[b] These requests function differently than the other requests (see 6.1.24 and FC-GS-7).

Table 150 – Fabric Configuration Service Command Codes for dMS (Part 4 of 4)

Encoded Value (hex)	Description	Mnemonic	Work Category	Payload Defined in	Capability See 6.1.22.4.3
0380	Deregister Platform	DPL	Local or 1-to-All	FC-GS-7	Platform
0391	Deregister Platform Node_Name	DPLN	Local or 1-to-All	FC-GS-7	Platform
0392	Deregister Platform Management Address	DPLM	Local or 1-to-All	FC-GS-7	Platform
0393	Deregister Platform Management Address List	DPLML	Local or 1-to-All	FC-GS-7	Platform
0394	Deregister Platform OS LUN Mappings	DPLI	Local or 1-to-All	FC-GS-7	Platform
0395	Deregister Platform Attribute Block	DPAB	Local or 1-to-All	FC-GS-7	Platform
039F	De-Register All Platform Information	DPALL	Local or 1-to-All	FC-GS-7	Platform
0400	FC Trace Route	FTR	NA[b]	FC-GS-7	Topology
0401	FC Ping	FPNG	NA[b]	FC-GS-7	Topology

[a] These requests are handled by the Entry Switch with no assistance from other Switches.

[b] These requests function differently than the other requests (see 6.1.24 and FC-GS-7).

9.4.4 Unzoned Name Service

The Distributed Service associated with the Unzoned Name Service, shall be identical to the services defined by the dNS in 9.3. These services are classified as the Basic Unzoned Name service.

9.4.5 Fabric Zone Service

The Fabric Zone Service is described in clause 10.

9.4.6 Fabric-Device Management Service

9.4.6.1 Operational Characteristics of the FDMI Server

As with the other servers, the FDMI server is distributed, but it has additional requirements placed upon it because an HBA may register through ports attached to different switches. This raises the possibility that multiple switches in the Fabric may be able to manage information for the same HBA. Since it is desirable that only one Switch manage an HBA's information, the FDMI server requires that a mechanism be provided above and beyond the normal distribution and caching mechanisms provided for other servers.

Switches in the Fabric may contain and manage a portion of the FDMI database. Each Switch that manages a portion of the FDMI database participates with other switches that manage portions of the FDMI database through the exchange of a new FDMI Inter-Switch messages. Each Switch that manages a portion of the FDMI database may also maintain cached information associated with portions of the FDMI database on other switches.

Each HBA attached to the Fabric has one Switch that functions as its Principal Manager. This ensures that only one Switch manages information on behalf of a given HBA. Through the exchange of FDMI Inter-Switch messages, switches resolve which one becomes the Principal Manager for each HBA. This requires that each Switch with HBAs attached maintain a map that associates its attached HBA's to the Principal Manager for each HBA.

The GS Client refers to the entity that issues an FDMI request to the HBA Management Server via the well-known management server address.

9.4.6.2 Registration Scenarios

9.4.6.2.1 HBA Attached to a Single Switch

In the simple case, an HBA is attached to only one Switch, possibly through multiple ports. The HBA attempts to register with the Switch and the Switch performs the checks as mandated by the FDMI interface specification in FC-GS-7. The Switch uses information contained in its local database and its cache entries to perform these checks. If the checks complete successfully then the HBA information is registered with the Switch and the HBA is notified of successful registration. If subsequent registrations are attempted over additional ports attached to the same Switch, the Switch would reject those requests because the HBA information is already contained in its FDMI database. Following the successful registration of HBA information with the Switch, the Switch may forward the information to other switches in the Fabric allowing other switches to update their caches.

9.4.6.2.2 HBA Attached to Multiple Switches

In the more complicated case, an HBA is attached to multiple switches through multiple ports. Since the switches at this point may have not had time to update each others caches between registrations, the HBA registration may pass the checks in multiple switches. This means that multiple switches are managing the information for the same HBA in their respective databases. This is not desirable because inconsistencies may be introduced into the FDMI database when multiple switches manage information for the same HBA.

9.4.6.2.3 Resolution of the Principal HBA Manager

When multiple switches allow the registration of information for a particular HBA, the switches shall resolve which Switch acts as the principal manager for the HBA. The Switch with the lowest Switch_Name shall become the principal HBA manager. Switches that have accepted registrations from an HBA exchange FDMI messages that contain the associated Switch_Name. Switches participating in the protocol determine from the Switch_Names which Switch serves as the principal HBA manager for the HBA. All this ensures that one and only one Switch serves as the principal HBA manager for a given HBA, even if the HBA is attached to other switches. The protocol that determines the principal manager runs immediately following a successful HBA registration (see annex C).

9.4.6.3 FDMI Inter-Switch Messages

9.4.6.3.1 General Format

FDMI Inter-Switch messages are exchanged between switches using the Inter-Switch FC-CT. The general format of the FDMI Inter-Switch message is depicted below:

Table 151 – FDMI Inter-Switch Message

Item	Size Bytes
FC-CT Header	See FC-GS-7
FDMI Header	28
Payload	n

9.4.6.3.2 FC-CT Header

The FC-CT header follows the format specified in FC-GS-7. The following values are specified for the GS_Type, GS_Subtype, and Command/Response fields:

GS_Type: FAh

GS_Subtype: 10h

Command/Response Code: See 9.4.6.4.

9.4.6.3.3 FDMI Header

The format of the FDMI header is described below:

Table 152 – FDMI Header

Item	Size Bytes
FDMI Version	1
Reserved	3
Switch_Name	8
Vendor Specified	16

FDMI Version: This field represents the version of the FDMI Header. The only value allowed is 01h. All other values are reserved.

Switch_Name: This field contains the Switch_Name for the Switch that originated the FDMI CT operation.

Vendor Specified Field:

The format of the vendor specified field is depicted in table 153.

Table 153 – Vendor Specified

Item	Size (Bytes)
Vendor ID	8
Vendor Specified Information	8

Vendor ID: Contains the T10 Vendor ID of the vendor that defines the content of Vendor Specified Information field.

Vendor Specified Information: This field contains 8 bytes of vendor specified information. The processing of the Vendor Specified information shall be subject to the following rules:

a) If the information contained in the Vendor Specified Information field is not recognized or processed by the server, then the command proceeds as defined;

b) For any FDMI command defined in the standard, the Vendor Specified information shall not cause the server to exhibit any behavior different from that defined for the command.

9.4.6.3.4 Payload

The Payload field is either null or contains the GS Client Payload depending on the FDMI Inter-Switch request see table 154. The GS Client payload includes the entire CT Request that was received by the entry Switch from the HBA, including the CT Header.

9.4.6.4 FDMI Inter-Switch Requests

FDMI Inter-Switch requests are mapped to Request CT_IUs. The following table indicates the operations performed by the HBA Management Server and indicates their associated command codes and payload contents.

Table 154 – FDMI Fabric Internal Command Codes (Part 1 of 2)

Code	Mnemonic	Description	Request Attributes	Accept Attributes
E100	FDRN	De-Registration Notification	FDMI Header, GS Client Payload	Null
E101	FRN	Registration Notification	FDMI Header, GS Client Payload	Null
E102	FUN	Update Notification	FDMI Header, GS Client Payload	Null

Table 154 – FDMI Fabric Internal Command Codes (Part 2 of 2)

Code	Mnemonic	Description	Request Attributes	Accept Attributes
E103	FDRF	De-Registration Forward	FDMI Header, GS Client Payload	Null
E104	FUF	Update Forward	FDMI Header, GS Client Payload	Null
E105	FETCH	Fetch	FDMI Header	HBA/Port List
E106-E10F		Reserved		

9.4.6.5 FDMI Inter-Switch Responses

9.4.6.5.1 Reject Response

When the destination Switch is unable to perform a requested operation, an HBA Management Server Reject CT_IU is sent to the originating Switch. HBA Management Server Reject CT_IUs specify a reason code of x'09' (Unable to perform command request).

Table 155 – Reason Code Explanation

Encoded Value (hex)	Description
00	No Additional Explanation
E0	Fetch Unsuccessful
others	*Reserved*

9.4.6.5.2 Accept Response

When the destination Switch has successfully performed the requested operation, an HBA Management Accept CT_IU is sent to the originating Switch indicating completion of the requested operation, and containing any response information associated with the requested operation.

9.4.6.6 FDMI Inter-Switch Operations

9.4.6.6.1 Registration Notification (FRN) Operation

When the HBA Management Server on the entry Switch registers HBA information in its FDMI database, the HBA Management Server shall send the FRN request to all switches in the Fabric. The FRN request payload shall specify the FDMI header, and the original CT request from the GS client. The FRN accept payload shall be null.

9.4.6.6.2 De-Register Notification (FDRN) Operation

When the HBA Management Server on the entry Switch de-registers HBA information in its FDMI database, the HBA Management Server shall send the FDRN request to all switches in the Fabric. The FDRN request payload shall specify the FDMI header, and the original CT request from the GS client. The FDRN accept payload shall be null.

9.4.6.6.3 Update Notification (FUN) Operation

When the HBA Management Server on the entry Switch updates HBA information in its FDMI database, the HBA Management Server shall send the FUN request to all switches in the Fabric. The FUN request payload shall specify the FDMI header, and the original CT request from the GS client. The FUN accept payload shall be null.

9.4.6.6.4 Update Forward (FUF) Operation

When the HBA Management Server on the entry Switch receives a request to update HBA information, but the Switch is not the Principal HBA Manager for the specified HBA, the HBA Management Server shall send the FUF request to the Switch that is the Principal HBA Manager for the specified HBA. The FUF request payload shall specify the FDMI header, and the original CT request from the GS client. The FUF accept payload shall be null.

9.4.6.6.5 De-Register Forward (FDRF) Operation

When the HBA Management Server on the entry Switch receives a request to de-register HBA information, but the Switch is not the Principal HBA Manager for the specified HBA, the HBA Management Server shall send the FDRF request to the Switch that is the Principal HBA Manager for the specified HBA. The FDRF request payload shall specify the FDMI header, and the original CT request from the GS client. The FDRF accept payload shall be null.

9.4.6.6.6 Fetch

When a Switch becomes part of the Fabric (e.g., result of a Merge), the HBA Management Server shall send the FETCH request to all Switches in the Fabric to obtain their Registered HBA/Port lists. The FETCH request payload shall specify the FDMI header. The FETCH accept payload shall return the Registered HBA/Port list. The format of the Registered HBA/Port list is shown below.

Table 156 – Registered HBA/Port List

Item	Size (Bytes)
Number of HBA Entries (n)	4
HBA Entry 1	x
HBA Entry 2	y
...	...
HBA Entry n	z

Number of HBA Entries: This field specifies the number of HBA entries contained in the Registered HBA/Port list.

HBA Entry: The format of the HBA Entry is depicted in table 157 below.

Table 157 – HBA Entry

Item	Size (Bytes)
HBA Identifier	8
Number of Port Entries (m)	4
Port Entry 1	8
Port Entry 2	8
...	...
Port Entry n	8

Number of Port Entries: This field specifies the number of Port entries for the specified HBA.

Port Entry: The format of the Port Entry is depicted in table 158 below.

Table 158 – Port Entry

Item	Size (Bytes)
Port_Name	8

9.4.6.7 GS Client Initiated FDMI Requests

In addition to the Fabric originated FDMI operations, there are GS client initiated FC-CT commands that are forwarded to other switches in the Fabric by the entry Switch. The FC-CT Command Codes

defined for use by Fabric Device Management Interface requests of the Distributed HBA Management Server are summarized in table 159.

Table 159 – Fabric Device Management Interface CT Commands for the dMS

Encoded Value (hex)	Description	Mnemonic	Work Category	Payload Defined in
0100	Get Registered HBA List	GRHL	Local or 1 to Many	FC-GS-7
0101	Get HBA Attributes	GHAT	Local or 1-to-1	FC-GS-7
0102	Get Registered Port List	GRPL	Local or 1-to-1	FC-GS-7
0110	Get Port Attributes	GPAT	Local or 1-to-1	FC-GS-7
0200	Register HBA	RHBA	Local	FC-GS-7
0201	Register HBA Attributes	RHAT	Local	FC-GS-7
0210	Register Port	RPRT	Local	FC-GS-7
0211	Register Port Attributes	RPA	Local	FC-GS-7
0300	De-Register HBA	DHBA	Local	FC-GS-7
0301	De-Register HBA Attributes	DHAT	Local	FC-GS-7
0310	De-Register Port	DPRT	Local	FC-GS-7
0311	De-Register Port Attributes	DPA	Local	FC-GS-7

9.4.7 Other Fabric Internal Services

9.4.7.1 Fabric Internal Requests

Fabric internal FC-CT for the distributed Management Server are shown in table 160.

Table 160 – Fabric Internal Management Server Operations

Code	Mnemonic	Description	Request Attributes	Accept Attributes
E020	GCAP	Get Management Server Capabilities	None	Management Server Capabilities
E100-E10F		Reserved for Inter-Switch FDMI Use (See 9.4.6.4)		

9.4.7.2 Get Management Server Capabilities (GCAP) Operation

The GCAP operation allows a management server instance on one Switch to query the management server capabilities of a management server instance on another Switch in the Fabric.

The responding distributed Management Server shall, when it receives a GCAP operation request, return its capabilities. The GCAP request payload shall be null. The GCAP accept payload contains the requested Management Server Capabilities.

Table 161 – GCAP Request Payload

Item	Size (Bytes)
Null	0

Table 162 – GCAP CT_ACC Payload

Item	Size (Bytes)
Number of Capability Entries (n)	4
Capability Entry 1	8
Capability Entry 2	8
...	8
Capability Entry n	8

9.4.7.2.1 Capability Entry

The format of the capability entry is shown below. The Management Server Subtype indicates the service. The Capability mask designates the supported capabilities of the service.

Table 163 – Capability Entry

Item	Size (Bytes)
Management Server GS_Subtype	1
Vendor Specific Capability Bit Mask	3
Subtype Capability Bit Mask	4

Management Server GS_Subtype: This field shall indicate the Management Server associated with the capabilities in the entry.

Vendor Specific Capability Bit Mask: This field shall indicate any vendor specific capabilities associated with the designated Management Server. The format of the field is not defined by this standard.

Subtype Capability Bit Mask: This field shall indicate capabilities associated with the designated Management Server.

9.4.7.2.2 Subtype Capability Bit Masks

The Capability Bit Masks currently defined by this standard are listed below.

Table 164 – Fabric Configuration Server (CT_Subtype 01h)

Option Bit(s)	Description (see 6.1.22.4.3)
0	Basic Configuration Services
1	Platform Configuration Services
2	Topology Discovery Configuration Services
3	Enhanced Configuration Services
4-31	Reserved

Table 165 – Unzoned Name Server (CT_Subtype 02h)

Option Bit(s)	Description (see 9.4.4)
0	Basic Unzoned Name Services
1-31	Reserved

9.4.8 Security Information Server

The FC-CT command codes defined for use by Security Information Server requests are summarized in table 166.

Table 166 – Security Information Server Command Codes for dMS

Encoded Value (hex)	Description	Mnemonic	Work Category	Payload Defined in
0001	Get Authorization State for Port Identifier	GAS_ID	Local or 1 to 1	FC-GS-7

9.5 Distributed Event Server

9.5.1 General Behavior

The Distributed Event Server is provided as follows:

a) Each Switch contains its own resident Event Server, called a distributed Event Server (dES);

b) Each dES within a Switch is responsible for the registrations and notifications associated with the Domain(s) assigned to the Switch;

c) Each dES within a Switch shall only originate events associated with the Domain(s) for which the Switch is responsible;

d) A client Nx_Port communicates its Event Server requests (as defined in FC-GS-7) to the Entry Switch via the well-known address.

9.5.2 FC-CT for Distributed Event Server

9.5.2.1 FC-CT Header Parameters

The following FC-CT Header parameters, beyond those defined in 9.2.2.2, shall be used for dES frames:

CT_Type: F4h (Event Service);

CT_Subtype: 01h

9.5.2.2 dES Command Codes

The Command Codes for FC-CT requests defined for dES use are summarized in table 167.

Table 167 – FC-CT Command Codes for dES

Encoded Value (Hex)	Description	Work Category	Payload Defined in
0100	Event Registration	1-to-All	FC-GS-7
0101	Event Notification	1-to-All	FC-GS-7

10 Switch Zone Exchange & Merge

10.1 Overview

This clause describes a mechanism for Switches to exchange zoning data. FC-GS-7 contains a description of the Fabric Zoning Service architecture and management requests for administering zoning.

When link parameters have been established for a link and the Switches have a Domain_ID, the two Switches joined by this link exchange Zoning Configuration information to make the information consistent across the Fabric. The Fabric Management inter-switch messages that are addressed to a Fabric Controller are the Merge request and Merge response. These messages are used to resolve the Zoning Configuration in a Fabric when two Switches are joined. Each Switch determines if the Zoning Configuration from the adjacent Switch may be merged with its local Zoning Configuration. The rules for merging Zoning Configurations are described in 10.5.2.

NOTE 37 – This protocol is designed to work when a single inter-switch link is established. Establishing more than one inter-switch link at the same time may lead to unpredictable effects over the Fabric.

10.2 Joining Switches

Merge request and Merge response messages are used to merge and propagate Zoning Configurations when an inter-Switch link becomes available. A merge operation is performed with the adjacent Switch when an inter-Switch link becomes available, and with all adjacent Switches when changes are made to the local Zoning Configuration as a result of merging the local Zoning Configuration with an adjacent Zoning Configuration.

A Merge request message contains the local Zoning Configuration of the Switch that is generating the Merge request, together with a Protocol Version field that defines the format of the Zoning Protocol used by the Switch, (e.g., Enhanced or Basic).

When a Merge request is received from an adjacent Switch, the receiving Switch determines if the request may be accepted and executed. If the Switch is not busy, but there is a Protocol Version mismatch, or, when the Protocol version matches, the merge is unable to be executed according to the rules described in 10.5.2, a Merge response is returned indicating that the Zone Configurations are unable to be merged, and the inter-Switch link is isolated. If the Switch is not busy and the merge may be executed, a Merge response is returned indicating that the Zone Configurations were successfully merged. After the successful merge, the Zone Set Name is changed to "Successful Zone Set Merge: Active Zone Set Name has changed".

This information exchange may start by sending Merge Request Resource Allocation messages to allocate the resources needed to process the Merge Request Sequences. An example of the Merge data flow is provided in 10.5.1.

10.3 Enhanced Zoning Support Determination

When a Switch supporting Enhanced Zoning joins a Fabric, it shall use the ESS SW_ILS (see 6.1.22.4.4) to determine the Enhanced Zoning capabilities of the other Switches. By doing so, the Switch also announces its Enhanced Zoning capabilities to the other Switches of the Fabric.

Each Switch supporting Enhanced Zoning shall maintain the information about the Enhanced Zoning support by all Switches in the Fabric. This information is updated whenever a Switch joins or leaves the Fabric, and it is used to reply to the GFEZ request.

If all the Switches in the Fabric support Enhanced Zoning, then the Enhanced Zoning supported bit (bit 0) of the Fabric Enhanced Zoning support flags of the GFEZ Accept shall be set to one, otherwise it shall be set to 0.

The value of the Enhanced Zoning enabled bit (bit 1) of the Fabric Enhanced Zoning support flags of the GFEZ Accept is instead determined with the Zone Merge protocol because a Switch is able to join a Fabric only if it is working in the same mode of the Fabric see 6.1.15).

If all the Switches in the Fabric support the Zone Set Database, then the Zone Set Database supported bit (bit 4) of the Fabric Enhanced Zoning support flags of the GFEZ Accept shall be set to one, otherwise it shall be set to 0.

10.4 Zoning Framework and Data Structures

10.4.1 Basic Zoning Framework

This clause provides an overview of the Basic Zoning framework associated with the Switch Fabric. The Basic Zoning framework describes zoning entities such as Zoning Objects, Object Members, Member types, and their relationships. The figures below depict the Basic Zoning framework.

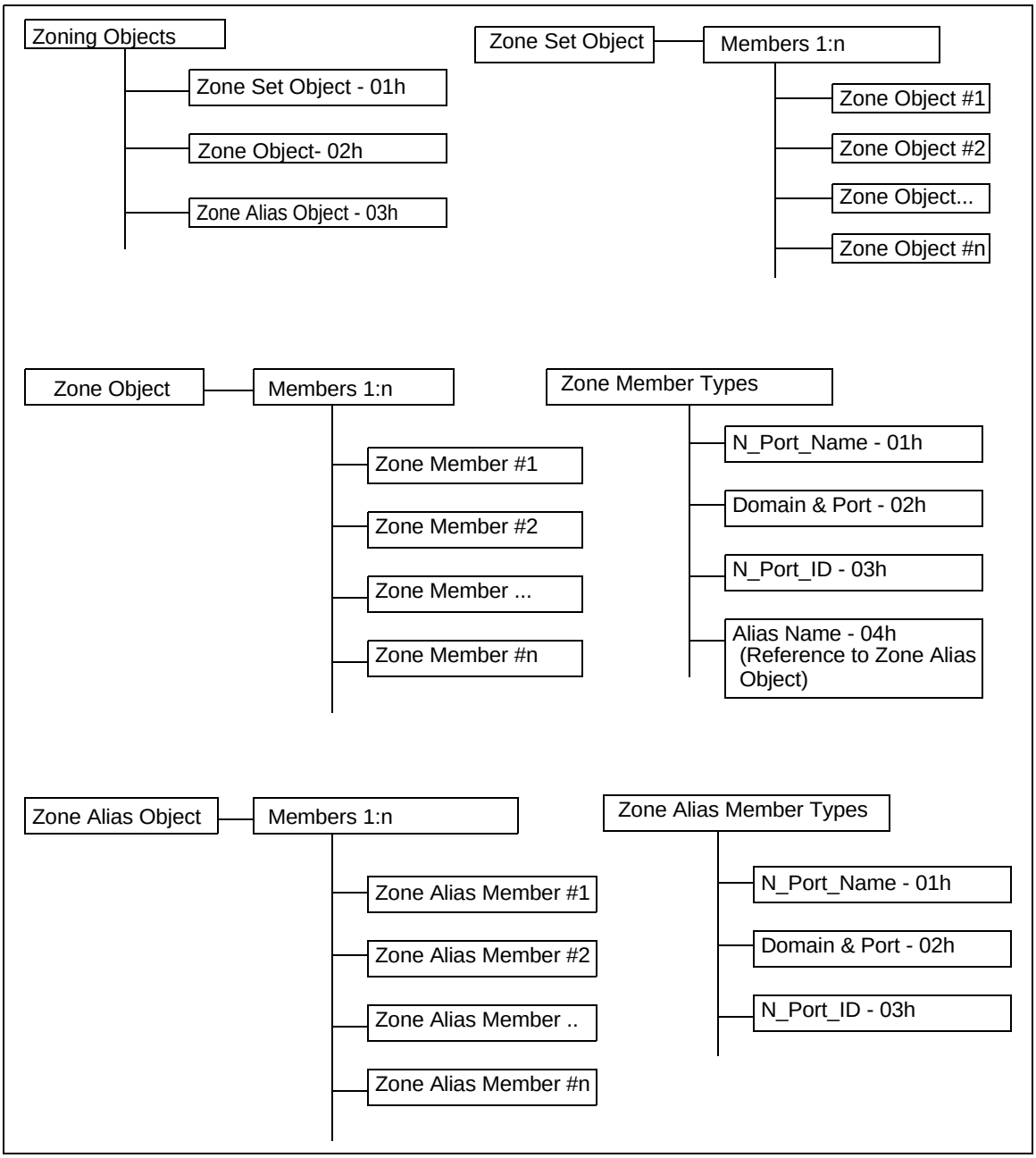

Figure 22 – Basic Zoning Framework

Zoning Objects: Three types of Zoning Objects are defined. They are:

a) Zone Set;

b) Zone;

c) Zone Alias.

Zone Set Object: The Zone Set object defines a group of Zones. A Zone Set object contains one or more members that are Zone objects. In addition, the Zone Set object has two attributes:

a) Name;

b) Number of Members.

Zone Object: The Zone Object defines a Zone and its members. A Zone object contains one or more Zone Members. Currently defined Zone Member Types are listed below:

a) N_Port_Name;

b) Domain_ID and physical port;

c) N_Port_ID;

d) Zone Alias Name;

The Zone Object has three attributes:

a) Name;

b) Protocol Type;

c) Number of Members.

Zone Alias Object: A Zone Alias object defines a Zone Alias and its members. A Zone Alias object contains one or more Zone Alias Members. Currently defined Zone Alias Member Types are listed below:

a) N_Port_Name;

b) Domain_ID and physical port;

c) N_Port_ID;

The Zone Alias Name specified as a Zone Member serves as a reference to a Zone Alias object.

The Zone Alias object has two attributes:

a) Name;

b) Number of Members.

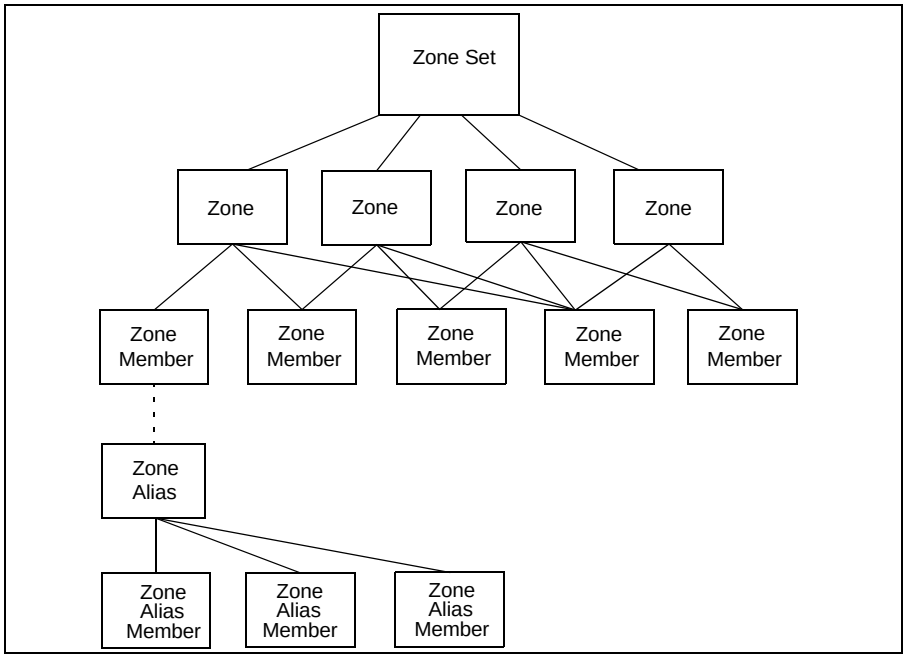

Figure 23 – Basic Zoning Hierarchy

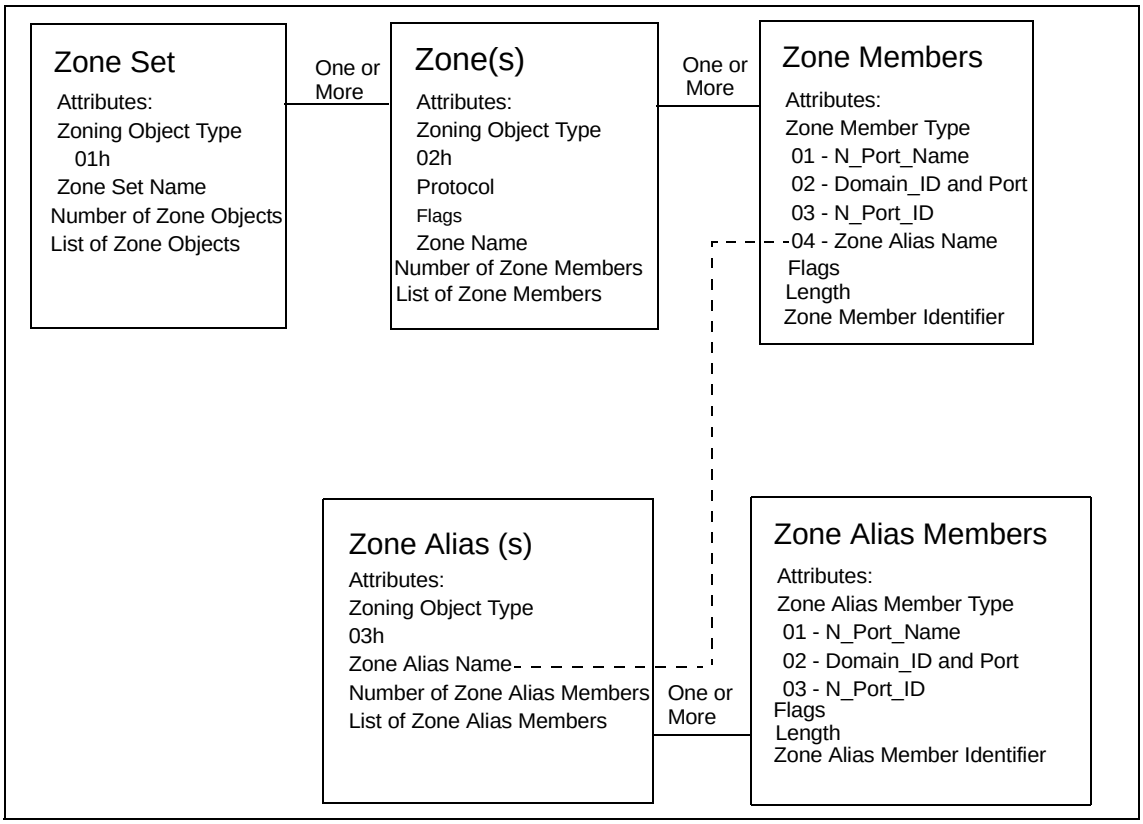

Figure 24 – Basic Zoning Object Structure

10.4.2 Basic Zoning Data Structures

10.4.2.1 Zoning Object List

The format of the Zoning Object List is depicted in table 168.

Table 168 – Zoning Object List

Item	Size
Number of Zoning Objects	4
Zoning Object 1	x
Zoning Object 2	y
...	...
Zoning Object n	z

Number of Zoning Objects: This field contains a value that specifies the number of Zoning objects contained in the Zoning Object list.

Zoning Object: This entry contains a Zoning object as described in 10.4.2.2.

10.4.2.2 Zoning Object Format

The general format of a Zoning object is depicted in table 169.

Table 169 – Zoning Object

Item	Size
Object Type	1
Protocol	1
Reserved	2
Object Name	a
Number of Object Members	4
Object Member 1	x
Object Member 2	y
...	...
Object Member n	z

Object Type: Valid values for defined Zoning object types are listed in table 170 below.

Table 170 – Zoning Object Types

Description	Value (hex)
Reserved	00
Zone Set	01
Zone	02
Zone Alias	03
Reserved	04-DF
Vendor Specified	E0-FF

Protocol: The Protocol attribute shall be reserved for Zone Set and Zone Alias objects. For Zone Objects, if the Protocol field is non-zero, Device_Data and FC-4 Link_Data frames not having the specified protocol value shall not be transmitted between members of the Zone. All other frames shall be transmitted between members of the zone. The values of the Protocol Format are shown in table 171:

Table 171 – Protocol Format

Encoded Value (hex)	Description
00	No Protocol Zoning
01-FF	Non-zero values are taken from FC-FS-3.

Object Name: This attribute specifies the name of the object. The format of this attribute is described in 10.4.2.3.

Number of Object Members: This field indicates the number of object members.

Object Members: One or more object members are contained in the Zoning object. Object members may be other Zoning objects or Zone Members.

10.4.2.3 General Name Format

All Name attributes pertaining to Zoning shall use the General Name Format. Examples include Zone Set names, Zone names, and Zone Alias names. The General Name Format is described in FC-GS-7.

10.4.2.4 Zone Member Format

Zone objects shall have Zone Members. The format of a Zone Member is depicted in table 172 below.

Table 172 – Zone Member Format

Item	Size
Zone Member Type	1
Reserved	1
Flags	1
Identifier Length	1
Identifier	x

Zone Member Type: Valid Zone Member types are shown in table 173.

Flags Field: Implementation is vendor specific.

Identifier Length: The identifier length is determined by the Zone Member Type as specified in table 173.

Identifier: The description of the Identifier fields for valid Zone Member Types are depicted in table 173.

Table 173 – Zone Member Type and Identifier Formats

Type (hex)	Identifier	Size (Bytes)
00	Reserved	
01	N_Port_Name: The format of the Zone Member information is a N_Port_Name.	8
02	Domain_ID & Physical Port Number: The format of the Zone Member Information is a combination of a Domain_ID + Physical Port Number. (i.e., 00DDPPPPh; where DD is the Domain_ID and PPPP is the Physical Port Number).	4
03	N_Port_ID: Address identifier format (00ddaapp). Valid address identifiers are those assignable to F and FL port attached devices.	4

Table 173 – Zone Member Type and Identifier Formats

Type (hex)	Identifier	Size (Bytes)
04	Alias Name	Variable
05-DF	Reserved	
E0-FF	Vendor Specific	

NOTE 38 – Name field describes either an Alias or Zone name in the format described in FC-GS-7.

10.4.3 Enhanced Zoning Framework

10.4.3.1 Introduction

In the Enhanced Zoning Framework more Zoning objects are defined to those defined in Basic Zoning. This subclause delineates the structures of the Enhanced Active Zone Set and of the Enhanced Zone Set Database.

10.4.3.2 Zone Set Database

The Zone Set Database may contain Zoning Object types as defined in the following subclauses. The currently defined Zoning Object types are:

a) Zone Set;

b) Zone;

c) Zone Alias;

d) Zone Attribute.

The Zone Set Database shall not contain the Active Zone Set. The Zone Set Database may contain multiple Zoning objects. Objects defined in the Zone Set Database need not be referenced. In the Zone Set Database the Zoning objects may reference each other using names formatted as specified in FC-GS-7.

Figure 25 depicts the logical structure of the Zone Set Database.

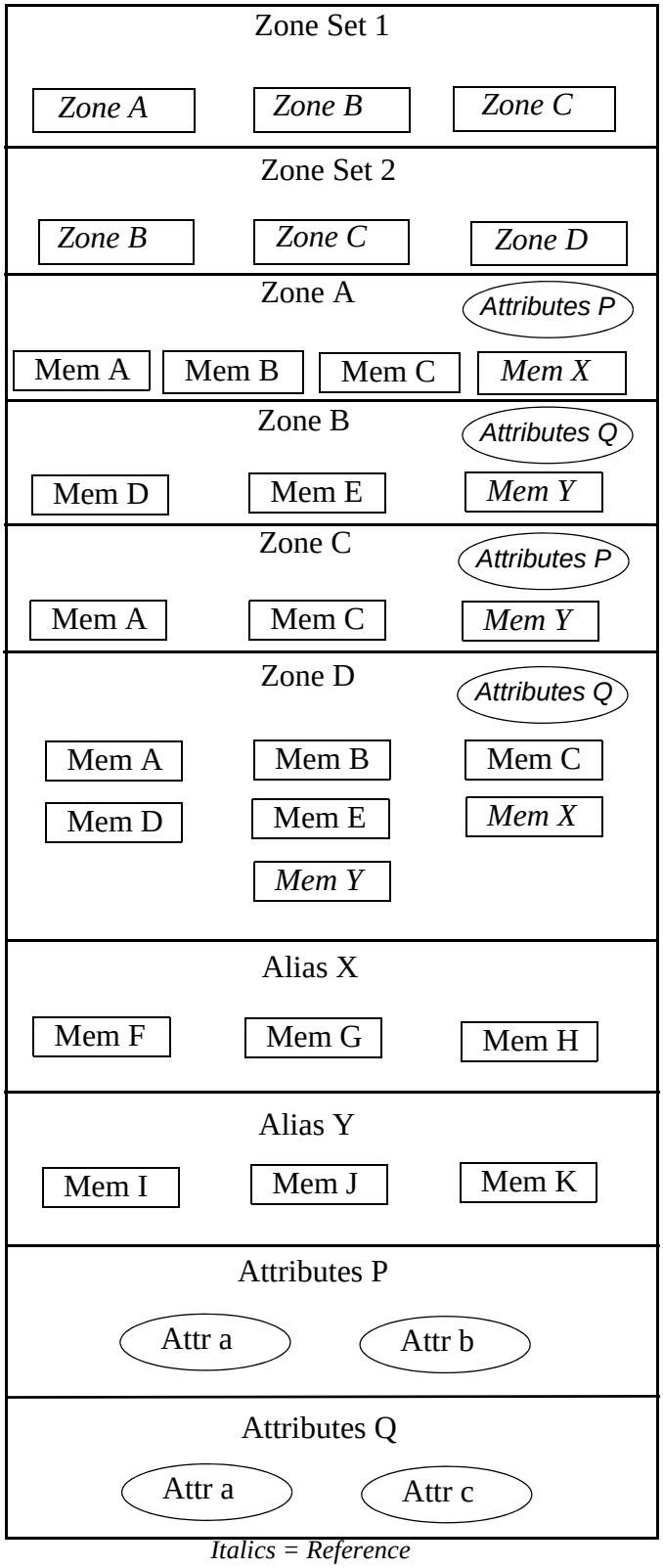

Figure 25 – Logical Structure of the Zone Set Database

Each Zone Set definition references its Zone objects. Each Zone may reference a Zone Attribute object or, in the member definitions, one or more Zone Alias objects.

10.4.3.3 Active Zone Set

References are not allowed in the Active Zone Set. At activation time any reference present in a Zone Set or Zone definition in the Zone Set Database shall be resolved. The resulting logical structure of the Active Zone Set is depicted in figure 26.

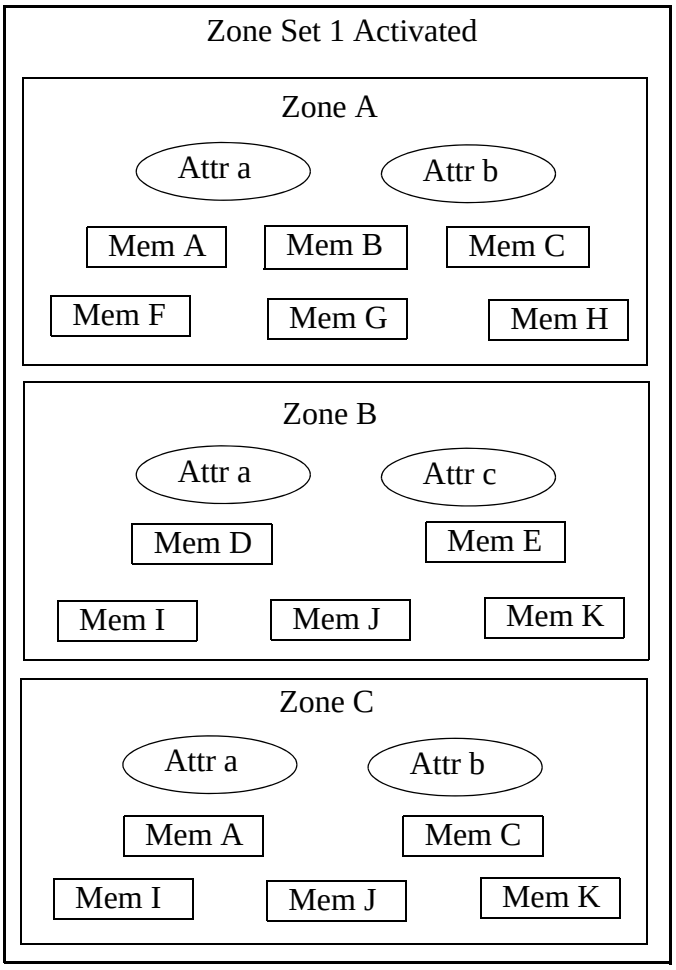

Figure 26 – Logical Structure of the Active Zone Set

10.4.4 Enhanced Zoning Data Structures

10.4.4.1 Zoning Object List

The format of the Zoning Object List is depicted in table 174.

Table 174 – Zoning Object List

Item	Size
Number of Zoning Objects	4
Zoning Object 1	x
Zoning Object 2	y
...	...
Zoning Object n	z

Number of Zoning Objects: This field contains a value that specifies the number of Zoning objects contained in the Zoning Object list.

Zoning Object: This entry contains a Zoning object. Valid Zoning objects are described in the following clauses.

10.4.4.2 Zoning Object Types

Currently defined Zoning Object types are listed in table 175 below.

Table 175 – Zoning Object Types

Description	Value (hex)
Reserved	00
Zone Set	01
Zone	02
Zone Alias	03
Zone Reference	04
Zone Attribute Object	05
Reserved	06-DF
Vendor Specified	E0-FF

10.4.4.3 Zone Set Object

10.4.4.3.1 Zone Set Object in the Zone Set Database

The Zone Set object used in the zone set database has the attributes described below and the format depicted in table 176.

Table 176 – Zone Set Object Format in the Zone Set Database

Item	Size (bytes)
Zone Set Object Identifier	1
Reserved	3
Zone Set Name	a
Number of Zone References	4
Zone Reference #1	x
Zone Reference #2	y
...	...
Zone Reference #n	z

Zone Set Name

The Zone Set Name attribute shall follow the general name format described in FC-GS-7.

Number of Zone References

This attribute shall contain the integer number of zone references in the Zone Set.

Zone Reference

Only Zone Reference objects (Zone Set Member Type 04h) are allowed in the Zone Set Object when used in the Zone Set Database.

10.4.4.3.2 Zone Set Object in the Active Zone Set

The Zone Set object used in the active zone set has the attributes described below and the format depicted in table 177.

Table 177 – Zone Set Object Format in the Active Zone Set

Item	Size (bytes)
Zone Set Object Identifier	1
Reserved	3
Number of Zones	4
Zone #1	x
Zone #2	y
...	...
Zone #n	z

Number of Zones

This attribute shall contain the integer number of zones in the Zone Set.

Zone

Only Zone objects (Zone Set Member Type 02h) are allowed in the Zone Set Object when used in the Active Zone Set.

NOTE 39 – The Active Zone Set does not have a name because such a name is not significant when a Zone Merge occurs.

10.4.4.4 Zone Reference Object

The Zone Reference object format is depicted in table 178.

Table 178 – Zone Reference Object Format

Item	Size (bytes)
Zone Reference Object Identifier	1
Reserved	3
Zone Name	x

Zone Name

The Zone Name attribute shall follow the general name format described in FC-GS-7.

10.4.4.5 Zone Object in the Zone Set Database

The Zone object used in the Zone Set Database allows references to other objects. It has the attributes described below and the format depicted in table 179.

Table 179 – Zone Object Format in the Zone Set Database

Item	Size (bytes)
Zone Object Identifier	1
Reserved	3
Zone Name	a
Zone Attribute Object Name	b
Number of Zone Members	4
Zone Member #1	x
Zone Member #2	y
...	...
Zone Member #n	z

Zone Name

The format of the Zone Name attribute shall follow the general name format described in FC-GS-7.

Zone Attribute Object Name

The format of the Zone Attribute Object Name attribute shall follow the general name format described in FC-GS-7. Its value references the Zone Attribute Object whose attributes apply for the Zone. A null value (‘00 00 00 00 00 00 00 00’) indicates that no attributes are associated with that Zone definition.

Number of Zone Members

This attribute shall contain the integer number of Zone Members in the Zone.

Zone Member

The zone member format is described in 10.4.4.6.1.

10.4.4.6 Zone Object in the Active Zone Set

The Zone object used in the Active Zone Set does not allow references to other objects. It has the attributes described below and the format depicted in table 180.

Table 180 – Zone Object Format in the Active Zone Set

Item	Size (bytes)
Zone Object Identifier	1
Reserved	3
Zone Name	a
Number of Zone Attribute Entries	4
Zone Attribute Entry #1	b
Zone Attribute Entry #2	c
...	...
Zone Attribute Entry #m	d
Number of Zone Members	4
Zone Member #1	x
Zone Member #2	y
...	...
Zone Member #n	z

Zone Name

The format of the Zone Name attribute shall follow the general name format described in FC-GS-7.

Number of Zone Attribute Entries

This attribute shall contain the integer number of Zone Attribute Entries in the Zone.

Zone Attribute Entry

The format of the Zone Attribute entry is described in 10.4.4.8.1.

Number of Zone Members

This attribute shall contain the integer number of Zone Members in the Zone.

Zone Member

The Zone Member format is described in 10.4.4.6.1.

10.4.4.6.1 Zone Member Format

Zone Objects shall have Zone Members. The format of a Zone Member is depicted in table 181 below.

Table 181 – Zone Member Format

Item	Size
Zone Member Type	1
Reserved	2
Identifier Length	1
Identifier	x

Zone Member Type

Valid Zone Member types are shown in table 182.

Identifier Length

The identifier length is determined by the Zone Member Type as specified in table 182.

Identifier

The description of the Identifier fields for valid Zone Member Types are depicted in table 182.

Table 182 – Zone Member Type and Identifier Formats (Part 1 of 2)

Type (hex)	Identifier	Size (Bytes)
00	Reserved	
01	N_Port_Name: The format of the Zone Member information is a N_Port_Name.	8
02	Domain_ID & Physical Port Number: The format of the Zone Member Information is a combination of a Domain_ID + Physical Port Number. (i.e., 00DDPPPPh; where DD is the Domain_ID and PPPP is the Physical Port Number).	4
03	N_Port_ID: Address identifier format (00ddaapp). Valid address identifiers are those assignable to F and FL port attached devices.	4

Table 182 – Zone Member Type and Identifier Formats (Part 2 of 2)

Type (hex)	Identifier	Size (Bytes)
04	Alias Name: The format of the Zone Member information is a General Name.	a
05	Node_Name: The format of the Zone Member information is a Node_Name.	8
06	F_Port_Name: The format of the Zone Member information is a F_Port_Name.	8
07	Reserved	8
08-DF	Wildcard	
E0-FF	Vendor Specific	

Wildcard Zone Member Format

The Wildcard Zone Member Identifier format shall be as shown in table 183. See FC-GS-7 for how to use the Wildcard Zone Member.

Table 183 – Zone Member Identifier Format - Wildcard

Item	Size (Bytes)
Subtype	2
Flags	2
Parameter	4

Subtype: see FC-GS-7.

Flags: see FC-GS-7.

Parameter: see FC-GS-7.

Vendor Specified Zone Member Format

The Vendor Specified Zone Member Identifier format is depicted in table 184.

Table 184 – Zone Member Identifier Format - Vendor Specified

Item	Size (Bytes)
Vendor ID	8
Vendor Specified Value	n
Pad	m

Vendor ID: Contains the T10 Vendor ID of the vendor that defines the content of the Vendor Specified Value field.

Vendor Specified Value: This field contains the Vendor Specified Value.

Pad: Fill bytes are added as necessary to the end of the Vendor Specified Value in order to ensure that the total length of the Vendor Specified Zone Member is a multiple of four. Fill bytes shall be nulls (00h). The number of fill bytes (m) is zero, one, two, or three depending on the length of the actual value (n).

10.4.4.7 Zone Alias Object

The Zone Alias object has the attributes described below and the format depicted in table 185.

Table 185 – Zone Alias Object Format

Item	Size (bytes)
Zone Alias Object Identifier	1
Reserved	3
Zone Alias Name	a
Number of Zone Alias Members	4
Zone Alias Member #1	x
Zone Alias Member #2	y
...	...
Zone Alias Member #n	z

Zone Alias Name

The format of the Zone Alias Name attribute shall follow the general name format described in FC-GS-7.

Number of Zone Alias Members

This attribute shall contain the integer number of Zone Alias Members in the Zone Alias.

Zone Alias Member

The Zone Alias Member has the format described in 10.4.4.6.1. All Zone Member Identifier Types may be used, with the exception of the Alias Name member (type '04').

10.4.4.8 Zone Attribute Object

The Zone Attribute object is a variable length structure that contains extensible attributes that may be associated with a Zone. The format of the Zone Attribute object is depicted in table 186 and table 187.

Table 186 – Zone Attribute Object Format

Item	Size (bytes)
Zone Attribute Object Identifier	1
Reserved	3
Zone Attribute Object Name	x
Zone Attribute Block	w

Table 187 – Zone Attribute Block Format

Item	Size (Bytes)
Number of Zone Attribute Entries	4
Zone Attribute Entry #1	x
Zone Attribute Entry #2	y
...	...
Zone Attribute Entry #n	z

Zone Attribute Object Name

The format of the Zone Attribute Object Name attribute shall follow the general name format described in FC-GS-7.

Number of Zone Attribute Entries

This field specifies the number of Zone Attribute Entries contained in the Zone Attribute Block. A value of zero in this field shall indicate that no attributes are registered.

10.4.4.8.1 Zone Attribute Entry Format

The format of the Zone Attribute Entry is depicted in table 188.

Table 188 – Zone Attribute Entry Format

Item	Size (Bytes)
Zone Attribute Type	2
Zone Attribute Length	2
Zone Attribute Value	x

Zone Attribute Type: This field indicates the attribute entry type. Valid Zone Attribute Types are depicted in table 189 and shall be restricted to a value between 0000h and 00FFh

Table 189 – Zone Attribute Types

Zone Attribute Type (hex)	Description
0001	Protocol
0002	Broadcast Zone
0003	Hard Zone
0004	IFR Zone[a]
0005	Peer Zone
00E0	Vendor Specific
other values	Reserved
[a] For a definition of the IFR Zone attribute type, see FC-IFR.	

Zone Attribute Length: This field indicates the total length in bytes of the Zone Attribute Entry. This length shall be a multiple of four and includes the following fields

a) Zone Attribute Type;

b) Zone Attribute Length;

c) Zone Attribute Value.

Zone Attribute Value: This field specifies the actual attribute value. If present, Attribute Values shall be at least four bytes in length and the length shall be a multiple of four. For Attribute Value fields, fill bytes are added as necessary to the end of the actual value in order to ensure that the length of the value field is a multiple of four. Fill bytes shall be nulls (00h). The number of fill bytes (m) is zero, one, two, or three depending on the length of the actual value (n). Therefore the total length of the value field is (n+m).

10.4.4.8.1.1 Protocol Attribute

When a Protocol Attribute Type is specified, the Protocol Attribute Value specifies the FC-4 type for which protocol zoning is enforced. Valid values are 01h-FFh. If the FC-4 Type is non-zero, Device_Data and FC-4 Link_Data frames not having the specified FC-4 Type value shall not be transmitted between members of the Zone. All other frames shall be transmitted between members of the zone. The format of the Protocol Attribute Value is depicted in table 190.

Table 190 – Protocol Attribute Value

Item	Size (Bytes)
FC-4 Type	1
Reserved	3

10.4.4.8.1.2 Broadcast Zone Attribute

Broadcast Zoning is enabled by setting the Broadcast Zone attribute on Zones that contain ports that send or receive broadcast frames. Use of the Broadcast Zone attribute allows multi-protocol devices to send broadcast frames to some devices but not to others. The Broadcast Zone attribute only affects the processing of broadcast frames. The Broadcast Zone attribute has no effect on Zoning enforcement for Name Server, RSCN, or hard zoning.

When Zoning is active, broadcast frames are delivered to all logged in Nx_Ports that share a Broadcast Zone with the source of the frame (as indicated by the S_ID of the frame). This implies that if Zoning is active and no Zones have the Broadcast Zone attribute set, then no broadcast frames are delivered. Zoning is enforced at the destination Switch. Broadcast Zoning shall not have any affect on the Switch to Switch routing of frames.

If any NL_Port attached to an FL_Port shares a Broadcast Zone with the source of the broadcast frame, or Zoning is not active, the frame shall be sent to the all the devices on the loop using the OPNfr open replicate primitive. Use of the OPNyr selective replicate is prohibited (see FC-DA).

Broadcast Zoning shall be enforced on the destination switches, using the S_ID of the broadcast frame to determine the Zones. Some Zone Member types are unable to be directly mapped to an address identifier. For these types a Switch shall use the Name Server to get the address identifier associated with the Zone Member data. Similar discovery may be needed to implement Hard Zones. The address identifier(s) for specific Zone Member Identifier Types may be obtained as follows:

a) N_Port_Name or Node_Name Zone Member: Use the GID_PN or GID_NN Name Server commands;

b) Domain + Physical Port Zone Member: Use the GID_DP Name Server command;

c) Fabric Port_Name Zone Member: Use the GID_FPN Name Server command.

NOTE 40 – GID_NN and GID_DP may return multiple address identifiers.

If address discovery finds address identifiers for non address identifier Zone Members, ports in those Broadcast Zones become accessible to the ports whose address identifiers have been discovered. If ports become accessible to new broadcast sources, an RSCN shall be issued for those ports. If a Switch is issuing an RSCN for some other reason after address discovery, and the RSCN includes newly accessible ports, a separate RSCN for address discovery for those ports is not necessary.

If Zoning is inactive in the Fabric, then broadcast frames are delivered to all logged in Nx_Ports.

There is no value associated with the Broadcast Zone Attribute. Therefore the Zone Attribute Length shall be set to four.

10.4.4.8.1.3 Hard Zone Attribute

Hard zoning is specified by setting the Hard Zone attribute.

When the Hard Zone attribute is specified, the zone configuration shall use hard zoning enforcement see 3.1.52). If an implementation is unable to enforce the zone configuration using hard zoning enforcement (e.g., enforcing the zone configuration requires hardware resources that are not available), the activation of the zone configuration shall fail.

When the Hard Zone attribute is not specified, the zoning configuration shall be enforced in one of the following ways:

a) hard zoning enforcement; or

b) soft zoning enforcement (see 3.1.89).

The zone configuration should be enforced using hard zoning enforcement whenever possible.

If zoning for an Nx_Port is enforced using hard zoning enforcement for any zone, zoning for that Nx_Port shall be enforced using hard zoning enforcement for all zones in which it is a member.

The activation of a hard zone may succeed because some of the Nx_Ports specified in the zoning configuration are not connected to the Fabric at activation time. If a FLOGI request received from such an Nx_Port makes it impossible to enforce the zoning configuration using hard zoning enforcement, then the Fabric shall reject or discard, as appropriate for the class of service, any frames not addressed to Fabric Services to or from that Nx_Port.

Fabrics may be composed of Switches that support hard zoning and Switches that support only soft zoning. The Fabric administrator decides which Nx_Ports are to be managed with hard zoning, and which ones with soft zoning.

The Fabric administrator may manage the situation where some Nx_Ports need to be restricted with hard zoning and others with soft zoning by defining two overlapping zones, one with the Hard Zoning attribute and the other without the Hard Zoning attribute. The zone without the Hard Zoning attribute defines which Nx_Ports are allowed to communicate. The zone with the Hard Zoning attribute defines the subset of these Nx_Ports for which Zoning shall be enforced on a frame-by-frame basis.

There is no value associated with the Hard Zone Attribute. Therefore the Zone Attribute Length shall be set to four.

Hard zoning enforcement requires a mapping of zone members to N_Port_IDs. A Switch may use the following name server requests to perform this mapping:

a) N_Port_Name member type. Use the GID_PN Name Server command;

b) Node_Name member type. Use the GID_NN Name Server command (which may return multiple N_Port_IDs);

c) F_Port_Name member type. Use the GID_FPN Name Server command.

10.4.4.8.1.4 Peer Zone Attribute

The Peer Zone Attribute is used to define a Peer Zone (see FC-GS-7). A Peer Zone is a Zone with the Peer Zone Attribute. A Peer Zone identifies a Principal member through the Peer Zone Attribute and a list of Peer members as Zone members. The semantic of a Peer Zone is that:

a) Peer members are allowed to communicate with the Principal member; and

b) Peer members are not allowed to communicate among themselves (unless allowed by other Zones in the Zone Set).

The format of the Peer Zone Attribute Value is depicted in table 191.

Table 191 – Peer Zone Attribute Value

Item	Size (Bytes)
Principal N_Port_Name	8

Principal N_Port_Name: The N_Port_Name of the Principal member of a Peer Zone.

10.4.4.8.1.5 Vendor Specific Zone Attribute

The format of the The Vendor Specific Attribute Value is depicted in table 192.

Table 192 – Vendor Specific Attribute Value

Item	Size (Bytes)
Vendor ID	8
Vendor Specific Value	n
Pad	m

Vendor ID: Contains the T10 Vendor ID of the vendor that defines the content of the Vendor Specific Value field.

Vendor Specific Value: This field contains the Vendor Specific Value.

Pad: Fill bytes are added as necessary to the end of the Vendor Specific Value in order to ensure that the total length of the Vendor Specific Zone Member is a multiple of four. Fill bytes shall be nulls (00h). The number of fill bytes (m) is zero, one, two, or three depending on the length of the actual value (n).

10.5 Merge Zone

10.5.1 Example Merge Operation

Figure 27 shows how the Zones are merged when two Switches are joined.

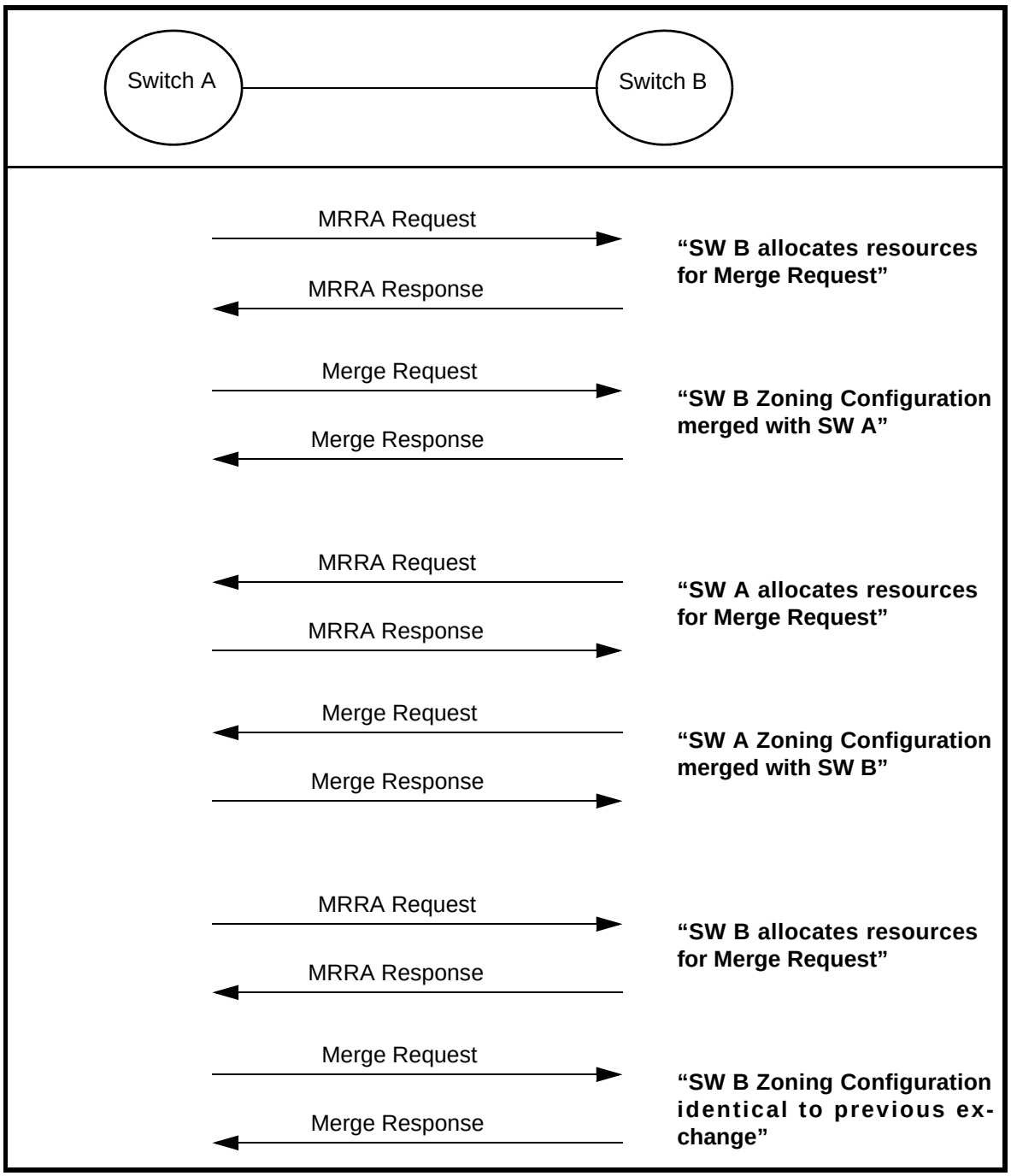

Figure 27 – Merge Operation Between Two Switches

The use of the MRRA SW_ILS is optional. Switch A may send a Merge Request Resource Allocation (MRRA) over the new link, to request Switch B to allocate the resource required to accept the following Merge Request. When the MRRA is accepted, a Merge Request conveying Switch A's Zoning configuration is sent to Switch B. Switch B merges the two Zoning configurations and changes the name of the Active Zone Set to "Successful Zone Set Merge: Active Zone Set Name has changed". Since Switch B's Zoning configuration has changed, it transmits its new Zoning configuration on all ISLs, including the ISL connecting with Switch A. Switch A receives Switch B's new Zoning configuration, that

it merges with its Zoning configuration and changes the Zone Set Name to "The Active Zone Set has changed due to a Zone Merge". After the merge, Switch A sends out its new Zoning configuration on all ISLs. When Switch B receives the new Zoning configuration, it confirms that it has an identical Zoning configuration and sends a Merge Response to end the Zone Merge.

Figure 28 shows the Zone Merge process when more than two Switches are involved. Note that initial state is Switches connected but no merge processing has begun.

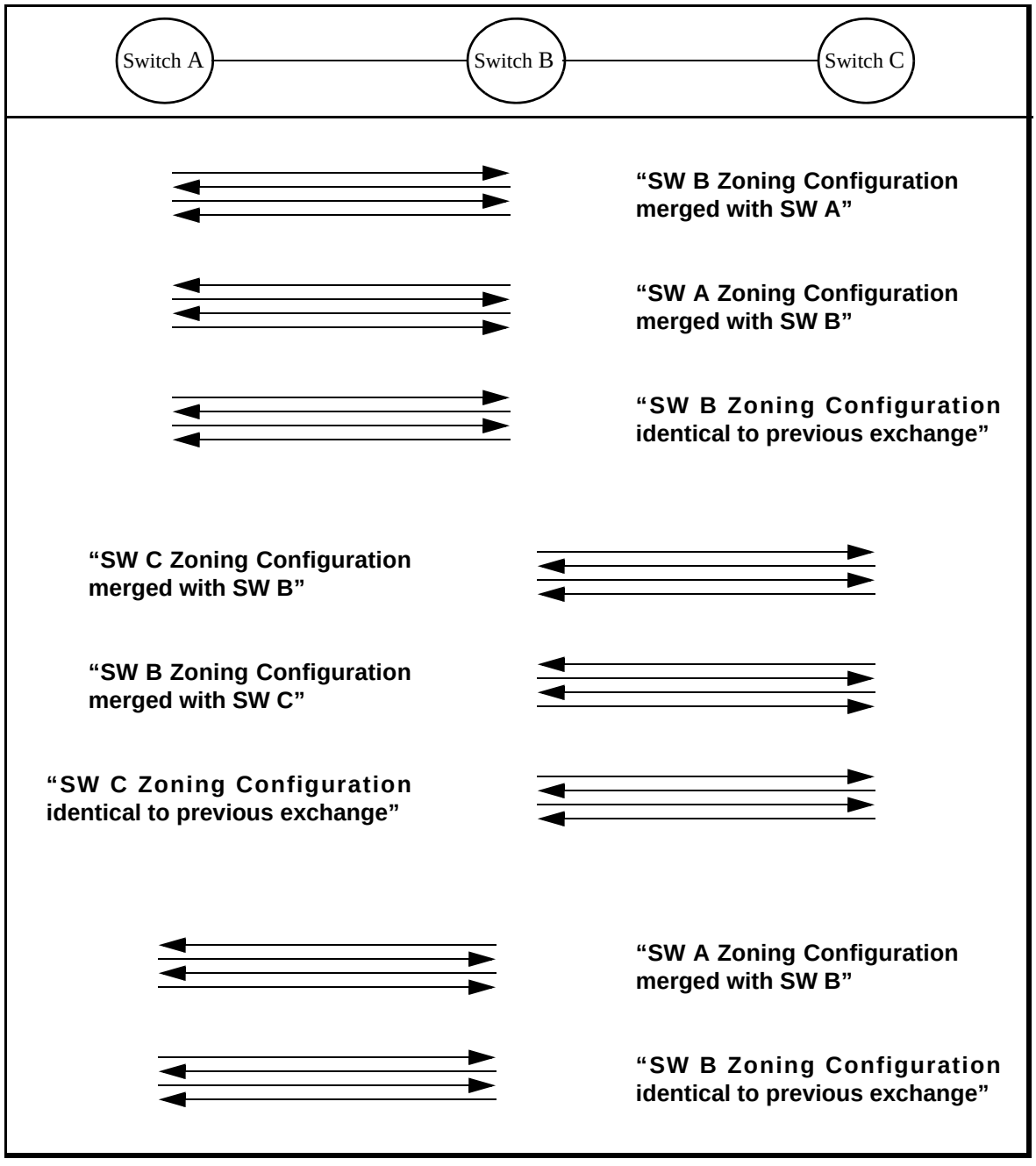

Figure 28 – Merge Operation Among Several Switches

10.5.2 Merge Zone Rules

10.5.2.1 Merge Rules in Basic Zoning

In Basic Zoning, when adjacent Switches exchange zoning information, they shall abide by the rules in table 193.

Table 193 – Basic Zoning Merge Rules

Adjacent Zoning Configuration	Local Zoning Configuration	Result in Local Switch
Zone Set State is Deactivated.	Zone Set State is Activated or Deactivated.	No change
Zone Set State is Activated.	Zone Set State is Deactivated.	Zone Set gets the Adjacent Zone Set State (i.e., the Zone Set is activated). Zone Set gets the adjacent Zone Set.
Adjacent Zone Set is equal to the Local Zone Set		No change.
Adjacent and Local Zone Sets contain a Zone with the same name but different members, or different Protocol Types		ISL Isolated.
Adjacent Zone Set contains Zones that are not included in the Local Zone Set, and/or Local Zone Set contains Zones that are not included in the Adjacent Zone Set.		Zone Set State becomes Activated. Zone Set is the merge of the local Zones plus the Adjacent Zones. The Zone Set Name is changed to "Successful Zone Set Merge: Active Zone Set Name has changed ".

NOTE 41 – To prevent potential ISL isolation, it is suggested that zoning be inactivated and a zone set propagated through the Fabric by the techniques documented in clause 10.6.

If a received Zoning Configuration contains some unsupported member types, the Zone Merge shall fail.

10.5.2.2 Merge Rules in Enhanced Zoning

In Enhanced Zoning the Merge behavior depends by the Merge Control setting. Merge Control is a Fabric wide setting that indicates the type of behavior two switches exhibit as a result of a merge operation. This setting may be set as Allow or Restrict. A setting of Allow means that the two databases are merged according to the specified merge rules, and the resulting database is the union of the two databases. A setting of Restrict indicates that if the two databases on the corresponding switches are not identical, then the links between the two switches shall become isolated.

Before the merge rules pertaining to the Zoning Database are applied, the Zone Merge flags shall be compared.

If one of the two Switches does not support the Zone Set Database and the other one is using it (as indicated by the Zone Merge flags) then no merge shall occur and the ISL shall become isolated.

If the Merge Control or Default Zone settings are not equal for both Switches, then no merge shall occur and the ISL shall become isolated. If the Merge Control and Default Zone settings are equal, ad-

jacent Switches exchanging zoning information shall abide by the rules in table 194. If the local Zoning Database is modified as a result of the merge, the new database is propagated to the neighboring Switches.

If a received Zoning Configuration contains some unsupported member types, the Zone Merge shall fail.

Table 194 – Enhanced Zoning Merge Rules

Condition Before Merge	Result in Local Switch
Adjacent Zoning Database is equal to the Local Zoning database. Merge Control = Restrict or Allow	No change.
The local and adjacent Zoning Databases contain a zone set, zone, or zone alias object with the same name but unlike members, Merge Control = Restrict or Allow	ISL Isolated.
The local and adjacent Zoning Databases contain references with the same name but with different definitions, Merge Control = Restrict or Allow	ISL Isolated
The local and adjacent Zoning Databases contain zones with the same name but with different attributes, Merge Control = Restrict or Allow	ISL Isolated
Adjacent Zoning Database contains zones or aliases that are not included in the local Zoning Database. Merge Control = Allow.	Zoning Database is the union of the local database plus the adjacent database
Local Zoning Database is empty, Adjacent Fabric Zone database contains data. Merge Control = Allow.	Data from Adjacent database is used to populate Local database
Adjacent Zoning Database contains zones or aliases that are not included in the local Zoning Database. Merge Control = Restrict.	ISL Isolated
Local Zoning Database is empty, Adjacent Fabric Zone database contains data. Merge Control = Restrict.	ISL Isolated
Adjacent Zoning Database is empty, Local Fabric Zone database contains data. Merge Control = Restrict.	ISL Isolated
Adjacent Zoning Database is empty, Local Fabric Zone database contains data. Merge Control = Allow.	No Change

10.6 Fabric Management Session Protocol

10.6.1 Fabric Management Session Protocol Overview

Requests to change a Fabric's Zoning Configuration or security policies are a result of an administration action or control requests via the Management Server. To keep consistency in the Fabric, the Fabric Management Session protocol ensures that only one management entity may change the Fabric at one time. Requests may be made to activate or deactivate a Zone Set, or security policy. The Switch that receives a change request is referred to as the Managing Switch. The Managing Switch validates the request and exchanges inter-Switch messages with the other Switches in the

Fabric. The protocol for locking the Fabric during policy changes is the Fabric Management Session Protocol.

The policy change requests serviced by the Managing Switch are used to:

a) reserve Change Authorization in each Switch in the Fabric (blocking any other changes from taking place while this change is in progress);

b) stage configuration changes in each Switch in the Fabric;

c) apply the staged configuration change in each Switch in the Fabric, and;

d) release Change Authorization in each Switch in the Fabric.

Acquire Change Authorization request and response messages are used to reserve Local Change Authorization in each Switch in the Fabric (i.e., to establish Fabric Management Sessions between the Managing Switch and the Managed Switches in the Fabric). Stage Fabric Configuration Update request and response messages are used to distribute a Configuration update to each Switch in the Fabric in order to verify that each Switch is able to implement the change. Update Fabric Configuration request and response messages are used to modify the Configuration on each Switch in the Fabric. Release Change Authorization request and response messages are used to release Local Change Authorization in each Switch in the Fabric (i.e., end the Fabric Management Sessions).

10.6.2 Reserving Fabric Change Authorization

The Managing Switch shall send an Acquire Change Authorization request to each Managed Switch in the Fabric to reserve the Fabric Change Authorization. Each Acquire Change Authorization request message includes a list of the Domain_IDs of the Managed Switches being included in the update.

Each Managed Switch that receives an Acquire Change Authorization request shall return an Acquire Change Authorization response to the Managing Switch. The Acquire Change Authorization response message is either an SW_ACC that indicates success, or an SW_RJT that specifies why the Acquire Change Authorization operation was unsuccessful. The Managed Switch returns an SW_RJT indicating “Fabric Changing” if the Domain_IDs the Managed Switch knows to be in the Fabric are not the same as the Domain_IDs identified in the request by the Managing Switch. The Managed Switch returns an SW_RJT indicating “Busy” if its Local Change Authorization is already reserved by another process. Otherwise, the Managed Switch reserves Local Change Authorization for the Managing Switch, and returns an SW_ACC indicating the operation was “Successful”.

The Managing Switch shall wait until an Acquire Change Authorization response has been received from each Managed Switch in the Fabric. If any of the responses indicate that a Managed Switch is Busy or that the Fabric is Changing, the Managing Switch shall initiate the process to release Fabric Change Authorization (see 10.6.5). If all the responses indicate that Local Change Authorization was successfully acquired, the Managing Switch shall initiate the process to stage the update (see 10.6.3).

10.6.3 Staging the Fabric Configuration

The Managing Switch shall send a Stage Fabric Configuration Update request to each Managed Switch in the Fabric to stage the Fabric Configuration. The Stage Fabric Configuration Update request includes a change command and the appropriate operation data.

Each Managed Switch that receives a Stage Fabric Configuration Update request shall return a Stage Fabric Configuration Update response to the Managing Switch. The Stage Fabric Configuration Update response message is either an SW_ACC that indicates success, or an SW_RJT that specifies why the Stage Fabric Configuration operation was unsuccessful.

If a received Zoning Configuration contains some unsupported member types, the Stage Fabric Configuration shall fail.

The Managing Switch shall wait until a Stage Fabric Configuration Update response has been received from each Managed Switch in the Fabric. If a staging error occurred, the process to release Fabric Change Authorization (see 10.6.5) shall be initiated. If the update was successfully staged in all the Switches in the Fabric, the process to update the Fabric Configuration (see 10.6.4) shall be initiated.

10.6.4 Updating the Fabric Configuration

The Managing Switch shall send an Update Fabric Configuration request to each Managed Switch in the Fabric to update the Fabric Configuration. There is no data included in the Update Fabric Configuration request message.

Each Managed Switch that receives an Update Fabric Configuration request shall return an Update Fabric Configuration response to the Managing Switch. The Update Fabric Configuration response message is either an SW_ACC that indicates success, or an SW_RJT that specifies why the Update Fabric Configuration operation was unsuccessful.

The Managing Switch shall wait until an Update Fabric Configuration response has been received from each Managed Switch in the Fabric. The Managing Switch shall then initiate the process to release Fabric Change Authorization (see 10.6.5).

10.6.5 Releasing Fabric Change Authorization

The Managing Switch shall send a Release Change Authorization request to each Managed Switch in the Fabric that has reserved Local Change Authorization for the Managing Switch. There is no data included in the Release Change Authorization request message.

Each Managed Switch that receives a Release Change Authorization request shall return a Release Change Authorization response to the Managing Switch. The Release Change Authorization response message is either an SW_ACC that indicates success, or an SW_RJT that specifies why the Release Change Authorization operation was unsuccessful.

The Managing Switch shall wait until a Release Change Authorization response has been received from each Managed Switch in the Fabric at which time the Fabric update is complete.

10.6.6 Mapping of a Server Session to a Fabric Management Session

In the context of Enhanced Zoning Management, a management action (i.e., write access to the Zoning Database) to the Zone Server shall occur only inside a server session. A server session is delimited by the CT requests Server Session Begin (SSB) and Server Session End (SSE), directed to the Management Service and with GS_Subtype specifying the Zone Server. Query requests that result in read access to the Zoning Database are not required to be issued inside a server session, although the information which they report is only consistent inside a server session.

The server session on the Zone Server side is translated in a lock of the Fabric on the Fabric side, using the Fabric Management Session Protocol. This ensures serialized management access to the Zoning Database by different management applications, and to guarantee a deterministic behavior.

The Switch handling the Zone Server CT requests, on receiving the SSB CT request shall try to become also the Managing Switch of the Fabric Management Session Protocol. As such it shall send the ACA SW_ILS to all the other Switches in the Fabric. The result of this action may be one of the following:

a) Each other Switch responds to the ACA with an SW_RJT indicating failure. This means that the Fabric is already locked by another Switch, and so somebody else is managing the Zoning Database. The Zone Server shall reject the SSB CT request. The managing application may retry later.

b) Some Switches respond to the ACA with an SW_ACC indicating success, some with an SW_RJT indicating failure. This means that another Switch is trying to lock the Fabric at the same moment. The Switch shall send an RCA SW_ILS to each Switch that accepted the ACA to release its attempt to lock the Fabric. The Zone Server shall reject the SSB CT request. The managing application may retry later.

c) Each other Switch responds to the ACA with an SW_ACC indicating success. This means that the Switch has been successful in locking the Fabric, and it owns the lock. The Switch shall prepare a copy of the Zoning Database for the subsequent management actions, and after that the Zone Server shall accept the SSB CT request.

At this point the management application may manage the Zoning Database using the Enhanced Zoning Commands. The Zone Server shall handle the requests applying them to the copy of the Zoning Database. The Zoning Database present in all the other Switches of the Fabric shall not be affected by these Enhanced Zoning Commands.

To apply the updated Zoning Database to the Fabric, the Management Application shall send a Commit (CMIT) CT request to the Zone Server. The Zone Server shall perform a consistency check of the updated Zoning Database. If the consistency check fails, then the Zone Server shall reject the CMIT CT request, with an appropriate reason code. Consistency checks may fail for many reasons including the following:

a) A Zone Set Object in the Zone Set Database references Zone Objects that do not exist;

b) A Zone Object in the Zone Set Database references Zone Alias Objects that do not exist;

c) A Zone Object in the Zone Set Database references Zone Attribute Objects that do not exist.

If the consistency check succeeds, then the Managing Switch shall stage the new Zoning Database to the other Switches of the Fabric, sending to each Switch the SFC SW_ILS.

a) If one or more Switches reject the SFC SW_ILS, then this may mean that these Switches are not able to support and enforce the new Zoning Database. The Fabric is unable to have a consistent Zoning Database, and so the Zone Server shall reject the CMIT CT request.

b) If all the Switches accept the SFC SW_ILS, then the Fabric is able to have a consistent Zoning Database. For certain operations the Managing Switch may need to send a second SFC message to each Managed Switch (see 10.6.7). When all switches have accepted the SFCs they are prepared to successfully update the zoning definitions. Then the Managing Switch shall

send to each other Switch the UFC SW_ILS to make the staged Zoning Database the Fabric Zoning Database.

To terminate the server session, the Management Application shall send a Server Session End (SSE) CT request to the Zone Server. Then the Managing Switch shall end the Fabric Management Session by sending an RCA SW_ILS to every Switch in the Fabric. When every RCA has been accepted, the Managing Switch shall destroy the copy of the Zoning Database and the Zone Server shall accept the SSE CT request.

If a management application does not send a CMIT request inside a server session, then every modification that it may have performed is not applied to the Fabric and is lost. The management Application may issue the CMIT request more than one time inside the same server session.

If after a successful SSB CT request a SSE CT request is never received (i.e, the management application locks the Fabric and then crashes), the Fabric Zone Server shall close the server session if it does not receive any Enhanced Zoning Commands for 2 minutes with a 10% tolerance. Consequently, the Managing Switch shall release the lock over the Fabric. Management applications are expected to keep the Fabric Management Session alive even in absence of management inputs.

If instead the Managing Switch crashes while locking the Fabric, the other Switches may detect the situation and release the lock.

10.6.7 Fabric Behavior to Handle the CT SFEZ Request

If the Fabric is functioning in Basic mode, and the SFEZ command has requested that the Zoning operational mode of the Fabric be changed to Enhanced, the Switch handling the CT SFEZ request shall initiate a Fabric Management Session. This causes a redistribution of the existing Zoning Database to the other Switches of the Fabric by using the operation request value ‘Activate Zone Set Enhanced’ in a Stage Fabric Configuration (SFC) SW_ILS. In this manner the Zoning Database is distributed using the Enhanced Zoning Data structures described in 10.4.4. If this step is successful, then the Zoning Policy Flags requested by the SFEZ command are propagated to the other Switches of the Fabric by using the operation request value ‘Set Zoning Policies’ in a second Stage Fabric Configuration (SFC) SW_ILS. If this step is successful, the UFC SW_ILS is used to apply the new configuration, changing the Zoning operational mode of all Switches in the Fabric to Enhanced mode. The Fabric Management Session shall then be released.

If the Fabric is functioning in Enhanced mode, then only the SFEZ command changes the Fabric’s Zoning Policies. This causes a Fabric Management Session to be initiated and an SFC SW_ILS to be sent to all Switches. The operation request value ‘Set Zoning Policies’ shall be used in this SFC SW_ILS. If a Fabric Management Session is already active, then the SFEZ command shall generate an immediate SFC SW_ILS with operation request value ‘Set Zoning Policies’ and an UFC SW_ILSs, keeping alive the existing Fabric Management Session.

NOTE 42 – Zoning structures managed within the Enhanced Zoning Framework may not be subsequently managed using management operations defined in the Basic Zoning Framework. In addition, no mechanism exists to convert from Enhanced Zoning mode to Basic Zoning mode.

10.6.8 Fabric Behavior to Handle the CT AAPZ and RAPZ Requests

The AAPZ and RAPZ CT Requests (see FC-GS-7) modify the Active Zone Set, therefore they require a lock of the Fabric, however they are not processed inside a Server Session. An implementation shall not wait more than one minute after receiving an AAPZ or RAPZ Request before attempting to acquire a Fabric lock to change the Active Zone Set. This enables coalescing multiple AAPZ and RAPZ Requests into a single change to the Active Zone Set through a single Fabric lock.

10.7 Switch Behaviors During Merge

To facilitate interoperability between newer and older Switches, an SW-4 Switch shall inspect the ELP Revision field and send the appropriate Merge Request Protocol Version. If the ELP Revision field is 2, the link should isolate or the Merge Request should have a Protocol Version = 0. If the ELP Revision field is greater than 2, the SW-4 Switch may send the configured Merge Request Protocol Version or the link shall be Isolated.

11 Distributed broadcast

11.1 Overview

Distributed broadcast provides a mechanism to distribute Broadcast frames without duplicating them or forming a loop. It is based on the FSPF tree with additional rules to provide one and only one broadcast path. The key is to build the same spanning tree with the same root for all switches.

Broadcast frames shall be Class 3 frames with a D_ID of FFFFFFh specified.

11.2 Spanning tree

The following list of rules are used to determine which ISLs are broadcast ISLs, and which are not:

1) Lowest Domain_ID Switch becomes the root of the tree;

2) Build the Shortest Path tree using FSPF cost as metric;

3) If a link is advertised with different costs in the two directions, the cost advertised by the Switch nearer to the root Switch (i.e., the upstream Switch) shall be used to determine the lowest cost path;

NOTE 43 – This is necessary when the two switches advertise different costs for ISLs;

4) If a Switch has multiple equivalent paths to the root of the tree, the ISL to the upstream Switch with the lowest Domain_ID shall be selected;

5) If there are multiple equivalent ISLs between a pair of switches, the ISL connected to the upstream Switches' lowest E_Port Index shall be selected;

6) The E_Ports selected in this process (broadcast member E_Ports) are used to forward broadcast frames;

The following set of rules are used for forwarding broadcast frames through the Fabric once the broadcast member E_Ports are identified:

a) A broadcast frame received on any Fx_Port is forwarded on all broadcast member E_Ports. In addition if zoning is enabled the broadcast frame is forwarded to all other Fx_ports in the broadcast zone on that Switch, or if zoning is not enabled on all other Fx_ports on that Switch;

b) A broadcast frame received on any broadcast member E_Port is forwarded on all other broadcast member E_Ports on that Switch. In addition if zoning is enabled the broadcast frame is forwarded to all other Fx_ports in the broadcast zone on that Switch, or if zoning is not enabled on all other Fx_ports on that Switch;

c) A broadcast frame received on any other port is discarded.

11.2.1 Spanning tree example

In the example in figure 29, the Switch with Domain_ID 5 is the root of the broadcast tree. From Domain_ID 7 to Domain_ID 5, there are 2 equal cost ISLs, so the path from Domain_ID 7 port 5 to Domain_ID 5 port 1 becomes the broadcast ISL. From Domain_ID 30 to the root, there are multiple equal cost paths. Domain_ID 7 is the lowest upstream Switch, therefore, this path is chosen. Secondly,

there are multiple equal cost ISLs to Domain_ID 7, the ISL from Domain_ID 30 port 1 to Domain_ID 7 port 6 is chosen because this is the lowest E_Port index on the upstream Switch.

A broadcast frame received on an Fx_Port on Domain_ID 30 is forwarded to Domain_ID 7 E_Port index 6. From there it is forwarded to Domain_ID 5 E_Port index 1, and from there to Domain_ID 12 E_Port index 2.

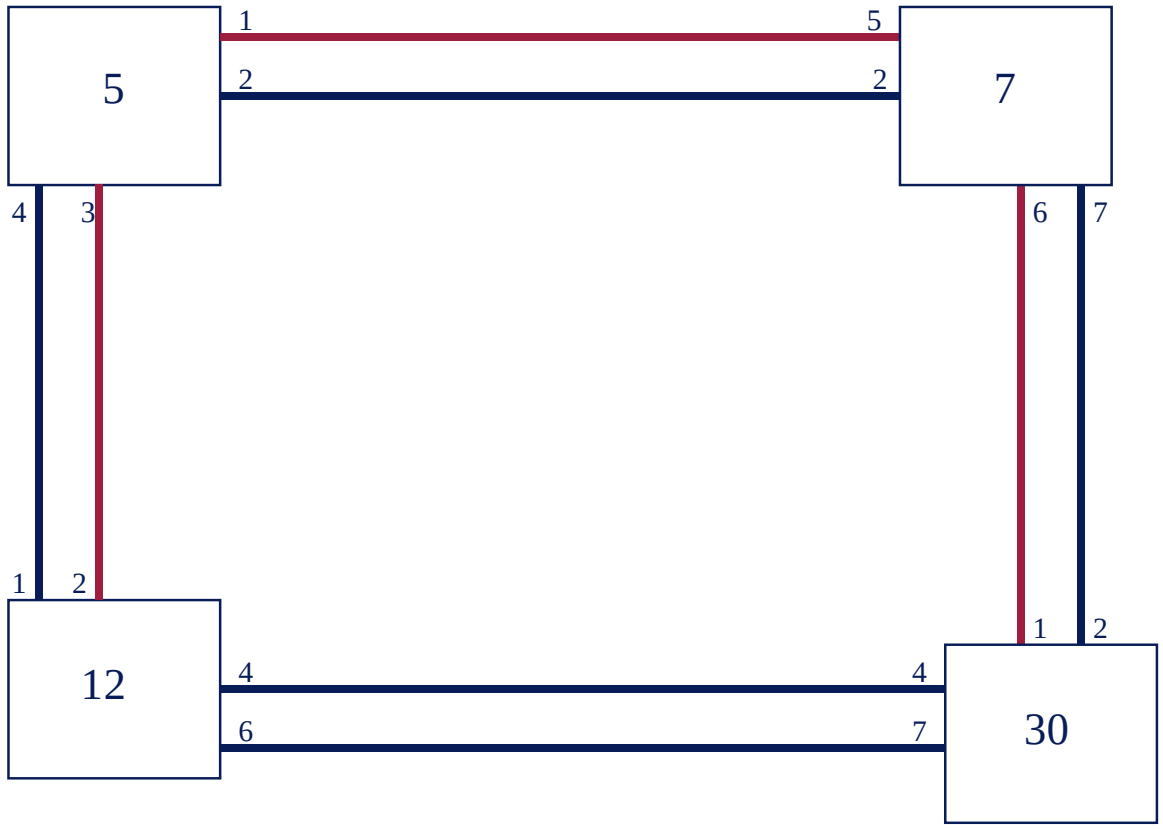

Figure 29 – Broadcast path selection example

12 Virtual Fabrics Switch Support

12.1 Overview

The Virtual Fabric Tagging Header (VFT_Header, see FC-FS-3) allows Fibre Channel frames to be tagged with the Virtual Fabric Identifier (VF_ID) of the Virtual Fabric (VF) to which they belong. Tagged frames (i.e., frames with a VFT_Header) belonging to different Virtual Fabrics may be transmitted over the same physical link. By combining VFT_Headers and other features, Virtual Fabrics provide compartmentalization of access and management. The VFT_Header may be supported by PN_Ports, PF_Ports and PE_Ports.

The use of VFT_Header between PE_Ports allows implementation of Virtual Fabrics without requiring any change in the PN_Port/Fx_Port interface, as shown in figure 30.

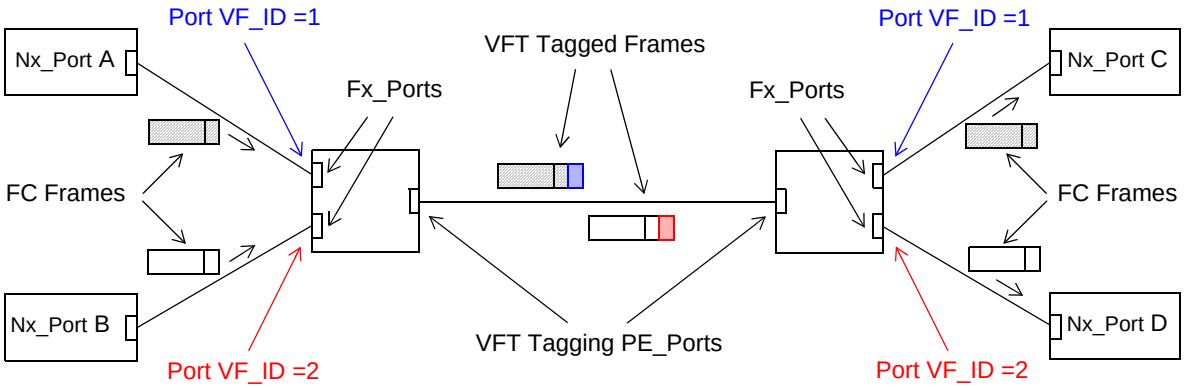

Figure 30 – Virtual Fabrics

Each Fx_Port of a VF capable Switch (i.e., a Switch supporting Virtual Fabrics) shall have a configurable Port VF_ID, so that Nx_Ports may access Virtual Fabric features without modifications. The Port VF_ID shall be associated to any untagged Fibre Channel frame received by the Fx_Port. A VF capable Switch shall perform frame forwarding by considering the Virtual Fabric the frame belongs to, as identified by the VF_ID. When transmitted between a pair of tagging PE_Ports (i.e., PE_Ports belonging to FC_Ports processing the VFT_Header), each Fibre Channel frame shall be tagged with a VFT_Header. When a VFT_Header tagged frame is received by a tagging PE_Port of a VF capable Switch, the VF_ID carried in the VFT_Header shall be used to perform frame forwarding, together with the D_ID carried in the Frame_Header.

As shown in figure 30, the Fibre Channel frames sent by Nx_Port A are associated with the Virtual Fabric having VF_ID 1 when received by the Fx_Port. The VF_ID is used by the Switch to perform frame forwarding. Frames transmitted over the tagging PE_Port are tagged with a VFT_Header carrying the VF_ID, and their CRC is recomputed (see FC-FS-3). The receiving tagging PE_Port retrieves from the VFT tagged frame the VF_ID and uses it together with the D_ID carried in the Frame_Header to route the frames to Nx_Port C. The Fx_Port connected to the destination Nx_Port C removes the VFT_Header, recomputes the CRC (see FC-FS-3) and delivers the original Fibre Channel frames.

NOTE 44 – Under rare circumstances VF_IDs may be aliased, resulting in multiple VF_IDs referring to the same Virtual Fabric.

12.2 VF Capable Switch Functional Model

A functional model of a VF capable Switch is shown in figure 31.

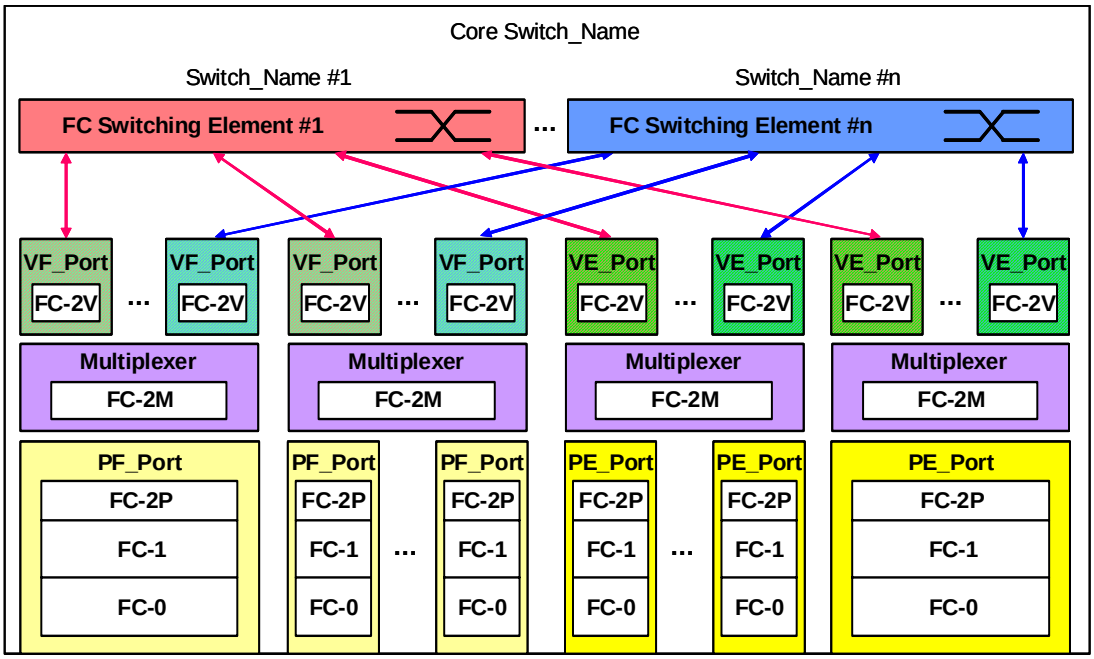

Figure 31 – Functional model of a VF capable Switch

A VF capable Switch is functionally a collection of multiple Switching Elements hosted in the same Core Switch. There is one Switching Element per each Virtual Fabric hosted on the Core Switch.

Each Switching Element is identified by an unique Switch_Name. In addition, the Core Switch is identified by a unique Core Switch_Name. Each Virtual Fabric is identified by a 12-bit Virtual Fabric Identifier (VF_ID). VF_ID Identifiers are administratively set using a management interface.

Each Switching Element is connected to one or more VF_Ports or VE_Ports. Each VF_Port or VE_Port is connected to a single Switching Element. Multiple VF_Ports belonging to different Virtual Fabrics may share one or more PF_Ports through the multiplexing and tagging functions of the Multiplexer. Multiple VE_Ports belonging to different Virtual Fabrics may share one or more PE_Ports through the multiplexing and tagging functions of the Multiplexer.

Physical links are shared across multiple Virtual Fabrics using the VFT_Header. The Multiplexer functions of multiplexing and tagging logic are driven by the VF_ID in the VFT_Header. Upon receiving a VFT tagged frame from a PF_Port or PE_Port, the Multiplexer logic delivers the frame to the appropriate VF_Port or VE_Port connected with the appropriate Switching Element. (i.e., the Switching Element associated with the Virtual Fabric whose VF_ID is carried in the VFT_Header).

When transmitted between a pair of tagging E_Ports, each Fibre Channel frame shall be tagged with a VFT_Header. A VF capable Switch shall perform frame forwarding by considering the Virtual Fabric the frame belongs to, as identified by the VF_ID.

Each Switch Port of a VF capable Switch shall have a configurable Port VF_ID. The Port VF_ID shall be associated to any untagged Fibre Channel frame received by the Switch Port. This allows the interconnection of VF capable Switches with non VF capable Switches. Any untagged Fibre Channel

frame received by an E_Port or Fx_Port on a VF capable Switch shall be implicitly associated with the Port VF_ID for processing. The Port VF_ID is then used by the tagging logic to deliver the frame to the appropriate Switching Element. In absence of any explicit configuration, the value 001h should be used as default Port VF_ID.

Switches supporting Virtual Fabrics may not receive VFT tagged frames on all Switch Ports. This may occur for the following reasons:

a) A Switch Port is administratively configured to not use VFT_Headers;

b) The port at the far end of a link is administratively configured to not use VFT Headers; or

c) The port at the far end of a link is not capable of processing VFT_Headers.

12.3 Switch_Names Usage

The Switch_Names of the Switching Elements and the Core Switch_Name shall be used as follows:

a) In state P5 (see figure 11), the Switch_Name of the Switching Element associated with the Port VF_ID shall be used when transmitting the ELP SW_ILS;

b) In state P17 (see figure 12), the Switch_Name shall abide by the rules in FC-SP; and

c) When a Switch Port initializes as an E_Port in state P10 (see figure 12), the Switch_Name of the Switching Element associated with the Port VF_ID shall be used for any subsequent operation or protocol.

12.4 Configuration Information

A VF capable Switch shall maintain the following configuration parameters per each Switch Port:

a) Tagging Administrative Status, used to negotiate the VFT tagging operational mode of the Switch Port (see 6.1.25.2.2);

b) Port VF_ID (see 12.2 and 6.1.25.2.3); and

c) Locally-Enabled VF_ID List, used to negotiate the list of Virtual Fabrics operational over the Switch Port (see 6.1.25.2.4).

12.5 Enabling VFT Tagging on Switch Ports

Figure 32 shows the extended Switch Port Initialization state machine enhanced to enable Virtual Fabrics. In state P13 (see figure 12), two Switch Ports may negotiate to perform the EVFP processing (see 12.6) if both of them support Virtual Fabrics. The support for Virtual Fabrics is indicated by the ‘Virtual Fabrics Supported’ Protocol ID value shown in table 71.

The Switch Port sending the ESC request indicates support for Virtual Fabrics by including the ‘Virtual Fabrics Supported’ Protocol ID shown in table 71 in the ESC payload. The replying Switch Port selects to negotiate Virtual Fabrics parameters by choosing the ‘Virtual Fabrics Supported’ Protocol ID shown in table 71 in the ESC SW_ACC payload. Then the Switch Port Initialization proceeds to state P21 or to state P22.

A Switch Port connected to a B_Port shall not indicate nor select the support for Virtual Fabrics in the ESC protocol if the directly connected B_Port did not announce support for Virtual Fabric Tagging by setting to one the ‘Bridge Virtual Fabrics’ flag in the ELP SW_ILS.

If one of the two Switch Ports does not support Virtual Fabrics, the Switch Port Initialization proceeds to state P17 (see figure 11).

When VFT tagging is enabled on a link, a Link Reset (see FC-FS-3) shall not change the tagging process, while a Link Initialization (see FC-FS-3) shall stop the tagging process and bring the involved Switch Ports to state P0 (see figure 12).

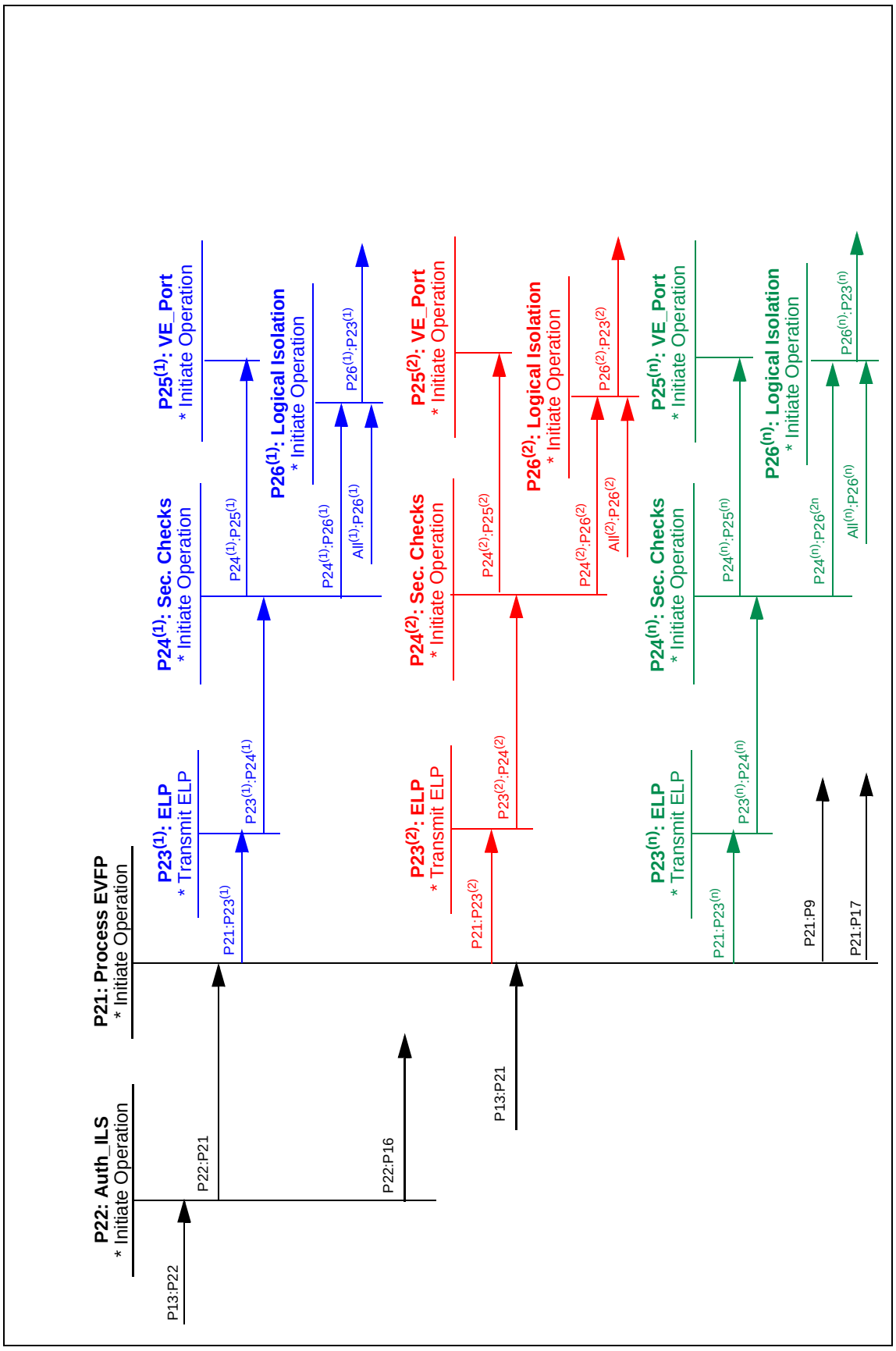

Figure 32 – Switch Port Mode Initialization State Machine - Virtual Fabric Support

Transition P13:P21. Occurs when the two Switch Ports negotiated to perform the EVFP processing, and neither Switch Port requires Authentication in state P22.

Transition P13:P22. Occurs when the two Switch Ports negotiated to perform the EVFP processing, and authentication is required by at least a Switch Port in state P22.

State P22: AUTH_ILS. While in this state an Authentication transaction (see FC-SP) shall be performed. If the Port receives an EVFP before authentication checks are complete, the Port shall respond with an SW_RJT, indicating a reason code of Logical Busy and a reason code explanation of Security Checks in Progress. Switch_Name usage shall abide by the rules defined in FC-SP.

NOTE 45 – If authentication is supported at state P22, implementations should provide the ability for an administrator to disable authentication in state P22.

Transition P22:P16. Occurs when the Authentication transaction performed in state P22 fails.

Transition P22:P21. Occurs when the Authentication transaction performed in state P22 completes successfully.

State P21: Process EVFP. The Switch Port shall perform EVFP processing as described in 12.6.

Transition P21:P17. Occurs when the EVFP processing determined that VFT tagging is not performed and the two Switch Ports have the same Port VF_ID.

Transition P21:P9. Occurs when the EVFP processing determined that VFT tagging is not performed and the two Switch Ports have a different Port VF_ID.

Transition P21:P23$^{(k)}$. Occurs when the EVFP processing determined that VFT tagging is performed. There is a different state for each Virtual Fabric negotiated to be used on the link. The state for Virtual Fabric K is denoted P23$^{(k)}$.

State P23$^{(k)}$: ELP. In this state the Fibre Channel frames transmitted by the Switch Port are tagged with the VFT_Header carrying VF_ID K. An ELP, tagged with VF_ID K, is transmitted. This ELP shall carry the Switch Name of the Switching Element associated with VF_ID K and the operational parameters (e.g., timeout values, Classes of service) of Virtual Fabric K. No flow control configuration is required in this state, because it is performed in state P5.

Transition P23$^{(k)}$:P24$^{(k)}$. Occurs when the ELP processing in state P23 is completed.

State P24$^{(k)}$: Security Checks. In this state the Fibre Channel frames transmitted by the Switch Port are tagged with the VFT_Header carrying VF_ID K. The Switch Port initiates and responds to all required security checks (see FC-SP), if any. If the Port receives an EFP before authentication checks are complete, the Port shall respond with an SW_RJT, indicating a reason code of Logical Busy and a reason code explanation of Security Checks in Progress.

NOTE 46 – If authentication is supported at state P24, implementations should provide the ability for an administrator to disable authentication in state P24.

Transition P24$^{(k)}$:P25$^{(k)}$. Occurs when the Security Checks performed in state P24$^{(k)}$ complete successfully.

State P25$^{(k)}$: VE_Port. In this state the Switch Port operates as VFT tagging PE_Port. Fibre Channel frames transmitted by the Switch Port are tagged with the VFT_Header carrying VF_ID K. The

VE_Port shall participate in the next phase of Fabric Configuration in Virtual Fabric K. The Switch_Name of the Switching Element associated with VF_ID K shall be used for any subsequent operation or protocol in Virtual Fabric K.

State P26$^{(k)}$: Logical Isolation. In this state the VE_Port corresponding to Virtual Fabric K becomes logically Isolated (i.e., in Virtual Fabric K no Class N traffic flows and only the SW_ILSs specified in 7.6 may be communicated).

Transition P24$^{(k)}$:P26$^{(k)}$. Occurs when the Security Checks performed in state P24$^{(k)}$ complete unsuccessfully.

Transition AII$^{(k)}$:P26$^{(k)}$. Occurs when a protocol in Virtual Fabric K causes the corresponding VE_Port to go in Logical Isolation state for any of the reasons listed in 7.6.

Transition P26$^{(k)}$:P23$^{(k)}$. Occurs when the Logically Isolated VE_Port corresponding to Virtual Fabric K receives or transmits an ELP.

12.6 Exchange Virtual Fabrics Parameters Processing

12.6.1 Overview

The Exchange Virtual Fabrics Parameters (EVFP) protocol allows peers of Interconnect_Ports belonging to VF capable Switches to:

a) Negotiate the VFT Tagging operational mode;

b) Verify the consistency of the two Port VF_IDs; and

c) Establish the list of operational Virtual Fabrics across the Inter Switch Link.

An EVFP transaction occurs between an EVFP Initiator and an EVFP Responder. An EVFP transaction (see figure 33) is identified by a unique Transaction Identifier (T_ID), and consists of a synchronizing phase (EVFP_SYNC) followed by a commit phase (EVFP_COMMIT).

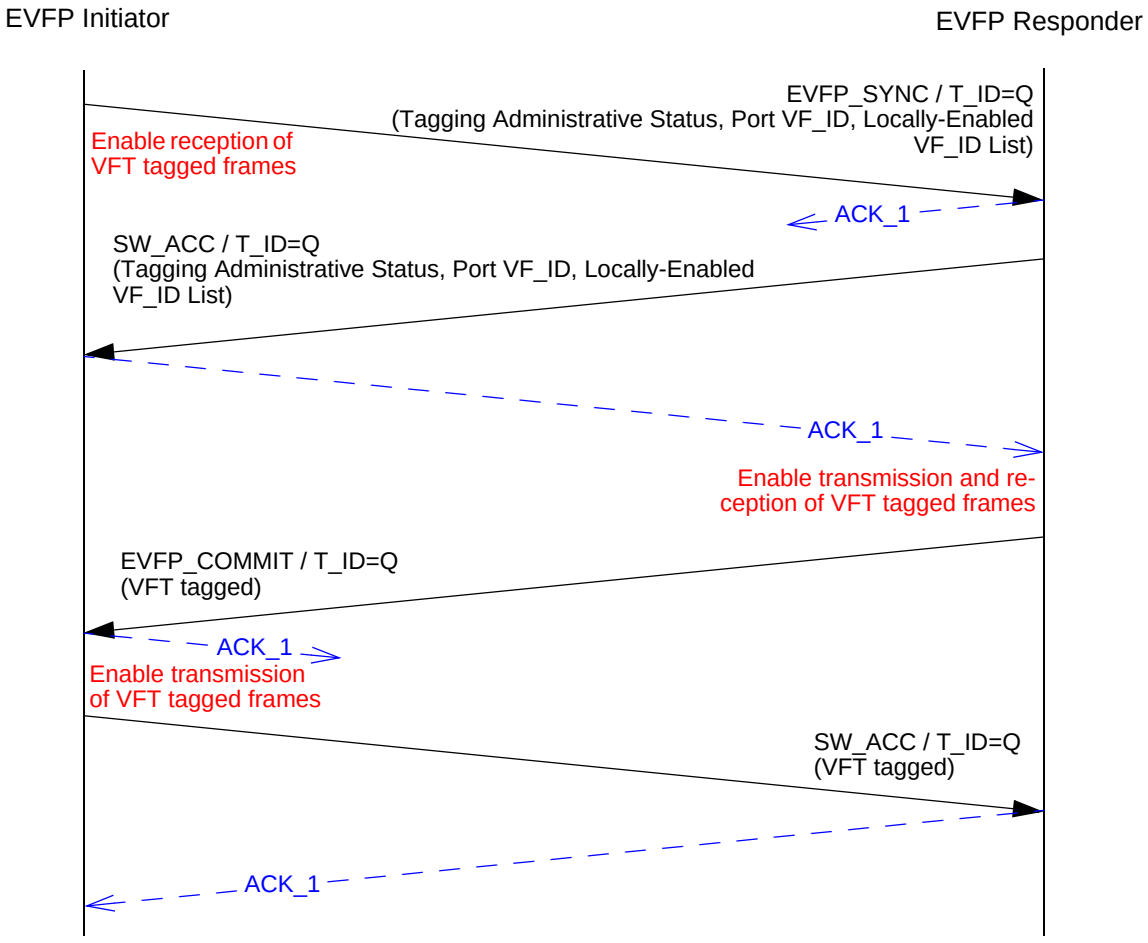

Figure 33 – A Generic EVFP Transaction

The VF_ID value FEFh is used by the EVFP protocol for certain operations and is referred to as Control VF_ID. The EVFP protocol, during the Switch Port Initialization, proceeds as follows:

1) The EVFP Initiator shall start the EVFP transaction by sending the EVFP_SYNC message (see 6.1.25.2) to the EVFP Responder. In the EVFP_SYNC message, the EVFP Initiator shall specify the Transaction Identifier, and shall send its Core Switch_Name, together with its Tagging Administrative Status (see 6.1.25.2.2), Port VF_ID (see 6.1.25.2.3) and Locally-Enabled VF_ID List (see 6.1.25.2.4). On sending the EVFP_SYNC message the EVFP Initiator enables the reception of VFT tagged frames;

2) The EVFP Responder shall reply with an EVFP_SYNC SW_ACC carrying its Tagging Administrative Status, Port VF_ID and Locally-Enabled VF_ID List. Then the EVFP Responder shall determine if VFT Tagging is to be enabled on the link, according to table 104. If VFT Tagging is to be enabled, the EVFP Responder shall go to step 3. If VFT Tagging is not to be enabled, the EVFP Responder shall check the received peer's Port VF_ID:

   a) if the peer's Port VF_ID is not equal to the local Port VF_ID, on completion of the Exchange (i.e., on receiving the ACK_1 for the EVFP_SYNC SW_ACC) the EVFP protocol terminates and the EVFP Responder goes in Isolated state (transition P21:P9, see 12.5); or

   b) if the peer's Port VF_ID is equal to the local Port VF_ID, on completion of the Exchange (i.e., on receiving the ACK_1 for the EVFP_SYNC SW_ACC) the EVFP protocol terminates and the EVFP Responder goes in state P17 (transition P21:P17, see 12.5).

   On receiving the EVFP_SYNC SW_ACC, the EVFP Initiator shall determine if VFT Tagging is to be enabled on the link, according to table 104. If VFT Tagging is to be enabled, on completion of the Exchange (i.e., on sending the ACK_1 for the EVFP_SYNC SW_ACC) the EVFP Initiator shall enable the reception of VFT tagged frames in its Port VF_ID and shall go to step 4. If VFT Tagging is not to be enabled, the EVFP Initiator disables the reception of VFT tagged frames and shall check the received peer's Port VF_ID:

   a) if the peer's Port VF_ID is not equal to the local Port VF_ID, on completion of the Exchange (i.e., on sending the ACK_1 for the EVFP_SYNC SW_ACC) the EVFP protocol terminates and the EVFP Initiator goes in Isolated state (transition P21:P9, see 12.5); or

   b) if the peer's Port VF_ID is equal to the local Port VF_ID, on completion of the Exchange (i.e., on sending the ACK_1 for the EVFP_SYNC SW_ACC) the EVFP protocol terminates and the EVFP Initiator goes in state P17 (transition P21:P17, see 12.5);

3) On completion of the EVFP_SYNC Exchange (i.e., on receiving the ACK_1 for the EVFP_SYNC SW_ACC), the EVFP Responder shall enable both transmission and reception of VFT tagged frames for the Virtual Fabrics operational on the link, computed as explained in 6.1.25.2.4. Transmission and reception of VFT tagged frames for the Control VF_ID shall be implicitly enabled. Transmission and reception of VFT tagged frames for the EVFP Initiator's Port VF_ID shall be also enabled on the link, to allow a successful completion of the EVFP protocol. Then the EVFP Responder shall send an EVFP_COMMIT message (see 6.1.25.3), tagged with the EVFP Initiator's Port VF_ID; and

4) On receiving the VFT tagged EVFP_COMMIT, the EVFP Initiator shall enable both transmission and reception of VFT tagged frames for the Virtual Fabrics operational on the link, computed as explained in 6.1.25.2.4. Transmission and reception of VFT tagged frames for the Control VF_ID shall be implicitly enabled. Transmission and reception of VFT tagged frames for the EVFP Initiator's Port VF_ID shall be also enabled on the link, to allow a successful com-

pletion of the EVFP protocol. Then the EVFP Initiator shall send an EVFP_COMMIT SW_ACC message tagged with its Port VF_ID.

When tagging is enabled the EVFP transaction completes successfully on completion of the EVFP_COMMIT Exchange, for both the EVFP Initiator and EVFP Responder. If the computed set of VF_IDs operational on the link does not include the EVFP Initiator's Port VF_ID, transmission and reception of VFT tagged frames for such VF_ID shall be disabled on the link upon completion of the EVFP transaction. When the EVFP transaction is completed the processing continues independently for each Virtual Fabric operational on the link, as shown by transitions P21:P23$^{(k)}$ (see 12.5). If the computed set of VF_IDs operational on the link is NULL, the involved Switch Ports remain in state P21 (see 12.5) until a new EVFP transaction is performed in the Control VF_ID.

If two Interconnect_Ports start an EVFP transaction at the same time, or if an Interconnect_Port is acting as an EVFP Initiator and receives an EVFP_SYNC message from the designated EVFP Responder, one of the two EVFP transactions shall be aborted. The Interconnect_Port that sent the EVFP_SYNC message with the numerically higher Core Switch_Name shall remain the EVFP Initiator, while the Interconnect_Port that sent the EVFP_SYNC message with the numerically lower Core Switch_Name shall become the EVFP Responder. The Interconnect_Port that remains the EVFP Initiator shall reply to the received EVFP_SYNC message with a 'EVFP collision' SW_RJT (see 6.1.25.1). The Interconnect_Port that becomes the EVFP Responder shall reply to the received EVFP_SYNC message and abort its own transaction upon receipt of the SW_RJT.

The EVFP protocol is used also when some Switch Port configuration information (see 12.4) are changed by a management action. The EVFP messages may be carried in Fibre Channel frames tagged with the Port VF_ID if the EVFP protocol begins while the link is not performing VFT tagging (see 12.6.1). The EVFP messages are carried in Fibre Channel frames tagged with the Control VF_ID if the EVFP protocol begins while the link is performing VFT tagging (see 12.6.2 and 12.6.3).

12.6.2 Changing the VFT Tagging Mode

When a management action changes the Administrative Tagging Mode of an E_Port belonging to a VF capable Switch that determined during initialization the peer supports the EVFP protocol, the E_Port shall determine if the link has to change its VFT Tagging mode (i.e., if it has to transition from tagging to untagging mode or from untagging to tagging mode) by acting as EVFP Initiator as follows. If the E_Port is currently performing tagging, all EVFP protocol messages shall be tagged with the Control VF_ID. If the E_Port is currently not performing tagging, all EVFP protocol messages shall be untagged.

1) The EVFP Initiator shall start the EVFP transaction by sending the EVFP_SYNC message to the EVFP Responder. The EVFP_SYNC message shall carry the updated Tagging Administrative Status (see 6.1.25.2.2), Port VF_ID, and the Locally-Enabled VF_ID List; and

2) The EVFP Responder shall reply with an EVFP_SYNC SW_ACC carrying its Tagging Administrative Status, Port VF_ID and Locally-Enabled VF_ID List. The EVFP Responder shall determine if VFT Tagging has to be changed on the link, according to table 104. The EVFP Responder:

    a) if VFT Tagging has not to be changed, on completion of the Exchange (i.e., on receiving the ACK_1 for the EVFP_SYNC SW_ACC) terminates the EVFP protocol; or

    b) if VFT Tagging has to be changed, on completion of the Exchange (i.e., on receiving the ACK_1 for the EVFP_SYNC SW_ACC) shall perform a link initialization.

On receiving the EVFP_SYNC SW_ACC, the EVFP Initiator shall determine if VFT Tagging has to be changed on the link, according to table 104. The EVFP Initiator:

a) if VFT Tagging has not to be changed, on completion of the Exchange (i.e., on sending the ACK_1 for the EVFP_SYNC SW_ACC) terminates the EVFP protocol; or

b) if VFT Tagging has to be changed, shall participate in the link initialization initiated by the EVFP Responder.

12.6.3 Adding or Removing Virtual Fabrics

When a management action changes the Locally-Enabled VF_ID List over a tagging E_Port, the E_Port shall initiate the EVFP protocol by acting as EVFP Initiator as follows. All EVFP protocol messages shall be tagged with the Control VF_ID.

1) The EVFP Initiator shall start the EVFP transaction by sending the EVFP_SYNC message to the EVFP Responder. The EVFP_SYNC message shall carry the Tagging Administrative Status, Port VF_ID, and the updated Locally-Enabled VF_ID List (see 6.1.25.2.4);

2) The EVFP Responder shall reply with an EVFP_SYNC SW_ACC carrying its Tagging Administrative Status, Port VF_ID and Locally-Enabled VF_ID List. The EVFP Responder, depending on the resulting operational VF_ID List (see 6.1.25.2.4):

   a) if the operational VF_ID List did not change, terminates the EVFP protocol on completion of the Exchange (i.e., on receiving the ACK_1 for the EVFP_SYNC SW_ACC) in the Control VF_ID; or

   b) if the operational VF_ID List did change, performs step 3 on completion of the Exchange (i.e., on receiving the ACK_1 for the EVFP_SYNC SW_ACC) in the Control VF_ID.

   On receiving the EVFP_SYNC SW_ACC in the Control VF_ID, the EVFP Initiator, depending on the resulting operational VF_ID List (see 6.1.25.2.4):

   a) if the operational VF_ID List did not change, terminates the EVFP protocol on completion of the Exchange (i.e., on sending the ACK_1 for the EVFP_SYNC SW_ACC) in the Control VF_ID; or

   b) if the operational VF_ID List did change, performs step 4 on completion of the Exchange (i.e., on sending the ACK_1 for the EVFP_SYNC SW_ACC) in the Control VF_ID.

3) On completion of the EVFP_SYNC Exchange (i.e., on receiving the ACK_1 for the EVFP_SYNC SW_ACC) in the Control VF_ID, the EVFP Responder shall apply the updated operational VF_ID List, enabling the added Virtual Fabrics and disabling the removed Virtual Fabrics. Then the EVFP Responder shall send an EVFP_COMMIT message; and

4) On receiving the EVFP_COMMIT message, the EVFP Initiator shall apply the updated operational VF_ID List, enabling the added Virtual Fabrics and disabling the removed Virtual Fabrics. Then the EVFP Initiator shall send an EVFP_COMMIT SW_ACC message.

When the operational VF_ID List changes, the EVFP transaction completes successfully on completion of the EVFP_COMMIT Exchange for both the EVFP Initiator and EVFP Responder. When the EVFP transaction is completed, the updated operational VF_ID List is operative.

12.6.4 Changing the Port VF_ID

When a management action changes the Port VF_ID of a tagging PE_Port, no changes are applied to the link.

When a management action changes the Port VF_ID of a non-tagging PE_Port, the PE_Port shall perform a link initialization.

When a management action changes the Port VF_ID of a Switch Port in Isolated state, the Switch Port shall go in state P0 (see figure 11).

13 Enhanced Commit Service

13.1 Overview

The Enhanced Commit Service (ECS) builds on the Fabric Management Session Protocol defined in clause 10 to provide a general mechanism to manage the serialization and updating of Fabric resources. ECS provides:

a) serialization and locking of resources on a per fabric application basis;

b) error recovery;

c) transaction semantics.

ECS may operate in assisted mode or in autonomous mode. When operating in assisted mode the protocol processing is controlled by a fabric application (see 13.2). In this case assisted Mode does not perform or enable the error recovery defined by the enhanced commit service. When operating in autonomous mode the protocol processing proceeds as described in 13.3 and provides error recovery during the commit process.

The enhanced commit service internal link services (EACA, ESFC, EUFC, ERCA, and TCO) are defined in this standard and are based on the zoning update commands (ACA, SFC, UFC, RCA) also defined in this standard (see 6.1).

13.2 Assisted Mode Protocol Operations

When operating in assisted mode (see 6.1.26.1.3) the ECS protocol processing begins, proceeds and terminates under the control of a fabric application.

A Switch begins the ECS protocol processing when requested by the fabric application (e.g., when an SSB CT Request is received by the Security Server, see FC-SP). The Switch begins the ECS protocol attempting to lock the Fabric for the specified fabric application, by sending an EACA Request to the Switches specified in the ECS Switch List, following the order of the ECS Switch List. If all these Switches accept the EACA Request, then the sending Switch becomes the ECS managing Switch for the transaction and shall return the control to the fabric application with a success indication (e.g., by replying with an SSB CT Accept). If one or more Switches rejects the EACA Request, then the sending Switch shall send an ERCA Request to the Switches that accepted the EACA Request and shall return the control to the fabric application with an error indication (e.g., by replying with an SSB CT Reject).

Once the Fabric is locked for the specified fabric application, the ECS protocol processing proceeds under the control of the fabric application. According to the requests received from the fabric application, the ECS managing Switch may distribute and commit information to the Switches participating in the ECS transaction by sending one or more ESFC and EUFC Requests to the Switches specified in the ECS Switch List, following the order of the ECS Switch List. A success indication shall be returned to the fabric application if all Switches participating in the ECS transaction accept the ESFC or EUFC Request. An error indication shall be returned to the fabric application if one ore more of the Switches participating in the ECS transaction reject an ESFC or EUFC Request.

The ECS protocol processing terminates when the fabric application requests to unlock the Fabric (e.g., when an SSE CT Request is received by the Security Server, see FC-SP). The ECS managing Switch unlocks the Fabric for the specified fabric application by sending an ERCA request to the Switches specified in the ECS Switch List, following the order of the ECS Switch List.

The TCO SW_ILS is not used by the ECS protocol when operating in assisted mode.

13.3 Autonomous Mode Protocol Operations

13.3.1 Protocol Phases

13.3.1.1 Overview

When operating in autonomous mode (see 6.1.26.1.3) the ECS protocol allows Fabric resources associated with a specific fabric application to be locked in each managed Switch. An ECS operation consists of one transaction bounded by an EACA request that begins the transaction, and an ERCA request that ends the transaction. Between the acceptance of the EACA by all managed Switches and the generation of the ERCA by the managing Switch, the ESFC and EUFC requests are used to update and commit application resources in each managed Switch.

An ECS operation affects a subset of Switches in the Fabric. The Switch List indicates the list of managed Switches for the ECS operation and indicates the order that the managing Switch sends ECS commands to the managed Switches. The Switch List also specifies which Switches are authorized to participate in ECS recovery processing.

13.3.1.2 Phase One

The managing Switch initiates the commit process by sending an EACA request to all Switches specified in the ECS Switch List. Once the EACA is accepted, the managing Switch and all the managed Switches of EACA, reserve the resources associated with the specified application. Once all managed Switches accept the EACA, the managing Switch transitions to phase two of the commit process.

If any Switch rejects the EACA, then the managing Switch aborts the commit process by sending an ERCA to all the Switches that accepted the EACA.

13.3.1.3 Phase Two

The managing Switch initiates the second phase of the commit process by sending an ESFC request to all managed Switches. The application specific data is validated and staged by the managed Switch receiving the ESFC. If all the managed Switches successfully complete the ESFC processing, then the managing Switch transitions to phase three of the commit process.

If the application data is invalid or consistency checks fail on the managed Switch, then the Switch rejects the ESFC. If a reject is received for an ESFC then the managing Switch performs an ERCA on all managed Switches and the commit process is aborted.

13.3.1.4 Phase Three

The managing Switch initiates phase three of the commit process by sending a EUFC request to all managed Switches. The data received from the prior ESFC request is committed by the managed Switch according to the requirements of the application. When all the managed Switches successfully complete the EUFC, then the managing Switch enters phase four of the commit process.

13.3.1.5 Phase Four

The managing Switch initiates phase four of the commit process by sending an ERCA request to all managed Switches. This causes the Switches to release the resources reserved by the relinquishes EACA and the commit process to conclude.

13.3.2 Handling Fabric Changes

If a Switch that was specified in the ECS Switch List leaves the Fabric after the EACA has been accepted, it is removed from the ECS Switch List and only those that are currently in the ECS Switch List are involved in the commit process. After the first ESFC is sent, the Managing Switch will continue the commit process regardless of changes to the Fabric. If a new Switch joins the Fabric during the commit process it would not participate in the commit process for this transaction. All operations relating to the application shall be held off until the commit process successfully completes or is aborted.

NOTE 47 – In certain cases, there is a slight chance that isolation of Switches in the Fabric during ECS may result in inconsistencies between Switches in the Fabric. These inconsistencies could prevent the Fabric from merging and user intervention may be required before the Fabric can merge.

13.3.3 Error Recovery

13.3.3.1 Overview

When the ECS protocol operates in autonomous mode, a mechanism is provided to select a new managing switch without interaction with the fabric application. Only Switches in the ECS Switch List are authorized to participate in the ECS recovery processing.

13.3.3.2 Managing Switch Not Functional

When the EACA is accepted by each managed Switch, each authorized Switch in the ECS Switch List begins monitoring for a domain unreachable condition associated with the managing Switch. When an authorized managed Switch determines that the managing Switch is unreachable, a new managing Switch shall be selected.

13.3.3.2.1 Dead Man Timer

When each authorized managed Switch determines that the managing Switch is unreachable and a transaction has been initiated (i.e., EACA received), a Dead Man timer is started that is based on the following equation:

Timer Value = 30sec * 'Switch position in list'.

The first authorized Switch in the ECS Switch List following the managing Switch shall assume a switch position of zero. Other authorized Switches after that assume positions in increasing order from zero. The Dead Man timer in each authorized Switch is set according to its position in the ECS Switch List. The Dead Man timer in each authorized Switch is kept alive until an ECS request is received from a Switch different than the previous managing Switch with the same Transaction_ID.

13.3.3.2.2 Basic Procedure

Typically, the first authorized Switch in the ECS Switch List following the managing Switch assumes the role of the new managing Switch. However, if that Switch is also non-operational then the next authorized Switch will assume the role of managing Switch and so forth. This will occur when the timer expires on a given Switch and no ECS requests are received with the same Transaction_ID.

When the new Switch assumes the role of managing Switch, it shall either issue an ERCA, or replay the current ECS phase with all remaining Switches specified in the ECS Switch List. When all Switches complete the current phase under the control of the new managing Switch, then the managing Switch proceeds to the next phase.

In the case where the new managing Switch determines that the process is in the EACA phase or the ERCA phase, then the new managing Switch shall send an ERCA to all remaining Switches in the ECS Switch List.

13.3.3.3 Resolution of Multiple Managing Switches

13.3.3.3.1 Two Managing Switches - Same Commit Phase

When two Switches have assumed the role of managing Switch and they are in the same commit phase, the managing Switch with the lowest domain_ID assumes the role of managing Switch and the other Switch relinquishes its role as managing Switch. This condition is detected when multiple ECS requests are received with the same Transaction_ID.

13.3.3.3.2 Two Managing Switches - Different Commit Phases

When two Switches have assumed the role of managing Switch, and one Switch is in a higher phase than the other, the managing Switch in the higher commit phase retains the role of managing Switch. To accomplish this the Switch in the higher commit phase returns an SW_RJT Reason Code of "Unable to perform command request and an SW_RJT Reason Code Explanation of "In Advanced Phase". When the Switch receives such a reject for an ECS request, it sends a TCO request to the Switch that rejected the ECS request for this phase. The Switch that receives the TCO then retains the role of the managing Switch and continues with the commit process. The managing Switch that originated the TCO relinquishes its role as a managing Switch and becomes a managed Switch.

13.3.4 Ladder Diagrams

13.3.4.1 Normal Case

The diagram below depicts the interactions for the successful case.

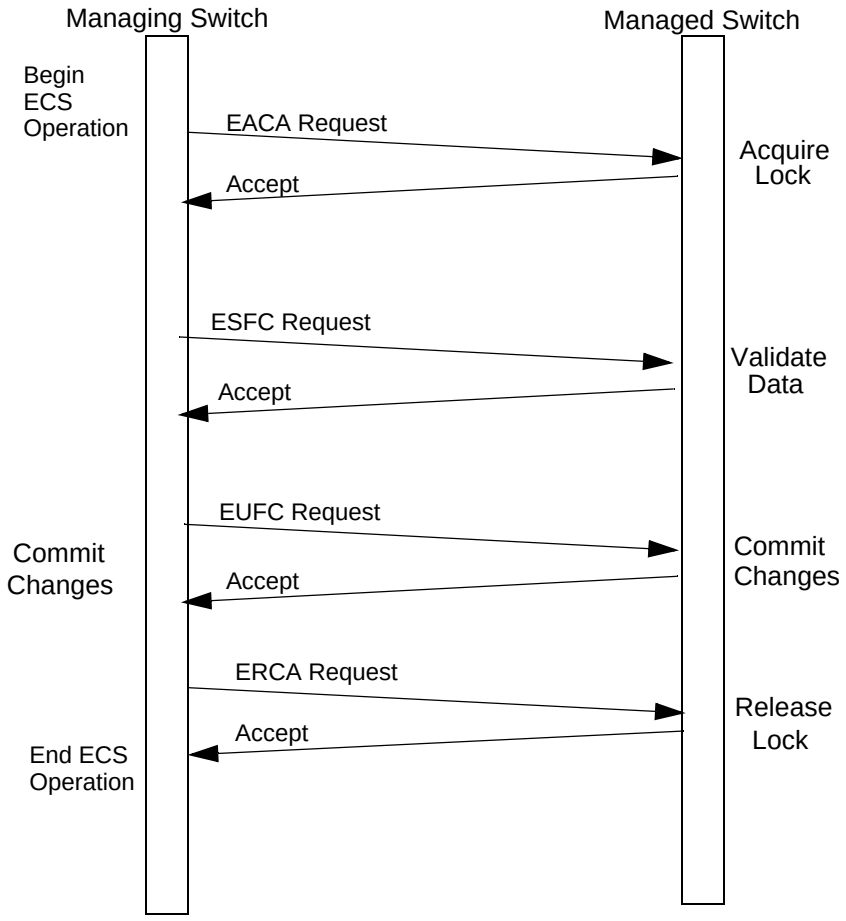

Figure 34 – Normal Commit Ladder Diagram

13.3.4.2 Unsuccessful Case

The diagram below depicts the interactions for the unsuccessful case.

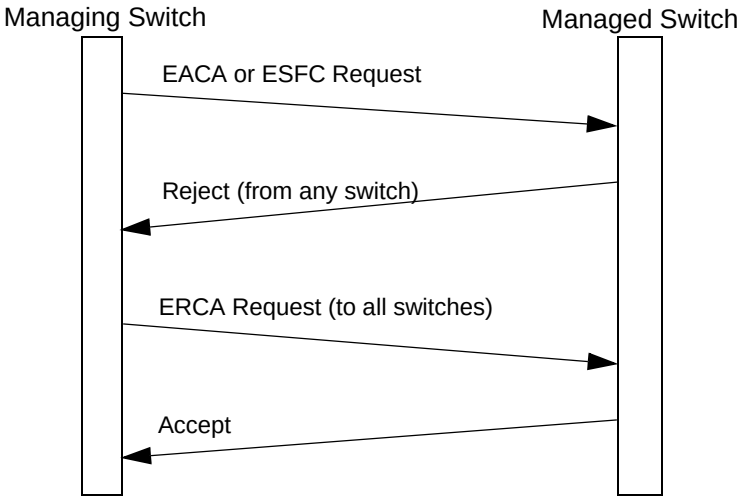

Figure 35 – Unsuccessful Commit Ladder Diagram

13.3.4.3 Transfer Ownership Case - Recovery Processing Enabled

The diagram below depicts the interactions for the transfer ownership case.

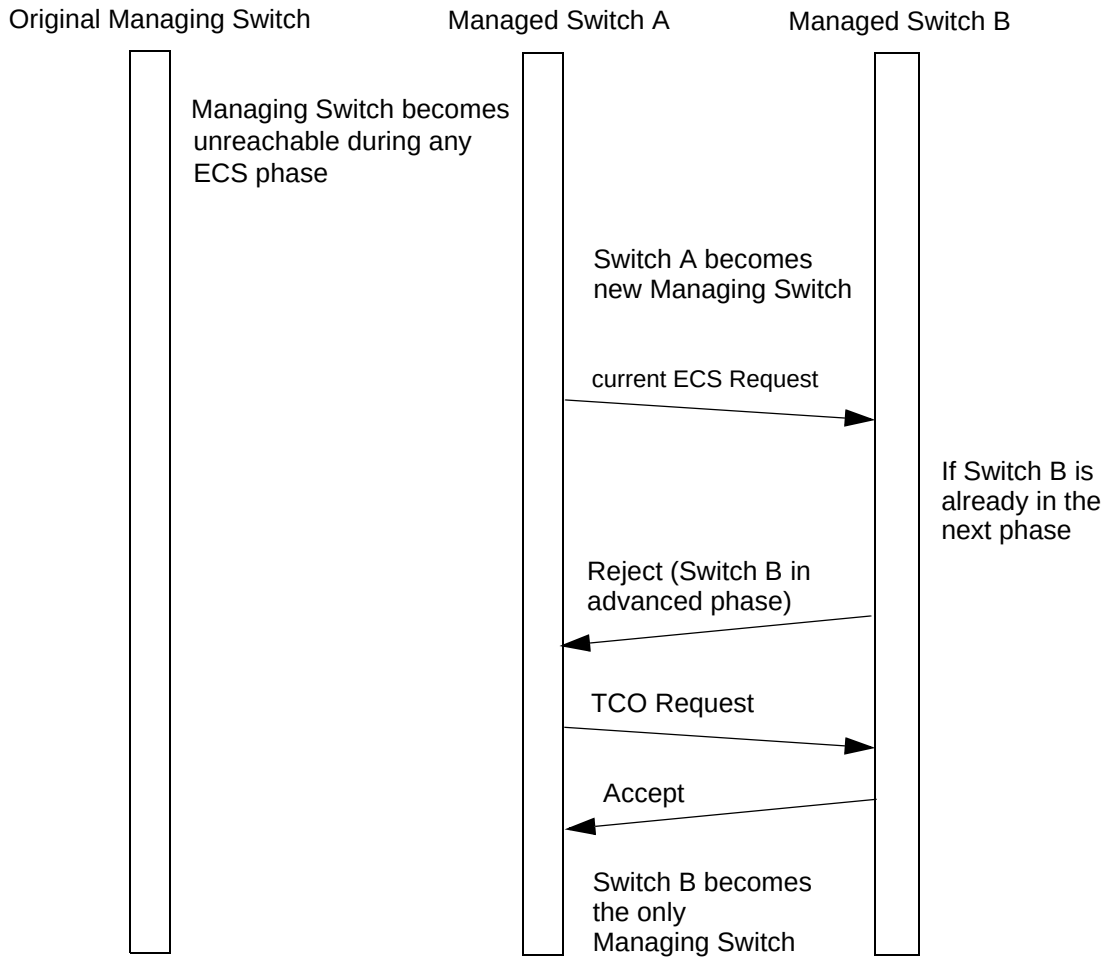

Figure 36 – Transfer Ownership Ladder Diagram

13.3.5 State Machines

13.3.5.1 Overview

State machines are shown for both a managing Switch and a managed Switch. State Machines are in the context of a specific service or application.

Figure 37 shows the state machine for a managing Switch. Figure 38 shows the state machine for a managed Switch. Figure 39 shows the state machine for the Transfer Commit Ownership case.

13.3.5.2 States and Transitions for the Managing Switch

The states and transitions for the managing Switch are shown below:

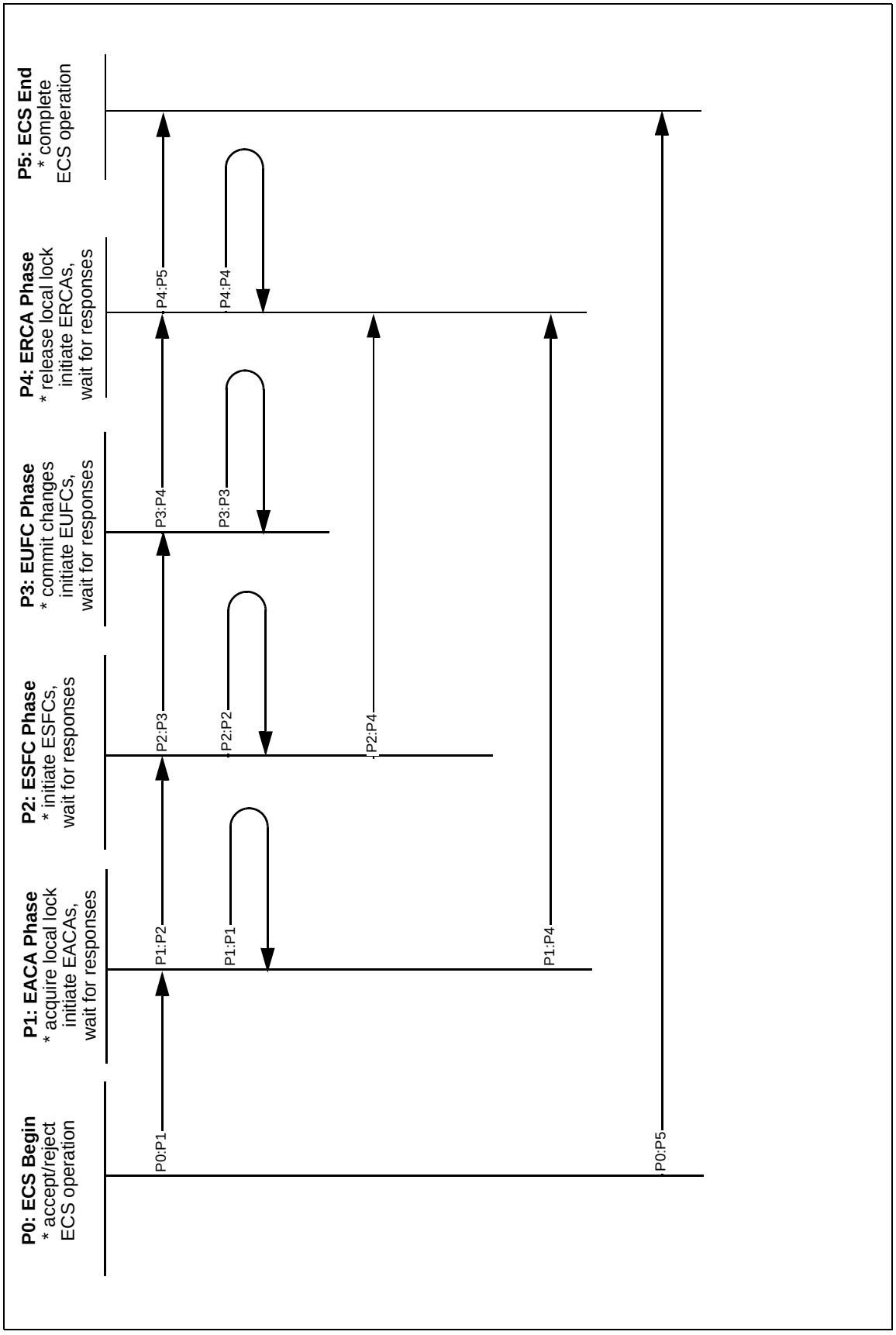

Figure 37 – ECS Managing Switch State Machine

State P0: ECS Begin. This state marks the beginning of an ECS operation due to ECS being invoked. ECS may be invoked due to internal distribution requirements or an externally initiated Fabric Management Session.The managing Switch determines whether the operation can be accepted or not.

Transition P0:P1. The managing Switch is not busy with another ECS operation for the same service or application.

Transition P0:P5. The managing Switch is busy with another ECS operation for the same service or application.

State P1: EACA Phase. The managing Switch initiates EACAs to all Switches participating in the ECS operation and waits for responses.

Transition P1:P1. This transition occurs whenever a response to an EACA is received.

Transition P1:P2. This transition occurs when responses to the EACAs have been received from all Switches and all the responses indicated that the EACAs were accepted.

Transition P1:P4. This transition occurs if any of the responses to the EACAs indicated that the EACA was rejected.

State P2: ESFC Phase. The managing Switch initiates ESFCs to all Switches participating in the ECS operation and waits for responses.

Transition P2:P2. This transition occurs whenever a response to an ESFC is received.

Transition P2:P3. This transition occurs when responses to the ESFCs have been received from all Switches and all the responses indicated that the ESFCs were accepted.

Transition P2:P4. This transition occurs if any of the responses to the ESFCs indicated that the ESFC was rejected.

State P3: EUFC Phase. The managing Switch initiates EUFCs to all Switches participating in the ECS operation and waits for responses.

Transition P3:P3. This transition occurs whenever a response to an EUFC is received.

Transition P3:P4. This transition occurs when responses to the EUFCs have been received from all Switches and there are no other ESFC requests to be performed as part of the ECS operation.

State P4: ERCA Phase. The managing Switch initiates ERCAs to all Switches participating in the ECS operation and waits for responses.

Transition P4:P4. This transition occurs whenever a response to an ERCA is received.

Transition P4:P5. This transition occurs when responses to the ERCAs have been received from all Switches.

State P5: ECS End. This state marks the end of an ECS operation. The managing Switch reports any errors that occurred during the ECS operation to the requesting service or application.

13.3.5.3 States and Transitions for the Managed Switch

The states and transitions for the managed Switch are shown below:

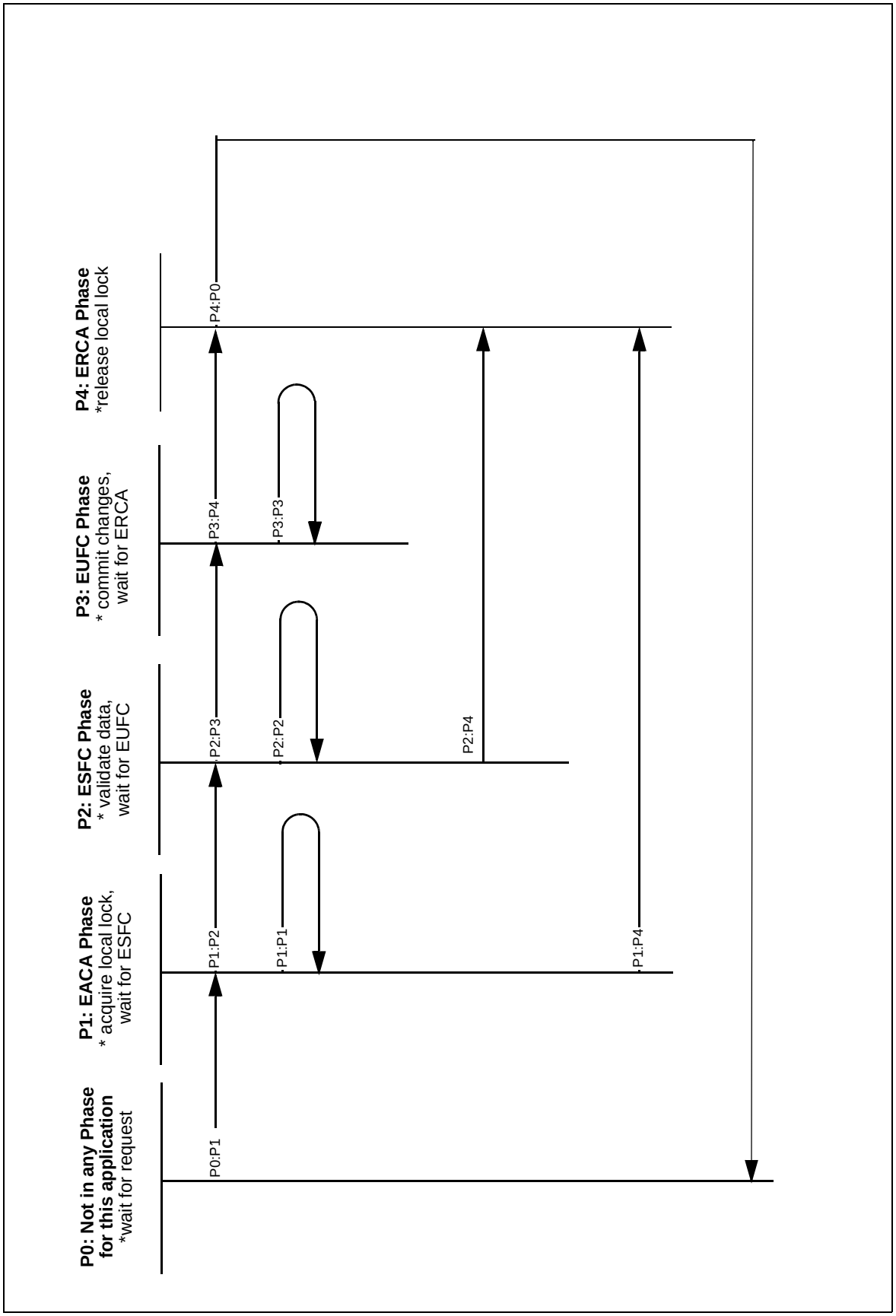

Figure 38 – ECS Managed Switch State Machine

State P0: Not in an ECS Phase. The Switch is not a managed Switch for a given service or application. The Switch is waiting for an ECS request for a given service or application.

Transition P0:P1. This transition occurs whenever an EACA for a given service or application is received. It becomes a managed Switch with respect to the specified transaction.

State P1: EACA Phase. The managed Switch acquires the appropriate lock and waits for the ESFC request.

Transition P1:P1. This transition occurs when an ECS request other than an ESFC or ERCA is received (see table 195).

Table 195 – EACA Phase - Events and Actions

Event	Action
1. EACA Received for same transaction	EACA accepted. If the request is from a new managing Switch, the identity of the new managing Switch is noted.
2. EUFC Received	Reject EUFC - Reason “Invalid phase transition within transaction”
3. EACA Received from a Switch that is not Authorized	Reject EACA - “Switch Not Authorized”

Transition P1:P2. This transition occurs when an ESFC for the given transaction is received and the data is valid.

Transition P1:P4. This transition occurs when an ERCA for the current transaction is received.

State P2: ESFC Phase. The managed Switch validates the received data in the ESFC and waits for the EUFC request.

Transition P2:P2. This transition occurs when an ECS request other than an EUFC or ERCA is received (see table 196).

Table 196 – ESFC Phase - Events and Actions

Event	Action
1. ESFC Received for same transaction	ESFC accepted. If the request is from a new managing Switch, the identity of the new managing Switch is noted.
2. EACA Received for same transaction	EACA Rejected - Reason “In advanced phase”
3. ESFC Received from a Switch that is not Authorized	Reject ESFC - “Switch Not Authorized”

Transition P2:P3. This transition occurs when an EUFC for the given transaction is received.

Transition P2:P4. This transition occurs when an ERCA for the current transaction is received.

State P3: EUFC Phase. The managed Switch commits the application data and waits for the ERCA request.

Transition P3:P3.This transition occurs when an ECS request other than an ERCA is received (see table 197).

Table 197 – EUFC Phase - Events and Actions

Event	Action
1. EUFC Received for same transaction	EUFC accepted. If the request is from a new managing Switch, the identity of the new managing Switch is noted.
2. ESFC Received for same transaction	ESFC Rejected - Reason “In advanced phase”
3. EACA Received for same transaction	EACA Rejected - Reason “In advanced phase”
4. EUFC Received from a Switch that is not authorized.	Reject EUFC - “Switch Not Authorized”

Transition P3:P4. This transition occurs when an ERCA for the current transaction is received.

State P4: ERCA Phase. The managed Switch releases the appropriate lock and concludes the ECS operation. Current transaction ends.

Transition P4: P0. This transition concludes ECS operation processing.

13.3.5.4 States and Transitions for Transfer Commit Ownership

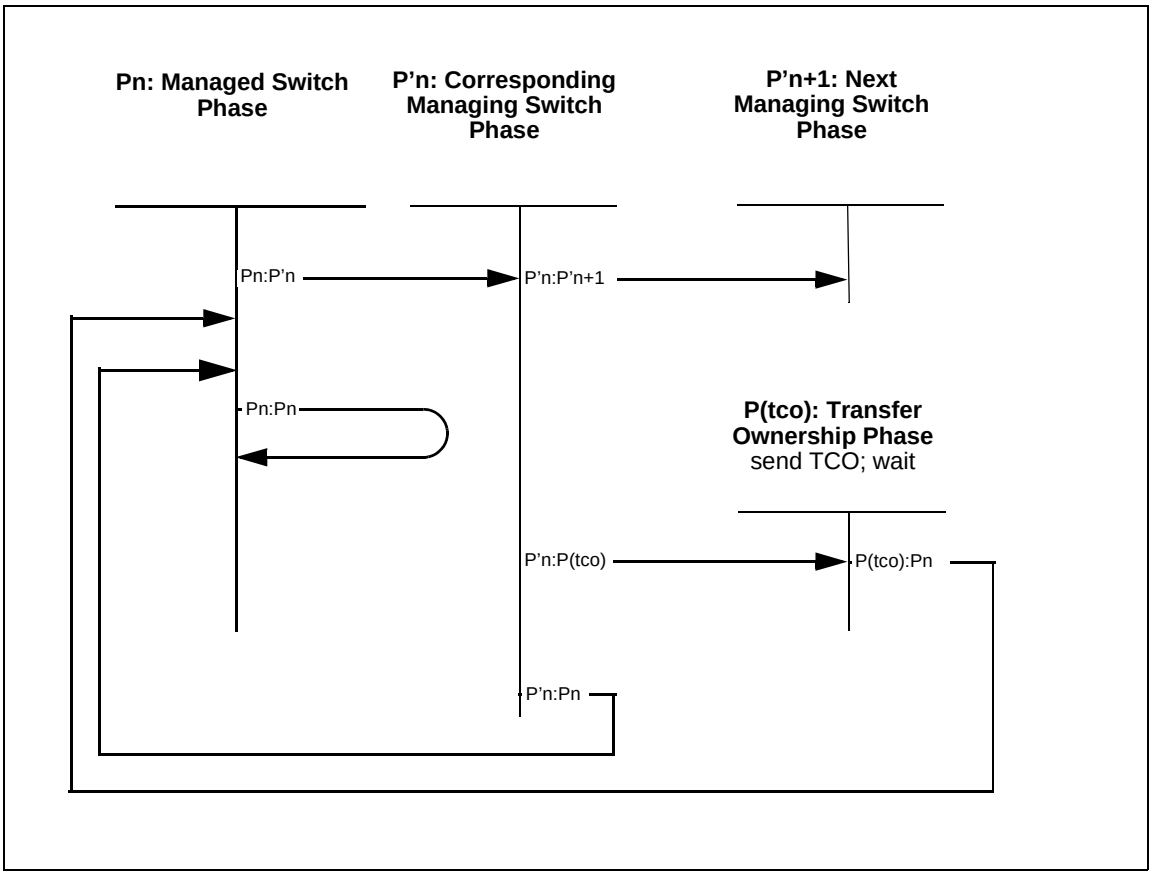

Figure 39 – ECS Transfer Commit Ownership (TCO) State Machine

All managed Switches currently involved in the ECS operation monitor for domain unreachable for the managing Switch. If domain unreachable is detected for the managing Switch then the detecting managed Switch starts a deadman timer with the appropriate value based on its own position in the ECS operation's Switch List. Managed Switches keep track of the current managing Switch by examining the source of ECS requests.

State Pn: Managed Switch Phase. This is the current state of the managed Switch. This can be P1: EACA Phase, P2: ESFC Phase, or P3: EUFC Phase.

Transition Pn:P'n. This transition occurs whenever the deadman timer expires, or a TCO is received due to an "In advanced phase" response to an ECS request. The managed Switch assumes the role of managing Switch and transitions to the corresponding managing Switch phase.

Transition Pn:Pn. This transition occurs whenever an ECS request that is older than the current request is received. Reject reason "In advanced phase" is returned.

State P'n: Corresponding Managing Switch Phase. This is the managing Switch state that corresponds to the managed Switch state that the new managing Switch was in. The new managing Switch initiates ECS requests corresponding to that phase and waits for responses.

Transition P'n:P'n+1. This is the normal transition that occurs when responses to the ECS requests have been received from all Switches and all the responses indicated that the ECS requests were accepted.

Transition P'n:Pn. This transition occurs if an ECS request for this phase is received from a switch with a lower domain ID. The managing Switch goes back to its role as managed Switch.

Transition P'n:P(tco). This transition occurs if any of the responses to the ECS requests indicated "In advanced phase."

State P'n+1: Next Managing Switch Phase. This is the next state that the managing Switch transitions to if all ECS requests are accepted.

State P(tco): Transfer Commit Ownership Phase. The managing Switch initiates TCO to Switch that returned "In advanced phase" to the ECS operation and waits for response.

Transition P(tco):Pn. This transition occurs if the TCO was accepted by the managed Switch that indicated it was "In advanced phase." The managing Switch goes back to its role as managed Switch.

14 Virtual Channels for Switched Fabric

14.1 Overview

The Virtual Channel (VC) Architecture enables different data flows to be identified that in turn allows them to be differentiated and subjected to different service policies. Virtual Channels are allocated to group traffic based on destination, control traffic, and intra-fabric traffic. Virtual Channels operate on individual ISLs and are not required to operate across the entire fabric.

14.2 Assignment of Virtual Channels

14.2.1 Overview of Assignment

Frames are assigned to Virtual Channels based on the assignment scheme and the number of virtual channels for that assignment scheme as determined during link initialization. The assignment schemes are described below.

14.2.2 Simple

In the Simple assignment scheme, frames are assigned to Virtual Channels as depicted in table 198. Class F frames shall always be assigned to VC 0 and all other classes of frames shall be assigned to VC 1.

Table 198 – Simple Assignment Scheme

Virtual Channel Number	Description
0	Class F Frames
1	All Other Frames

14.2.3 Fixed

In the Fixed assignment scheme, Class F frames shall be assigned to VC 0 and Broadcast frames shall be assigned to the highest VC as determined during link initialization. Frames other than Class F and Broadcast shall be assigned to Virtual Channels based on D_ID as depicted in table 199.

Table 199 – Fixed- Assignment Scheme

Virtual Channel Number	Description
0	Class F Frames
1	Reserved
2 - (n-3)	Based on D_ID (see table 200)
n-2	Reserved
n-1	Broadcast frames

Table 200 – VC Assignments - Fixed

Number of VCs (n)	VC mapped to D_ID bits
8	D_ID(9:8)
12	D_ID(10:8)
20	D_ID(11:8)
36	D_ID(12:8)
68	D_ID(13:8)
132	D_ID(14:8)

If Virtual Channels are supported, implementation of the Fixed Assignment scheme shall be mandatory when eight or greater Virtual Channels are available.

14.2.4 Variable

In the Variable assignment scheme, frames are allocated to VCs according to the D_ID. In this scheme Class F and Broadcast frames are not allocated to dedicated Virtual Channels. Class F and Broadcast frames are transported over Virtual Channels according to D_ID and are given priority over other frames within each Virtual Channel. The assignment is shown in table 201 below.

Table 201 – Variable Assignment Scheme

Virtual Channel Number	Description (see table 202)
0	Based on D_ID
1	Based on D_ID
...	
n-2	Based on D_ID
n-1	Based on D_ID

Table 202 – VC Assignments - Variable

Number of VCs (n)	VC mapped to D_ID bits
4	D_ID(9:8)
8	D_ID(10:8)
16	D_ID(11:8)
32	D_ID(12:8)
64	D_ID(13:8)
128	D_ID(14:8)
256	D_ID(15:8)

If Virtual Channels are supported and less than eight Virtual Channels are available, implementation of the variable assignment scheme shall be mandatory.

14.3 VC Parameter Negotiation

14.3.1 Agreement of Assignment Schemes

If VC_RDY flow control mode is specified, then the originator of the ELP and the recipient of the ELP shall agree on the assignment scheme. If the recipient of the ELP does not support the assignment scheme specified in the ELP by the originator, then the recipient rejects the ELP request with a Reason Code Explanation of “Invalid Flow Control Parameters”. The originator of the ELP may then request another assignment scheme defined for VC_RDY flow control mode. In the event that an assignment scheme cannot be agreed upon, the originator shall request that R_RDY flow control mode be used.

14.3.2 Negotiation of Number of VCs

Once the assignment scheme has been agreed upon, the recipient checks the number of VCs specified by the VC value. If the recipient of the ELP supports a VC value less than that specified by the originator, it shall transmit the lower value that it supports in the SW_ACC to the ELP. The originator will then use this value representing the number of VCs and their assignment for operation. The originator may use the VC_Credit assigned to the valid VCs in the original ELP, or it may originate a new ELP with the reduced number of VCs and a different VC_Credit distribution.

The recipient of the ELP shall not send an SW_ACC indicating a higher VC Value than what was advertised in the originator’s ELP.

In the event that the originator and responder cannot agree upon the number of VCs, the originator shall request that R_RDY flow control be used.

14.4 Credit Management

14.4.1 Overview

Each VC maintains a set of VC Buffer credits. The credit for each VC is set to the value exchanged in the ELP SW_ILS at the completion of Link Initialization for the ISL. The VC Credit shall be decrement-

ed for every frame transmitted that is associated with that VC. VC Credit shall be incremented for each VC_RDY received for that VC. Frames shall not be transmitted for a VC for which there is no VC Credit even if there is Buffer-to-buffer Credit. Frames shall not be transmitted on any VC if there is not Buffer-to-buffer Credit for that ISL. If a frame is received on a Virtual Channel with no buffer-to-buffer Credit, the frame shall be discarded and no VC_RDY shall be returned.

After negotiating VC_RDYs in ELP and following the link reset, the VC_RDYs shall be sent and R_RDYs shall not be sent.

14.4.2 VC_RDY Primitive Signals

VC_RDY primitive signals are used for buffer-to-buffer flow control on ISLs that support Virtual Channels. The general format for the VC_RDY Primitive Signal is the same as defined for Class 4 use in FC-FS. The general format is shown in the figure below.

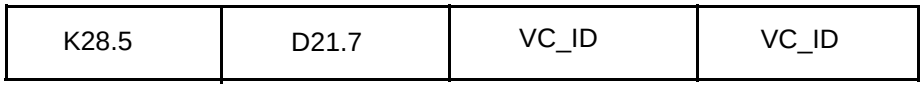

Figure 40 – VC_RDY Primitive Signal Format

Table 203 provides the VC_ID values used in the VC_RDY Primitive Signals for each virtual channel number.

Table 203 – VC_ID Values for VC_RDY Primitive Signals

Virtual Channel Number	VC_ID Value
00h	D0.0
01h	D1.0
02h	D2.0
...	see FC-FS-3
FDh	D29.7
FEh	D30.7
FFh	D31.7

15 Inter-Fabric Routing Support

15.1 F_RJT and F_BSY processing for Class 2/F

15.1.1 Overview

Switches may support either or both of:

a) Inter-Fabric Routers operating according to the FC-IFR simple mode of operation (see FC-IFR), or

b) Inter-Fabric Routers operating according to the FC-IFR NAT mode of operation (see FC-IFR).

When a Switch supporting Inter-Fabric Routers operating according to the FC-IFR simple mode of operation (see FC-IFR) determines a Class 2/F F_RJT or Class 2/F F_BSY needs to be generated in response to a received frame, the Switch shall:

a) generate an encapsulated Class 2/F F_RJT or Class 2/F F_BSY as specified in 15.1.2 if the received frame contains an Enc_Header; or

b) generate a Class 2/F F_RJT or Class 2/F F_BSY as specified in FC-FS-3 if the received frame does not contain an Enc_Header.

When a Switch not supporting Inter-Fabric Routers operating according to the FC-IFR simple mode of operation (see FC-IFR) determines a Class 2/F F_RJT or Class 2/F F_BSY needs to be generated in response to a received frame, the Switch shall generate a Class 2/F F_RJT or Class 2/F F_BSY as specified in FC-FS-3.

15.1.2 Encapsulated Class 2 F_RJT or Class 2 F_BSY frame format

15.1.2.1 Overview

An encapsulated Class 2/F F_RJT or Class 2/F F_BSY frame shall be formatted as specified in figure 41.

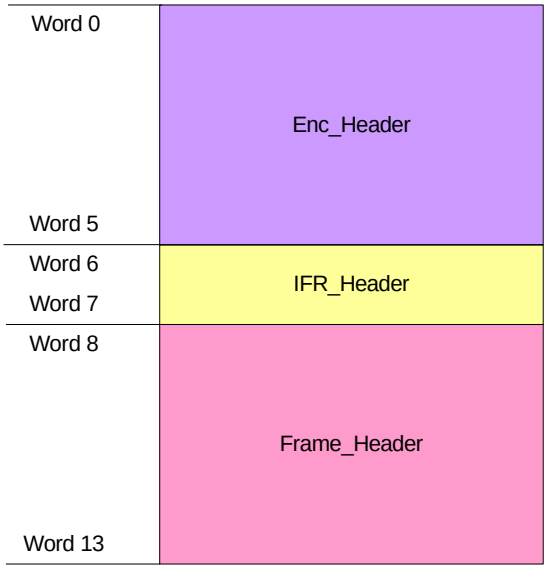

Figure 41 – Encapsulated Class 2/F F_RJT and Class 2/F F_BSY frame format

15.1.2.2 Encapsulated Enc_Header field values

The Enc_Header field values for an encapsulated Class 2/F F_RJT or Class 2/F F_BSY frame shall be set as specified in table 204.

Table 204 – Encapsulated Class 2/F F_RJT/F_BSY Enc_Header field values

Field	Value
R_CTL	52h
D_ID	S_ID field in received Enc_Header
CS_CTL	CS_CTL field in received Enc_Header
S_ID	D_ID field in received Enc_Header
TYPE	TYPE field from Frame_Header (see 15.1.2.4)
F_CTL	F_CTL field from Frame_Header (see 15.1.2.4)
SEQ_ID	SEQ_ID field in received Enc_Header
DF_CTL	DF_CTL field from Frame_Header (see 15.1.2.4)
SEQ_CNT	SEQ_CNT field in received Enc_Header
OX_ID	OX_ID field in received Enc_Header
RX_ID	RX_ID field in received Enc_Header
Parameter	Parameter field from Frame_Header (see 15.1.2.4)

15.1.2.3 Encapsulated IFR_Header field values

The IFR_Header field values for an encapsulated Class 2/F F_RJT or Class 2/F F_BSY frame shall be set as specified in table 205.

Table 205 – Encapsulated Class 2/F F_RJT/F_BSY IFR_Header field values

Field	Value
R_CTL	51h
DF_ID	SF_ID field in received IFR_Header
Exp_Time	a) zero; or b) valid expiration time if port is IFR capable
SF_ID	DF_ID field in received IFR_Header
Ver	Ver field in received IFR_Header
Pri	Pri field in received IFR_Header
ETV	a) zero; or b) one if valid expiration time
HCV	a) zero; or b) one if valid Hop_Cnt
Hop_Cnt	a) zero; or b) valid Hop_Cnt if port is IFR capable

15.1.2.4 Encapsulated Frame_Header field values

The Frame_Header field values for an encapsulated Class 2/F F_RJT or Class 2/F F_BSY frame shall be set as specified in FC-FS-3 and this standard.

16 Timers and Constants

16.1 General Timers and Constants

General timers and constants referenced in FC-SW-6 are summarized in table 206.

Table 206 – Timers and Constants for FC-SW-6

Timer/Constant	Value	Description
F_S_TOV	5 Seconds	Fabric Stability timeout value. Ensures that Fabric stability has been achieved during various aspects of Fabric Configuration
D_S_TOV	5 Seconds	Distributed Service timeout value. A value that indicates the maximum time that a Distributed Service requestor shall wait for a response.
Min_LS_Arrival	1 seconds	The minimum amount of time that shall pass before a Switch shall accept updates of any given LSR via flooding.
Min_LS_Interval	5 seconds	The minimum amount of time that shall pass before a Switch is allowed to send an LSR update via flooding.
Check_Age	5 minutes	The minimum amount of time between verification checks of LSRs in a Switch's database.
Max_Age_Diff	15 minutes	If the age of two instances of the same LSR differ by more than this amount, they are considered to come from different incarnations. The LSR with the smaller age field is considered to be more current.
LS_Refresh_Time	30 minutes	The maximum interval between transmission of refresh LSRs.
Max_Age	1 hour	The maximum age that an LSR may reach. When an LSR reaches this age, it is removed from the database.
Initial_Message_Number	80000001h	The initial value for the LSR Incarnation field.
Max_Message_Number	7FFFFFFFh	The maximum value for the LSR Incarnation field.
Rxmt_Interval	5 seconds	The maximum time period for which an LSR may go unacknowledged. If an LSR is not acknowledged within this time period, it shall be retransmitted.
Hello_Interval	20 seconds	The minimum interval between HELLOs sent by a Switch on a link to verify link health.
Dead_Interval	80 seconds	The maximum interval for which no HELLO may be received on a link. If no HELLO is received on a link after this time period the link is considered broken and thus removed from the database.

16.2 SW_ILS Time-Out Values

SW_ILS time-out values and the recommended actions for them are provided in table 207.

Table 207 – SW_ILS Time-Out Values

SW_ILS	Timeout (SW_ACC or SW_RJT)	Recommended Action when Timeout Expires
ELP	E_D_TOV + 4 secs	Go to state P11
ESC	E_D_TOV + 4 secs	Go to state P11
BF	F_S_TOV + E_D_TOV	Restart BF
RCF	F_S_TOV + E_D_TOV	Restart RCF
EFP	2*F_S_TOV	Restart BF or Retransmit
DIA	F_S_TOV	Restart BF or Retransmit
RDI	R_A_TOV	Retransmit
MR	R_A_TOV + 70 secs	Retransmit
MRRA	R_A_TOV	Retransmit
ACA	R_A_TOV	Retransmit
RCA	R_A_TOV	Retransmit
SFC	R_A_TOV + 20 secs	Retransmit
UFC	R_A_TOV + 20 secs	Retransmit
EACA	R_A_TOV	Retransmit
ERCA	R_A_TOV	Retransmit
ESFC	R_A_TOV + 20 secs	Retransmit
EUFC	R_A_TOV + 20 secs	Retransmit
TCO	R_A_TOV	Retransmit
SW_RSCN	R_A_TOV	Retransmit
DRLIR	R_A_TOV	Retransmit
ESS	R_A_TOV	Retransmit
CEC	R_A_TOV	Retransmit
EVFP	2*R_A_TOV	Retransmit
STR	R_A_TOV	Retransmit

17 Distributed Switch Environment

17.1 Overview

A Distributed Switch is a set of FCDFs associated with at least one Controlling Switch, that controls the operations of the set of FCDFs. Figure 42 shows an example of Distributed Switch composed of a Controlling Switch and two FCDFs.

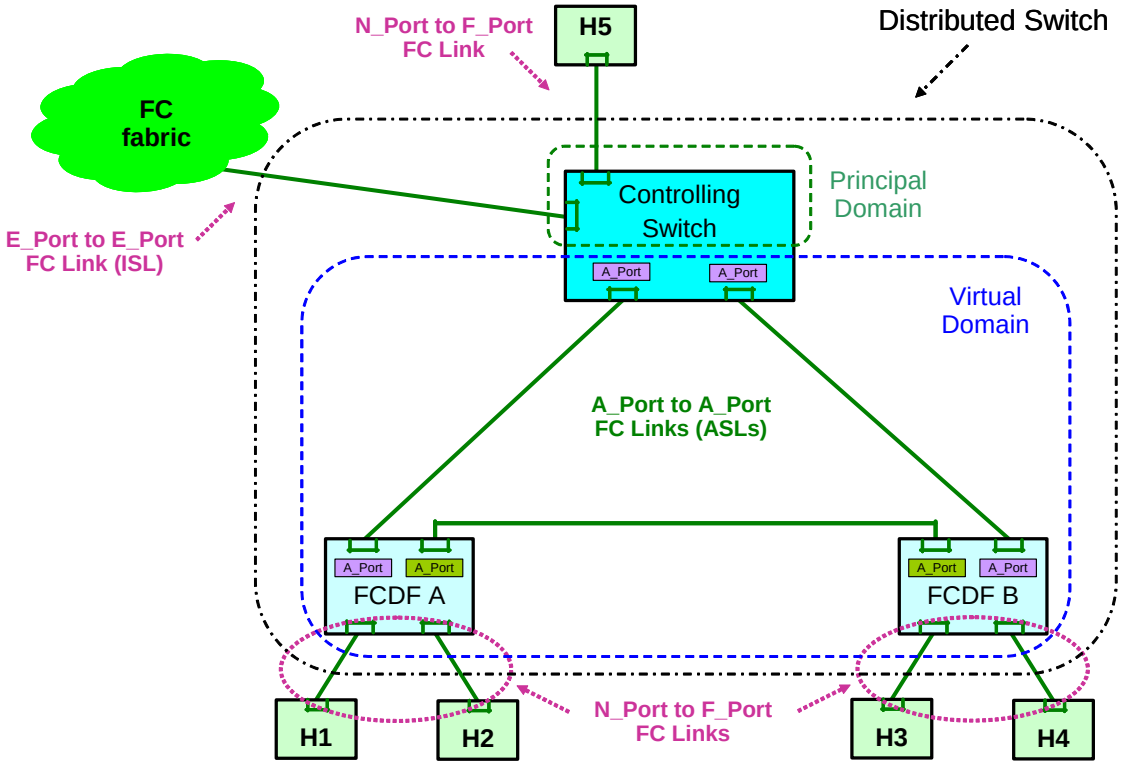

Figure 42 – Example of Distributed Switch

From an external point of view (i.e., outside the dotted and dashed black line in figure 42), a Distributed Switch behaves as a Fibre Channel Switch. In particular, a Distributed Switch supports the instantiation of N_Port to F_Port links and of E_Port to E_Port links (ISLs). N_Port to F_Port links are supported by both FCDFs and Controlling Switches, while ISLs are supported only by Controlling Switches. This means that it is possible to connect a Distributed Switch to another Switch only through a Controlling Switch, not through an FCDF.

From an internal point of view (i.e., inside the dotted and dashed black line in figure 42), A_Port to A_Port links (ASLs) enable FC frames forwarding between Controlling Switch and FCDFs, as well as between FCDFs. ASLs are also used to exchange control information between Controlling Switch and FCDFs.

The Controlling Switch uses one or more Virtual Domain_IDs to perform N_Port_ID allocations for N_Ports connected to the FCDF Set of the Distributed Switch (i.e., a Virtual Domain_ID is used as the most significant byte in the N_Port_IDs allocated to N_Ports that are attached to the FCDF Set). The Controlling Switch uses also another Domain_ID, called Principal Domain, for its normal functions as a Fibre Channel Switch. As a result, a Distributed Switch such as the one shown in figure 42 uses at least two Domain_IDs: one for the Principal Domain and one or more for the Virtual Domain.

To properly support the operations of a Virtual Domain, a Controlling Switch shall have at least one Switch_Name to associate with the Virtual Domain, in addition to its own Switch_Name.

FCDFs are not able to operate properly without a Controlling Switch, therefore the Controlling Switch is a single point of failure in a Distributed Switch configuration with only one Controlling Switch, as the one shown in figure 42. To avoid this issue, Distributed Switches may support a redundant configuration of two Controlling Switches, a Primary one and a Secondary one. The Secondary Controlling Switch keeps its state synchronized with the Primary and is able to take its place in case of failure according to the Controlling Switch Redundancy Protocol.

Figure 43 shows an example of Distributed Switch including a redundant pair of Controlling Switches.

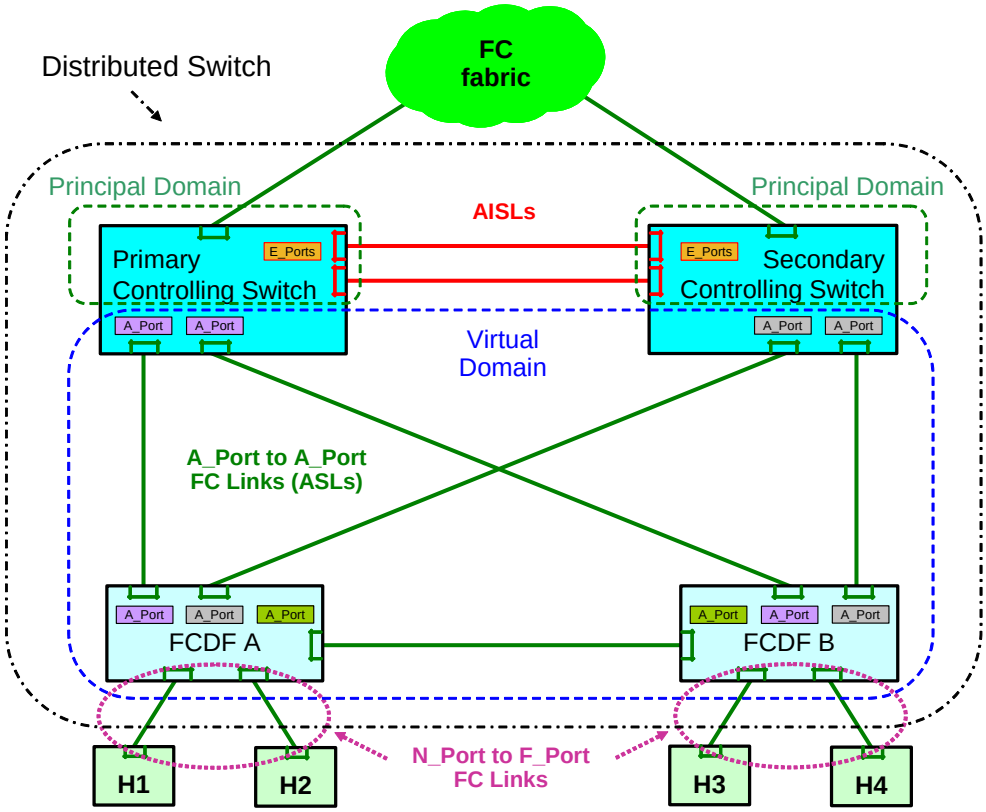

Figure 43 – Example of Redundant Distributed Switch

The two Controlling Switches in a redundant Distributed Switch instantiate at least two Augmented ISLs (AISLs) between themselves, where the term ‘augmented’ indicates that link is used also for the Redundancy protocol, in addition to normal E_Port operation.

The Controlling Switches use one or more Virtual Domain_IDs to perform N_Port_ID allocations for N_Ports connected to the FCDF Set of the Distributed Switch (i.e., a Virtual Domain_ID is used as the most significant byte in the N_Port_IDs allocated to N_Ports that are attached to the FCDF Set). Using Virtual Domain_IDs to assign N_Port_IDs enables seamless operation in case of failures of one of the two redundant Controlling Switches. Each Controlling Switch uses also another Domain_ID, called Principal Domain, for its normal functions as a Fibre Channel Switch. As a result, a redundant Distributed Switch typically uses three or more Domain_IDs: one for each Principal Domain and one or more for the Virtual Domain. To properly support the operations of a Virtual Domain, a Controlling Switch shall have at least a Switch_Name to associate with the Virtual Domain, in addition to its own Switch_Name.

The two redundant Controlling Switches instantiate ASLs to enable the forwarding of FC frames and the communication of control information between Controlling Switches and FCDFs. In a redundant configuration, FCDFs instantiate ASLs to each of the Controlling Switches and between themselves.

A Distributed Switch may have a cascaded FCDF configuration. Figure 44 shows an example of such a configuration.

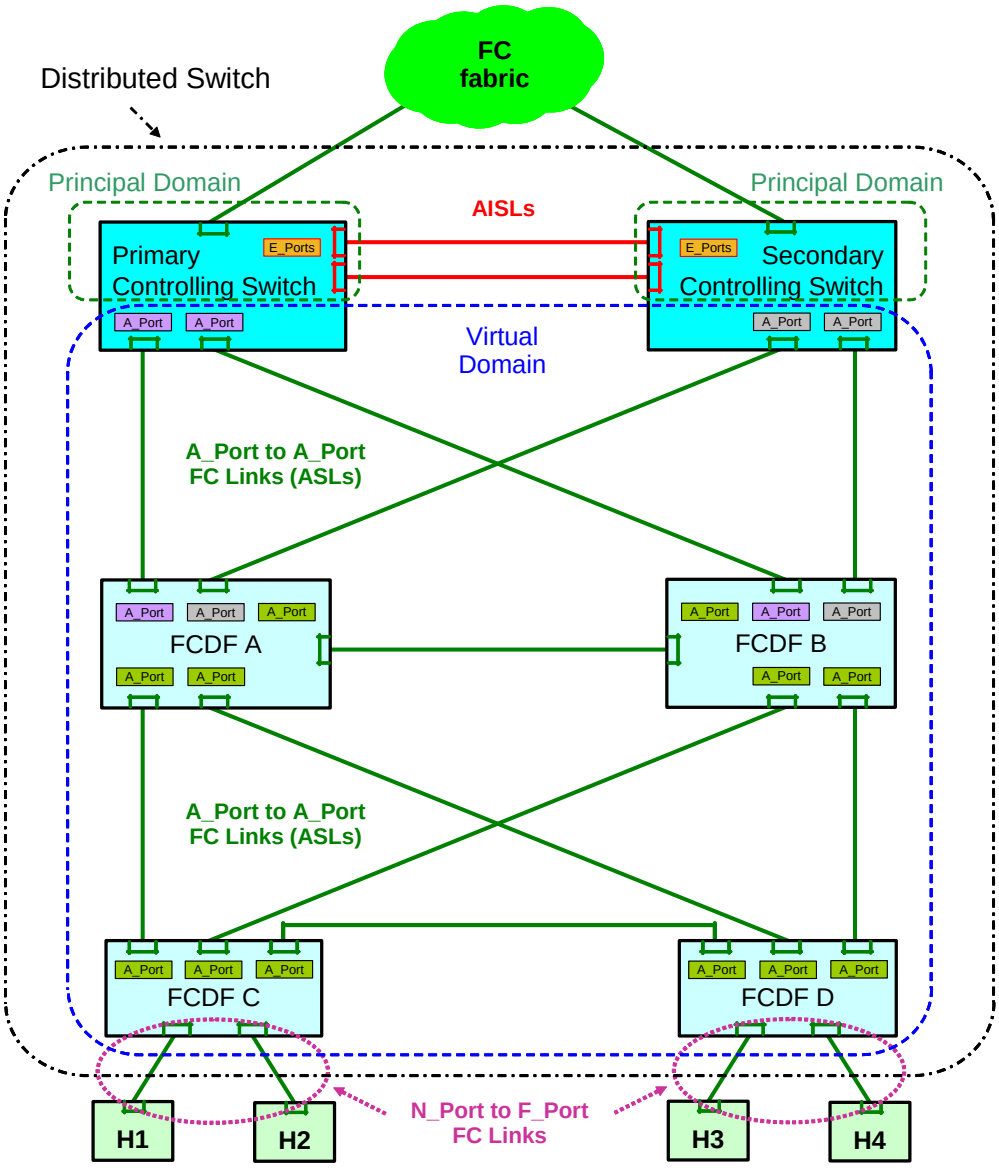

Figure 44 – Example of Distributed Switch with Cascaded FCDFs

A Controlling Switch is uniquely identified by its Switch_Name Name_Identifier, as an FC Switch. An FCDF is uniquely identified by its Switch_Name Name_Identifier. A Distributed Switch is defined by an administrative configuration on the Controlling Switches, listing:

a) the Switch_Names of the two Controlling Switches that act as the Primary/Secondary pair for that Distributed Switch (i.e., the Controlling Switch Set); and

b) the Switch_Names of the FCDFs that are part of that Distributed Switch (i.e., the FCDF Set).

17.2 Controlling Switch Functional Model

Figure 45 shows the functional model of a Controlling Switch.

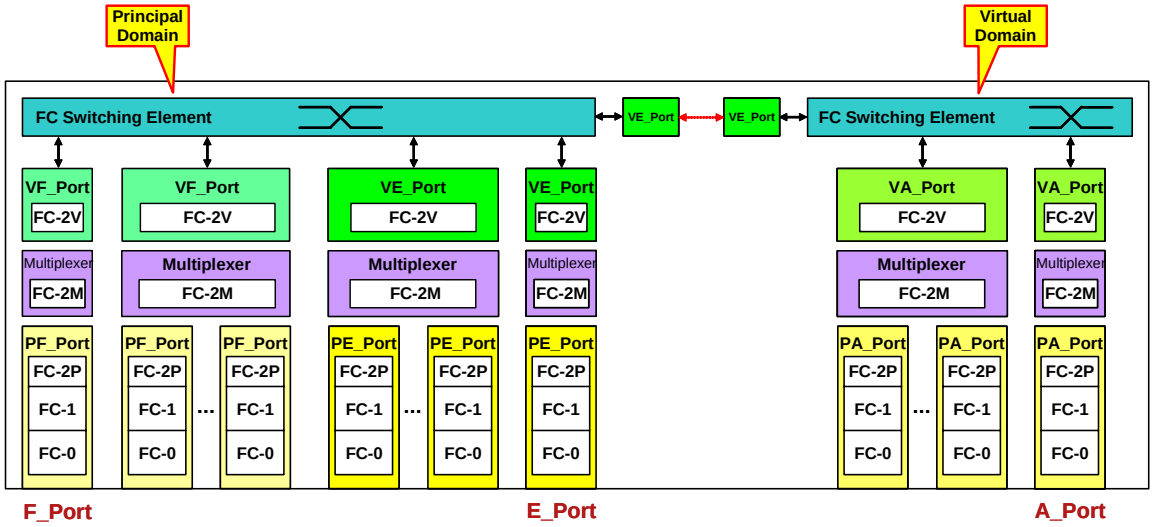

Figure 45 – Controlling Switch Functional Model

A Controlling Switch is an FC Switch that supports the instantiation of VA_Ports, in addition to VF_Ports and VE_Ports. As any FC Switch, a Controlling Switch is able to aggregate its physical ports in sets that behave as virtual ports, providing higher bandwidth than the one available to a single physical port.

For a Controlling Switch, a physical port is an LCF (see FC-FS-3), that may behave as a Physical F_Port (PF_Port), as a Physical E_Port (PE_Port), or as a Physical A_Port (PA_Port). A virtual port is an instance of the FC-2V sublevel of Fibre Channel (see FC-FS-3), that may behave as a Virtual F_Port (VF_Port), as a Virtual E_Port (VE_Port), or as a Virtual A_Port (VA_Port).

As shown in figure 45, a Controlling Switch is functionally modeled as having two FC Switching Elements, one for the Principal Domain and one for the Virtual Domain, connected by an internal VE_Port to VE_Port link. The Switching Element associated with the Principal Domain supports the intantiation of VF_Ports and VE_Ports, the Switching Element associated with the Virtual Domain supports the intantiation of VA_Ports.

17.3 FCDF Functional Model

Figure 46 shows the functional model of an FCDF.

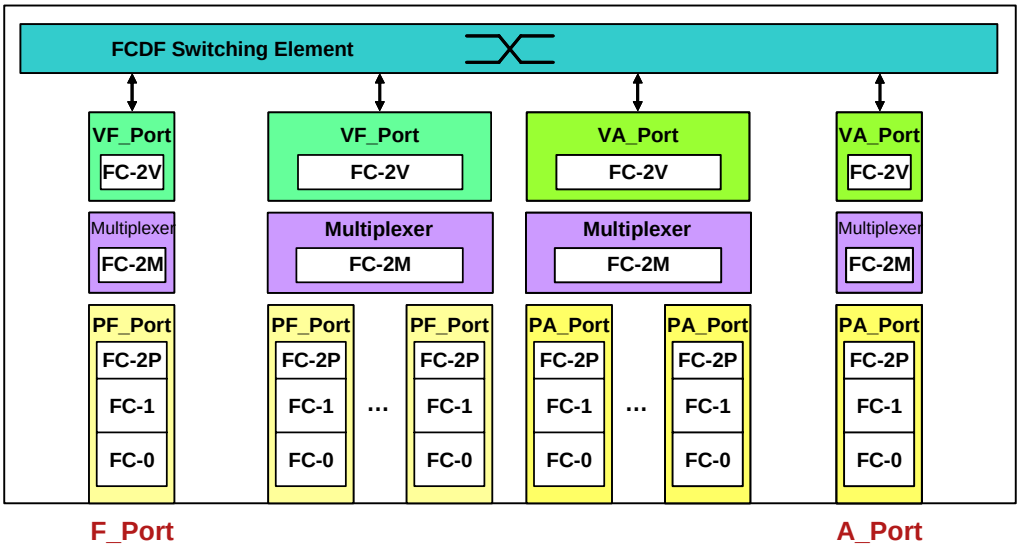

Figure 46 – FCDF Functional Model

An FCDF is a simplified FC switching entity that forwards FC frames among VA_Ports and VF_Ports through a FCDF Switching Element. As any FC Switch, an FCDF is able to aggregate its physical ports in sets that behave as virtual ports, providing higher bandwidth than the one available to a single physical port. An FCDF shall support at least one VA_Port operating together with a PA_Port (i.e., an A_Port) and may support one or more F_Ports.

For an FCDF, a physical port is an LCF (see FC-FS-3), that may behave as a Physical F_Port (PF_Port) or as a Physical A_Port (PA_Port). A virtual port is an instance of the FC-2V sublevel of Fibre Channel (see FC-FS-3), that may behave as a Virtual F_Port (VF_Port) or as a Virtual A_Port (VA_Port).

Figure 47 shows the model of the FCDF Switching Element, composed by a Switch Construct, a Routing Table Update function, and a FCDF Controller function.

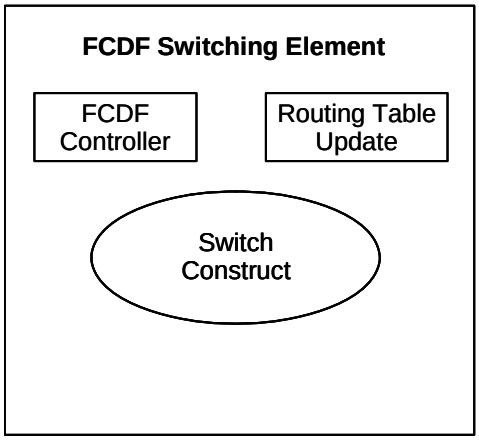

Figure 47 – FCDF Switching Element

The Switch Construct is the entity performing FC frames forwarding based on the FC frame's D_ID field according to a routing table. The structure of the Switch Construct is undefined and beyond the scope of this document.

The Routing Table Update is a logical entity that updates the Switch Construct's routing table through the VA_Port protocols.

The FCDF Controller is a logical entity that performs the management of the FCDF through the VA_Port protocols. The FCDF Controller has the characteristics of a VN_Port.

17.4 FCDF Handling of Well Known Addresses

N_Ports use Well Known Addresses (WKAs) and Domain Controller address identifiers to exchange information with the Fabric, either through ELSs or through the Common Transport protocol.

An FCDF supports VF_Ports, therefore it shall terminate FC frames destined to the F_Port Controller WKA. This implies local processing by the FCDF of the FLOGI, FDISC, LOGO, and RLS ELSs.

The handling of other WKAs and Domain Controllers address identifiers is performed by the Primary Controlling Switch, therefore an FCDF shall forward all FC frames having as D_ID the address iden-

tifiers listed in table 208 to the Primary Controlling Switch through a VA_Port. The NPRD SW_ILS provides to FCDFs the routing information needed to reach the Primary Controlling Switch.

Table 208 – Forwarded Domain Controller and Well Known Address Identifiers

Address Value	Description
FFFC01h .. FFFCFEh	Domain Controller Address Identifiers
FFFFF4h	Event Service WKA
FFFFF6h	Clock Synchronization Service WKA
FFFFF7h	Security Key Distribution Service WKA
FFFFFAh	Management Service WKA
FFFFFBh	Time Service WKA
FFFFFCh	Directory Service WKA
FFFFFDh	Fabric Controller WKA

The AISLs used for the redundancy protocol between the Primary and Secondary Controlling Switch are used as paths to reach the Primary Controlling Switch when an FCDF is connected to the Secondary Controlling Switch but not anymore to the Primary one. In order to do so, the Secondary Controlling Switch shall forward to the Primary Controlling Switch over the AISLs:

a) any FC frame destined to the address identifier FFFFF9h (i.e., the VA_Port Controller); and

b) any FC frame destined to the address identifiers shown in table 208 when they are received from a VA_Port.

17.5 A_Port Operation

An A_Port is the combination of one PA_Port and one VA_Port operating together. A PA_Port is the LCF within the Fabric that attaches to another PA_Port through a link. A VA_Port is an instance of the FC-2V sublevel of Fibre Channel that connects to another VA_Port. A VA_Port is uniquely identified by an A_Port_Name Name_Identifier and is addressable by the VA_Port connected to it through the A_Port Controller address identifier (i.e., FFFFF9h).

An A_Port is the point at which a Controlling Switch is connected to an FCDF to create a Distributed Switch. Also, an A_Port is the point at which an FCDF is connected to another FCDF. It normally functions as a conduit among FCDFs and between FCDFs and Controlling Switches for frames destined for remote N_Ports and NL_Ports. An A_Port is also used to carry frames between Controlling Switch and FCDFs for purposes of configuring and maintaining the Distributed Switch.

An A_Port shall support the Class F service. An A_Port shall also be capable of forwarding one or more of the following classes of service: Class 2 service, Class 3 service. An A_Port shall not admit to its FCDF or Controlling Switch any Primitive Sequences, or any Primitive Signals other than Idle, that the A_Port receives on its inbound fibre.

The model of an A_Port on an FC-FS-3 Transport is shown in figure 48.

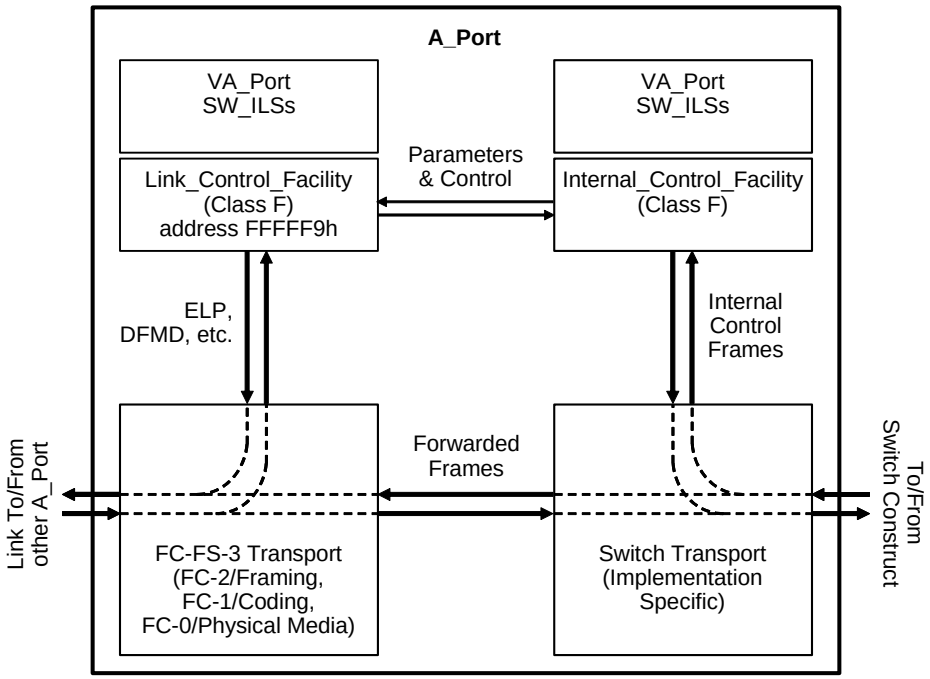

Figure 48 – A_Port Model

An A_Port contains an FC-FS-3 Transport element through which all frames are passed, and Primitives are transferred across the Link to and from the other A_Port. Frames received from the other A_Port are either directed to the Switch Construct via the Switch Transport element, or directed to the Link_Control_Facility. The Link_Control_Facility receives frames related to the VA_Port SW_ILSs, and transmits responses to those frames.

Frames received from the FC-FS-3 Transport element that are destined for other ports are directed by the Switch Transport to the Switch Construct for further forwarding. Frames received from the Switch Construct by the Switch Transport are directed either to the FC-FS-3 Transport for transmission to the other A_Port, or to the Internal_Control_Facility. The Internal_Control_Facility receives frames related to VA_Port SW_ILSs, and transmits responses to those frames.

Information is passed between the Internal_Control_Facility and the Link_Control_Facility to effect the control and configuration of the Transport elements.

17.6 A_Port to A_Port Links (ASLs)

An ASL becomes operational on successful completion of an ELP Exchange between a Controlling Switch and a FCDF or between two FCDFs. Two additional bits in the flags field of the ELP payload indicate if the originator of the ELP Request or SW_ACC is a Controlling Switch or an FCDF.

Bits 13 and 12 in the flags field of the ELP payload indicate if the originator of the ELP Request or SW_ACC is a Controlling Switch or an FCDF.Bits 13 and 12 in the flags field of the ELP payload indicate if the originator of the ELP Request or SW_ACC is a Controlling Switch or an FCDF (see 6.1.4).

A received ELP Request having both these bits set to one is invalid, shall be rejected, and the link shall be Isolated. A received SW_ACC having both these bits set to one is invalid and the link shall be Isolated. Table 209 shows the meaning of the values of these bits.

Table 209 – VA_Port ELP Flags

Bit 13 value	Bit 12 value	Description
0b	0b	The originator of the ELP Request or SW_ACC is a normal FC Switch or FCF
0b	1b	The originator of the ELP Request or SW_ACC is an FCDF or an FDF
1b	0b	The originator of the ELP Request or SW_ACC is a Controlling Switch or a Controlling FCF
1b	1b	Invalid combination

A port of a Controlling Switch shall transmit an ELP Request after completing Link Initialization. This ELP Request has the Controlling FCF/Switch flag set to one and the FDF/FCDF flag set to zero.

If the ELP is accepted by the neighbor and

a) the received ELP SW_ACC has both the Controlling FCF/Switch flag and the FDF/FCDF flag set to zero; or

b) the received ELP SW_ACC has the Controlling FCF/Switch flag set to one and the neighbor Switch is not the peer Controlling Switch of this Distributed Switch

then the Controlling Switch port behaves as an E_Port (i.e., an ISL is instantiated).

If the ELP is accepted and the received ELP SW_ACC has the Controlling FCF/Switch flag set to one and the neighbor Switch is the peer Controlling Switch of this Distributed Switch then the Controlling Switch port behaves as an Augmented E_Port for this Distributed Switch (i.e., an AISL is instantiated), used for the redundancy protocol of the Distributed Switch (see 17.8).

If the ELP is accepted and the received ELP SW_ACC has the FDF/FCDF flag set to one and the neighbor FCDF is part of this Distributed Switch FCDF Set, then the Controlling Switch port behaves as an A_Port (i.e., an ASL is instantiated) when the Controlling Switch is operational (i.e., when in state P2 or S2 of of the Controlling Switch Redundancy Protocol, see 17.8), otherwise (i.e., when the Controlling Switch is not yet operational) the Controlling Switch port shall transition to the Isolated state.

If the ELP is accepted and the received ELP SW_ACC has the FDF/FCDF flag set to one and the neighbor FCDF is not part of this Distributed Switch FCDF Set, then the Controlling Switch port shall go in Isolated state.

A port of a Controlling Switch shall reject a received ELP Request with the FDF/FCDF flag set to one with Reason Code ‘Protocol Error’ and Reason Code Explanation ‘Invalid Request’.

A port of a Controlling Switch shall reply to a received ELP Request with the FDF/FCDF flag set to zero according to the normal ELP rules (i.e., acceptance or rejection includes considering the involved Switch_Names). If the ELP Request is accepted and

a) the received ELP Request has both the Controlling FCF/Switch flag and the FDF/FCDF flag set to zero; or

b) the received ELP Request has the Controlling FCF/Switch flag set to one and the neighbor Switch is not the peer Controlling Switch of this Distributed Switch

then the Controlling Switch port behaves as an E_Port (i.e., an ISL is instantiated).

If the ELP is accepted and the received ELP Request has the Controlling FCF/Switch flag set to one and the neighbor Switch is the peer Controlling Switch of this Distributed Switch, then the Controlling Switch port behaves as an Augmented E_Port for this Distributed Switch (i.e., an AISL is instantiated), used for the Redundancy protocol of the Distributed Switch (see 17.8).

The ports of an FCDF that has not yet received from the Primary Controlling Switch the Distributed Switch's FCDF Set through the DFMD SW_ILS shall wait to receive an ELP Request after completing Link Initialization.

After having received from the Primary Controlling Switch the Distributed Switch's FCDF Set through the DFMD SW_ILS, the ports of an FCDF that have completed Link Initialization, except the one from which the DFMD Request has been received, shall transmit an ELP Request with the FDF/FCDF flag set to one.

On Receiving an ELP Request with the Controlling FCF/Switch flag set to one or the FDF/FCDF flag set to one, the FCDF port shall process it irrespective of the value of the Switch_Name field in the ELP Request payload (i.e., acceptance or rejection shall be based on the other ELP parameters, not on the involved Switch_Names). If the ELP is accepted then the FCDF Port behaves as an A_Port (i.e., an ASL is instantiated).

NOTE 48 – These rules enable an ordered establishments of ASLs from the Controlling Switch(es) to the peripheral FCDFs in a Distributed Switch with cascaded FCDFs.

An FCDF does not support E_Ports, therefore a port of an FCDF shall reject a received ELP Request with both Controlling FCF/Switch flag and FDF/FCDF flag set to zero (i.e., a ELP Request coming from a Switch that is not a Controlling Switch) with Reason Code 'Protocol Error' and Reason Code Explanation 'Invalid Request'. After an FCDF has received the Distributed Switch's FCDF Set from the Primary Controlling Switch, that FCDF shall reject received ELP Requests coming from a Controlling Switch other than the Controlling Switches in the Controlling Switch set, with Reason Code 'Logical Error' and Reason Code Explanation 'Not Authorized'.

17.7 VA_Port SW_ILSs

17.7.1 Overview

The VA_Port SW_ILSs are used to exchange information between Controlling Switches and FCDFs (i.e., they are not used to exchange information between FCDFs). When a Distributed Switch includes cascaded FCDFs, the intermediate FCDFs relay the SW_ILSs as shown in figure 49.

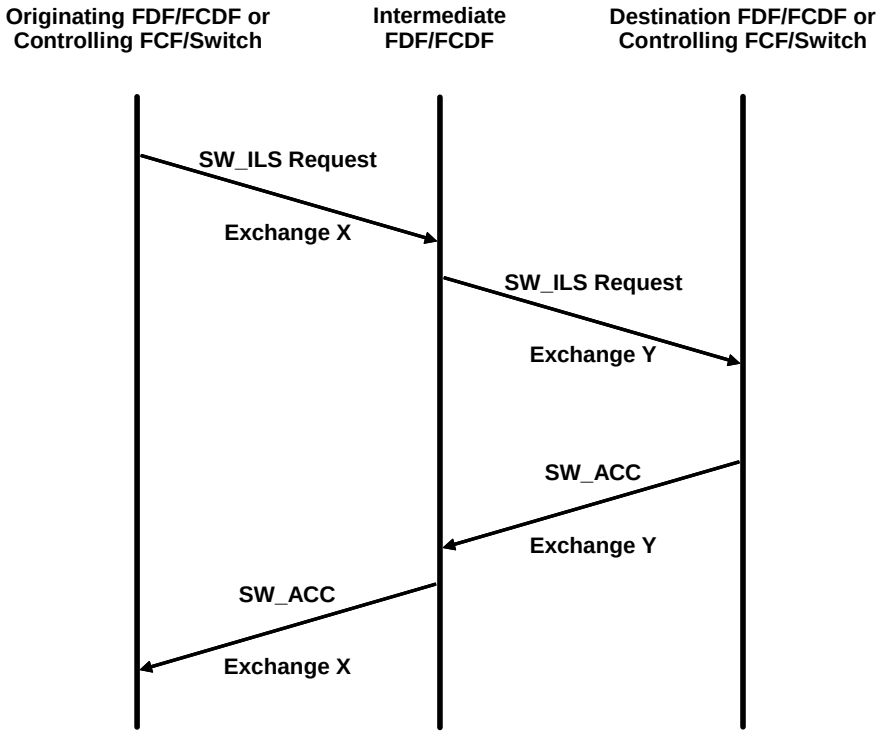

Figure 49 – VA_Port SW_ILS Relay

To enable this relay, all VA_Port SW_ILSs include the originating and destination FCDF or Controlling Switch Switch_Names in the first two fields of their payload. The subsequent part of a VA_Port SW_ILS is a list of self-identifying descriptors, as defined in 17.7.2. The descriptor list may be null.

The need for the originating and destination FCDF or Controlling Switch Switch_Names in the first two fields of the payload requires the definition of an updated SW_RJT, called here VA_RJT (see 17.7.3.1).

The VA_Port SW_ILSs have the same high-order byte in their command code, denoted here as XXh. Table 210 shows the VA_Port SW_ILSs command codes.

Table 210 – VA_Port SW_ILSs Command Codes

Encoded Value	Description	Abbreviation
XX00 0001h	VN_Port Reachability Notification	VNRN
XX00 0002h	VN_Port Unreachability Notification	VNUN
XX00 0003h	FCDF Reachability Notification	FDRN
XX00 0004h	FCDF Unreachability Notification	FDUN

Table 210 – VA_Port SW_ILSs Command Codes

Encoded Value	Description	Abbreviation
XX00 0005h	N_Port_ID Route Distribution	NPRD
XX00 0006h	N_Port_ID and Zoning ACL Distribution	NPZD
XX00 0007h	Active Zoning ACL Distribution	AZAD
XX00 0008h	Distributed Switch Membership Distribution	DFMD

17.7.2 VA_Port SW_ILS Descriptors

17.7.2.1 Descriptor Format

Each VA_Port SW_ILS descriptor has the format shown in table 211. This format applies also to the descriptors for the Redundancy Protocol SW_ILSs (see 17.8).

Table 211 – Descriptor Format

Item	Size (Bytes)
Descriptor Tag	4
Descriptor Length	4
Descriptor Value	variable

Descriptor Tag: the two most significant bytes of this field are reserved, the two least significant bytes contain the tag value. The defined tag values are shown in table 212.

Table 212 – Descriptor Tags

Tag Value	Descriptor	Reference
0001h	VN_Port Reachability	17.7.2.2
0002h	FLOGI/NPIV FDISC Parameters	17.7.2.3
0003h	VN_Port Unreachability	17.7.2.4
0004h	FCDF Reachability	17.7.2.5
0005h	Sequence Number	17.7.2.6
0006h	Controlling Switch Reachability	17.7.2.7
0007h	N_Port_IDs Reachability	17.7.2.8
0008h	Domain_IDs Reachability	17.7.2.9
0009h	Allocation Status	17.7.2.10
000Ah	Peering Status	17.7.2.11
000Bh	Membership Set	17.7.2.12
000Ch	Integrity	17.7.2.13
000Dh	FCDF Identification	17.7.2.14
000Eh	Reject	17.7.2.15
0011h	Controlling Switch State	17.8.2.2
0012h	FCDF Topology	17.8.2.3

Table 212 – Descriptor Tags

Tag Value	Descriptor	Reference
0013h	FCDF N_Port_IDs	17.8.2.4
0014h	RHello Interval	17.8.2.5
all others	Reserved	

Descriptor Length: contains the length in bytes of the Descriptor Value.

Descriptor Value: contains the specific information carried in the descriptor.

17.7.2.2 VN_Port Reachability Descriptor

The format of the VN_Port Reachability descriptor is shown in table 213.

Table 213 – VN_Port Reachability Descriptor Format

Item	Size (Bytes)
Tag Value = 0001h	4
Length = 12	4
F_Port_Name	8
Physical Port Number	4

F_Port_Name: contains the F_Port_Name of the VF_Port to which the newly reachable VN_Port is being associated.

Physical Port Number: contains the physical port number where an FLOGI or NPIV FDISC Request has been received.

17.7.2.3 FLOGI/NPIV FDISC Parameters Descriptor

The format of the FLOGI/NPIV FDISC Parameters descriptor is shown in table 214.

Table 214 – FLOGI/NPIV FDISC Parameters Descriptor Format

Item	Size (Bytes)
Tag Value = 0002h	4
Length = 116	4
FLOGI/NPIV FDISC Parameters	116

FLOGI/NPIV FDISC Parameters Descriptor: contains the payload of an FLOGI or NPIV FDISC (see FC-LS-2).

17.7.2.4 VN_Port Unreachability Descriptor

The format of the VN_Port Unreachability descriptor is shown in table 215.

Table 215 – VN_Port Unreachability Descriptor Format

Item	Size (Bytes)
Tag Value = 0003h	4
Length = 20	4
Flags	1
Unreachable N_Port_ID	3
Unreachable N_Port_Name	8
F_Port_Name	8

Flags: 8 flag bits. The following flag bits are defined:

Bit 8 .. 1: reserved.

Bit 0: indicates if only one VN_Port is unreachable or if all the VN_Ports associated to a VF_Port are unreachable. This flag is set to zero to indicate that only one VN_Port is unreachable and to one to indicate that all the VN_Ports associated to a VF_Port are unreachable.

Unreachable N_Port_ID: when bit 0 of the flag field is set to zero contains the N_Port_ID of the unreachable VN_Port. When bit 0 of the flag field is set to one contains 000000h.

Unreachable N_Port_Name: when bit 0 of the flag field is set to zero contains the N_Port_Name of the unreachable VN_Port. When bit 0 of the flag field is set to one contains 0000 0000 0000 0000h.

F_Port_Name: contains the F_Port_Name of the involved VF_Port.

17.7.2.5 FCDF Reachability Descriptor

The format of the FCDF Reachability descriptor is shown in table 216.

Table 216 – FCDF Reachability Descriptor Format

Item	Size (Bytes)
Tag Value = 0004h	4
Length = 28	4
FCDF or Controlling Switch Switch_Name	8
Local A_Port_Name	8
Adjacent A_Port_Name	8
Reserved	2
Link Cost	2

FCDF or Controlling Switch Switch_Name: contains the Switch_Name of the adjacent entity with which an ASL has been instantiated or deinstantiated.

Local A_Port_Name: contains the local A_Port_Name of the instantiated or deinstantiated ASL.

Adjacent A_Port_Name: contains the adjacent A_Port_Name of the instantiated or deinstantiated ASL.

Link Cost: contains the cost of the instantiated or deinstantiated ASL.

17.7.2.6 Sequence Number Descriptor

The format of the Sequence Number descriptor is shown in table 217.

Table 217 – Sequence Number Descriptor Format

Item	Size (Bytes)
Tag Value = 0005h	4
Length = 8	4
Sequence Number	8

Sequence Number: contains a monotonically increasing sequence number. When the sequence number reaches the value FFFFFFFF FFFFFFFFh it wraps to 00000000 00000000h.

17.7.2.7 Controlling Switch Reachability Descriptor

The format of the Controlling Switch Reachability descriptor is shown in table 218.

Table 218 – Controlling Switch Reachability Descriptor Format

Item	Size (Bytes)
Tag Value = 0006h	4
Length = variable	4
Controlling Switch Switch_Name	8
Number of Paths to the Controlling Switch (j)	4
Next-hop Switch_Name #1	8
Local A_Port_Name #1	8
Path #1 cost	4
Next-hop Switch_Name #2	8
Local A_Port_Name #2	8
Path #2 cost	4
...	
Next-hop Switch_Name #j	8
Local A_Port_Name #j	8
Path #j cost	4

Controlling Switch Switch_Name: contains the Switch_Name of the Controlling Switch.

Number of Paths to the Controlling Switch: contains the number of paths toward the Controlling Switch. Each path that follows is expressed by the Switch_Name of the next-hop FCDF or Controlling Switch followed by the local A_Port_Name of the involved ASL and by the path cost.

17.7.2.8 N_Port_IDs Reachability Descriptor

The format of the N_Port_IDs Reachability descriptor is shown in table 219.

Table 219 – N_Port_IDs Reachability Descriptor Format

Item	Size (Bytes)
Tag Value = 0007h	4
Length = variable	4
Number of N_Port_ID Reachability Entries (p)	4
N_Port_ID Reachability Entry #1	see table 220
N_Port_ID Reachability Entry #2	see table 220
…	
N_Port_ID Reachability Entry #p	see table 220

Number of N_Port_ID Reachability Entries: contains the number of N_Port_ID Reachability Entries that follow. There shall be an N_Port_ID Reachability Entry for each FCDF currently belonging the Distributed Switch. The N_Port_ID Reachability Entry format is shown in table 220.

Table 220 – N_Port_ID Reachability Entry Format

Item	Size (bytes)
Reachable FCDF Switch_Name	8
Number of Equal Cost Paths to the Reachable FCDF (w)	4
Next-hop Switch_Name #1	8
Local A_Port_Name #1	8
Next-hop Switch_Name #2	8
Local A_Port_Name #2	8
...	
Next-hop Switch_Name #w	8
Local A_Port_Name #w	8
Number of N_Port_ID Ranges (q)	4
N_Port_ID Range #1	4
N_Port_ID Range #2	4
…	
N_Port_ID Range #q	4

Reachable FCDF Switch_Name: contains the Switch_Name of the FCDF to which the subsequent next-hops and N_Port_ID Ranges refer.

Number of Equal Cost Paths to the Reachable FCDF: contains the number of equal cost paths having the lowest cost toward the destination FCDF. Each path that follows is expressed as the Switch_Name of the next-hop FCDF or Controlling Switch followed by the local A_Port_Name of the involved ASL.

Number of N_Port_ID Ranges: contains the number of N_Port_ID Range Entries that follow. The N_Port_ID Range is defined by an N_Port_ID in the least significant three bytes, and by the number of bits defining the range in the most significant byte (e.g., the range 020200h .. 02027Fh is expressed as '7 || 020200h'). The set of N_Port_ID Range Entries encodes in a compact form all the N_Port_IDs currently allocated to VN_Ports logged into the reachable FCDF.

17.7.2.9 Domain_IDs Reachability Descriptor

The format of the Domain_IDs Reachability descriptor is shown in table 221.

Table 221 – Domain_IDs Reachability Descriptor Format

Item	Size (Bytes)
Tag Value = 0008h	4
Length = variable	4
Number of Reachable Domain_ID Entries (r)	4
Reachable Domain_ID Entry #1	see table 222
Reachable Domain_ID Entry #2	see table 222
…	
Reachable Domain_ID Entry #r	see table 222

Number of Reachable Domain_ID Entries: contains the number of Reachable Domain_ID Entries that follow. The Reachable Domain_ID Entry format is shown in table 222.

Table 222 – Reachable Domain_ID Entry Format

Item	Size (bytes)
Reachable Domain_ID	4
Number of Equal Cost Paths to the Reachable Domain_ID (y)	4
Next-hop Switch_Name #1	8
Local A_Port_Name #1	8
Next-hop Switch_Name #2	8
Local A_Port_Name #2	8
...	
Next-hop Switch_Name #y	8
Local A_Port_Name #y	8

Reachable Domain_ID: contains the reachable Domain_ID. The three most significant bytes of this field are reserved.

Number of Equal Cost Paths to the Reachable Domain_ID: contains the number of equal cost paths having the lowest cost toward the destination Domain_ID. Each path that follows is expressed as the Switch_Name of the next-hop FCDF or Controlling Switch followed by the local A_Port_Name of the involved ASL.

17.7.2.10 Allocation Status Descriptor

The format of the Allocation Status descriptor is shown in table 223.

Table 223 – Allocation Status Descriptor Format

Item	Size (Bytes)
Tag Value = 0009h	4
Length = variable	4
Number of Allocation / Deallocation Entries (z)	4
Allocation / Deallocation Entry #1	see table 224
Deallocation Entry #2	see table 224
...	
Deallocation Entry #z	see table 224

Number of Allocation / Deallocation Entries: contains the number of Allocation / Deallocation Entries that follow. Only one Allocation Entry may be present, multiple Deallocation Entries may be present. The Allocation / Deallocation Entry format is shown in table 224.

Table 224 – Allocation / Deallocation Entry Format

Item	Size (bytes)
Flags	4
Allocated / Deallocated N_Port_ID	4
N_Port_Name associated with the Allocated/Deallocated N_Port_ID	8
Switch_Name of the FCDF associated with the Allocated/Deallocated N_Port_ID	8
FLOGI / NPIV FDISC LS_ACC Parameters	116

Flags: 32 flag bits. The following flag bits are defined:

Bit 32 .. 2: reserved.

Bit 1: indicates if the FLOGI / NPIV FDISC LS_ACC Parameters field is present in the payload. The field is present when this flag is set to one and not present when this flag is set to zero. This flag shall not be set to one when bit 0 indicates deallocation (i.e., the FLOGI / NPIV FDISC LS_ACC Parameters field may be present only when an N_Port_ID allocation is performed).

Bit 0: indicates if the operation is an allocation or a deallocation. This flag is set to zero to indicate allocation and to one to indicate deallocation.

Allocated / Deallocated N_Port_ID: contains the N_Port_ID that the Primary Controlling Switch allocated or deallocated in the least significant three bytes. The most significant byte is reserved.

N_Port_Name associated with the Allocated/Deallocated N_Port_ID: contains the N_Port_Name of the VN_Port for which an N_Port_ID is allocated or deallocated.

Switch_Name of the FCDF associated with the Allocated/Deallocated N_Port_ID: contains the Switch_Name of the FCDF associated with the VN_Port for which an N_Port_ID is allocated or deallocated.

FLOGI / NPIV FDISC LS_ACC Parameters: this field is present when bit 1 of the flags field is set to one. It contains the payload of the LS_ACC generated by the Primary Controlling Switch in response to the FLOGI or NPIV FDISC payload provided in the VNRN Request Sequence.

17.7.2.11 Peering Status Descriptor

The format of the Peering Status descriptor is shown in table 225.

Table 225 – Peering Status Descriptor Format

Item	Size (Bytes)
Tag Value = 000Ah	4
Length = variable	4
Number of Peering Entries (h)	4
Peering Entry #1	see table 226
Peering Entry #2	see table 226
…	
Peering Entry #h	see table 226

Number of Peering Entries: contains the number of Peering Entries that follow. Each Peering Entry contains a complete list of the Peer N_Port_IDs with which the Principal N_Port_ID is allowed to communicate according to the current fabric zoning configuration. The Peering Entry format is shown in table 226.

Table 226 – Peering Entry Format

Item	Size (bytes)
Principal N_Port_ID	4
Number of Allowed Peers (k)	4
Peer N_Port_ID #1	4
Peer N_Port_ID #2	4
…	
Peer N_Port_ID #q	4

Principal N_Port_ID: contains the N_Port_ID to which the subsequent Peer N_Port_IDs refer.

Number of Allowed Peers: contains the number of N_Port_IDs to which the Principal N_Port_ID is allowed to communicate.

Peer N_Port_ID: contains an N_Port_ID in the least significant three bytes and the most significant byte is reserved.

17.7.2.12 Membership Set Descriptor

The format of the Membership Set descriptor is shown in table 227.

Table 227 – Membership Set Descriptor Format

Item	Size (Bytes)
Tag Value = 000Bh	4
Length = variable	4
Primary Controlling Switch Switch_Name	8
Secondary Controlling Switch Switch_Name	8
Number of FCDFs (n)	4
FCDF Switch_Name #1	8
FCDF Switch_Name #2	8
…	
FCDF Switch_Name #n	8

Primary Controlling Switch Switch_Name: contains the Switch_Name of the Primary Controlling Switch.

Secondary Controlling Switch Switch_Name: contains the Switch_Name of the Secondary Controlling Switch. This field shall be set to 00000000 00000000h when there is no Secondary Controlling Switch.

Number of FCDFs: contains the number of FCDF Switch_Names that follow. This list of FCDF Switch_Names is the FCDF Set of the Distributed Switch. If the number of FCDF Switch_Names is zero, then any FCDF is allowed in the Distributed Switch.

17.7.2.13 Integrity Descriptor

The format of the Integrity descriptor is shown in table 228.

Table 228 – Integrity Descriptor Format

Item	Size (Bytes)
Tag Value = 000Ch	4
Length = variable	4
Integrity Type	4
Integrity Check Value Length	4
Integrity Check Value	variable

Integrity Type: indicates, in the least significant byte, the type of cryptographic integrity that protects the payload. The defined values are:

00h: No integrity

01h: HMAC-SHA-256-128 integrity

02h .. FFh: Reserved

Integrity Check Value Length: contains the length expressed in bytes of the Integrity Check Value.

Integrity Check Value: contains the cryptographic hash of the payload computed using the shared key according to the specified Integrity Type.

17.7.2.14 FCDF Identification Descriptor

The format of the FCDF Identification descriptor is shown in table 229.

Table 229 – FCDF Identification Descriptor Format

Item	Size (Bytes)
Tag Value = 000Dh	4
Length = 36	4
Number of Physical Ports	4
RNID Specific Node-Identification Data	32

Number of Physical Ports: contains the number of physical ports that the FCDF has.

RNID Specific Node-Identification Data: see FC-SB-4.

17.7.2.15 Reject Descriptor

The format of the Reject descriptor is shown in table 230.

Table 230 – Reject Descriptor Format

Item	Size (Bytes)
Tag Value = 000Eh	4
Length = 4	4
Reserved	1
Reason Code	1
Reason Code Explanation	1
Vendor Specific	1

17.7.3 VA_Port SW_ILSs Definition

17.7.3.1 VA_RJT

The VA_RJT SW_ILS is used in place of an SW_RJT as a reply Sequence to a VA_Port SW_ILS Request to reject that request.

Addressing: The S_ID field shall be set to the value of the D_ID field in the SW_ILS request. The D_ID field shall be set to the value of the S_ID field in the SW_ILS request.

Payload: the format of the VA_RJT Payload is shown in table 231.

Table 231 – VA_RJT Payload

Item	Size (bytes)
SW_ILS Code = 0300 0000h	4
Destination Switch_Name	8
Originating Switch_Name	8
Descriptor List Length	4
Reject Descriptor	see 17.7.2.15

Destination Switch_Name: contains the Switch_Name of the destination entity.

Originating Switch_Name: contains the Switch_Name of the originating entity.

Descriptor List Length: contains the length in bytes of the subsequent list of descriptors.

Reject Descriptor: see 17.7.2.15

17.7.3.2 VN_Port Reachability Notification (VNRN)

The VN_Port Reachability Notification SW_ILS is used by an FCDF to communicate to the Primary Controlling Switch that a VN_Port is attempting Fabric login through an FLOGI Request or a NPIV FDISC Request. If the FCDF does not have an ASL with the Primary Controlling Switch, the VNRN SW_ILS is relayed to the Primary Controlling Switch by the intermediate FCDFs.

VNRN Request Sequence

Addressing: the S_ID field shall be set to FFFFF9h, indicating the originating VA_Port, and the D_ID field shall be set to FFFFF9h, indicating the destination VA_Port.

Payload: the format of the VNRN Request Sequence Payload is shown in table 232.

Table 232 – VNRN Request Payload

Item	Size (bytes)
SW_ILS Code = XX00 0001h	4
Destination Controlling Switch Switch_Name	8
Originating FCDF Switch_Name	8
Descriptor List Length	4
VN_Port Reachability Descriptor	see 17.7.2.2
FLOGI/NPIV FDISC Parameters Descriptor	see 17.7.2.3

Destination Controlling Switch Switch_Name: contains the Switch_Name of the destination Controlling Switch.

Originating FCDF Switch_Name: contains the Switch_Name of the originating FCDF.

Descriptor List Length: contains the length in bytes of the subsequent list of descriptors.

VN_Port Reachability Descriptor: see 17.7.2.2.

FLOGI/NPIV FDISC Parameters Descriptor: contains the payload of the received FLOGI or NPIV FDISC Request (see FC-LS-2).

VNRN Reply Sequence

VA_RJT: indicates the rejection of the VNRN Request Sequence. As a result, a FLOGI LS_RJT or a NPIV FDISC LS_RJT is sent as response to the FLOGI Request or NPIV FDISC Request that caused the issuance of the VNRN Request.

SW_ACC: indicates the acceptance of the VNRN Request Sequence. The format of the VNRN SW_ACC Payload is shown in table 233.

Table 233 – VNRN SW_ACC Payload

Item	Size (bytes)
SW_ILS Code = 0200 0000h	4
Destination FCDF Switch_Name	8
Originating Controlling Switch Switch_Name	8
Descriptor List Length	4
FLOGI / NPIV FDISC Parameters Descriptor	see 17.7.2.3

FLOGI / NPIV FDISC Parameters Descriptor: this descriptor contains the payload of the LS_ACC generated by the Primary Controlling Switch in response to the FLOGI or NPIV FDISC payload provided in the VNRN Request Sequence.

17.7.3.3 VN_Port Unreachability Notification (VNUN)

The VN_Port Unreachability Notification SW_ILS is used by an FCDF to communicate to the Primary Controlling Switch that one or more of its VN_Ports have been logged out. If the FCDF does not have an ASL with the Primary Controlling Switch, the VNUN SW_ILS is relayed to the Primary Controlling Switch by the intermediate FCDFs.

VNUN Request Sequence

Addressing: the S_ID field shall be set to FFFFF9h, indicating the originating VA_Port, and the D_ID field shall be set to FFFFF9h, indicating the destination VA_Port.

Payload: the format of the VNUN Request Sequence Payload is shown in table 234.

Table 234 – VNUN Request Payload

Item	Size (bytes)
SW_ILS Code = XX00 0002h	4
Destination Controlling Switch Switch_Name	8
Originating FCDF Switch_Name	8
Decriptor List Length	4
VN_Port Unreachability Descriptor	see 17.7.2.4

Destination Controlling Switch Switch_Name: contains the Switch_Name of the destination Controlling Switch.

Originating FCDF Switch_Name: contains the Switch_Name of the requesting FCDF.

Descriptor List Length: contains the length in bytes of the subsequent list of descriptors.

VN_Port Unreachability Descriptor: see 17.7.2.4**.**

VNUN Reply Sequence

VA_RJT: indicates the rejection of the VNUN Request Sequence.

SW_ACC: indicates the acceptance of the VNUN Request Sequence. The format of the VNUN SW_ACC Payload is shown in table 235.

Table 235 – VNUN SW_ACC Payload

Item	Size (bytes)
SW_ILS Code = 0200 0000h	4
Destination FCDF Switch_Name	8
Originating Controlling Switch Switch_Name	8
Descriptor List Length = 0000 0000h	4

17.7.3.4 FCDF Reachability Notification (FDRN)

The FCDF Reachability Notification SW_ILS is used by an FCDF to communicate to the Primary Controlling Switch that it has instantiated an ASL with another FCDF or with the Secondary Controlling Switch. If the FCDF does not have an ASL with the Primary Controlling Switch, the FDRN SW_ILS is relayed to the Primary Controlling Switch by the intermediate FCDFs.

The FDRN SW_ILS is also used between Primary and Secondary Controlling Switch to keep their state synchronized.

FDRN Request Sequence

Addressing: when used between a FCDF and the Primary Controlling Switch the S_ID field shall be set to FFFFF9h, indicating the originating VA_Port, and the D_ID field shall be set to FFFFF9h, indi-

cating the destination VA_Port. When used between the two Controlling Switches the S_ID field shall be set to FFFFFDh, indicating the originating VE_Port, and the D_ID field shall be set to FFFFFDh, indicating the destination VE_Port.

Payload: the format of the FDRN Request Sequence Payload is shown in table 236.

Table 236 – FDRN Request Payload

Item	Size (bytes)
SW_ILS Code = XX00 0003h	4
Destination Controlling Switch Switch_Name	8
Originating FCDF Switch_Name	8
Descriptor List Length	4
FCDF Reachability Descriptor	see 17.7.2.5

Destination Controlling Switch Switch_Name: contains the Switch_Name of the destination Controlling Switch.

Originating FCDF Switch_Name: contains the Switch_Name of the requesting FCDF.

Descriptor List Length: contains the length in bytes of the subsequent list of descriptors.

FCDF Reachability Descriptor: describes the instantiated ASL **(**see 17.7.2.5).

FDRN Reply Sequence

VA_RJT: indicates the rejection of the FDRN Request Sequence.

SW_ACC: indicates the acceptance of the FDRN Request Sequence. The format of the FDRN SW_ACC Payload is shown in table 237.

Table 237 – FDRN SW_ACC Payload

Item	Size (bytes)
SW_ILS Code = 0200 0000h	4
Destination FCDF Switch_Name	8
Originating Controlling Switch Switch_Name	8
Descriptor List Length = 0000 0000h	4

17.7.3.5 FCDF Unreachability Notification (FDUN)

The FCDF Unreachability Notification SW_ILS is used by an FCDF to communicate to the Primary Controlling Switch that it has deinstantiated an ASL with another FCDF or with the Secondary Controlling Switch. If the FCDF does not have an ASL with the Primary Controlling Switch, the FDUN SW_ILS is relayed to the Primary Controlling Switch by the intermediate FCDFs.

The FDUN SW_ILS is also used between Primary and Secondary Controlling Switch to keep their state synchronized.

FDUN Request Sequence

Addressing: when used between a FCDF and the Primary Controlling Switch the S_ID field shall be set to FFFFF9h, indicating the originating VA_Port, and the D_ID field shall be set to FFFFF9h, indicating the destination VA_Port. When used between the two Controlling Switches the S_ID field shall be set to FFFFFDh, indicating the originating VE_Port, and the D_ID field shall be set to FFFFFDh, indicating the destination VE_Port.

Payload: the format of the FDUN Request Sequence Payload is shown in table 238.

Table 238 – FDUN Request Payload

Item	Size (bytes)
SW_ILS Code = XX00 0004h	4
Destination Controlling Switch Switch_Name	8
Originating FCDF Switch_Name	8
Descriptor List Length	4
FCDF Reachability Descriptor	see 17.7.2.5

Destination Controlling Switch Switch_Name: contains the Switch_Name of the destination Controlling Switch.

Originating FCDF Switch_Name: contains the Switch_Name of the requesting FCDF.

Descriptor List Length: contains the length in bytes of the subsequent list of descriptors.

FCDF Reachability Descriptor: describes the deinstantiated ASL **(**see 17.7.2.5)..

FDUN Reply Sequence

VA_RJT: indicates the rejection of the FDUN Request Sequence.

SW_ACC: indicates the acceptance of the FDUN Request Sequence. The format of the FDUN SW_ACC Payload is shown in table 239.

Table 239 – FDUN SW_ACC Payload

Item	Size (bytes)
SW_ILS Code = 0200 0000h	4
Destination FCDF Switch_Name	8
Originating Controlling Switch Switch_Name	8
Descriptor List Length = 0000 0000h	4

17.7.3.6 N_Port_ID Route Distribution (NPRD)

The N_Port_ID Route Distribution SW_ILS is used by the Primary Controlling Switch to communicate to an FCDF the N_Port_ID routing information for the Distributed Switch. If the Primary Controlling Switch does not have an ASL with the destination FCDF, the NPRD SW_ILS is relayed to the destination FCDF by the intermediate FCDFs.

NPRD Request Sequence

Addressing: the S_ID field shall be set to FFFFF9h, indicating the originating VA_Port, and the D_ID field shall be set to FFFFF9h, indicating the destination VA_Port.

Payload: the format of the NPRD Request Sequence Payload is shown in table 240.

Table 240 – NPRD Request Payload

Item	Size (bytes)
SW_ILS Code = XX00 0005h	4
Destination FCDF Switch_Name	8
Originating Controlling Switch Switch_Name	8
Descriptor List Length	4
Sequence Number Descriptor	see 17.7.2.6
Primary Controlling Switch Reachability Descriptor	see 17.7.2.7
Secondary Controlling Switch Reachability Descriptor	see 17.7.2.7
N_Port_IDs Reachability Descriptor	see 17.7.2.8
Domain_IDs Reachability Descriptor	see 17.7.2.9

Destination FCDF Switch_Name: contains the Switch_Name of the destination FCDF.

Originating Controlling Switch Switch_Name: contains the Switch_Name of the requesting Controlling Switch.

Descriptor List Length: contains the length in bytes of the subsequent list of descriptors.

Sequence Number Descriptor: see 17.7.2.6.

Primary Controlling Switch Reachability Descriptor: contains the reachability information toward the Primary Controlling Switch.

NOTE 49 – Paths toward the Primary Controlling Switch are fundamental for the operation of an FCDF. Specifying higher cost paths enables more redundancy, because if the lowest cost path toward the Primary Controlling Switch fails, a higher cost path may be used.

Secondary Controlling Switch Reachability Descriptor: contains the reachability information toward the Secondary Controlling Switch.

N_Port_IDs Reachability Descriptor: see 17.7.2.8.

Domain_IDs Reachability Descriptor: see 17.7.2.9.

NPRD Reply Sequence

VA_RJT: indicates the rejection of the NPRD Request Sequence.

SW_ACC: indicates the acceptance of the NPRD Request Sequence. The format of the NPRD SW_ACC Payload is shown in table 241.

Table 241 – NPRD SW_ACC Payload

Item	Size (bytes)
SW_ILS Code = 0200 0000h	4
Destination Controlling Switch Switch_Name	8
Originating FCDF Switch_Name	8
Descriptor List Length = 0000 0000h	4

17.7.3.7 N_Port_ID and Zoning ACL Distribution (NPZD)

The N_Port_ID and Zoning ACL Distribution SW_ILS is used by the Primary Controlling Switch to communicate to the Secondary Controlling Switch the allocation of an N_Port_ID and/or the deallocation of one or more N_Port_IDs and to communicate to an FCDF the allocation of an N_Port_ID and its associated Zoning ACL information and/or the deallocation of one or more N_Port_IDs and their associated Zoning ACL information. Upon receiving an NPZD Request, an FCDF shall update its Zoning enforcement according to the received Zoning ACLs only for the listed Principal N_Port_IDs. If the Primary Controlling Switch does not have an ASL with the destination FCDF, the NPZD SW_ILS is relayed to the destination FCDF by the intermediate FCDFs.

NPZD Request Sequence

Addressing: when used between a FCDF and the Primary Controlling Switch the S_ID field shall be set to FFFFF9h, indicating the originating VA_Port, and the D_ID field shall be set to FFFFF9h, indicating the destination VA_Port. When used between the two Controlling Switches the S_ID field shall be set to FFFFFDh, indicating the originating VE_Port, and the D_ID field shall be set to FFFFFDh, indicating the destination VE_Port.

Payload: the format of the NPZD Request Sequence Payload is shown in table 242.

Table 242 – NPZD Request Payload

Item	Size (bytes)
SW_ILS Code = XX00 0006h	4
Destination FCDF or Controlling Switch Switch_Name	8
Originating Controlling Switch Switch_Name	8
Descriptor List Length	4
Sequence Number Descriptor	see 17.7.2.6
Allocation Status Descriptor	see 17.7.2.10
Peering Status Descriptor	see 17.7.2.11

Destination FCDF or Controlling Switch Switch_Name: contains the Switch_Name of the destination FCDF or Controlling Switch.

Originating Controlling Switch Switch_Name: contains the Switch_Name of the requesting Controlling Switch.

Descriptor List Length: contains the length in bytes of the subsequent list of descriptors.

Sequence Number: see 17.7.2.6.

Allocation Status Descriptor: see 17.7.2.10.

When an N_Port_ID is deallocated and allocated at the same time (e.g., as a result of a re-login), that N_Port_ID is listed only in the allocation entry (i.e., there is no deallocation entry for that N_Port_ID in the Allocation Status descriptor).

Peering Status Descriptor: see 17.7.2.11.

When present, the Peering Status descriptor contains Peering entries per each VN_Port currently logged into the destination FCDF and with which the allocated N_Port_ID is allowed to communicate or with which the deallocated N_Port_IDs were allowed to communicate, according to the current fabric zoning configuration. In case of allocation, the Peering Status descriptor for the FCDF that receives the allocated N_Port_ID also contains a Peering Entry with a Principal N_Port_ID equal to the allocated N_Port_ID. In case of deallocation, the Zoning ACLs for the deallocated N_Port_IDs are implicitly removed and the Peering Status descriptor for the FCDF that had the deallocated N_Port_IDs does not contain Peering Entries with a Principal N_Port_ID equal to any the deallocated N_Port_IDs.

NPZD Reply Sequence

VA_RJT: indicates the rejection of the NPZD Request Sequence.

SW_ACC: indicates the acceptance of the NPZD Request Sequence. The format of the NPZD SW_ACC Payload is shown in table 243.

Table 243 – NPZD SW_ACC Payload

Item	Size (bytes)
SW_ILS Code = 0200 0000h	4
Destination Controlling Switch Switch_Name	8
Originating FCDF Switch_Name	8
Descriptor List Length = 0000 0000h	4

17.7.3.8 Active Zoning ACL Distribution (AZAD)

The Active Zoning ACL Distribution SW_ILS is used by the Primary Controlling Switch to communicate to an FCDF new Zoning ACL information when a new Zone Set is activated in the fabric. Upon receiving an AZAD Request, an FCDF shall completely replace its Zoning enforcement according to the received Zoning ACLs. If the Primary Controlling Switch does not have an ASL with the destination FCDF, the AZAD SW_ILS is relayed to the destination FCDF by the intermediate FCDFs.

AZAD Request Sequence

Addressing: the S_ID field shall be set to FFFFF9h, indicating the originating VA_Port, and the D_ID field shall be set to FFFFF9h, indicating the destination VA_Port.

Payload: the format of the AZAD Request Sequence Payload is shown in table 244.

Table 244 – AZAD Request Payload

Item	Size (bytes)
SW_ILS Code = XX00 0007h	4
Destination FCDF Switch_Name	8
Originating Controlling Switch Switch_Name	8
Descriptor List Length	4
Sequence Number Descriptor	see 17.7.2.6
Peering Status Descriptor	see 17.7.2.11

Destination FCDF Switch_Name: contains the Switch_Name of the destination FCDF.

Originating Controlling Switch Switch_Name: contains the Switch_Name of the requesting Controlling Switch.

Descriptor List Length: contains the length in bytes of the subsequent list of descriptors.

Sequence Number: see 17.7.2.6.

Peering Status Descriptor: see 17.7.2.11.

AZAD Reply Sequence

VA_RJT: indicates the rejection of the AZAD Request Sequence.

SW_ACC: indicates the acceptance of the AZAD Request Sequence. The format of the AZAD SW_ACC Payload is shown in table 245.

Table 245 – AZAD SW_ACC Payload

Item	Size (bytes)
SW_ILS Code = 0200 0000h	4
Destination Controlling Switch Switch_Name	8
Originating FCDF Switch_Name	8
Descriptor List Length = 0000 0000h	4

17.7.3.9 Distributed Switch Membership Distribution (DFMD)

The Distributed Switch Membership Distribution SW_ILS is used by the Primary Controlling Switch to communicate to an FCDF the identities of the Primary and Secondary Controlling Switches and of all the FCDFs that compose the Distributed Switch. The DFMD payload may be integrity protected by a cryptographic hash; in this case the involved entities shall be provided with a shared key. If the Primary Controlling Switch does not have an ASL with the destination FCDF, the DFMD SW_ILS is relayed to the destination FCDF by the intermediate FCDFs.

DFMD Request Sequence

Addressing: the S_ID field shall be set to FFFFF9h, indicating the originating VA_Port, and the D_ID field shall be set to FFFFF9h, indicating the destination VA_Port.

Payload: the format of the DFMD Request Sequence Payload is shown in table 246.

Table 246 – DFMD Request Payload

Item	Size (bytes)
SW_ILS Code = XX00 0008h	4
Destination FCDF Switch_Name	8
Originating Controlling Switch Switch_Name	8
Descriptor List Length	4
Membership Set Descriptor	see 17.7.2.12
Integrity Descriptor	see 17.7.2.13

Destination FCDF Switch_Name: contains the Switch_Name of the destination FCDF.

Originating Controlling Switch Switch_Name: contains the Switch_Name of the originating Controlling Switch.

Descriptor List Length: contains the length in bytes of the subsequent list of descriptors.

Membership Set Descriptor: see 17.7.2.12.

Integrity Descriptor: see 17.7.2.13.

DFMD Reply Sequence

VA_RJT: indicates the rejection of the DFMD Request Sequence.

SW_ACC: indicates the acceptance of the DFMD Request Sequence. The format of the DFMD SW_ACC Payload is shown in table 247.

Table 247 – DFMD SW_ACC Payload

Item	Size (bytes)
SW_ILS Code = 0200 0000h	4
Destination Controlling Switch Switch_Name	8
Originating FCDF Switch_Name	8
Descriptor List Length	4
FCDF Identification Descriptor	see 17.7.2.14

17.7.4 VA_Port SW_ILS Timeouts

Table 248 shows the timeouts associated to each VA_Port SW_ILS.

Table 248 – VA_Port SW_ILSs Timeouts

Description	Abbreviation	Timeout
VN_Port Reachability Notification	VNRN	2000 ms
VN_Port Unreachability Notification	VNUN	500 ms
FCDF Reachability Notification	FDRN	500 ms
FCDF Unreachability Notification	FDUN	500 ms
N_Port_ID Route Distribution	NPRD	1000 ms
N_Port_ID and Zoning ACL Distribution	NPZD	1000 ms
Active Zoning ACL Distribution	AZAD	1000 ms
Distributed Switch Membership Distribution	DFMD	1000 ms

17.8 Redundancy Protocol SW_ILSs

17.8.1 Overview

The Redundancy Protocol SW_ILSs are used to exchange redundancy information between Controlling Switches. Redundancy Protocol SW_ILSs include the originating and destination Controlling Switch Switch_Names in the first two fields of their payload. The subsequent part of a Redundancy Protocol SW_ILS is a list of self-identifying descriptors, as defined in 17.8.2. The descriptor list may be null.

The Redundancy Protocol SW_ILSs have the same high-order byte in their command code, denoted here as YYh. Table 249 shows the Redundancy Protocol SW_ILSs command codes.

Table 249 – Redundancy Protocol SW_ILSs Command Codes

Encoded Value	Description	Abbreviation
YY00 0001h	Exchange Redundancy Parameters	ERP
YY00 0002h	Get FCDF Topology State	GFTS
YY00 0003h	Get FDCF N_Port_IDs State	GFNS
YY00 0004h	Secondary Synchronization Achieved	SSA
YY00 0005h	Redundancy Hello	RHello

17.8.2 Redundancy Protocol Descriptors

17.8.2.1 Descriptor Format

The Redundancy Protocol descriptors have the same format of the VA_Port SW_ILS descriptors (see 17.7.2.1). Descriptor tags are shown in table 212.

17.8.2.2 Controlling Switch State Descriptor

The format of the Controlling Switch State descriptor is shown in table 250.

Table 250 – Controlling Switch State Descriptor Format

Item	Size (Bytes)
Tag Value = 0011h	4
Length = variable	4
Originating Controlling Switch Priority	4
Number of Allocated N_Port_ID Ranges (q)	4
Allocated N_Port_ID Range #1	4
Allocated N_Port_ID Range #2	4
…	
Allocated N_Port_ID Range #q	4

Originating Controlling Switch Priority: contains the operational Priority of the originating Controlling Switch in the least significant byte and three reserved bytes in the three most significant bytes.

Number of Allocated N_Port_ID Ranges: contains the number of Allocated N_Port_ID Range Entries that follow. This list of Allocated N_Port_ID Ranges identifies the N_Port_IDs allocated by the originating Controlling Switch. The N_Port_ID Range is defined by an N_Port_ID in the least significant three bytes, and by the number of bits defining the range in the most significant byte (e.g., the range 020200h .. 02027Fh is expressed as ‘7 || 020200h’).

17.8.2.3 FCDF Topology Descriptor

The format of the FCDF Topology descriptor is shown in table 251.

Table 251 – FCDF Topology Descriptor Format

Item	Size (Bytes)
Tag Value = 0012h	4
Length = variable	4
Number of FCDF Connectivity Records (n)	4
FCDF Connectivity Record #1	see table 252
FCDF Connectivity Record #2	see table 252
...	
FCDF Connectivity Record #n	see table 252

Number of FCDF Connectivity Records: contains the number of FCDF Connectivity Records that follow. The format of the FCDF Connectivity Record is shown in table 252.

Table 252 – FCDF Connectivity Record Format

Item	Size (bytes)
FCDF Switch_Name	8
Number of ASL Records (m)	4
ASL Record #1	28
ASL Record #2	28
...	
ASL Record #m	8

FCDF Switch_Name: contains the Switch_Name of the FCDF whose ASLs are being described.

Number of ASL Records: contains the number of ASL Records that follow. The format of the ASL Record is shown in table 252

Table 253 – FCDF Connectivity Record Format

Item	Size (bytes)
Switch_Name of Neighbor	8
Local A_Port_Name	8
Adjacent A_Port_Name	8
Link Cost	4

Switch_Name of Neighbor: contains the Switch_Name of the FCDF or Controlling Switch at the other end of the described ASL.

Local A_Port_Name: contains the local A_Port_Name of the described ASL.

Adjacent A_Port_Name: contains the adjacent A_Port_Name of the described ASL.

Link Cost: contains the link cost of the described ASL in the two least significant bytes and two reserved bytes in the two most significant bytes.

17.8.2.4 FCDF N_Port_IDs Descriptor

The format of the FCDF N_Port_IDs descriptor is shown in table 217.

Table 254 – FCDF N_Port_IDs Descriptor Format

Item	Size (Bytes)
Tag Value = 0013h	4
Length = variable	4
Number of Virtual Domain_ID Records (z)	4
Virtual Domain_ID Record #1	see table 255
Virtual Domain_ID Record #2	see table 255
...	
Virtual Domain_ID Record #z	see table 255
Number of FCDF Allocation Records (n)	4
FCDF Allocation Record #1	see table 256
FCDF Allocation Record #2	see table 256
...	
FCDF Allocation Record #n	see table 256

Number of Virtual Domain_ID Records: contains the number of Virtual Domain_ID Records that follow. The format of the Virtual Domain_ID Record is shown in table 256.

Table 255 – Virtual Domain_ID Record Format

Item	Size (bytes)
Virtual Domain_ID Value	4
Distributed Switch Switch_Name	8

Virtual Domain_ID Value: contains a Virtual Domain_ID for the Distributed Switch in the least significant byte and three reserved bytes in the three most significant bytes.

Distributed Switch Switch_Name: contains a Switch_Name for the Distributed Switch, Switch_Name associated with the Virtual Domain_ID value.

Number of FCDF Allocation Records: contains the number of FCDF Allocation Records that follow. The format of the FCDF Allocation Record is shown in table 256.

Table 256 – FCDF Allocation Record Format

Item	Size (bytes)
FCDF Switch_Name	8
Number of Allocated N_Port_ID Ranges (s)	4
Allocated N_Port_ID Range #1	4
Allocated N_Port_ID Range #2	4
...	
Allocated N_Port_ID Range #s	4

FCDF Switch_Name: contains the Switch_Name of the FCDF whose N_Port_IDs allocation is provided.

Number of Allocated N_Port_ID Ranges: contains the number of Allocated N_Port_ID Range Entries that follow. This list of Allocated N_Port_ID Ranges identifies the N_Port_IDs allocated to the described FCDF. The N_Port_ID Range is defined by an N_Port_ID in the least significant three bytes, and by the number of bits defining the range in the most significant byte (e.g., the range 020200h .. 02027Fh is expressed as ‘7 || 020200h’).

17.8.2.5 RHello Interval Descriptor

The format of the RHello Interval descriptor is shown in table 257.

Table 257 – RHello Interval Descriptor Format

Item	Size (Bytes)
Tag Value = 0014h	4
Length = 4	4
RHello_Interval	4

RHello_Interval: contains the RHello_Interval value expressed in ms.

17.8.3 Redundancy Protocol SW_ILSs

17.8.3.1 Exchange Redundancy Parameters (ERP)

The Exchange Redundancy Parameter (ERP) SW_ILS is used by the redundancy protocol to determine which Controlling Switch behaves as Primary and which one behaves as Secondary.

ERP Request Sequence

Addressing: the S_ID field shall be set to FFFFFDh, indicating the originating VE_Port, and the D_ID field shall be set to FFFFFDh, indicating the destination VE_Port.

Payload: The format of the ERP Request Sequence Payload is shown in table 258.

Table 258 – ERP Request Payload

Item	Size (bytes)
SW_ILS Code = YY00 0001h	4
Destination Controlling Switch Switch_Name	8
Originating Controlling Switch Switch_Name	8
Descriptor List Length	4
Controlling Switch State Descriptor	see 17.8.2.2

Destination Controlling Switch Switch_Name: contains the Switch_Name of the destination Controlling Switch.

Originating Controlling Switch Switch_Name: contains the Switch_Name of the requesting Controlling Switch.

Descriptor List Length: contains the length in bytes of the subsequent list of descriptors.

Controlling Switch State Descriptor: see 17.8.2.2.

ERP Reply Sequence

SW_RJT: indicates the rejection of the ERP Request Sequence.

SW_ACC: indicates the acceptance of the ERP Request Sequence. The format of the ERP SW_ACC Payload is shown in table 259.

Table 259 – ERP SW_ACC Payload

Item	Size (bytes)
SW_ILS Code = 0200 0000h	4
Destination Controlling Switch Switch_Name	8
Originating Controlling Switch Switch_Name	8
Descriptor List Length	4
Controlling Switch State Descriptor	see 17.8.2.2

17.8.3.2 Get FCDF Topology State (GFTS)

The Get FCDF Topology State (GFTS) SW_ILS is used by the Secondary Controlling Switch to request to the Primary the Virtual Domain_ID value(s) and the current FCDF topology, in order to synchronize its state with the one of the Primary.

GFTS Request Sequence

Addressing: the S_ID field shall be set to FFFFFDh, indicating the originating VE_Port, and the D_ID field shall be set to FFFFFDh, indicating the destination VE_Port.

Payload: The format of the GFTS Request Sequence Payload is shown in table 260.

Table 260 – GFTS Request Payload

Item	Size (bytes)
SW_ILS Code = YY00 0002h	4
Destination Controlling Switch Switch_Name	8
Originating Controlling Switch Switch_Name	8
Descriptor List Length = 0000 0000h	4

Destination Controlling Switch Switch_Name: contains the Switch_Name of the destination Controlling Switch.

Originating Controlling Switch Switch_Name: contains the Switch_Name of the originating Controlling Switch.

Descriptor List Length: contains the length in bytes of the subsequent list of descriptors.

GFTS Reply Sequence

SW_RJT: indicates the rejection of the GFTS Request Sequence.

SW_ACC: indicates the acceptance of the GFTS Request Sequence. The format of the GFTS SW_ACC Payload is shown in table 261.

Table 261 – GFTS SW_ACC Payload

Item	Size (bytes)
SW_ILS Code = 0200 0000h	4
Destination Controlling Switch Switch_Name	8
Originating Controlling Switch Switch_Name	8
Descriptor List Length	4
FCDF Topology Descriptor	see 17.8.2.3

FCDF Topology Descriptor: see 17.8.2.3.

17.8.3.3 Get FCDF N_Port_IDs State (GFNS)

The Get FDCF N_Port_IDs State (GFNS) SW_ILS is used by the Secondary Controlling Switch to request to the Primary the current allocation of N_Port_IDs to each FCDF of the Distributed Switch, in order to synchronize its state with the one of the Primary.

GFNS Request Sequence

Addressing: the S_ID field shall be set to FFFFFDh, indicating the originating VE_Port, and the D_ID field shall be set to FFFFFDh, indicating the destination VE_Port.

Payload: The format of the GFNS Request Sequence Payload is shown in table 262.

Table 262 – GFNS Request Payload

Item	Size (bytes)
SW_ILS Code = YY00 0003h	4
Destination Controlling Switch Switch_Name	8
Originating Controlling Switch Switch_Name	8
Descriptor List Length = 0000 0000h	4

Destination Controlling Switch Switch_Name: contains the Switch_Name of the destination Controlling Switch.

Originating Controlling Switch Switch_Name: contains the Switch_Name of the originating Controlling Switch.

Descriptor List Length: contains the length in bytes of the subsequent list of descriptors.

GFNS Reply Sequence

SW_RJT: indicates the rejection of the GFNS Request Sequence.

SW_ACC: indicates the acceptance of the GFNS Request Sequence. The format of the GFNS SW_ACC Payload is shown in table 263.

Table 263 – GFNS SW_ACC Payload

Item	Size (bytes)
SW_ILS Code = 0200 0000h	4
Destination Controlling Switch Switch_Name	8
Originating Controlling Switch Switch_Name	8
Descriptor List Length	4
FCDF N_Port_IDs Descriptor	see 17.8.2.4

FCDF N_Port_IDs Descriptor: see 17.8.2.4.

17.8.3.4 Secondary Synchronization Achieved (SSA)

The Secondary Synchronization Achieved (SSA) SW_ILS is used by the Secondary Controlling Switch to communicate to the Primary that it achieved state synchronization.

SSA Request Sequence

Addressing: the S_ID field shall be set to FFFFFDh, indicating the originating VE_Port, and the D_ID field shall be set to FFFFFDh, indicating the destination VE_Port.

Payload: The format of the SSA Request Sequence Payload is shown in table 264.

Table 264 – SSA Request Payload

Item	Size (bytes)
SW_ILS Code = YY00 0004h	4
Destination Controlling Switch Switch_Name	8
Originating Controlling Switch Switch_Name	8
Descriptor List Length = 0000 0000h	4

Destination Controlling Switch Switch_Name: contains the Switch_Name of the destination Controlling Switch.

Originating Controlling Switch Switch_Name: contains the Switch_Name of the originating Controlling Switch.

Descriptor List Length: contains the length in bytes of the subsequent list of descriptors.

SSA Reply Sequence

SW_RJT: indicates the rejection of the SSA Request Sequence.

SW_ACC: indicates the acceptance of the SSA Request Sequence. The format of the SSA SW_ACC Payload is shown in table 265.

Table 265 – SSA SW_ACC Payload

Item	Size (bytes)
SW_ILS Code = 0200 0000h	4
Destination Controlling Switch Switch_Name	8
Originating Controlling Switch Switch_Name	8
Descriptor List Length = 0000 0000h	4

17.8.3.5 Redundancy Hello (RHello)

The Redundancy Hello (RHello) SW_ILS is used by the redundancy protocol. The Rhello SW_ILS is transmitted in a unidirectional Exchange (i.e., it does not have a Reply Sequence).

RHello Request Sequence

Addressing: when used over an AISL, the S_ID field shall be set to FFFFFDh, indicating the originating VE_Port, and the D_ID field shall be set to FFFFFDh, indicating the destination VE_Port. When used over an ASL, the S_ID field shall be set to FFFFF9h, indicating the originating VA_Port, and the D_ID field shall be set to FFFFF9h, indicating the destination VA_Port.

Payload: The format of the RHello Request Sequence Payload is shown in table 266.

Table 266 – RHello Request Payload

Item	Size (bytes)
SW_ILS Code = YY00 0005h	4
Destination Controlling Switch Switch_Name	8
Originating Controlling Switch Switch_Name	8
Descriptor List Length	4
RHello Interval Descriptor	see 17.8.2.5

Destination Controlling Switch Switch_Name: contains the Switch_Name of the destination Controlling Switch.

Originating Controlling Switch Switch_Name: contains the Switch_Name of the originating Controlling Switch.

Descriptor List Length: contains the length in bytes of the subsequent list of descriptors.

RHello_Interval: see 17.8.2.5.

17.8.4 Redundancy Protocol Timeouts

Table 267 shows the timeouts associated to each Redundancy Protocol SW_ILS.

Table 267 – Redundancy Protocol SW_ILSs Timeouts

Description	Abbreviation	Timeout
Exchange Redundancy Parameter	ERP	1000 ms
Get FCDF Topology State	GFTS	1000 ms
Get FDCF N_Port_IDs State	GFNS	1000 ms
Secondary Synchronization Achieved	SSA	1000 ms

17.9 Distributed Switch Operations

17.9.1 Overview

In a Distributed Switch, the Primary Controlling Switch defines the routes for the FCDF topology and performs N_Port_ID allocations and deallocations for all its controlled FCDFs. When two Controlling Switches are present in a Distributed Switch, the two Controlling Switches keep their state synchronized.

17.9.2 FCDF Routing

When becoming operational (i.e., when in state P2 or S2 of of the Controlling Switch Redundancy Protocol, see 17.8), a Controlling Switch instantiates ASLs with the FCDFs that are directly reachable and are part of its FCDF Set.

Upon instantiating an ASL with an FCDF, the Primary Controlling Switch shall initiate an FDRN Exchange describing that link with the Secondary Controlling Switch, if available, to keep the state synchronized. Upon completion of this FDRN Exchange, the Primary Controlling Switch shall provide to that FCDF the Distributed Switch Membership information through a DFMD Exchange. At this point the instantiated ASL becomes part of the Distributed Switch internal topology (i.e., the set of ASLs internal to the Distributed Switch). The Primary Controlling Switch shall recompute the N_Port_ID routes and distribute them to each FCDF belonging to the Distributed Switch through NPRD Exchanges.

Upon deinstantiating an ASL with an FCDF, the Primary Controlling Switch shall initiate an FDUN Exchange describing that disappeared link with the Secondary Controlling Switch, if available, to keep the state synchronized. Upon completion of this FDUN Exchange, the Primary Controlling Switch shall recompute the N_Port_ID routes and distribute them to each FCDF belonging to the Distributed Switch through NPRD Exchanges.

When becoming operational, an FCDF waits for a Controlling Switch or another FCDF to initiate an ELP Exchange with it, in order to set up a ASL. Upon completing the DFMD Exchange with the Primary Controlling Switch, the FCDF becomes able to initiate ELP Requests to instantiate other ASLs with other FCDFs. Upon completing the NPRD Exchange with the Primary Controlling Switch, an FCDF becomes able to set up proper forwarding tables to forward FC frames inside and outside the Distributed Switch. At this point the FCDF enables its ports for logins from Nodes; any FLOGI received on a FCDF port before this point is responded by the FCDF with a LS_RJT having reason code ‘Logical busy’ and reason code explanation ‘No additional explanation’.

Upon instantiating a ASL with another FCDF or with the Secondary Controlling Switch, an FCDF shall perform a FDRN Exchange with the Primary Controlling Switch to inform it of the new link. Upon completing a FDRN Exchange with an FCDF, the Primary Controlling Switch shall initiate another FDRN Exchange with the same parameters with the Secondary Controlling Switch, if available, to keep the state synchronized. After completing this FDRN Exchange the primary Controlling Switch shall provide to the newly reported FCDF the Distributed Switch Membership information through a DFMD Exchange, if that FCDF did not not already receive a DFMD Exchange in a previous step. At this point the instantiated ASL becomes part of the Distributed Switch internal topology (i.e., the set of ASLs internal to the Distributed Switch). Upon completion of this DFMD Exchange, the Primary Controlling Switch shall recompute the N_Port_ID routes and distribute them to each FCDF belonging to the Distributed Switch through NPRD Exchanges.

NOTE 50 – An ASL with the Secondary Controlling Switch may be instantiated before the ASL with the Primary Controlling Switch. The FCDF recognizes the Primary Controlling Switch because it is the one from which it receives the DFMD Request. In this case, the FCDF initiates with the Primary Controlling Switch the FDRN Exchange describing the link with the Secondary Controlling Switch upon completing the DFMD Exchange.

Upon deinstantiating an ASL with another FCDF or with the Secondary Controlling Switch, an FCDF shall perform a FDUN Exchange with the Primary Controlling Switch to inform it of the disappeared link. Upon completing a FDUN Exchange with an FCDF, the Primary Controlling Switch shall initiate another FDUN Exchange with the same parameters with the Secondary Controlling Switch, if available, to keep the state synchronized. Upon completion of this FDUN Exchange, the Primary Controlling Switch shall recompute the N_Port_ID routes and distribute them to each FCDF belonging to the Distributed Switch through NPRD Exchanges.

17.9.3 N_Port_ID Handling

Upon receiving on a port a FLOGI Request or a NPIV FDISC Request from a Node, an FCDF shall send a VNRN Request to the Primary Controlling Switch to inform it of the newly reachable VN_Port. If the Primary Controlling Switch rejects the VNRN Request, the FCDF shall also reject the FLOGI

Request or NPIV FDISC Request. Upon receiving the VNRN Request, the Primary Controlling Switch performs the following processing:

a) if the VNRN Request carried a FLOGI Request and that VN_Port was not already logged in or if the VNRN Request carried a NPIV FDISC Request, then the Primary Controlling Switch shall allocate to the newly reachable VN_Port an N_Port_ID from a Virtual Domain_ID; or

b) if the VNRN Request carried a FLOGI Request and that VN_Port was already logged in, then the Primary Controlling Switch shall implicitly log out that VN_Port and all the VN_Ports associated to the VF_Port that VN_Port was associated with and then allocate to that VN_Port an N_Port_ID from a Virtual Domain_ID.

The Primary Controlling Switch shall also recompute the Zoning ACLs for the affected N_Port_IDs, generate appropriate RSCN(s), and update the Fibre Channel Name Server. The Primary Controlling Switch shall distribute the Zoning ACLs and N_Port_ID allocation/deallocation information to the Secondary Controlling Switch, if available, and to each FCDF belonging to the Distributed Switch through an appropriate NPZD Exchange. The NPZD Request sent to the Secondary Controlling Switch shall carry no Peering Entries. The NPZD Requests sent to the Secondary Controlling Switch shall include the FLOGI / NPIV FDISC LS_ACC Parameters; the NPZD Requests sent to the FCDFs shall not include them. Upon receiving the NPZD SW_ACC from the Secondary Controlling Switch and from the FCDF that sent the VNRN Request, the Primary Controlling Switch shall send the VNRN SW_ACC to the FCDF that sent the VNRN Request. Upon receiving the VNRN SW_ACC, containing the FLOGI / NPIV FDISC LS_ACC Parameters, the FCDF that sent the VNRN Request shall accept the FLOGI Request or FIP NPIV FDISC Request and complete the N_Port login.

When a VN_Port is logged out or when a VF_Port is deinstantiated, an FCDF shall perform a VNUN Exchange with the Primary Controlling Switch to inform it that the VN_Port is now unreachable or that all the VN_Ports associated with that VF_Port are unreachable. Upon completing a VNUN Exchange, the Primary Controlling Switch shall deallocate the N_Port_ID(s) assigned to the affected VN_Port(s), recompute the Zoning ACLs for the affected N_Port_IDs, generate appropriate RSCN(s), and update the Fibre Channel Name Server. The Primary Controlling Switch shall then distribute this information to the Secondary Controlling Switch, if available, and to each FCDF belonging to the Distributed Switch through NPZD Requests indicating N_Port_ID(s) deallocation.

When a new Zone Set is activated in the Fabric, the Primary Controlling Switch shall recompute the Zoning ACLs for all N_Port_IDs allocated in the Virtual Domain and distribute them to the FCDFs of the Distributed Switch through AZAD Exchanges.

Upon receiving on a port a FLOGI Request or a NPIV FDISC Request from a Node, a Controlling Switch shall allocate to the newly reachable VN_Port an N_Port_ID from the Principal Domain_ID if it accepts the received FLOGI or NPIV FDISC Request.

17.10 Distributed Switch Redundancy Protocol

17.10.1 Redundancy Protocol Overview

The purpose of the Controlling Switch Redundancy protocol is to avoid any single point of failure in a Distributed Switch. Figure 50 shows an example of redundant Distributed Switch, including the two Principal Domains and the Virtual Domain.

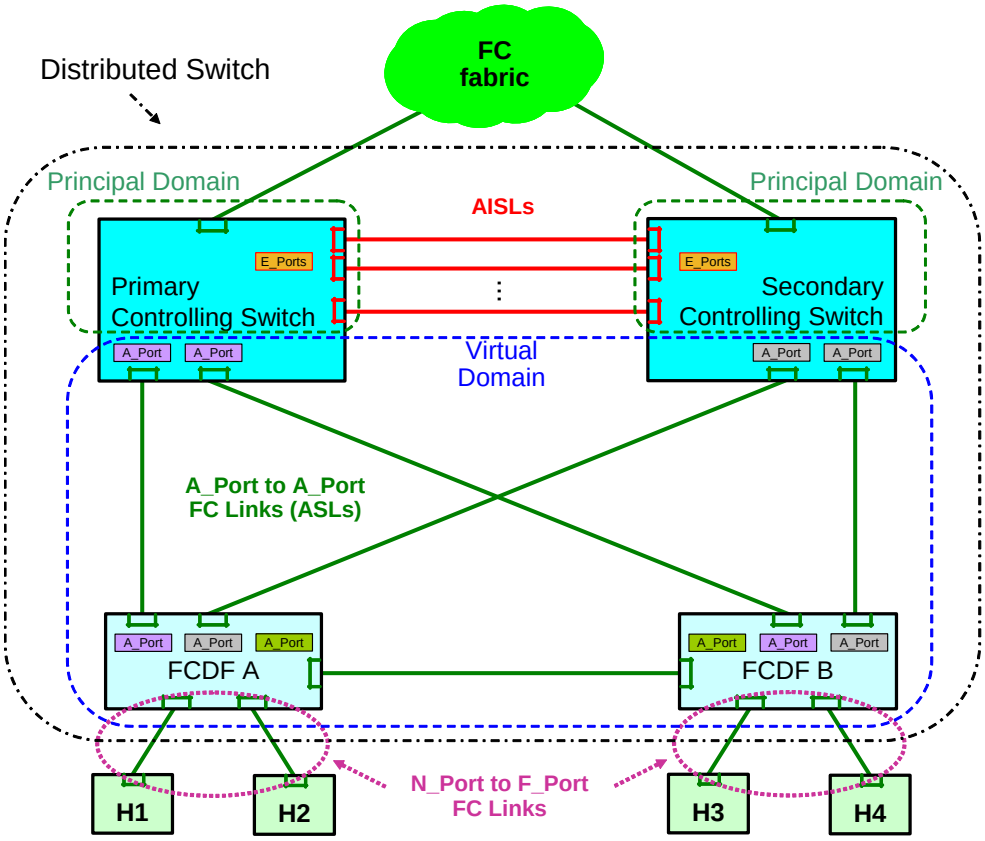

Figure 50 – Example of Redundant Distributed Switch

The Controlling Switch Redundancy protocol uses a set of Augmented E_Port to E_Port links (AISLs) between the Primary and Secondary Controlling Switches. This set is referred to as the AISL Set. It is strongly recommended to deploy at least two AISLs in the AISL Set, in order to distinguish the case of an AISL failure from the case of a Controlling Switch failure. Additional AISLs provide additional resiliency.

In a Redundant Distributed Switch the Primary Controlling Switch generates the LSR(s) describing the Virtual Domain in the Distributed Switch. In addition, both Primary and Secondary Controlling Switch list the Virtual Domain as a directly attached Domain in their LSR. The resulting FSPF topology is depicted in figure 51, where Z1 .. Zn are the Domain_IDs belonging to the Virtual Domain and X

and Y are the Domain_IDs of the Principal Domains of the two Controlling Switches. X and Y are also connected between themselves by virtue of the AISLs.

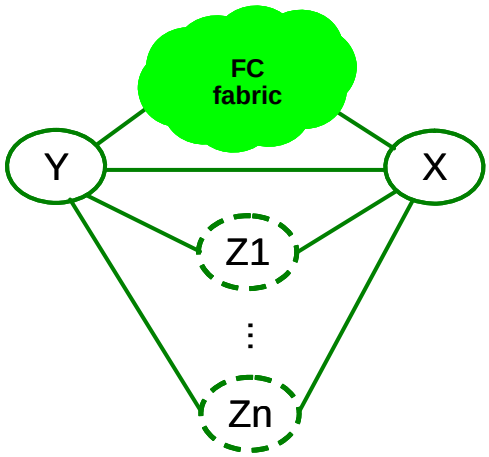

Figure 51 – Distributed Switch FSPF Topology

17.10.2 Redundancy Protocol State Machine

The redundancy protocol state machine reacts to AISLs failures in a timed fashion. To this end, the redundancy protocol state machine uses indications from the physical layer to determine if a link failed together with periodic Redundancy Hello messages (RHello) to verify the health of the Controlling Switches. The redundancy protocol state machine uses the following time intervals and timers:

RHello_Interval: Time interval between RHellos, expressed in milliseconds. The default value is 200 ms.

Down_Interval: Time interval for a Controlling Switch to declare the other one down. Calculated as 2.5 * RHello_Interval.

To determine which Controlling Switch behaves as Primary and which one as Secondary, the redundancy protocol uses a Priority value associated to each Controlling Switch. Priority values are shown in table 268.

Table 268 – Controlling Switch Priority Values

Value	Description
00h	Reserved
01h	Highest Priority value. This value is administratively configured to force the election of a Controlling Switch to Primary.
02h[a]	Primary Controlling Switch priority. This value is used by the Redundancy protocol to identify a Controlling Switch as Primary.
03 .. FEh	Higher to lower Priority values. The default value is 128.
FFh[a]	This value indicates that a Controlling Switch is not willing to operate as Primary. This is used by the Primary Controlling Switch to trigger a transition of the Secondary Controlling Switch to Primary without having to wait for the current Primary to timeout, if appropriate.
[a] These values are used by the Redundancy protocol and not available to an administrator.	

Figure 52 shows the redundancy protocol state machine.

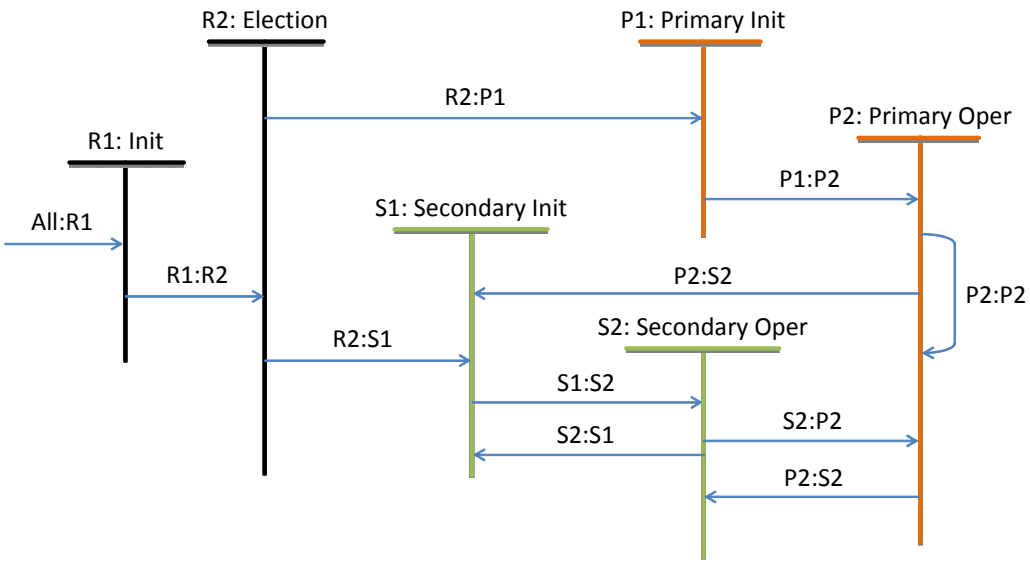

Figure 52 – Redundancy Protocol State Machine

State R1:Init. In this state a Controlling Switch clears its state and waits to begin the processing for the redundancy protocol.

Transition R1:R2. Occurs when processing for the redundancy protocol begins. The redundancy protocol processing begins when:

a) the redundancy protocol is enabled;

b) the Controlling Switch Set and the FCDF Set are configured; and

c) Fabric configuration is completed.

Transition All:R1. Occurs when the redundancy protocol is disabled.

State R2:Election. In this state a Controlling Switch determines if it operates as Primary or Secondary. If the AISL Set is NULL, then the Controlling Switch exits this state. If the AISL Set is not NULL, then an ERP Exchange is performed.

NOTE 51 – In this state the ERP payload does not contain N_Port_ID Ranges, given that the Controlling Switch cleared its state in state R1.

If the ERP Exchange shows that the local Controlling Switch Priority is 01h and the remote Controlling Switch Priority is 01h (i.e., both Controlling Switches are manually configured to be Primary) then the Redundancy protocol is disabled and an error is logged.

Transition R2:P1. Occurs when:

a) the AISL Set is NULL;

b) the AISL Set is not NULL and the ERP Exchange showed that the local Controlling Switch Priority is lower than the remote Controlling Switch Priority; or

c) the AISL Set is not NULL, the ERP Exchange showed that the local Controlling Switch Priority is equal to the remote Controlling Switch Priority, and the local Switch_Name is lower than the remote Switch_Name.

Transition R2:S1. Occurs when:

a) the AISL Set is not NULL and the ERP Exchange showed that the local Controlling Switch Priority is higher than the remote Controlling Switch Priority; or

b) the AISL Set is not NULL, the ERP Exchange showed that the local Controlling Switch Priority is equal to the remote Controlling Switch Priority, and the local Switch_Name is greater than the remote Switch_Name.

State P1:Primary Initialization. In this state a Controlling Switch performs the operations to become the Primary Controlling Switch of the Distributed Switch. To this end the Controlling Switch sets its Priority to 02h and obtains an additional Domain_ID value (a Virtual Domain_ID) from the Principal Switch of the fabric by generating an RDI Request on behalf of the Virtual Domain Switch_Name.

Transition P1:P2. Occurs when the Virtual Domain_ID is available.

State P2:Primary Operational. In this state the Controlling Switch is operational as Primary. On entering this state the Controlling Switch:

a) sets its Priority to 02h;

b) initiates an ERP Exchange with the Secondary Controlling Switch, if available;

c) sends a DFMD SW_ILS to all reachable FCDF of the FCDF Set declaring itself as Primary Controlling Switch;

d) on native Fibre Channel links that were Isolated because connected to FCDFs, if any, it performs an ELP; and

e) on FCoE interfaces, it establishes VA_Port to VA_Port Virtual Links with neighbor FDFs belonging to the FDF Set to which no VA_Port to VA_Port Virtual Links has been established, if any.

While in this state, the Controlling Switch:

a) performs the Distributed Switch operations (see 17.9);

b) generates the FSPF LSR(s) describing the Virtual Domain(s) in the Distributed Switch and lists the Virtual Domain(s) as a directly attached Domain(s) in its FSPF LSR;

c) on receiving an SSA SW_ILS (i.e., when the Secondary Controlling Switch completed its state synchronization) sends a DFMD SW_ILS to all reachable FCDFs of the FCDF Set declaring itself as Primary and the Secondary as Secondary;

d) if the Secondary Controlling Switch is available sends RHello Requests every RHello_Interval over each of its AISLs and over each ASL through which the Secondary is reachable;

e) resets the Down_Timer to Down_Interval everytime an RHello Request from the Secondary Controlling Switch is received over at least one AISL or ASL;

f) when the Secondary Controlling Switch is not anymore available (i.e., when Down_Timer expires) sends a DFMD SW_ILS to all reachable FCDFs of the FCDF Set declaring itself as Primary; and

g) if the AISL Set goes from NULL to not-NULL (i.e., a Controlling Switch becomes available), it performs an ERP Exchange with the other Controlling Switch.

Transition P2:P2: Occurs following the ERP Exchange performed when the AISL Set went from NULL to not-NULL if:

a) there is no allocated N_Port_IDs conflict between the two Controlling Switches and this Switch is the one selected to remain Primary; or

b) there is an allocated N_Port_IDs conflict between the two Controlling Switches. In this case the AISL shall be Isolated.

Transition P2:S2: Occurs following the ERP Exchange performed when the AISL Set went from NULL to not-NULL if there is no allocated N_Port_IDs conflict between the two Controlling Switches and this Switch is the one selected to become Secondary.

State S1:Secondary Initialization. In this state a Controlling Switch waits for at least an AISL to be available. When an AISL is available it performs the operations to become the Secondary Controlling Switch of the Distributed Switch. The Controlling Switch has to synchronize its state with the one of the Primary Controlling Switch. To this end the Controlling Switch:

1) Requests to the Primary the FCDF topology through the GTFS (Get FCDF Topology State) SW_ILS;

2) Requests to the Primary the Virtual Domain_IDs and N_Port_IDs Allocation state in the Distributed Switch through the GFNS (Get FDCF N_Port_IDs State) SW_ILS;

3) Obtains the information associated with each N_Port_ID in the Name Server through the GE_ID CT Request; and

4) Communicates the achieved state synchronization to the Primary through the SSA (Secondary Synchronization Achieved) SW_ILS.

While in this state the Controlling Switch:

a) processes FDUN, FDRN, and NPZD Requests coming over the AISL from the Primary Controlling Switch, if at least an AISL is available;

b) sends RHello Requests every RHello_Interval over each of its AISLs and over each ASL through which the Primary is reachable; and

c) resets the Down_Timer to Down_Interval everytime an RHello Request is received over at least one AISL or ASL.

Transition S1:S2. Occurs when the Secondary Controlling Switch has synchronized its state with the Primary.

State S2:Secondary Operational. In this state the Controlling Switch is operational as Secondary. On entering this state the Controlling Switch:

a) sets its Priority to its configured value;

b) initiates an ERP Exchange with the Primary Controlling Switch;

c) on native Fibre Channel links that were Isolated because connected to FCDFs, if any, it performs an ELP; and

d) on FCoE interfaces, it establishes VA_Port to VA_Port Virtual Links with neighbor FDFs belonging to the FDF Set to which no VA_Port to VA_Port Virtual Links has been established, if any.

While in this state, the Secondary Controlling Switch:

a) participates in the Distributed Switch operations (see 17.9);

b) lists the Virtual Domain(s) as a directly attached Domain(s) in its FSPF LSR;

c) sends RHello Requests every RHello_Interval over each of its AISLs and over each ASL through which the Primary is reachable; and

d) resets the Down_Timer to Down_Interval everytime an RHello Request is received over at least one AISL or ASL.

Transition S2:S1. Occurs when when all AISLs are down and Down_Timer is not expired (i.e., the Secondary Controlling Switch is receiving RHello Requests from ASLs. This indicates that the Primary Controlling Switch is still operational, although not reachable through the AISLs).

Transition S2:P2. Occurs when the Secondary Controlling Switch becomes Primary. This occurs when:

a) the Primary Controlling Switch is not anymore available (i.e., when Down_Timer expires); or

b) the Priority field in a received ERP Request has a value of FFh. This is an indication that the Primary Controlling Switch determined to become Secondary.

Transition P2:S2. Occurs when the Primary Controlling Switch determines to become Secondary by setting its Priority to FFh. This may happen as result of an administrative action.

Annex A
(informative)

Examples of Switch Port Initialization

A.1 Introduction

This annex presents some example scenarios that may occur during Switch Port Initialization (see 7.2). It is expected that the reader is familiar with Loop Initialization as defined in FC-AL-2, and with Link Initialization as defined in FC-FS-3. Loop Initialization states as defined in FC-AL-2 are referenced here to facilitate the understanding of the process.

A.2 Example 1: two E/F/FL_Port-capable Switch Ports

In this example, two Switch Ports that are E/F/FL_Port-capable are attached to each other. Figure A.1 illustrates this example.

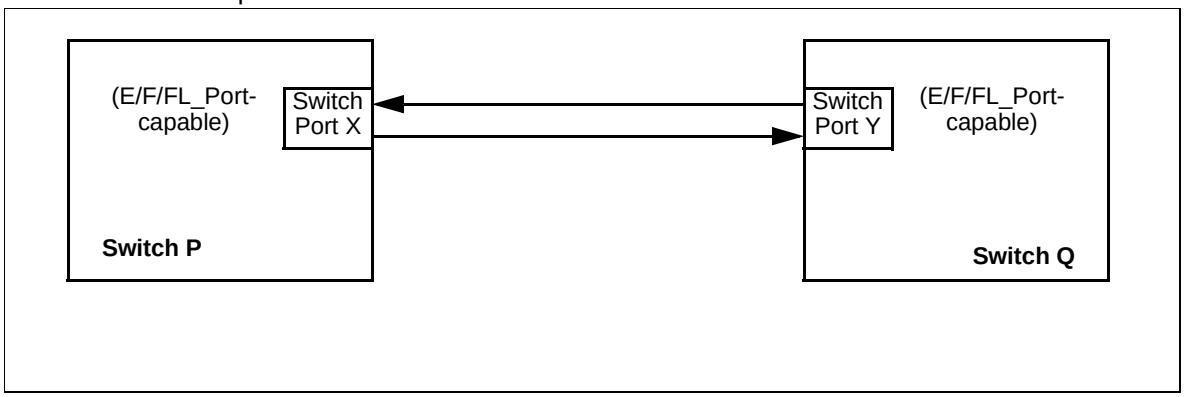

Figure A.1 – Initialization example 1

According to the initialization algorithm, since each Switch Port is E/F/FL_Port-capable, they start the process with Loop Initialization, as defined in FC-AL-2. LIP Primitive Sequences are sent and received, and each Switch Port starts sending LISM frames. When Switch Port X receives LISM from Switch Port Y, it sees that its Port_Name is lower than the Port_Name in the Payload, and continues sending the same LISM.

On the other hand, when Switch Port Y receives LISM from Switch Port X, it sees that its Port_Name is higher than the Port_Name in the Payload. This causes Switch Port Y to start sending the LISM it received, with the Port_Name belonging to Switch Port X. Switch Port Y also transitions to the MONITORING state with PARTICIPATE = FALSE (0), because only one FL_Port may be participating on a loop.

Switch Port X receives its LISM and assumes the role of Loop Master. Switch Port X then proceeds to send all of the other Loop Initialization Sequences, and by the end of Loop Initialization, discovers that it is the only L_Port on the Loop.

Because there may be a Non-Participating Switch Port on the Loop, Switch Port X knows it is required to attempt Link Initialization. Switch Port X begins Link Initialization by REQ(old-port). Switch Port X transitions to the OLD-PORT-REQ state and begins transmitting LIP; this causes Switch Port Y to be-

gin Loop Initialization. Switch Port Y transmits a minimum of 12 of the received LIPs in the OPEN-INIT-START state and transitions to the OPEN-INIT-SELECT-MASTER state. When Switch Port X recognizes LIP, it transitions to the OLD-PORT state and transmits OLS for minimum(2xAL_TIME). After a maximum (1xAL_TIME), Switch Port Y recognizes Primitive Sequences (OLS, NOS) defined in FC-FS-3 and transitions from the FL_Port operating mode to E/F_Port mode. The Link protocol continues to completion and a point-to-point Link is now active.

Switch Port X and Switch Port Y may now attempt to Exchange Link Parameters and establish an Inter-Switch Link.

A.3 Example 2: two E/F/FL_Port-capable Switch Ports and one PN_Port

In this example, two Switch Ports that are E/F/FL_Port-capable are attached to each other as in the first example, but there is also a PN_Port on the loop. Figure A.2 illustrates this example.

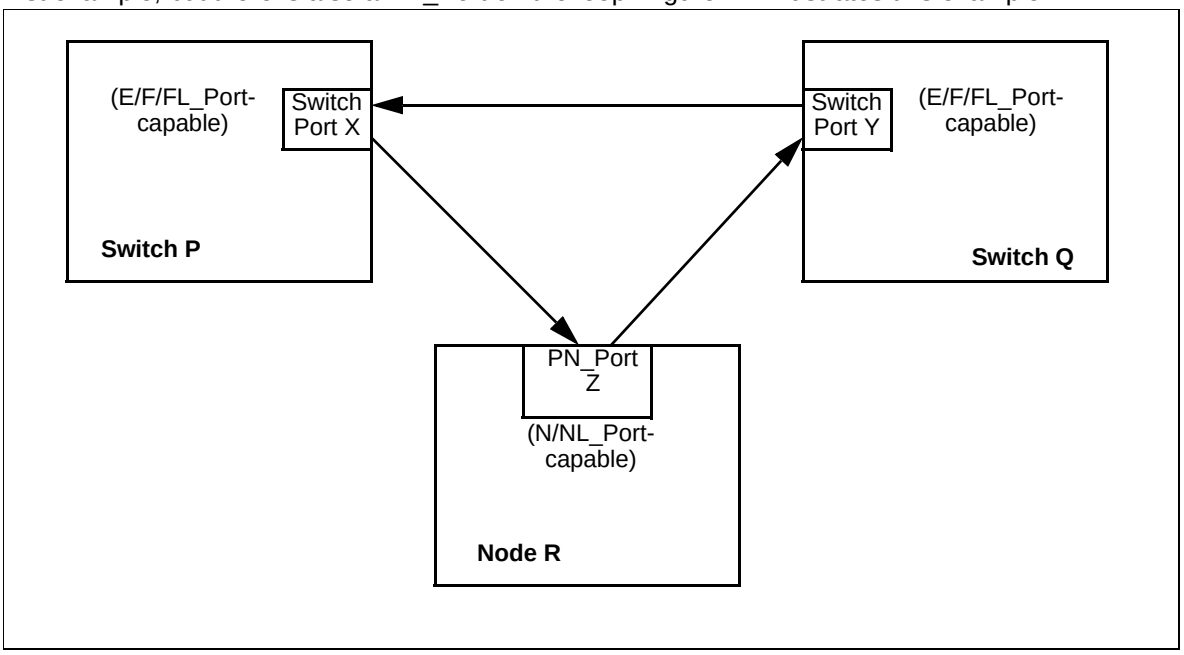

Figure A.2 – Initialization example 2

According to the initialization algorithm, since each Switch Port is E/F/FL_Port-capable and PN_Port Z is N/NL_Port-capable, they start the process with Loop Initialization, as defined in FC-AL-2. LIP Primitive Sequences are sent and recognized, and each Switch Port and the PN_Port start sending LISM frames. As in the first example, Switch Port X receives LISM from Switch Port Y, it sees that its Port_Name is lower than the Port_Name in the Payload, and continues sending the same LISM.

When PN_Port Z receives the LISM from Switch Port X, it finds a D_ID of zero, meaning that the originator is an FL_Port. Since an FL_Port always wins Loop Master, the PN_Port begins sending the received LISM from Switch Port X.

When Switch Port Y receives Switch Port X's LISM from Port Z, it sees that its Port_Name is higher than the Port_Name in the Payload. This causes Switch Port Y to start sending the LISM it received, with the Port_Name belonging to Switch Port X. Switch Port Y also transitions to the MONITORING state with PARTICIPATE = FALSOE (0), because only one FL_Port may be participating on a loop.

Switch Port X receives its LISM and assumes the role of Loop Master. Switch Port X then proceeds to send all of the other Loop Initialization Sequences, and by the end of Loop Initialization, discovers that there is only one other L_Port on the loop. Because that one other port may be capable of point-to-point operation, Switch Port X knows it may attempt Link Initialization.

Switch Port X begins Link Initialization by asserting REQ (old-port) causing the transmission of LIP in the OLD-PORT-REQ state, and causes PN_Port Z to begin Loop Initialization. PN_Port Z transmits a minimum of 12 received LIPs in the OPEN-INITSTART state causing Switch Port Y to begin Loop Initialization and transitions to either the OPEN-INIT-SELECTMASTER or the SLAVE-WAIT-FOR-MASTER state. Switch Port Y transmits a minimum of 12 received LIPs in the OPENINIT-START state and transitions to the OPEN-INIT-SELECT-MASTER state. Switch Port X recognizes LIP, transitions to the OLD-PORT state and transmits OLS for minimum (2xAL_TIME). If after minimum (1xAL_TIME) PN_Port Z recognizes OLS and reacts to it, it transitions to the OLD-PORT state and transmits LR in response. Switch Port Y being in the OPEN-INIT-SELECT-MASTER state does not recognize LR and continues with the INITIALIZATION process thereby blocking LR to Switch Port X. When Switch Port X fails Link Initialization it should remove REQ (old-port) to allow Loop Initialization to complete. When Loop Initialization completes successfully, Switch Port X operates as an FL_Port, and PN_Port Z operates as an L_Port. Switch Port Y remains Non-Participating.

NOTE 52 – If PN_Port Z had been bypassed, the process would have completed as in Example 1, because the Primitive Sequences would have been ignored by PN_Port Z. At a later time, if PN_Port Z is then enabled, Loop Initialization begins (i.e., PN_Port Z starts sending LIP to get an AL_PA), and things sort themselves out as described for Example 2. If Switch Port Y had been bypassed, then Switch Port X would have become an F_Port in a point-to-point Link with PN_Port Z.

NOTE 53 – If PN_Port Z was L_Port capable only when it went to the OPEN-INIT-START state, it would stall in either the OPEN-INITSELECT-MASTER or SLAVE-WAIT-FOR-MASTER state transmitting LISM or waiting for an ARB(F0). This would cause Switch Port X to fail at Link Initialization, and then go back to Loop Initialization. Again, Switch Port Y stays Non-Participating.

A.4 Example 3: one E/F/FL_Port-capable Port and one E/F_Port-capable Port

In this example, a Switch Port that is E/F/FL_Port-capable is attached to a Switch Port that is E/F_Port-capable. Figure A.3 illustrates this example.

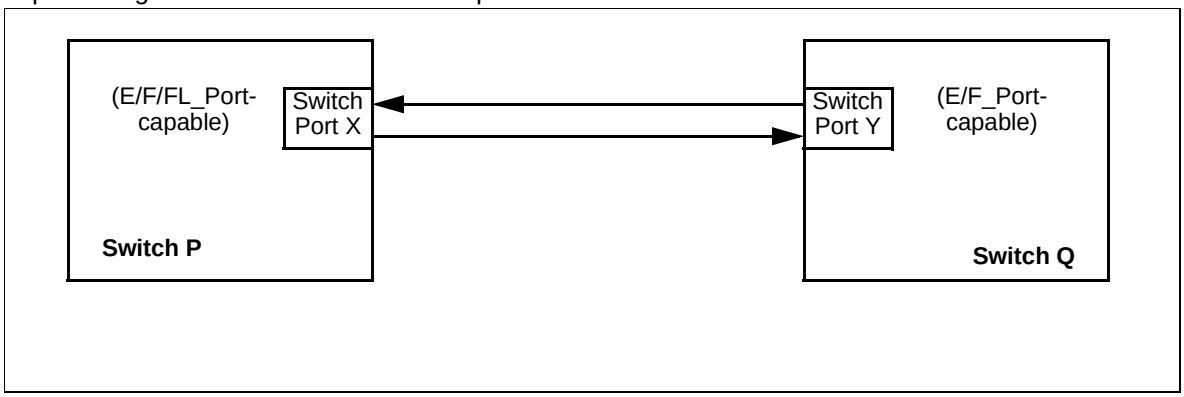

Figure A.3 – Initialization example 3

According to the initialization algorithm, the Switch Port that is E/F/FL_Port-capable starts the process with Loop Initialization as defined in FC-AL-2. However, the Switch Port that is E/F_Port-capable starts the process with Link Initialization as defined in FC-FS-3. Switch Port X sends LIP Primitive Sequences, and Switch Port Y sends OLS Primitive Sequences. If Switch Port X in the NORMAL-INITIALIZE state does not receive LIP before expire(3xAL_TIME), it transitions to the OLD-PORT state and completes Link Initialization.

Switch Port X and Switch Port Y may now attempt to Exchange Link Parameters and establish an Inter-Switch Link.

Annex B
(informative)

ELP Negotiation Example

B.1 Introduction

This annex presents an example of how ELP negotiation may be performed.

B.2 ELP Exchange Protocol

The following description is an extension of the ELP exchange described this standard. It allows for a negotiation of link parameters.

NOTE 54 – In the following discussion related to the ELP Protocol, the Reference Configuration given in figure B.1 is used.

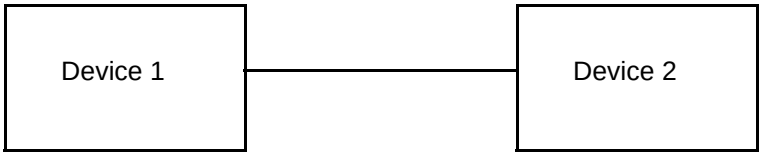

Figure B.1 – Reference ELP Configuration

Following is a summary of the resulting state at each E_Port after the ELP exchange with and without Parameter Negotiation.

B.2.1 ELP Exchange without Parameter Negotiation

– The ELP originating port (Device 1) shall consider the exchange of Link Parameters complete and successful, when it has transmitted an ACK_1 for the SW_ACC it has received.

– The ELP originating port (Device 1) shall consider the exchange of Link Parameters complete but not successful, when it has transmitted an ACK_1 for the SW_RJT it has received. The originating port now goes into isolation.

– The responding port (Device 2) shall consider the exchange of Link Parameters complete and successful, when it has received the ACK_1 for the SW_ACC it has transmitted.

– The responding port (Device 2) shall consider the exchange of Link Parameters complete but not successful, when it has received the ACK_1 for the SW_RJT it has transmitted. The responding port now goes into isolation.

Figures B.2 and B.3 illustrate two complete ELP exchanges between two E_Ports, one successful and the other unsuccessful without negotiation. This is the current ELP operation.

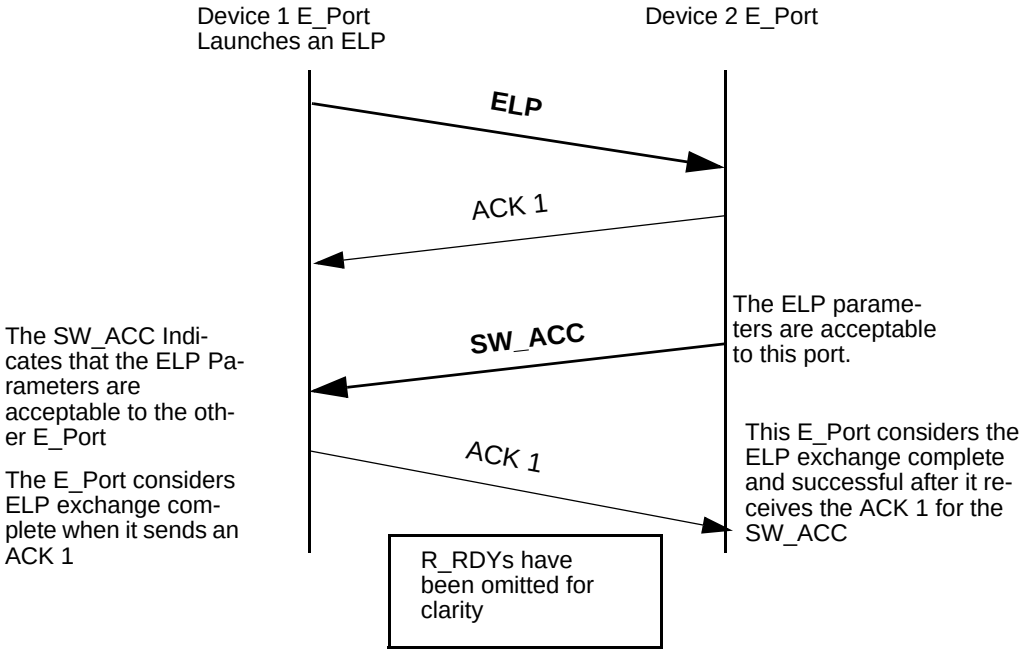

Figure B.2 – A Successful and Complete ELP Exchange

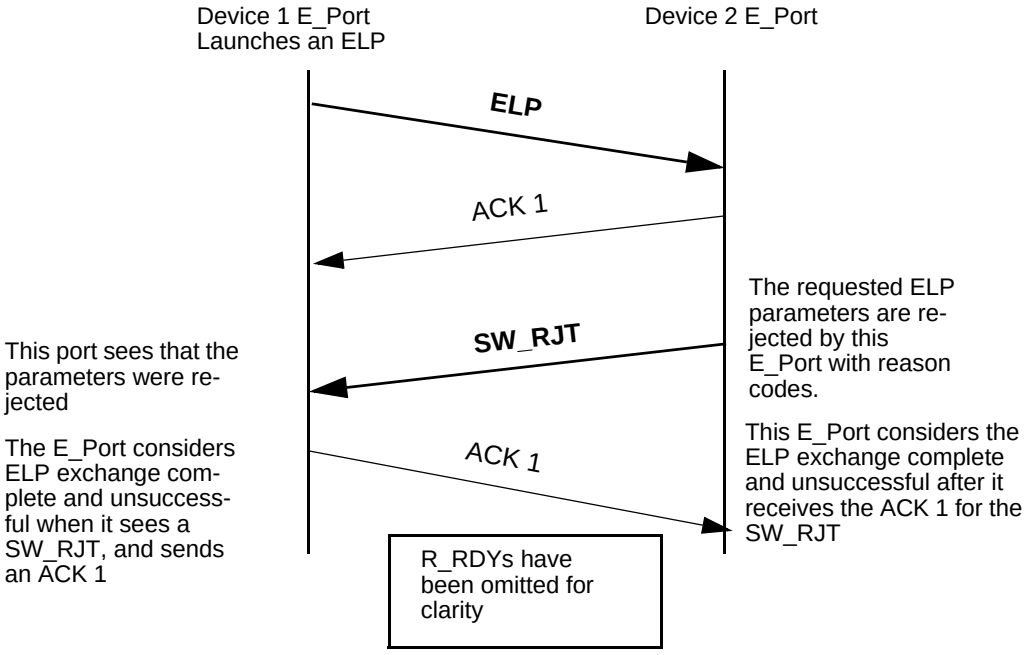

Figure B.3 – An Unsuccessful but Complete ELP Exchange

B.2.2 ELP Exchange with Parameter Negotiation

– The ELP parameter negotiation process always converges with the final reply being a SW_ACC for a successful exchange or a SW_RJT for an unsuccessful exchange.

– If a responding port (Device 2) is unable to agree to the received parameters, it sends out a SW_RJT after which it may issue a new ELP with modified parameters. This responding port now becomes the new ELP originator (Device 2).

– The old originating port (Device 1) after receiving a SW_RJT shall send an ACK_1 and waits for the possible arrival of a new ELP with modified parameters; this device (Device 1) now becomes the new responder. Until a new ELP is received this device is isolated on this link.

– If the new responder (Device 1) finds the parameters acceptable, then it sends out a SW_ACC and waits for an ACK_1.

– If the new responder (Device 1) finds the parameters unacceptable, then it sends out a SW_RJT and waits for an ACK_1 after which it goes into isolation.

–There is a maximum of 2 exchanges of link parameters, after which the ports are either operational or isolated.

Figures B.4 and B.5 illustrates a complete ELP exchanges between two E_Ports with negotiation, one successful and the other unsuccessful with negotiation**:**

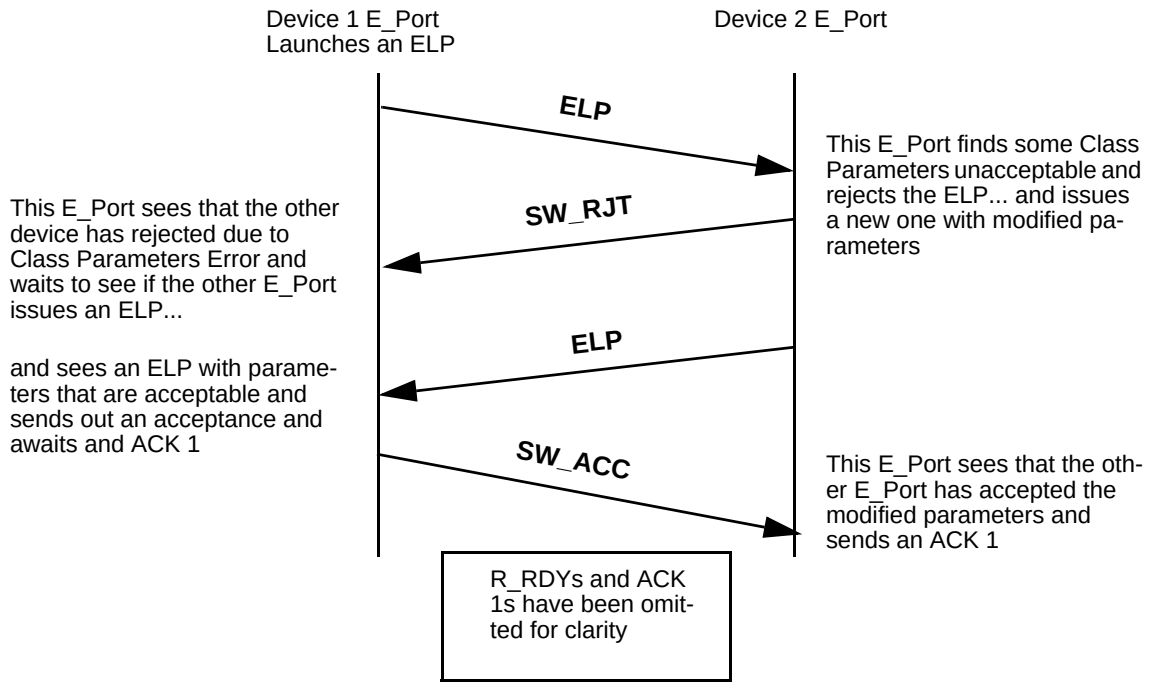

Figure B.4 – A successful ELP Exchange Protocol Parameter Negotiation

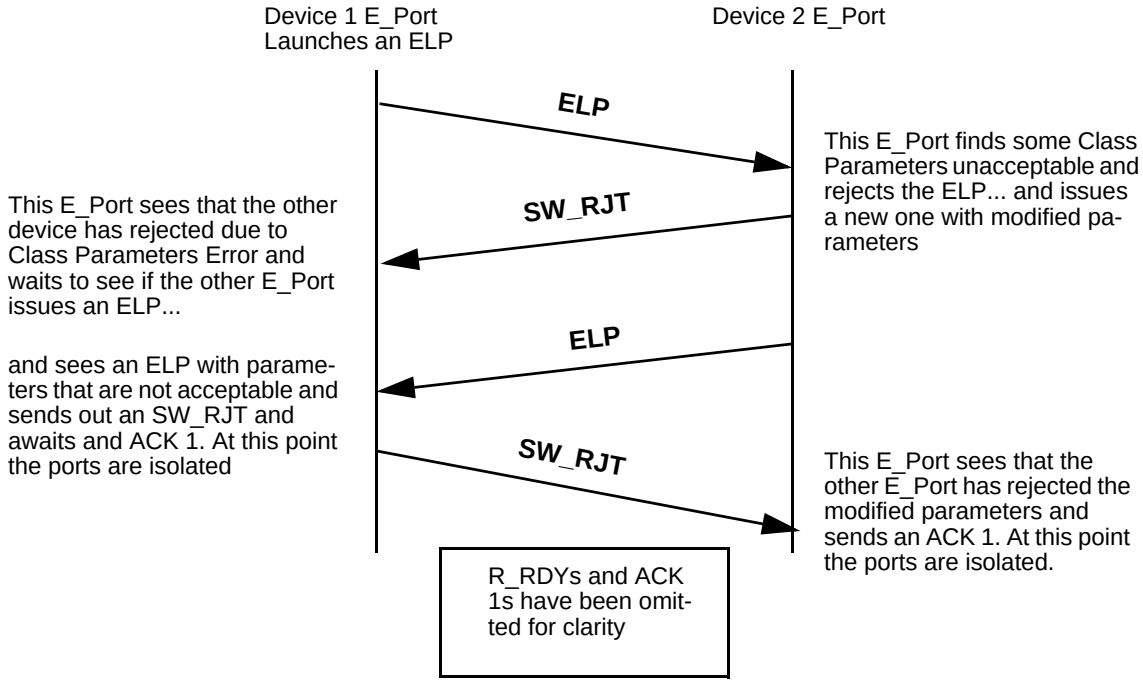

Figure B.5 – An Unsuccessful ELP Exchange Protocol Parameter Negotiation

Annex C
(informative)

Fabric Device Management interface-Sample Flows

C.1 Introduction

This annex presents sample flows for the Fabric Device Management Interface (FDMI).

C.2 Sample Flows

C.2.1 HBA Registration - Single Switch

In Figure C.1 below, the switch interactions are shown for HBA registration to a single switch.

1) The HBA attempts registration by sending an RHBA to Switch 1. Since the HBA has not already registered with Switch 1, the registration completes successfully and Switch 1 becomes the HBA's primary manager.

2) Switch 1 sends a Registration Notification to Switches 2 and 3. Switches 2 and 3 now update their caches such that subsequent queries to Switches 2 and 3 regarding the HBA may be handled locally.

3) Some time later the HBA requests registration by sending an RHBA through another port to Switch 1. This time the registration fails because the HBA is already registered with Switch 1. Since the HBA has not registered with any other switches, Switch 1 becomes the HBA's primary manager.

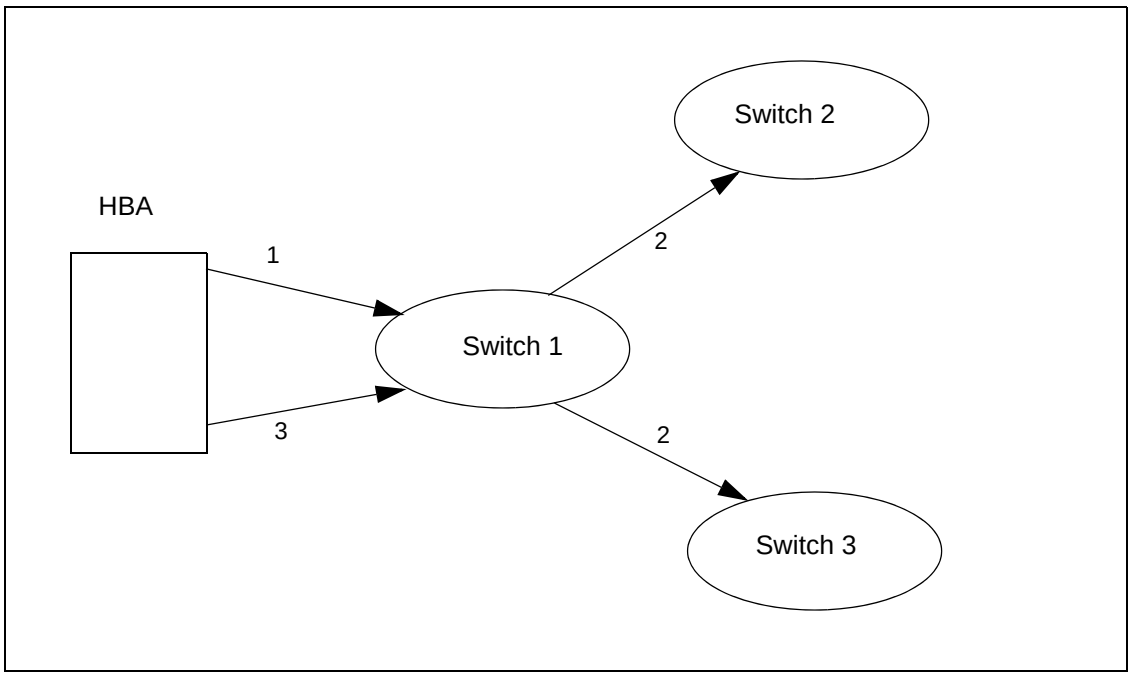

Figure C.1 – Registration of HBA Information - Single Switch

C.2.2 HBA Registration - Multiple Switches - Caches Updated

In Figure C.2 below, the switch interactions are shown for HBA registration to multiple switches when the caches are updated.

1) The HBA attempts registration by sending an RHBA to Switch 1. Since the HBA has not already registered with Switch 1, the registration completes successfully.

2) Switch 1 sends a Registration Notification to Switches 2 and 3. Switches 2 and 3 now update their caches such that subsequent queries to Switches 2 and 3 regarding the HBA may be handled locally.

3) Some time later the HBA requests registration by sending an RHBA through another port to Switch 3. Switch 3 performs the checks against its FDMI database and its cached information. The cached information indicates that the HBA has already been registered in Switch 1 and the registration is rejected. Switch 1 becomes the HBA's primary manager.

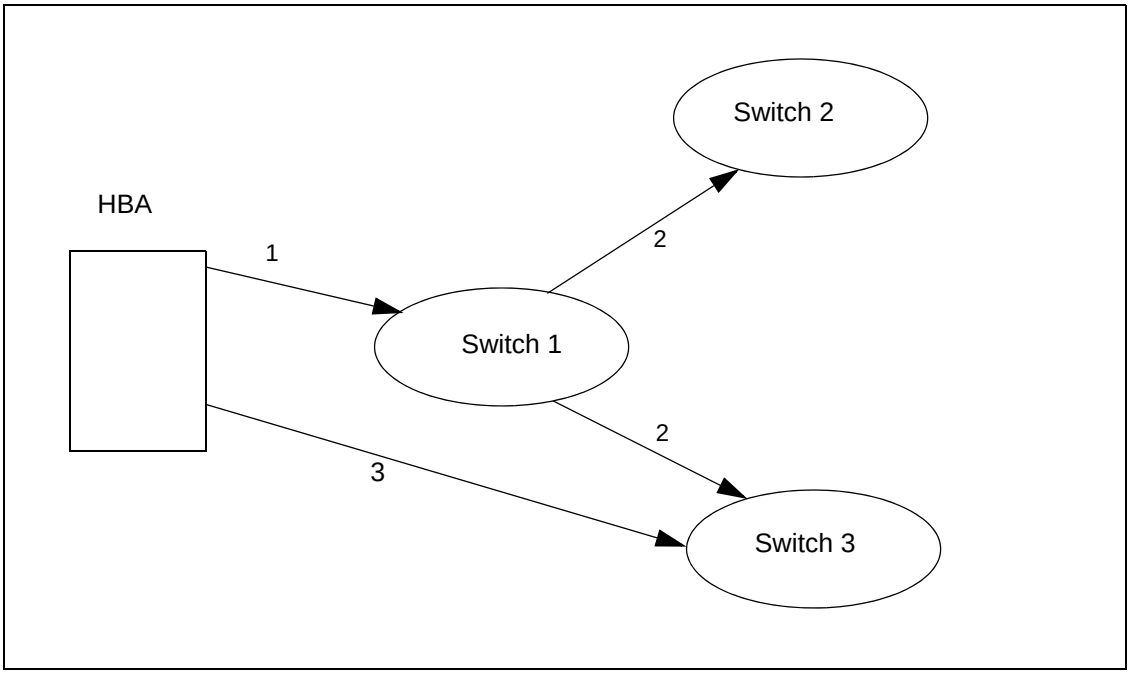

Figure C.2 – Registration of HBA Information - Multiple Switches Caches Updated

C.2.3 HBA Registration - Multiple Switches - Caches Not Updated

In Figure C.3 below, the switch interactions are shown for HBA registration to multiple switches when the caches are not updated.

1) The HBA attempts registration by sending an RHBA to Switch 1. Since the HBA has not already registered with Switch 1, the registration completes successfully.

2) Before Switch 1 sends a Registration Notification to Switches 2 and 3, the HBA requests registration by sending an RHBA through another port to Switch 3. Since Switch 3 does not have an updated cache, Switch 3 assumes that the HBA is not registered and accepts the registration.

3) Switch 1 sends a Registration Notification to Switches 2 and 3.

4) Switch 3 send a Registration Notification to Switches 1 and 2.

In this case Switch 1 has the lowest Switch_Name. Switch 2 receives the Registration Notification from Switches 1 and 3 and notes that Switch 1 has the lower Switch_Name. Switch 2 updates his cache with the information received from Switch 1 and designates Switch 1 has the primary manager for the HBA. Switch 3 receives the Registration Notification from Switch 1 and notes that Switch 1 has the lower Switch_Name. Switch 3 deletes the HBA information from its FDMI database and updates his cache with the information received from Switch 1. Switch 3 designates Switch 1 as the primary manager for the HBA.

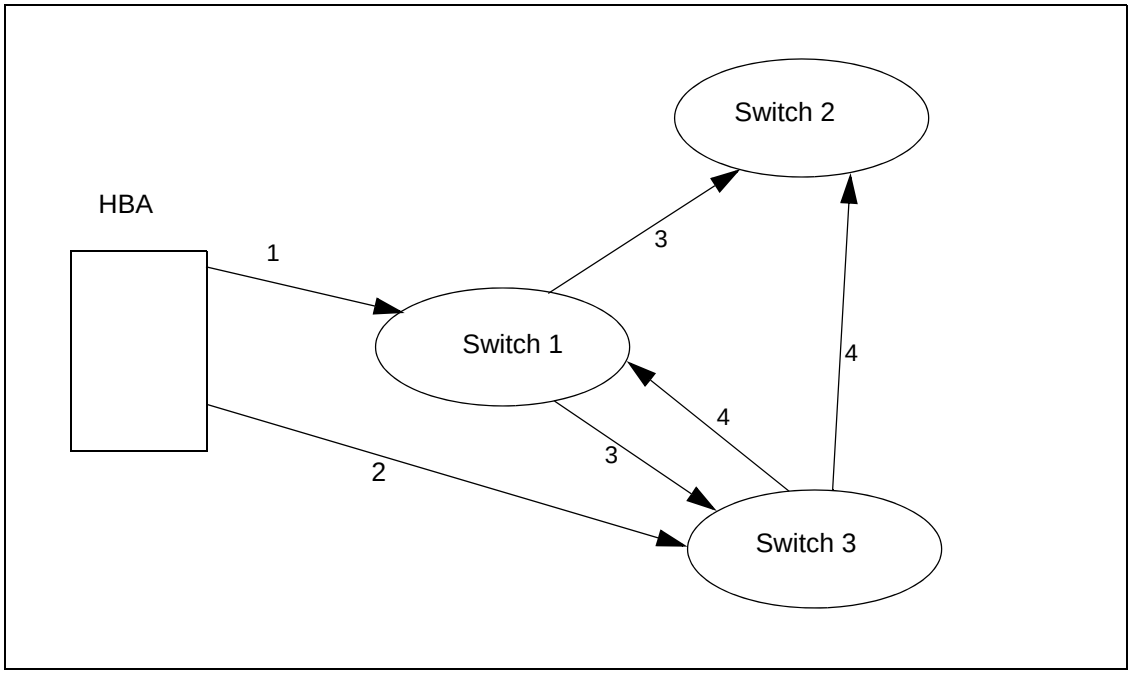

Figure C.3 – Registration of HBA Information - Multiple Switches Caches Not Updated

C.2.4 HBA De-Registration - Primary HBA Manager

In Figure C.4 below, the switch interactions are shown for HBA De-Registration to the primary HBA manager.

1) The HBA attempts de-registration by sending an DHBA to Switch 1. Since Switch 1 is the primary HBA manager, the de-registration completes successfully and the HBA information is removed from Switch 1's FDMI database.

2) Switch 1 sends a De-registration Notification to Switches 2 and 3. Switches 2 and 3 now delete the HBA's information from their caches.

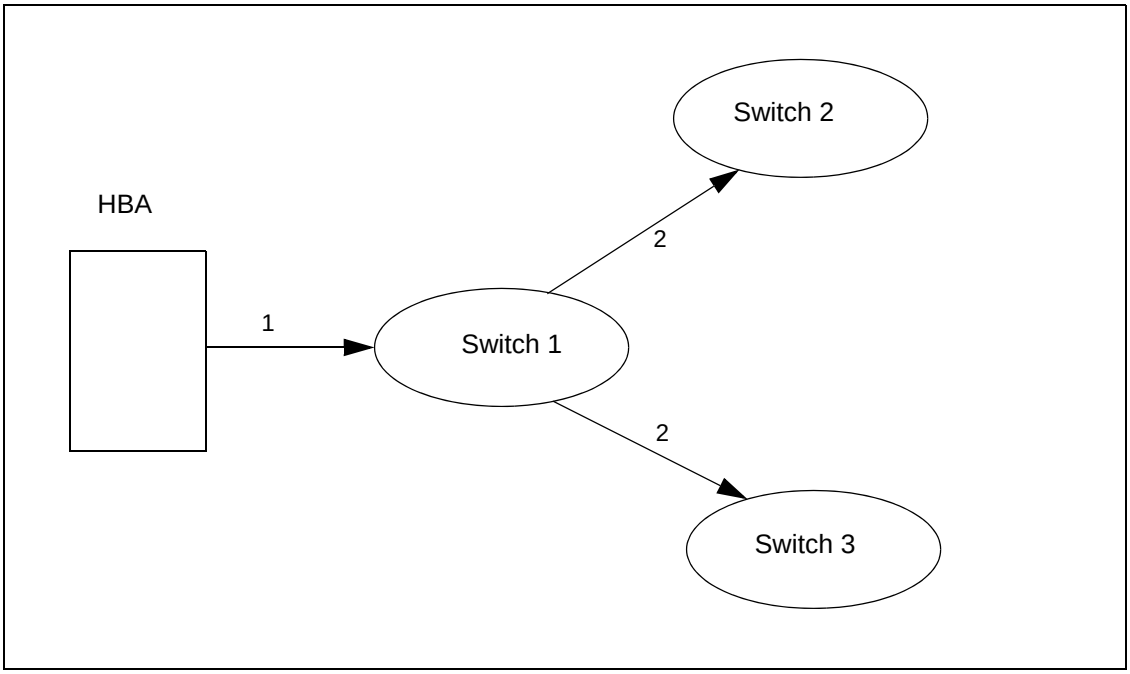

Figure C.4 – HBA De-Registration - Primary HBA Manager

C.2.5 HBA De-Registration - Non-Primary HBA Manager

In Figure C.5 below, the switch interactions are shown for HBA De-Registration to a switch that is not the primary HBA manager. In this case, Switch 1 in the primary manager for the HBA and the HBA is connected to Switch 1 and Switch 2.

1) The HBA attempts de-registration by sending an DHBA to Switch 2. Since Switch 2 is not the primary manager for the HBA Switch 2, it is required to inform Switch 1 of the de-registration request.

2) Switch 2 sends a De-Registration Forward to Switch 1 which is the HBA's primary manager. Switch 1 de-registers the HBA information and deletes the information from its FDMI database.

3) Switch 1 sends a De-registration Notification to Switches 2 and 3. Switches 2 and 3 now delete the HBA's information from their caches.

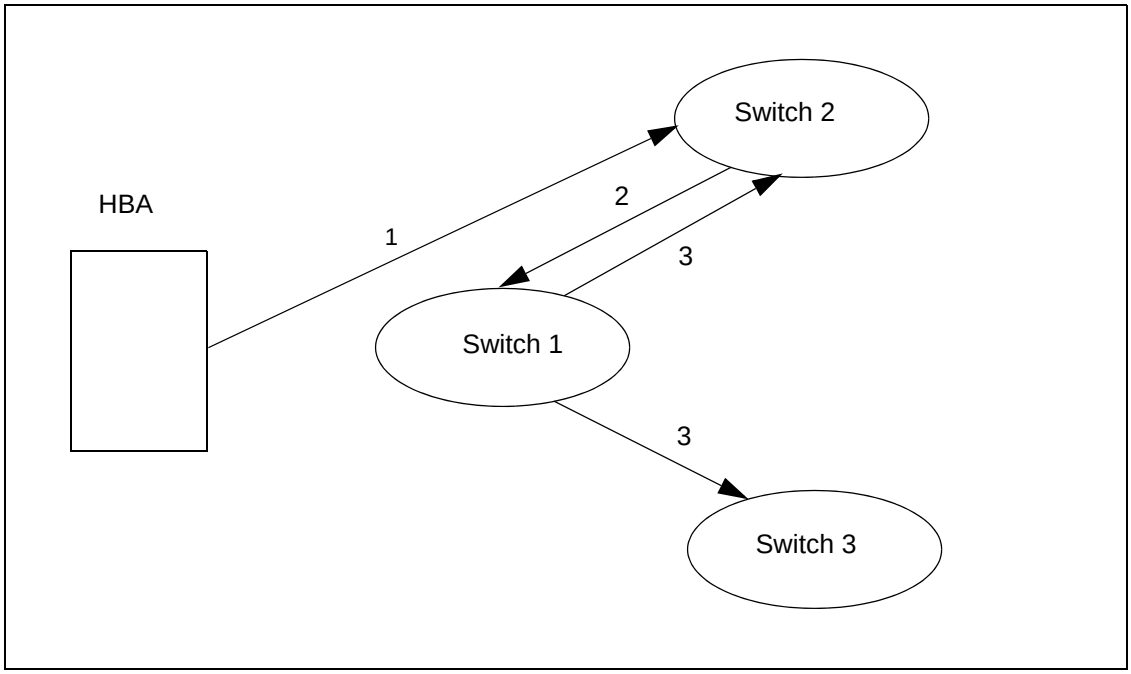

Figure C.5 – HBA De-Registration - Non-Primary HBA Manager

Annex D
(Normative)

Fast Fabric Initialization for AE-Capable Equipment

This Annex defines the minimal requirements for Fast Fabric Initialization (FFI). The FFI requirements apply to switches that are specified to be “AE-Capable”. An AE-Capable Switch is a Fabric element that supports at least one Avionics Expansion Port (AE_Port).

D.1 Background

Fibre Channel is growing in industry acceptance in the Avionics Environment. Initial systems were relatively simple, employing architectures containing no more than two switches. Systems being designed today for tomorrow’s applications will be complex by comparison, employing many switches. The real-time nature of modern avionics demands stringent requirements, especially during fabric initialization, for determinism, low latency, predictability, reliability, and fault tolerance.

Avionics Fibre Channel equipment differs from typical commercial equipment in several important ways. Foremost, the Avionics Fabric domain topology by definition will be predefined. The location of all domains and their associated Inter-Switch Links within the fabric topology, whether currently active or inactive, do not change for the duration of a mission.

Another important distinction in requirements between avionics systems and commercial systems is that commercial systems are plug-and-play with self-discovery, while avionics systems are inherently well-known systems with fixed configurations. The plug-and-play requirement in the commercial industry results in significant protocol mechanisms not necessary in the Avionics Environment. In Avionics all the switches, their Domain_IDs, Inter-Switch Links, and the topology and routing maps are well known per mission and not subject to change. By eliminating the protocol tools utilized in the commercial industry for discovering entities, Avionics systems can significantly simplify initialization protocols and lower system latency from power interruption to an active state.

This Annex formalizes the methods for accelerating the initialization of an Avionics Fabric by allowing certain AE-Capable Switches to have implicit knowledge of the entire domain topology before fabric initialization, which is then distributed throughout the fabric. Techniques used in commercial fabric initialization, such as Principal Switch Selection, and the Fabric Shortest Path First protocol are undesirable and omitted since they add overhead and uncertainty to the initialization time of Avionics systems. Avionics Fabrics must be initialized quickly, and in a deterministic and repeatable fashion.

D.2 Definitions

AE (Avionics Environment): Avionics Environment refers to hi-reliability applications in harsh environmental conditions.

AE-Capable Switch: A Fibre Channel Fabric Switch that is capable of supporting at least one AE_Port.

AE Principal Switch: An AE Switch has no Uplinks and assumes the primary role of distributing the Domain Topology Map in an Avionics Fabric.

AE Secondary Principal Switch: An AE Switch that is capable of becoming the AE Principal Switch.

AE Switch: An AE-Capable Switch that has activated at least one AE_Port. AE Switches are required to implement the requirements set forth in this Annex.

AE_Port: A Fabric Avionics Expansion Port that connects to another Avionics Expansion Port to create an Inter-Switch Link. AE_Ports are required to implement the requirements set forth in this Annex.

Active AE_Port: An AE_Port that has reached the AE0 state or subsequent states.

Avionics Fabric: A Fibre Channel Fabric that contains at least one AE Switch and supports all the requirements of this Annex. An Avionics Fabric may or may not support the requirements of the rest of the FC-SW-5 standard.

Domain Topology Map: An entity within the Avionics Fabric that unambiguously describes the Domain_IDs and all of the Inter-Switch Links of the Avionics Fabric. The Domain_IDs and all of the Inter-Switch Links shall remain unchanged for the duration of a mission.

Downlink: An ISL connected to an Active AE_Port that is not part of at least one path that leads to the AE Principal Switch.

FFI (Fast Fabric Initialization): A technique that provides accelerated initialization of an Avionics Fabric through the distribution of the Domain Topology Map. The Domain Topology Map is distributed to all AE Switches via the AE Principal Switch using the FFI request Sequence.

FFI Incarnation Number: A number within FFI request Sequence that uniquely identifies each version of the Domain Topology Map. The FFI Incarnation Number is required to be managed solely by the AE Principal Switch.

FFI Link Descriptor: A description of an individual AE_Port-to-AE_Port connection within the Avionics Fabric.

FFI Link State Record: For an individual AE Switch, a description of the Domain and all the AE_Port Inter-Switch Link connections of that Switch.

FFI SW_ILS: An AE specific SW_ILS command that distributes the Domain Topology Map throughout the Avionics Fabric or for reporting changes in link status and error conditions.

Inactive AE_Port: A switch port that is designated in the Domain Topology Map to be an AE_Port, but the switch port has not yet reached the AE0 state in its Port Mode Initialization.

Uplink: An ISL connected to an Active AE_Port that is part of at least one path that leads to the AE Principal Switch.

D.3 Characteristics of Avionics Fabrics

D.3.1 Overview

An Avionics Fabric differs from a normal Fibre Channel Fabric in that it shall support a method for initialization called Fast Fabric Initialization (FFI). Specified in this Annex, FFI amends the requirements defined in clause 7 (Fabric Configuration) of this standard for Avionics Fabrics.

Fast Fabric Initialization specifies a set of coherent behaviors for the establishment of communication between multiple switches within an Avionics Fabric. The complete definition of the fabric topology is

defined in the Domain Topology Map of the system. The AE Principal Switch shall be responsible for initiating the distribution the Domain Topology Map to all AE Switches in the Avionics Fabric.

The Domain Topology Map shall be distributed to all AE Switches in the Avionics Fabric using the FFI request Sequence. The FFI request Sequence shall use AE_Ports for distribution. In addition to the Domain Topology Map, the FFI request Sequence payload shall include the link status of all AE_Port links and the switch status of all AE Switches that are defined in the Domain Topology Map.

The method for the AE Principal Switch to obtain the Domain Topology Map can be implicit or through the reception of the FFI_DTM ELS Command from an Nx_Port.

D.3.2 AE Switch Port Mode Initialization

D.3.2.1 Overview

Switch ports that are AE-Capable shall negotiate their Port Mode via the Port Mode Initialization State Machine defined in clause 7 and as modified by this clause. Switch ports that are AE-Capable shall negotiate to become an AE_Port by following the rules of this clause.

Switch ports that are AE-Capable may be connected to other switch port types, but AE_Port functionality shall only be used when an AE_Port is directly connected to another AE_Port.

D.3.2.2 Switch Port Mode Initialization State Machine Modifications

Figure D.1 shows the modifications needed to the Switch Port Mode Initialization State Machine specified in clause 7 for AE-Capable Ports.

The AEP4 state replaces the P4 state that is described in clause 7. Therefore the transitions for P0:P4 and P1:P4 are identical except that they now point to state AEP4 as P0:AEP4 and P1:AEP4. The AEP4 state is described fully herein.

There are new Switch Port Mode Initialization states called AEP5, AEP6 and AE0. They are described fully herein. All other Switch Port Mode Initialization State Machine states, including P5, remain unmodified.

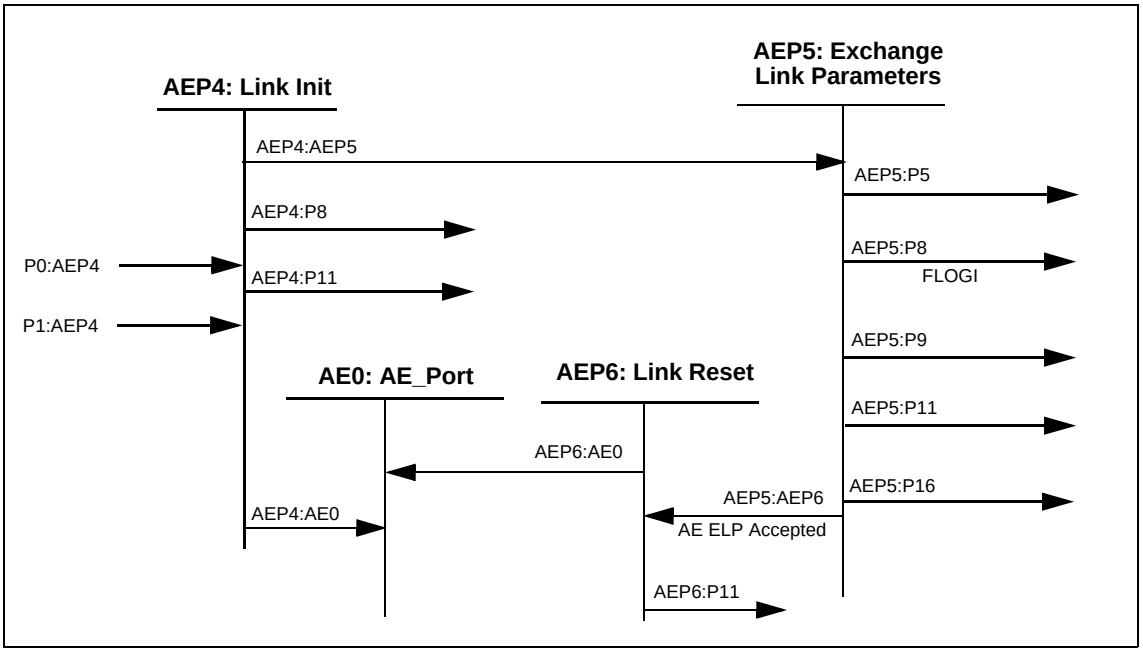

Figure D.1 – Modifications to Port Mode Initialization

The following text specifies modifications to the Port Mode Initialization State Machine for AE-Capable Ports.

Transition P0:AEP4. The Switch Port is not capable of becoming an FL_Port. Attempt Link Initialization. (AEP4 replaces P4.)

Transition P1:AEP4. This transition occurs if the Loop Initialization does not complete successfully. This may occur if the Switch Port is attached to a non-L_Port capable port, so the next thing to try is a Link Initialization. (AEP4 replaces P4.)

State AEP4: Link Initialization. The Switch Port shall attempt Link Initialization as defined in FC-FS-3.

Transition AEP4:AE0. This transition occurs if the Link Initialization procedure succeeds and the AE-Capable Port is programmed to immediately become an AE_Port.

Transition AEP4:AEP5. This transition occurs if the Link Initialization procedure succeeds and the AE-Capable Port is not programmed to immediately become an AE_Port.

Transition AEP4:P8. This transition occurs if the Link Initialization procedure succeeds and the AE-Capable Port is programmed to immediately become an F_Port.

Transition AEP4:P11. This transition occurs when the Link Initialization procedure fails.

State AEP5: Exchange Link Parameters. An AE-Capable Switch Port shall originate an AE-specific ELP SW_ILS Request Sequence by setting the ISL Flow Control Mode field in the ELP to a Vendor Specific value of AE02h. The minimum requirements of the ELP payload for AE-Capable Ports are provided in clause D.3.3.

Table D.1 describes the responses an Originator of an ELP may receive, and further actions and state transitions the originator shall make.

Table D.1 – Responses to ELP Request for Originating Interconnect_Port (Part 1 of 2)

Response to ELP	Indication	Originating Interconnect_Port Action
1. R_RDY	Request received at destination	Wait E_D_TOV+1 second for response frame. Do not transition
2. ACK_1	Request received at destination	Wait E_D_TOV+1 second for response frame. Do not transition.
3. SW_ACC (Flow Control = “AE02”)	Destination is an AE_Port and accepts all ELP parameters	Send ACK_1, Transition (AEP5:AEP6)
4. F_BSY or P_BSY	Destination is busy	Retry[a], Transition (AEP5:P11)
5. F_RJT or P_RJT	The frame is not acceptable	Respond accordingly[c], Transition (AEP5:P11)
6. ELP (rcvd Switch_Name > own Switch_Name)	Both Interconnect_Ports sent ELP at the same time. (Destination has control.)	Send SW_ACC or SW_RJT based on the values of the received ELP parameters. Transition (AEP5:AEP6)
7. ELP (rcvd Switch_Name < own Switch_Name)	Both Interconnect_Ports sent ELP at the same time. (Own switch has control.)	Send SW_RJT[b], Do not transition.
8. ELP (rcvd Switch_Name = own Switch_Name)	Interconnect_Port output is looped back to input	Remove loopback condition, Transition (AEP5:P9)
9. SW_RJT with Reason Code of “Command already in progress[d]”	Both Interconnect_Ports sent ELP at the same time. Destination has control (has greater Switch_Name) and has rejected own switch’s ELP.	Send SW_ACC or SW_RJT based on the values of the received ELP parameters. Do not transition.

[a] The retry is performed following a time-out period, as defined in P11. The R_A_TOV used for retry may be specifically set for Avionics Equipment.

[b] The Reason Code shall be “Unable to perform command request” with a Reason Explanation of “Command already in progress”.

[c] Response is defined in FC-FS-3.

[d] An SW_ACC is sent for the other ELP Exchange in progress if the received ELP parameters are acceptable. If the received ELP parameters are not acceptable, then an SW_RJT is sent.

Table D.1 – Responses to ELP Request for Originating Interconnect_Port (Part 2 of 2)

Response to ELP	Indication	Originating Interconnect_Port Action
10. SW_RJT with any other Reason Code	Both Interconnect_Ports sent ELP at the same time. Own switch has control, but the destination switch has rejected the ELP parameters	Try different ELP for E_Port if supported, Transition (AEP5:P5) Otherwise Isolate, Transition (AEP5:P9)
11. FLOGI	Destination is an N_Port	Respond accordingly[c], Transition (AEP5:P8)
12. any other frame	Indeterminate	Discard frame and Retry[a], Transition (AEP5:P11)
13. E_D_TOV+ 1 second expires	No response within timeout period	Retry[a], Transition (AEP5:P11)

[a] The retry is performed following a time-out period, as defined in P11. The R_A_TOV used for retry may be specifically set for Avionics Equipment.

[b] The Reason Code shall be “Unable to perform command request” with a Reason Explanation of “Command already in progress”.

[c] Response is defined in FC-FS-3.

[d] An SW_ACC is sent for the other ELP Exchange in progress if the received ELP parameters are acceptable. If the received ELP parameters are not acceptable, then an SW_RJT is sent.

The originating AE_Port shall consider the exchange of Link Parameters complete (but not necessarily successful) when it has received the SW_ACC or SW_RJT and has transmitted the ACK_1 for the SW_ACC or SW_RJT reply Sequence.

The responding AE_Port shall consider the exchange of Link Parameters complete when it has received the ACK_1 for the SW_ACC or SW_RJT.

The exchange of Link Parameters shall be considered successful when the exchange of Link Parameters is complete, and the reply to the ELP is an SW_ACC, and both AE_Ports agree that the parameters exchanged are acceptable.

Transition AEP5:P5. This transition occurs if the responding Interconnect_Port does not agree that the AE_Ports parameters are acceptable, and the Interconnect_Port is capable of becoming an E_Port. The responding Interconnect_Port shall return an SW_RJT reply Sequence with the Reason Code of “Unable to perform Command Request” and the Reason Code Explanation of “Class F Service Parameter Error”, and shall perform the entry conditions for State P5.

This transition may also occur if the originating AE-Capable Port does not agree that the parameters in the SW_ACC are acceptable, or it receives an SW_RJT indicating the parameters in the ELP request were not acceptable to the responding Interconnect_Port, and it is capable of originating a new ELP request Sequence for state P5.

Transition AEP5:AEP6. This transition is taken by the originator of the ELP exchange when the exchange of link parameters is complete. In order for this transition to occur, the received ISL Flow Control Mode field in the ELP SW_ACC shall be set to the Vendor Specific value of AE02h and all other ELP parameters shall be set to acceptable values.

Transition AEP5:P8. This transition occurs if the exchange of Link Parameters is unable to be completed and an explicit FLOGI ELS Request is received and the AE-Capable Switch port is capable of F_Port operation.

Transition AEP5:P9. This transition occurs if the originating AE-Capable Port does not agree that the parameters in the SW_ACC are acceptable, or it receives an SW_RJT indicating the parameters in the ELP request were not acceptable to the responding Interconnect_Port, and it is not capable of originating a new ELP request Sequence with modified parameters. (Note: The switch port remains isolated until the next initialization event.)

Transition AEP5:P11. This transition occurs if the ELP is rejected with “unable to perform command request” Reason Code, and no FLOGI is received. The Switch Port performs the Link Offline protocol as defined in FC-FS-3 during the transition.

Transition AEP5:P16. This transition is taken when authorization checks that are based on data from the ELP fail.

State AEP6: Link Reset. The AE_Port shall perform the Link Reset Protocol.

Transition AEP6:AE0. This transition occurs if the Link Reset Protocol is successful.

Transition AEP6:P11 This transition occurs if the Link Reset Protocol fails.

State AE0: AE_Port. The Switch Port has completed becoming an AE_Port and shall continue to operate as an AE_Port until the next initialization event. The AE_Port shall then participate in the next phase of Fast Fabric Initialization.

D.3.3 ELP Payload Requirements

Table D.2 provides the minimum requirements of the ELP Request and Accept Payload for AE-Capable Ports. Fields not specified below shall follow the definitions of the ELP SW_ILS in clause 6.

The flow control model requires the use of the R_RDY Primitive Signal to manage BB_Credit. The format of the Flow Control Parameters for an AE-Capable AE_Port shall be the same as the flow control parameters for the R_RDY mode of flow control (0002h). The Compatibility Parameters of the Flow Control Parameters shall be set to 0000h.

Table D.2 – ELP Required Payload Values for AE_Ports (Part 1 of 2)

ELP Request/Accept Payload	Value	Notes
Revision	3h	
Flags	0000h	B_Ports Prohibited
Class F Service Parameters		
VAL (Class Valid)	1b	
XII (X_ID Interlock)	0b	
Max BB Receive Data Field Size	$\geq$ 2048	
Class 1 Interconnect_Port Parameters		

Table D.2 – ELP Required Payload Values for AE_Ports (Part 2 of 2)

ELP Request/Accept Payload	Value	Notes
VAL (Class 1 Valid)	X	
Max BB Receive Data Field Size	≥ 2048	Valid only when Class 1 validity is set to 1b
Class 2 Interconnect_Port Parameters		
VAL (Class 2 Valid)	X	
SEQ (Sequential Delivery)	1b	Valid only when Class 2 validity is set to 1b
Max BB Receive Data Field Size	≥ 2048	Valid only when Class 2 validity is set to 1b
Class 3 Interconnect_Port Parameters		
VAL (Class 3 Valid)	1b	
SEQ (Sequential Delivery)	1b	
Max BB Receive Data Field Size	≥ 2048	
ISL Flow Control Mode	AE02h	Uniquely defines AE-Capable Ports
Flow Control Parameter Length	20	Follows the R_RDY Flow Control model
Compatibility Parameters	0000h	Not used
Legend: X = Don't Care		

D.3.4 AE Principal Switch

D.3.4.1 AE Principal Switch Initialization Process

An AE Principal Switch shall be required in order to support FFI.

The AE Principal Switch shall initiate fabric initialization through distribution of the Domain Topology Map. The Domain Topology Map is communicated to the AE Principal Switch either implicitly or through the reception of the FFI_DTM ELS Command from an Nx_Port. When the AE Principal Switch receives a valid FFI_DTM ELS Command from an Nx_Port, the AE Principal Switch shall update its own Domain Topology Map to the Domain Topology Map that is specified in the payload of the FFI_DTM ELS.

The AE Principal Switch shall be capable of retaining or reacquiring its Domain Topology Map through a power cycling event.

The AE Principal Switch initiates the distribution of the Domain Topology Map throughout the Avionics Fabric. The AE Principal Switch shall send the FFI SW_ILS request Sequence to each of its AE_Port ISLs after the ELP exchange has been successfully completed on that link.

D.3.4.2 Map Update Process

The AE Principal Switch shall also collect link status information from all AE Switches and distribute this information to all AE Switches using the Map Update process.

The AE Principal Switch receives Link Change Notification Flags and Problem Detected Notification Flags from downstream AE Switches via the FFI request Sequences. The AE Principal Switch collates this information, and then initiates a Map Update FFI request Sequence to all downstream switches in the Avionics Fabric. When the AE Principal Switch initiates a new Map Update FFI request Sequence, it shall also increment the FFI Incarnation Number in order to uniquely identify the Map Update. When the FFI Incarnation Number reaches the value of “FFFF FFFF”, it shall be allowed to roll over with the understanding that the value of “0000 0000” is greater than “FFFF FFFF”.

Only the AE Principal Switch is allowed to change the FFI Incarnation Number. The lower AE Switches shall use their current FFI Incarnation Number to identify any change in status. The AE Principal Switch may combine several Link Change Notifications before a new Map Update FFI request Sequence is initiated.

The procedure for distributing the Domain Topology Map throughout the Avionics Fabric is fully described in clause D.3.5.

D.3.4.3 AE Principal Switch Update Process

It is permitted that more than one switch have the capability to become the AE Principal Switch. Only one AE Principal Switch shall be active at any time.

The selection of the AE Principal Switch is beyond the scope of this Annex. Similarly, the selection of a replacement AE Principal Switch is beyond the scope of this Annex. In the case where an alternate AE Principal Switch is selected, this clause describes the mechanism for replacing the AE Principal Switch and for updating the AE Principal Uplink on each switch.

Any AE Switch that is capable of becoming the AE Principal Switch shall be marked as such using the AE Secondary Principal Switch Flag in the FFI Link State Record in the Domain Topology Map. Any AE Switch that is capable of becoming the AE Principal Switch shall be capable of retaining or reacquiring its Domain Topology Map through a power cycling event.

The selection of a replacement AE Principal Switch is accomplished implicitly or through the usage of the FFI_PSS ELS command. The FFI_PSS ELS command is addressed to an AE Secondary Principal Switch.

If the AE Secondary Principal Switch receives a valid FFI_PSS ELS command, it shall reply to the FFI_PSS ELS with the LS_ACC Reply Sequence and then become the new AE Principal Switch. The new AE Principal Switch shall send an FFI request Sequence with the ìPrincipal Updateî flag set, indicating that this FFI request Sequence is informing the Avionics Fabric of the change in the AE Principal Switch. The new FFI Incarnation number shall be considered valid. After each lower switch resets its own links to Downlinks as appropriate, the lower switch then sends a new FFI request Sequence with the Principal Update flag to its AE_Port Downlinks. In this manner, all lower switches will relearn their Uplinks and Downlinks in accordance with the new AE Principal Switch.

D.3.5 FFI Domain Topology Map Distribution

D.3.5.1 Overview

Domain_ID Assignment shall follow the requirements of this clause. This clause supersedes the Address Distribution procedure described in clause 7 of this document.

The Domain_IDs and all of the AE_Port Inter-Switch Links are defined in the Domain Topology Map. The Domain Topology Map is required to be sent in full in the payload of every FFI request Sequence. From the payload, any AE Switch that does not yet have a Domain_ID can immediately obtain one. In addition, when any AE Switch initiates a new New FFI request Sequence, it is also required to identify in the payload the intended Domain_ID recipient before beginning transfer of the Domain Topology Map. In this manner the recipient shall adopt or confirm its own Domain_ID before processing the Domain Topology Map.

D.3.5.2 FFI Domain Topology Map Distribution State Machine Diagram

Figure D.2 and figure D.3 represent the FFI Domain Topology Map Distribution State Machines for non-principal AE Switches and for the AE Principal Switch, respectively.

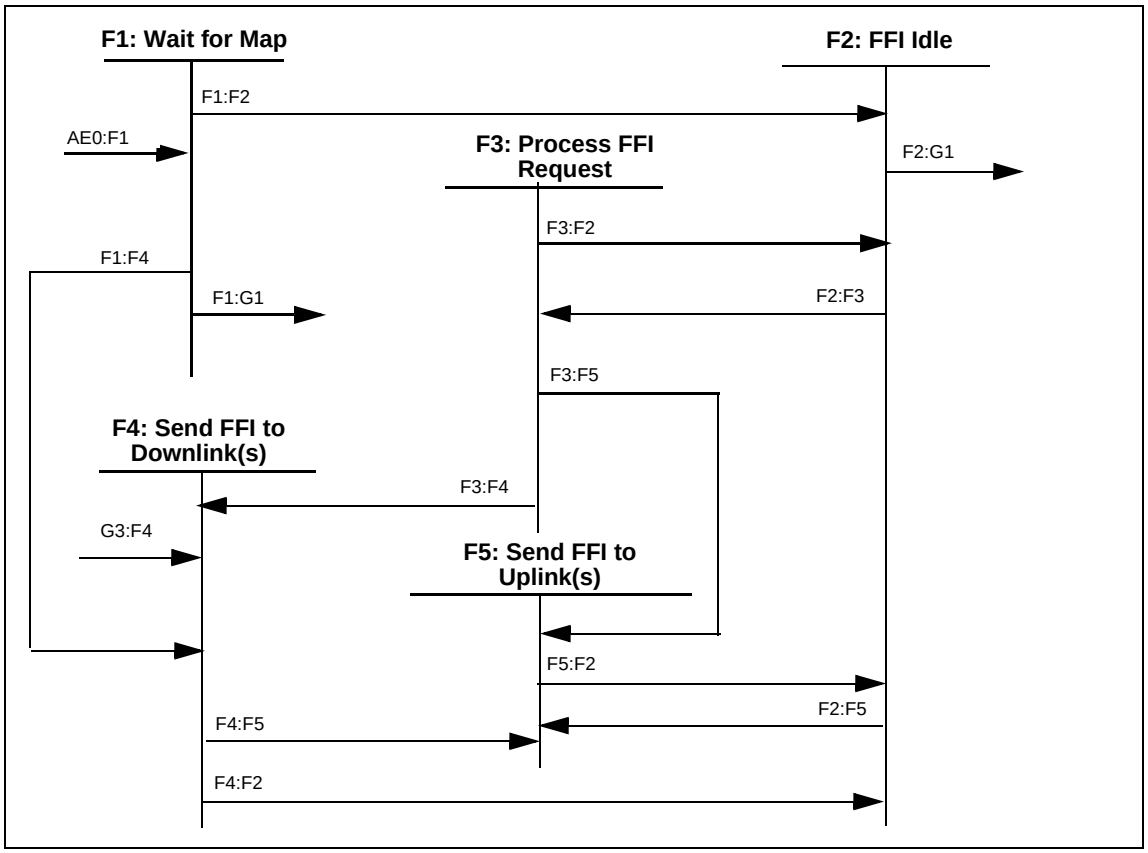

Figure D.2 – FFI Domain Topology Map Distribution State Machine, non-principal AE Switches

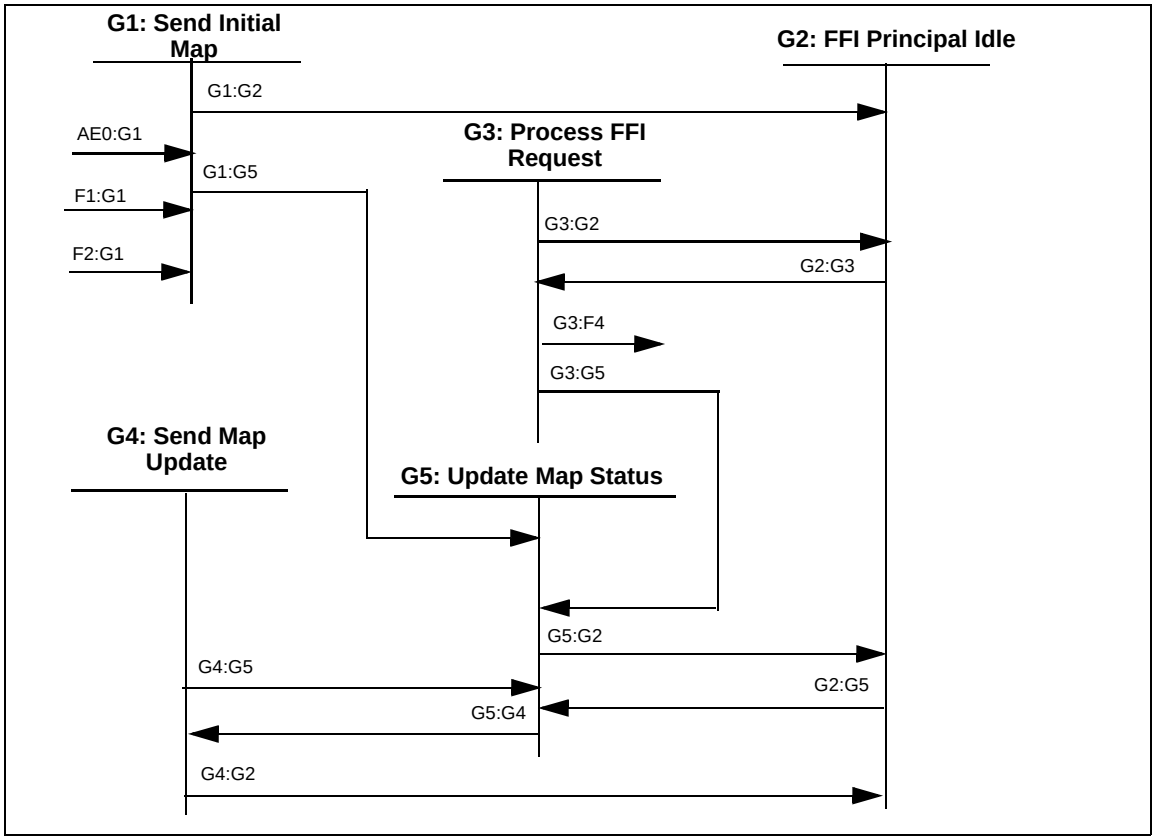

Figure D.3 – FFI Domain Topology Map Distribution State Machine, AE Principal Switch

D.3.5.3 FFI Domain Topology Map Distribution State Machine Text

The following text represents the FFI Address Distribution State Machine for AE Switches. A prerequisite to any state in the FFI Address Distribution State Machine is that the AE Capable Switch must have at least one Active AE_Port.

The FFI Address Distribution State Machine consists of two distinct sections. The "F" and "G" states are substates of State AE0. The “F” states shall be used when the AE Switch is not the AE Principal Switch. The “G” states shall be used when the AE Switch is the AE Principal Switch. If an AE Switch is not capable of becoming the AE Principal Switch, it is not required to implement the “G” states of the state machine.

Transition AE0:F1. This transition occurs after the AE_Port Mode initialization has been completed on the first active AE_Port and it is not pre-determined that the AE Switch is the AE Principal Switch.

Transition AE0:G1. This transition occurs after the AE_Port Mode initialization has been completed on the first active AE_Port and it is pre-determined that the AE Switch is the AE Principal Switch. In order to be the AE Principal Switch, the AE Switch must have an implicit Domain_ID and an implicit Domain Topology Map, and the Domain Topology Map must indicate that its own Domain_ID is designated as the AE Principal Switch.

State F1: Wait for Map. The AE Switch waits for the first FFI request Sequence to be received on one of its AE_Ports, or for an implicit Domain Topology Map to be received that indicates that this AE Switch is the AE Principal Switch.

Upon entry to this state, the AE Switch shall declare all of its AE_Ports to be Downlinks.

The received FFI request Sequence shall be checked for validity. This includes checking that either the Map Update Flag or the Override Incarnation Number Map Update Flag or the Override Domain Topology Map Update Flag or the Principal Update Flag is set. If the received FFI request Sequence is valid, the AE Switch declares this AE_Port Inter-Switch Link (ISL) to be an Uplink for FFI. The AE Switch shall reply with the SW_ACC.

If the AE Switch already has a Domain Topology Map, and an FFI request Sequence is received with either the Map Update Flag or Principal Update Flag or Override Incarnation Number Map Update Flag set, it shall compare the received Domain Topology Map to its own Domain Topology Map. If the Domain Topology Maps do not match, then the AE Switch shall reply with SW_RJT and shall remain in the F1 state. The Reason Code shall be “Logical Error” and the Reason Explanation shall be “Invalid Data”.

If the AE Switch receives an FFI request Sequence with the Override Domain Topology Map Update Flag set, it shall accept the Domain Topology Map from the FFI request Sequence. The AE Switch shall reply with SW_ACC.

Each FFI request Sequence that is received shall be checked for Map consistency. This includes checking the Originator Domain, Recipient Domain, Originator Port Index, and Recipient Port Index fields against the Domain Topology Map. If an inconsistency is detected, then the AE Switch shall send an SW_RJT and shall remain in the F1 state. The Reason Code shall be “Logical Error” and the Reason Explanation shall be “Invalid Data”.

If the FFI request Sequence has the Link Change Notification Flag or Problem Detected Notification Flag set, then the AE Switch shall send an SW_RJT and shall remain in the F1 state. The Reason Code shall be “Unable to Perform Command Request” and the Reason Explanation shall be “Unable to Verify Connection”.

In the case where the AE Switch receives a Domain Topology Map that indicates its FFI LSR has the AE Secondary Principal Switch Flag set, the AE Switch shall first verify that it can perform the role required by the AE Principal Switch. If not, then the AE Switch shall send an SW_RJT and shall remain in the F1 state. The Reason Code shall be “Logical Error” and the Reason Explanation shall be “Invalid Data”.

In all cases where the FFI request Sequence is acceptable, the AE Switch shall set its Domain_ID to the value in the FFI request Sequence. The AE Switch shall set its Domain Topology Map to be the map defined in the FFI request Sequence. The AE Switch shall set its FFI Incarnation Number to the value in the FFI request Sequence.

Transition F1:F2. This transition occurs after a SW_ACC reply to a valid FFI request Sequence has been sent, and there are no other active AE_Port Downlinks.

Transition F1:F4. This transition occurs after a SW_ACC reply to a valid FFI request Sequence has been sent, and there is at least one active AE_Port Downlink.

Transition F1:G1. This transition occurs when the AE Switch receives a Domain Topology Map that indicates that this AE Switch is the new AE Principal Switch. The AE Switch may learn this Domain Topology Map implicitly or by receiving a valid FFI_DTM ELS Command from an external Nx_Port.

State F2: FFI Idle. The AE Switch is responsible for continually monitoring the status of all of its AE_Ports, and for maintaining the consistency of the Domain Topology Map database. In this state, the AE Switch monitors all of its ports and takes the appropriate action as they become Active AE_Ports or Inactive AE_Ports. Also in this state, the AE Switch monitors all of its AE_Ports for any FFI request Sequences. Additionally, the AE Switch maintains the latest FFI Incarnation Number.

In the event that an AE Secondary Principal Switch receives a valid FFI_PSS ELS Command, then the switch shall set the AE Principal Update Flag before transitioning to state G1.

Transition F2:F3. This transition occurs when the AE Switch receives an FFI request Sequence on any of its AE_Ports.

Transition F2:F5. This transition occurs when the AE Switch has determined that a link status change has occurred on at least one of its AE_Ports.

Transition F2:G1. This transition occurs when the AE Switch that is designated as an AE Secondary Principal Switch receives a valid FFI_PSS ELS Command.

State F3: Process FFI Request The AE Switch processes the FFI request Sequence that has been received on an AE_Port. The following table describes the actions to be taken by the AE Switch.

Each FFI request Sequence that is received shall first and foremost be checked for Map consistency. This includes checking the Originator Domain, Recipient Domain, Originator Port Index, and Recipient Port Index fields against the Domain Topology Map.

Each FFI request Sequence that is received shall then be checked to insure that one and only one FFI Type Flag is set.

Other conditions detected for each FFI request Sequence and actions to be taken are described in table D.3

Table D.3 – Actions taken by a non-Principal AE Switch for an FFI request Sequence (Part 1 of 3)

Condition Detected	Actions taken
1. Map Update Flag set and Incarnation Number is incremented by one	Reply SW_ACC Mark link as Uplink Update AE Switch's local Incarnation Number Update AE Switch's local Domain Topology Map Create FFI request Sequence(s) using received payload Set Map Update Flag Transition F3:F4
2. Map Update Flag set and Incarnation Number matches	Reply SW_ACC Mark link as Uplink Transition F3:F2
[a] The Reason Code shall be "Protocol Error" with a Reason Explanation of "Unable to Merge". [b] The Reason Code shall be "Logical Error" with a Reason Explanation of "Invalid Data".	

Table D.3 – Actions taken by a non-Principal AE Switch for an FFI request Sequence (Part 2 of 3)

Condition Detected	Actions taken
3. Map Update Flag set and Incarnation Number is incremented by more than one	Reply SW_RJT[a] Create FFI request Sequence(s) using AE Switch's local Domain Topology Map as payload Set FFI Problem Detected Reason Code to Incarnation Number Error Set Problem Detected Notification Flag Transition F3:F5
4. Map Update Flag set and Incarnation Number is less than current	Reply SW_RJT[a] Create FFI request Sequence(s) using AE Switch's local Domain Topology Map as payload Set FFI Problem Detected Reason Code to Incarnation Number Error Set Problem Detected Notification Flag Transition F3:F5
5. Override Incarnation Number Map Update Flag set and Incarnation Number is different	Reply SW_ACC Mark link as Uplink Update AE Switch's local Incarnation Number Update AE Switch's local Domain Topology Map Create FFI request Sequence(s) using received payload Set Override Incarnation Number Map Update Flag Transition F3:F4
6. Override Incarnation Number Map Update Flag set and Incarnation Number matches	Reply SW_ACC Mark link as Uplink Transition F3:F2
7. Link Change Notification Flag set	Reply SW_ACC Mark link as Downlink Create FFI request Sequence(s) using received payload Set Link Change Notification Flag Transition F3:F5
8. Problem Detected Notification Flag set	Reply SW_ACC Mark link as Downlink Create FFI request Sequence(s) using received payload Copy FFI Problem Detected Reason Code from received payload Set Problem Detected Notification Flag Transition F3:F5
9. Principal Update Flag Set and Incarnation Number is different	Reply SW_ACC Mark link as Uplink Mark all other links as Downlinks Update AE Switch's local Incarnation Number Update AE Switch's local Domain Topology Map Create FFI request Sequence(s) using received payload Set Principal Update Flag Transition F3:F4

[a] The Reason Code shall be "Protocol Error" with a Reason Explanation of "Unable to Merge".

[b] The Reason Code shall be "Logical Error" with a Reason Explanation of "Invalid Data".

Table D.3 – Actions taken by a non-Principal AE Switch for an FFI request Sequence (Part 3 of 3)

Condition Detected	Actions taken
10. Principal Update Flag Set and Incarnation Number matches	Reply SW_ACC Mark link as Uplink Transition F3:F2
11. Override Domain Topology Map Update Flag Set and Incarnation Number is different	Reply SW_ACC Mark link as Uplink Update AE Switch's local Incarnation Number Update AE Switch's local Domain Topology Map Create FFI request Sequence(s) using received payload Set Override Domain Topology Map Update Flag Transition F3:F4
12. Override Domain Topology Map Update Set and Incarnation Number matches	Reply SW_ACC Mark link as Uplink Transition F3:F2
13. Map inconsistency	Reply SW_RJT[b] Create FFI request Sequence(s) using AE Switch's local Domain Topology Map as payload Set FFI Problem Detected Reason Code to Map Inconsistent Set Problem Detected Notification Flag Transition F3:F5
14. More than one flag set	Reply SW_RJT[b] Create FFI request Sequence(s) using AE Switch's local Domain Topology Map as payload Set FFI Problem Detected Reason Code to Multiple Flags Set Set Problem Detected Notification Flag Transition F3:F5
15. No flags set	Reply SW_RJT[b] Create FFI request Sequence(s) using AE Switch's local Domain Topology Map as payload Set FFI Problem Detected Reason Code to No Flags Set Set Problem Detected Notification Flag Transition F3:F5
16. Other	Reply SW_RJT[b] Create FFI request Sequence(s) using AE Switch's local Domain Topology Map as payload Set FFI Problem Detected Reason Code to Unexpected Condition Set Problem Detected Notification Flag Transition F3:F5

[a] The Reason Code shall be "Protocol Error" with a Reason Explanation of "Unable to Merge".

[b] The Reason Code shall be "Logical Error" with a Reason Explanation of "Invalid Data".

Transition F3:F4. This transition occurs when the processing of the required actions of the FFI request Sequence has been completed, and there is a need to forward new FFI request Sequences on all active AE_Port Downlinks.

Transition F3:F2. This transition occurs when the processing of the FFI request Sequence has been completed, and there is no need to propagate a new FFI request sequence.

Transition F3:F5. This transition occurs when the processing of the required actions of the FFI request Sequence has been completed, and there is a need to forward new FFI request Sequences on all active AE_Port Uplinks.

State F4: Send FFI to Downlink(s): The switch initiates a separate FFI request Sequence in a new Exchange on each active AE_Port Downlink and waits for a response or time-out from each active AE_Port Downlink.

If the AE Switch determines that there are no Active AE_Port Downlinks, then this state shall perform no action.

If one or more responses are not received within 2xE_D_TOV from when the last FFI request Sequence was sent, the AE Switch shall construct an FFI request Sequence with the Problem Detected Notification Flag set. The AE Switch shall set the Problem Detected Reason Code to Downlink Time-out occurred. The FFI Time-out Detected Flag shall be set in the appropriate Link Descriptor(s) in the Domain Topology Map.

Transition F4:F2. This transition occurs when the switch has received an SW_ACC or SW_RJT response to every FFI request Sequence. This transition also occurs if there are no Active AE_Port Downlinks.

Transition F4:F5. This transition occurs if one or more responses are not received within 2xE_D_TOV from the last FFI request Sequence for which a reply was not received.

State F5: Send FFI Uplink(s): The switch initiates a separate FFI request Sequence in a new Exchange on each active AE_Port Uplink and waits for a response or timeout from each active AE_Port Uplink.

Transition F5:F2. This transition occurs when the switch has received an SW_ACC or SW_RJT response to every FFI request Sequence. This transition also occurs if one or more responses are not received within 2xE_D_TOV from the last FFI request Sequence for which a reply was not received.

State G1: Send Initial Map The AE Principal Switch shall send an FFI request Sequence with the Override Incarnation Number Map Update Flag set or the Override Domain Topology Map Update Flag set on all active AE_Ports. The AE Principal Switch shall wait for a reply or time-out from each active AE_Port. All AE_Ports links shall be set to Downlinks.

If the AE Switch determines that there are no Active AE_Port Downlinks, then this state shall perform no action.

If a SW_RJT reply is received on any AE_Port ISL, then the FFI Reject Detected Flag shall be set in the FFI LSR Link Descriptor of the Domain Topology Map before transitioning.

If a time-out occurs (response is not received within 2xE_D_TOV) on any AE_Port ISL, then the FFI Time-out Detected Flag shall be set in the appropriate Link Descriptor(s) of the Domain Topology Map before transitioning.

Transition G1:G2. This transition occurs when the switch has received an SW_ACC response to every FFI request Sequence. This transition also occurs if there are no Active AE_Port Downlinks.

Transition G1:G5. This transition occurs when the switch has received an SW_RJT response to at least one of its FFI request Sequence(s), or if one or more responses are not received within 2xE_D_TOV from the last FFI request Sequence for which a reply was not received.

State G2: FFI Principal Idle. The AE Principal Switch monitors all of its ports and takes the appropriate action as they become Active AE_Ports or Inactive AE_Ports. Also in this state, the AE Switch monitors its AE_Ports for any FFI request Sequences received.

Transition G2:G3. This transition occurs when the AE Principal Switch receives an FFI request Sequence.

Transition G2:G5. This transition occurs when the AE Principal Switch has detected a link status change on at least one of its own AE_Ports.

State G3: Process FFI Request The AE Principal Switch processes an FFI request Sequence that has been received on an AE_Port. Table D.4 describes the action to be taken by the AE Principal Switch on the AE_Port that it received the FFI request Sequence.

Table D.4 – Action taken by AE Principal Switch for an FFI request Sequence

Condition Detected	Action taken
1. Link Change Notification Flag set	Reply SW_ACC, Update the master Domain Topology Map Transition G3:G5
2. Problem Detected Notification Flag set	Reply SW_ACC, Mark problems in the master Domain Topology Map[a] Transition G3:G5
3. Map Update Flag set	Reply SW_RJT[b] Transition G3:G2
4. Override Incarnation Number Map Update Flag set	Reply SW_RJT[b] Transition G3:G2
5. Override Domain Topology Map Update Flag set	Reply SW_RJT[b] Transition G3:G2
6. Principal Update Flag Set	Reply SW_ACC Mark link as Uplink Mark all other links as Downlinks Update AE Switch's local Incarnation Number Update AE Switch's local Domain Topology Map Create FFI request Sequence(s) using received payload Set Principal Update Flag Transition G3:F4

[a] Subsequent actions to be taken are implementation specific and are beyond the scope of this Annex.

[b] The Reason Code shall be "Logical Error" with a Reason Explanation of "Invalid Data".

Transition G3:G5. This transition occurs when the processing of the FFI request Sequence has been completed, and a change has occurred to the Domain Topology Map.

Transition G3:G2. This transition occurs when the processing of the FFI request Sequence has been completed, and no change has occurred to the Domain Topology Map.

Transition G3:F4. This transition occurs when the AE Switch has received a valid FFI request Sequence with the Principal Update Flag set. This AE Switch is therefore no longer acting as the AE Principal Switch in the Avionics Fabric.

State G4: Send Map Update. The AE Principal Switch sends a new FFI request Sequence on all active AE_Port Downlinks. The AE Principal Switch shall wait for a reply or time-out from each active AE_Port.

If the AE Switch determines that there are no Active AE_Port Downlinks, then this state shall perform no action.

The FFI request Sequence payload, the FFI Incarnation Number, and the FFI Type Flags that were previously set before this state are used to create the FFI request Sequence.

If a SW_RJT reply is received on any AE_Port ISL, then the FFI Reject Detected Flag shall be set in the FFI LSR Link Descriptor of the Domain Topology Map before transitioning.

If a timeout occurs on any AE_Port ISL, then the FFI Timeout Detected Flag shall be set in the FFI LSR Link Descriptor of the Domain Topology Map before transitioning.

Transition G4:G2. This transition occurs when the switch has received an SW_ACC response to every FFI request Sequence. This transition also occurs if there are no Active AE_Port Downlinks.

Transition G4:G5. This transition occurs when the switch has received an SW_RJT response to at least one of its FFI request Sequence(s), or if one or more responses are not received within 2xE_D_TOV from the last FFI request Sequence for which a reply was not received.

State G5: Update Map Status. The AE Principal Switch shall update the Domain Topology Map according to all of the link changes that have been detected or reported.

After all the changes are processed, the AE Principal Switch determines if it is necessary to distribute the Domain Topology Map to any registered Nx_Ports. If any Nx_Ports have registered for Map updates via the FFI_MUR ELS Command, then the AE Principal Switch shall initiate an FFI_RMUN Command to each of those Nx_Ports.

After all the changes are processed, the AE Principal Switch determines if it is necessary to redistribute the Domain Topology Map to other AE Switches. If the AE Principal Switch has received a valid FFI_SMU ELS Command, and has not subsequently received a valid FFI_RMU ELS Command, then AE Principal Switch shall not redistribute the Domain Topology Map before making the transition to State G2.

If the AE Principal Switch has not received a valid FFI_SMU ELS Command, or has last received a valid FFI_RMU ELS Command, then the AE Principal Switch shall construct the FFI request Sequence payload using the latest FFI Domain Topology Map, increment the FFI Incarnation Number by one, and set the Map Update Flag before making the transition to state G4.

Transition G5:G2. This transition occurs when the AE Principal Switch determines that it will not update the Domain Topology Map to all of its active AE_Ports.

Transition G5:G4. This transition occurs when the AE Principal Switch determines that it will update the Domain Topology Map to all of its active AE_Ports.

D.3.6 Fast Fabric Initialization (FFI) SW_ILS Definition

D.3.6.1 Overview

The Fast Fabric Initialization Switch Internal Link Service (FFI SW_ILS) provides a common mechanism for distributing the Domain Topology Map and AE_Port link status throughout the Avionics Fab-

ric. During initialization, the Domain Topology Map of the Avionics Fabric is distributed by the AE Principal Switch to all AE Switches in the Avionics Fabric using the FFI SW_ILS request Sequence. After initialization, the FFI SW_ILS request Sequence is used to communicate the link status of the Domain Topology Map within the Avionics Fabric.

Protocol:

The Fast Fabric Initialization (FFI) Request Sequence
Accept (SW_ACC) Reply Sequence

Addressing: For use in Fabric Configuration, the S_ID field shall be set to FFFFFDh, indicating the Fabric Controller of the originating Switch. The D_ID field shall be set to FFFFFDh, indicating the Fabric Controller of the destination Switch.

Payload: The format of the FFI request Sequence Payload is shown in table D.5.

Table D.5 – FFI Request Payload

Item	Size Bytes
50000000h	4
Originator Switch Domain	1
Originator Port Index	1
Responder Switch Domain	1
Responder Port Index	1
FFI Incarnation Number	4
Reserved	2
FFI Type Flags	1
FFI Problem Detected Reason Code	1
Number of FFI Link State Records	4
FFI Link State Records	n

Originator Switch Domain: This field shall contain the Domain_ID of the Switch that originated the FFI request Sequence.

Originator Port Index: This field shall contain the port index of the Switch that originated the FFI request Sequence.

Responder Switch Domain: This field shall contain the Domain_ID of the neighboring Switch that responds to the FFI request Sequence.

Responder Port Index: This field shall contain the port index of the neighboring Switch that responds to the FFI request Sequence.

FFI Incarnation Number: This field contains the current incarnation of the FFI Domain Topology Map.

FFI Type Flags: This field shall define the type of FFI request Sequence. Each FFI sequence is required to set one and only one of these flags. The types are listed in table D.6.

Table D.6 – FFI Type Flags Definition

Bit	Description
0	Map Update Flag
1	Override Incarnation Number Map Update Flag
2	Override Domain Topology Map Update Flag
3	Principal Update Flag
4	Link Change Notification Flag
5	Problem Detected Notification Flag
6-7	Reserved

The Map Update Flag is used to indicate that this FFI request Sequence is being sent downstream to update the link status of the system.

The Override Incarnation Number Map Update is used to indicate that the normal Incarnation Number checks on this Map Update are to be ignored on this downstream FFI request Sequence.

The Override Domain Topology Map Update Flag is used to indicate that this FFI request Sequence is being sent downstream to replace the previous Domain Topology Map of the Avionics Fabric.

The Principal Update Flag is used to indicate that the originator of this specific FFI request Sequence is a new AE Principal Switch, and that all Uplinks and Downlinks need to be recalculated.

The Link Change Notification Flag is used to indicate that this FFI request Sequence is being sent upstream in order to indicate that a change in link status of one or more AE_Port ISLs has been detected. The change in link status is defined as an AE_Port ISL that has become active or has become inactive.

The Problem Detected Notification Flag is used to indicate that this FFI request Sequence is being sent upstream in response to some error that has been detected in the Domain Topology Map.

FFI Problem Detected Reason Code: This field shall contain a code for the error that was detected in the FFI request Sequence. This field is not meaningful unless the Problem Detected Notification Flag is set. The following table defines the codes: The types are listed in table D.6.

Table D.7 – FFI Problem Detected Reason Codes

Value	Description
0	No information
1	Multiple Flag Bits detected
2	No Flag Bit detected
3	Invalid Incarnation Number detected
4	Downlink Timeout Occurred
5	Uplink Timeout Occurred
6	Map Inconsistency
7	Multiple AE Principal Switches detected
8	Unexpected Condition
All others	Reserved

Number of FFI Link State Records: This field shall specify the number of FFI Link State Records that follow this field.

FFI Link State Records: This field contains all of the individual FFI Link State Records that describe the Domain Topology Map of the Avionics Fabric. The format of the FFI Link State Record is described in clause D.3.6.2.

Reply Switch Fabric Internal Link Service Sequence:

Service Reject (SW_RJT)
  Signifies the rejection of the FFI request Sequence
Accept (SW_ACC)
  Signifies acceptance of the FFI request Sequence.
  Accept Payload

Payload: The format of the FFI accept Payload is shown in table D.8

Table D.8 – FFI Accept Payload

Item	Size Bytes
02000000h	4
Originator Switch Domain	1
Originator Port Index	1
Recipient Switch Domain	1
Recipient Port Index	1

D.3.6.2 Fast Fabric Initialization Link State Record (FFI LSR) Format

FFI LSR: There is one format for the FFI LSR. The format is shown in table D.9. One or more FFI Link Descriptors may be contained in a single FFI LSR.

Table D.9 – FFI Link State Record - Link Descriptor Form

Item	Size Bytes
FFI Link State Record Identifier	1
FFI Link State Record Flags	1
Number of FFI Link Descriptors	2
Link Descriptor #1	4
...	4
...	4
Link Descriptor #n	4

FFI Link State Record Identifier: This field contains the Domain_ID of the Switch that owns the FFI LSR.

FFI Link State Record Flags: This field shall contain the FFI Link State Record Flags to be used with-in the FFI request Sequence. Table D.10 defines the FFI LSR Flags:

Table D.10 – FFI LSR Flags Definition

Bit	Description
0	FFI LSR Active Flag
1	FFI LSR Inconsistency Flag
2	AE Principal Switch Flag
3	AE Secondary Principal Switch Flag
4-7	Reserved

The FFI LSR Active Flag is used to indicate the current status of the particular AE Switch. When set to 1b, this indicates that the particular AE Switch is currently active and has correctly initialized. When set to 0b, this indicates that the particular AE Switch has not correctly initialized.

The FFI LSR Inconsistency Flag is used to indicate a conflict between two FFI request Sequences that have been received on different AE_Ports. If the Domain_ID of the topology does not properly match, or if the number and position of AE_Port ISLs defined for a particular Domain are not consistent, then

this bit is set to indicate the location of the inconsistency. In addition, the AE Switch is required to forward this inconsistency by initiating an FFI request Sequence to its AE Principal Uplink.

The AE Principal Switch Flag is used to identify which switch in the fabric is currently performing the role of the AE Principal Switch.

The AE Secondary Principal Switch Flag is used to identify which switches in the system have the capability to perform the role of the AE Principal Switch.

Number of FFI Link Descriptors: This field specifies the number of FFI Link Descriptors contained in the FFI Link State Record.

FFI Link Descriptor: The format of the FFI Link Descriptor is described in clause D.3.6.3.

D.3.6.3 Fast Fabric Initialization Link Descriptor Format

FFI Link Descriptor: The FFI Link Descriptor is a description of an individual AE_Port to AE_Port ISL within the Avionics Fabric. The format of the FFI Link Descriptor is shown in table D.11.

Table D.11 – FFI Link Descriptor Format

Item	Size Bytes
FFI Link Descriptor Flags	1
FFI Neighbor Link ID	1
Output Port Index	1
Neighbor Port Index	1

FFI Link Descriptor Flags: This field shall contain the link descriptor flags to be used within the FFI request Sequence. Table D.12 defines the FFI Link Descriptor flags:

Table D.12 – FFI Link Descriptor Flags Definition

Bit	Description
0	FFI Link Active Flag
1	FFI Timeout Detected Flag
2	FFI Reject Detected Flag
3-7	Reserved

The FFI Link Active Flag is used to indicate the current status of the particular ISL. When set to 1b, this indicates that the particular ISL is currently active and has correctly initialized. When set to 0b, this indicates that the particular ISL is either inactive or has not yet correctly initialized.

The FFI Timeout Detected Flag is used to indicate that an FFI request Sequence has not received the required SW_ACC or SW_RJT within the specified timeout period.

The FFI Reject Detected Flag is used to indicate that an SW_RJT response was received during the initialization process of that specific ISL.

FFI Neighbor Link Identifier: This field identifies the link and contains the Domain_ID of the neighbor Switch at the other end of the ISL, relative to the owning Switch.

Output Port Index: This field shall specify the source AE_Port Index.

Neighbor Port Index: The field shall specify the destination AE_Port Index.

D.4 FFI Domain Topology Map Distribution (Informative)

D.4.1 Sample Configuration

Figure D.4 shows an example Avionics Fabric. All the switches shown are AE-Capable Switches. In order to improve readability, some of the SW_ILS reply Sequences (SW_ACC or SW_RJT with Reason Codes) and all of the Acknowledgment frames have been omitted.

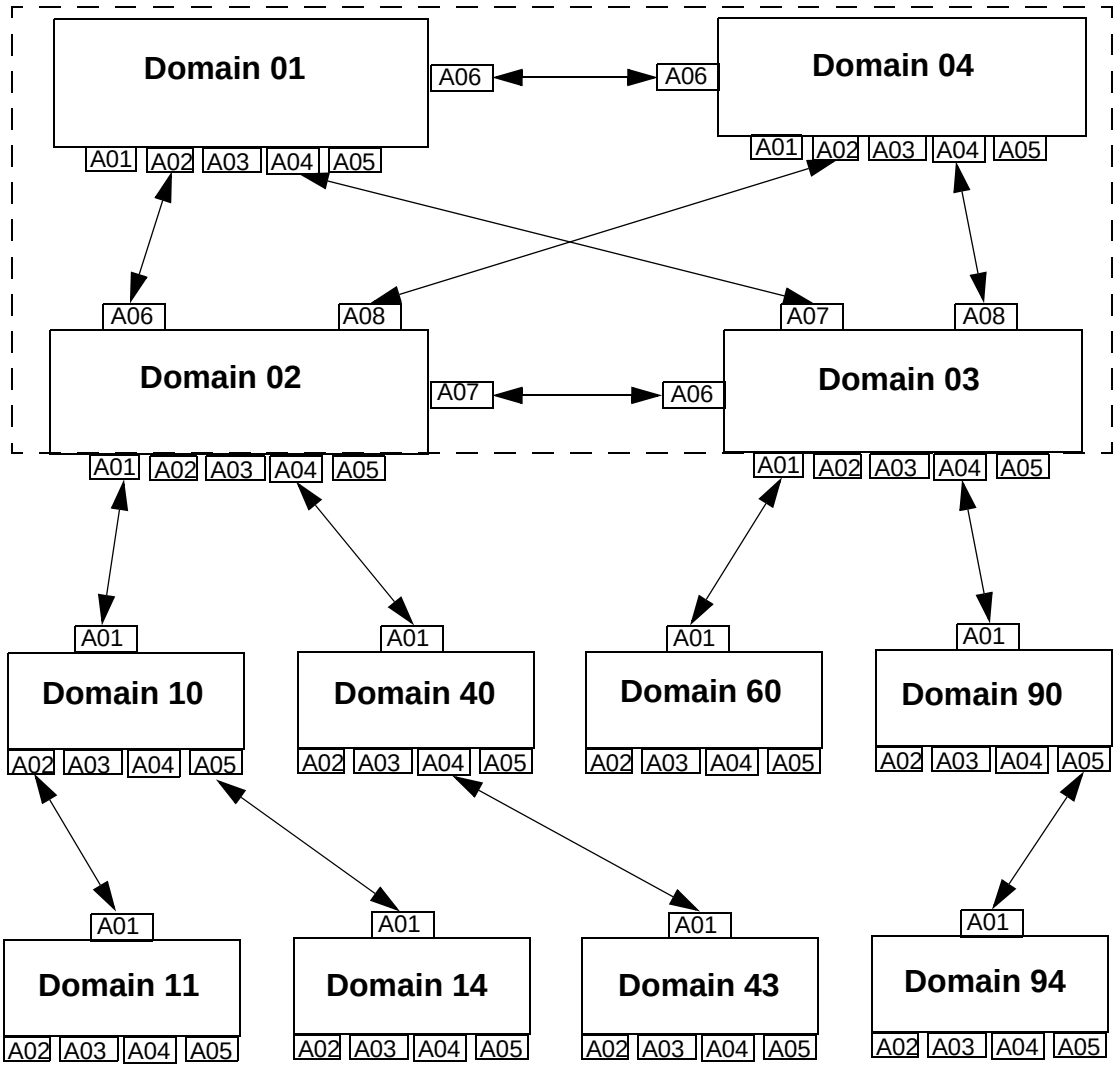

Figure D.4 – Example Avionics Fabric

In this example, Domains 01, 02, 03 and 04, know their Domain_IDs implicitly upon power up. All other switches do not know their domains upon power up. Domain 01 is the designated AE Principal Switch.

D.4.2 Initialization Procedure Example

The example begins with Domains 01, 02, 03 and 04 being powered up simultaneously. The other switches will be powered up later. These four switches will complete Switch Port Mode Initialization implicitly or explicitly (it does not matter for this example) and the inter-connected Switch Ports arrive at State AE0 (AE_Port). The next phase of FFI, Domain Topology Map distribution, can then begin.

Domains 02, 03 and 04 transition to State F1 (Wait for Map). All of their connected AE_Ports are initially defined to be Downlinks.

Domain 01 (the AE Principle Switch) transitions to State G1 (Send Initial Map). Domain 01 then sends out the “master” Domain Topology Map with the Incarnation Number set to 1 to Domains 02, 03 and

04 via three separate FFI request Sequences. Domains 02, 03 and 04 receive an FFI request Sequence, mark those Inter-Switch Links (ISLs) as Uplinks, store the Incarnation Number, and compare the received Domain_ID and Topology Map with their implicitly defined Domain_ID and Topology Map. If there are no discrepancies, Domains 02, 03 and 04 send back an SW_ACC to Domain 01; otherwise, they send back an SW_RJT.

If there are no discrepancies in the received Domain Topology Map (i.e., “the Map”), Domains 02, 03 and 04 then send the Map out on all of their Active AE_Port Downlinks via FFI request Sequences. Because of this, Domains 02, 03 and 04 may receive multiple copies of the Map. If this occurs, the recipient marks these links as Uplinks and compares the received Map with its own Map. If there are no errors and the received Incarnation Number is the same as the recipient’s current Incarnation Number, the recipient replies with an SW_ACC. If there are any discrepancies, the recipient responds with an SW_RJT.

Now assume that power is applied to the AE-Capable Switches labeled Domain 10, Domain 11 and Domain 14 simultaneously. These switches have no prior knowledge of their Domain_IDs or the Topology Map so adjacent AE_Ports exchange parameters and go through Port Mode Initialization explicitly and arrive at State AE0. The AE_Ports are all initialized as Downlinks and these three switches transition to State F1 (Wait for Map).

Domain 02, sensing a change of status on its ISL to the switch at Domain 10, sends out an FFI request Sequence on all of its Active AE_Port Uplinks with:

1. the Link Change Notification Flag set,

2. the FFI LSR Active Flag set for this Link State Record (i.e., Domain), and

3. the FFI Link Active Flag set for this specific Link Descriptor.

One copy of this FFI request Sequence will reach Domain 01 (the AE Principle Switch) directly. Other copies may arrive indirectly via Domains 03 and 04.

Domain 01 receives this FFI request Sequence, updates the master Domain Topology Map to show that this Domain is now active and that the ISL between Domain 02, Port Index 01 and Domain 10, Port Index 01 is now active. Domain 01 increments the Incarnation Number by 1 and sends out the new Map on all of its Active AE_Port Downlinks.

Domain 02 receives the FFI request Sequence with the new Map and Incarnation Number, updates its internal Map and Incarnation Number and proceeds to send the Map to all of its Active AE_Port Downlinks (in this case just switch at Domain 10) via another FFI request Sequence.

When the switch at Domain 10 receives the Map, it marks the ISL as an Uplink, adopts the Domain_ID of 10, stores the Map and Incarnation Number (without question since there is nothing to compare it with yet) and responds with SW_ACC.

Now that Domain 10 is active and has an Uplink, it is obligated to report the change in status on its ISLs to the switches at Domain 11 and Domain 14. An FFI request Sequence will be sent from Domain 10 up to Domain 02 followed by another FFI request Sequence from Domain 02 up to Domain 01 with the appropriate Flags set to indicate that these switches and links are now active. As before, Domain 01 will update the master Domain Topology Map, increment the Incarnation Number by 1, and send the new map out on all of its Active AE_Port Downlinks. The new map is received by Domain 02, then by Domain 10, and finally by the switches at Domain 11 and Domain 14.

This procedure is repeated whenever a new AE-Capable Switch joins the Avionics Fabric.

A similar procedure is followed whenever an AE_Switch or one of its ISLs become inactive. In that case, the Link Change Notification Flag would be set on the FFI request Sequence, and the corresponding FFI LSR Active Flag and/or the FFI Link Active Flag would not be set.

D.4.3 AE Principal Switch Update Example

This example describes an orderly transfer of the role of AE Principal Switch from one switch to another.

In this example, assume the Avionics Fabric described above has been fully initialized and is stable. Assume that a problem has been identified with Domain 01 and Domain 04 must now become the new AE Principal Switch. Note that the Domain Topology Map does not change when a new AE Principal Switch is selected. Also note that Domain 04 must have previously indicated that it is capable of being the AE Principal Switch by having the AE Secondary Principal Switch Flag set in its FFI Link State Record.

The process begins with Domain 04 receiving an updated Domain Topology Map that has its own FFI Link State Record marked with the AE Principal Switch Flag, or by receiving a valid FFI_PSS ELS Command from an external Nx_Port. Domain 04 immediately updates its own map, adopts the role of AE Principal Switch, and proceeds to State G1 (Send Initial Map). At this time Domain 04 redefines all of its ISLs to be Downlinks and then sends out an FFI request Sequence with the new Map, the Incarnation Number set to 1, and the AE Principal Update Flag set.

Similar to what was done in the previous example at Initialization, Domains 02 and 03 will receive the FFI request Sequence from Domain 04, see that the AE Principal Update Flag is set, and perform the following actions:

1. redefine these ISLs to be Uplinks and their other ISLs to be Downlinks,

2. store the Incarnation Number, and

3. compare the received Domain_ID and Topology Map with their implicitly defined Domain_ID and Topology Map.

If there are no discrepancies in the received Map, Domains 02 and 03 send back an SW_ACC to Domain 04; otherwise, they send back an SW_RJT. If there are no discrepancies, Domains 02 and 03 will then send the Map out, with the AE Principal Update Flag set, on all of their Active AE_Port Downlinks via FFI request Sequences.

The Switch at Domain 01, if active, will see the FFI request Sequence from Domain 04 with the AE Principle Update Flag set indicating that Domain 04 has become the new AE Principal Switch. Note that this is the only Update Flag that can be accepted by the AE Principle Switch. Upon receiving this FFI request Sequence, Domain 01 will revert to a non-principal AE Switch. It will declare this ISL to be an Uplink and propagate the FFI request Sequence to all of its Active AE_Port Downlinks.

When the other AE Switches receive the new Map with the AE Principal Update Flag set, they redefine the receiving ISL to be an Uplink and propagate the Map to all of their Downlinks.

This process continues until all switches have been informed that the AE Principal Switch has been changed.