ISO/IEC 14165-133

Edition 1.0 2010-02

INTERNATIONAL STANDARD

Information technology – Fibre channnel – Part 133: Fibre channel switch fabric-3 (FC-SW-3)

ISO/IEC 14165-133:2010(E)

IEC Central Office
3, rue de Varembé
CH-1211 Geneva 20
Switzerland
Email: inmail@iec.ch
Web: www.iec.ch

About the IEC

The International Electrotechnical Commission (IEC) is the leading global organization that prepares and publishes International Standards for all electrical, electronic and related technologies.

About IEC publications

The technical content of IEC publications is kept under constant review by the IEC. Please make sure that you have the latest edition, a corrigenda or an amendment might have been published.

- Catalogue of IEC publications: www.iec.ch/searchpub
The IEC on-line Catalogue enables you to search by a variety of criteria (reference number, text, technical committee,…). It also gives information on projects, withdrawn and replaced publications.

- IEC Just Published: www.iec.ch/online_news/justpub
Stay up to date on all new IEC publications. Just Published details twice a month all new publications released. Available on-line and also by email.

- Electropedia: www.electropedia.org
The world's leading online dictionary of electronic and electrical terms containing more than 20 000 terms and definitions in English and French, with equivalent terms in additional languages. Also known as the International Electrotechnical Vocabulary online.

- Customer Service Centre: www.iec.ch/webstore/custserv
If you wish to give us your feedback on this publication or need further assistance, please visit the Customer Service Centre FAQ or contact us:

Email: csc@iec.ch
Tel.: +41 22 919 02 11
Fax: +41 22 919 03 00

ISO/IEC 14165-133

Edition 1.0 2010-02

INTERNATIONAL STANDARD

Information technology – Fibre channnel – Part 133: Fibre channel switch fabric-3 (FC-SW-3)

INTERNATIONAL ELECTROTECHNICAL COMMISSION

PRICE CODE XC

ICS 35.200

ISBN 2-8318-1076-7

Contents

Figures

Tables

INFORMATION TECHNOLOGY – FIBRE CHANNEL –

Part 133: Fibre Channel Switch Fabric-3 (FC-SW-3)

FOREWORD

1) ISO (International Organization for Standardization) and IEC (International Electrotechnical Commission) form the specialized system for worldwide standardization. National bodies that are members of ISO or IEC participate in the development of International Standards. Their preparation is entrusted to technical committees; any ISO and IEC member body interested in the subject dealt with may participate in this preparatory work. International governmental and non-governmental organizations liaising with ISO and IEC also participate in this preparation.

2) In the field of information technology, ISO and IEC have established a joint technical committee, ISO/IEC JTC 1. Draft International Standards adopted by the joint technical committee are circulated to national bodies for voting. Publication as an International Standard requires approval by at least 75 % of the national bodies casting a vote.

3) The formal decisions or agreements of IEC and ISO on technical matters express, as nearly as possible, an international consensus of opinion on the relevant subjects since each technical committee has representation from all interested IEC and ISO member bodies.

4) IEC, ISO and ISO/IEC publications have the form of recommendations for international use and are accepted by IEC and ISO member bodies in that sense. While all reasonable efforts are made to ensure that the technical content of IEC, ISO and ISO/IEC publications is accurate, IEC or ISO cannot be held responsible for the way in which they are used or for any misinterpretation by any end user.

5) In order to promote international uniformity, IEC and ISO member bodies undertake to apply IEC, ISO and ISO/IEC publications transparently to the maximum extent possible in their national and regional publications. Any divergence between any ISO/IECpublication and the corresponding national or regional publication should be clearly indicated in the latter.

6) ISO and IEC provide no marking procedure to indicate their approval and cannot be rendered responsible for any equipment declared to be in conformity with an ISO/IEC publication.

7) All users should ensure that they have the latest edition of this publication.

8) No liability shall attach to IEC or ISO or its directors, employees, servants or agents including individual experts and members of their technical committees and IEC or ISO member bodies for any personal injury, property damage or other damage of any nature whatsoever, whether direct or indirect, or for costs (including legal fees) and expenses arising out of the publication of, use of, or reliance upon, this ISO/IEC publication or any other IEC, ISO or ISO/IEC publications.

9) Attention is drawn to the normative references cited in this publication. Use of the referenced publications is indispensable for the correct application of this publication.

International Standard ISO/IEC 14165-133 was prepared by subcommittee 25: Interconnection of information technology equipment, of ISO/IEC joint technical committee 1: Information technology.

The list of all currently available parts of the ISO/IEC 14165 series, under the general title *Information technology – Fibre Channel,* can be found on the IEC web site.

This International Standard has been approved by vote of the member bodies and the voting results may be obtained from the address given on the second title page.

This publication has been drafted in accordance with the ISO/IEC Directives, Part 2.

Introduction

This International Standard describes the requirements for an interconnecting Fabric consisting of multiple Fabric Switch elements to support the ISO/IEC Fibre Channel - Framing and Signaling (FC-FS) and ISO/IEC Fibre Channel - Physical Interface (FC-PI) standards.

FC-SW-3 is one of the Fibre Channel family of standards. ISO/IEC FC-GS-4, is a standard related to Generic Fabric Services and is closely tied to FC-SW-3. ISO/IEC FC-BB-2 describes how Fabrics are extended over transports complementary to Fibre Channel. ANSI/INCITS FC-MI-2 and ANSI/INCITS FC-DA describe interoperability profiles that assists in the interoperability of Switches. ISO/IEC FC-AL-2 specifies the arbitrated loop topology. ISO/IEC FC-SP describes the Security requirements and protocols associated with Fibre Channel networks.

FC-SW-3 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).

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. ISO and IEC shall not be held responsible for identifying any or all such patent rights.

The International Electrotechnical Commission (IEC) and the International Organization for Standardization (ISO) draw attention to the fact that it is claimed that compliance with this document may involve the use of patents as indicated below.

ISO and IEC take no position concerning the evidence, validity and scope of the putative patent rights. The holders of the putative patent rights have assured IEC and ISO that they are willing to negotiate free licences or licences under reasonable and non-discriminatory terms and conditions with applicants throughout the world. In this respect, the statements of the holders of the putative patent rights are registered with IEC and ISO.

Brocade Communications Systems Inc. has informed IEC and ISO that it has patent applications or granted patents as listed below.

6 765 919, 6 980 525 and 10/780 147

Information may be obtained from:
Brocade Communications Systems, Inc.
1745 Technology Drive
USA - San Jose, CA 95110

Cisco Systems Inc. has informed IEC and ISO that it has patent applications or granted patents.

Information may be obtained from:
Cisco Systems Inc.
170 West Tasman Drive
USA - San Jose, CA 95134

McData Corporation has informed IEC and ISO that it has patent applications or granted patents.

Information may be obtained from:
McData Corporation
11802 Ridge Parkway
USA - Broomfield, Colorado 80021

INFORMATION TECHNOLOGY
Fibre Channel —
Part 133: Fibre Channel Switch Fabric - 3 (FC-SW-3)

1 Scope

This part of ISO/IEC 14165-133 describes the operation and interaction of Fibre Channel Switches.

This part of ISO/IEC 14165-133 includes:

a) E_Port Operation and Fabric Configuration;

b) Path selection (FSPF and FSPF-Backbone);

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.

2 Normative references

The following referenced documents are indispensable for the application of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

The provisions of the referenced specifications other than ISO/IEC, IEC, ISO and ITU documents, as identified in this clause, are valid within the context of this International Standard. The reference to such a specification within this International Standard does not give it any further status within ISO or IEC. In particular, it does not give the referenced specification the status of an International Standard.

ISO/IEC 14165-122, *Information technology - Fibre channel - Part 122: Arbitrated loop-2 (FC-AL-2)*

ISO/IEC 14165-241, *Information technology - Fibre channel - Part 241: Backbone-2 (FC-BB-2)*

ISO/IEC 14165-251, *Information technology - Fibre channel - Part 251: Framing and signalling (FC-FS)*

ISO/IEC 14165-414, *Information technology - Fibre channel - Part 414: Generic services-4 (FC-GS-4)*

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.

3 Terms and conventions

For the purposes of this standard, the following terms, definitions, conventions, abbreviations, acronyms, and symbols apply.

3.1 Terms and definitions

3.1.1
Active Zone Set
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
process whereby addresses are dispensed to Switches and Switch Ports

3.1.3
Address identifier
as defined in FC-FS, 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
logical entity within a Switch that is responsible for address assignment

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

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

3.1.7
Adjacent
two Switches that have synchronized their databases are considered Adjacent

3.1.8
Area
second level in a three-level addressing hierarchy

3.1.9
Area Identifier
bits 15 through 8 of an address identifier

3.1.10
AR Number
unique number statically assigned to each autonomous region

3.1.11
AR0
special AR containing only the Switch backbone network, that in general may consist of point-to-point links and switching devices

3.1.12
AR0_SW
Switch inside AR0 and a component of the FSPF-Backbone layer

3.1.13
Autonomous Region
AR
encompasses one or more Fibre Channel Address Domains and consists of Switches all running a common routing protocol; an AR boundary is administratively defined

3.1.14
Border Switch
BSW
FC-SW-3 defined Switch that comprises the FSPF-Backbone network

3.1.15
Broadcast Address
FFFFFFh value in the D_ID field specifies that the frame be broadcast to all Nx_Ports

3.1.16
Broadcast Zone
zone with the Broadcast attribute specified

3.1.17
Broadcast Zoning Enforcement
zoning technique where the Fabric limits broadcast distribution among zone members using frame-by-frame filtering techniques

3.1.18
B_Port
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

3.1.19
Class F service
service that multiplexes frames at frame boundaries and is used for control and coordination of the internal behavior of the Fabric

3.1.20
Class N service
refers to any class of service other than Class F

3.1.21
Domain
highest level in a three-level addressing hierarchy

3.1.22
Domain Address Manager
Switch that is responsible for address assignment to other Switches outside of its Domain

3.1.23
Domain Identifier
bits 23 through 16 of an address identifier

3.1.24
Domain_ID_List
list where each record contains a Domain_ID value and the Switch_Name of the Switch assigned the Domain_ID (see 6.1.5)

3.1.25
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

3.1.26
Distributed Service
implementation of a Well-Known Service where certain components or functions of the service are distributed throughout the Fabric

3.1.27
Distributed Services Time-Out Value
D_S_TOV
value that indicates the maximum time that a distributed service requestor waits for a response

3.1.28
Entry Switch
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

3.1.29
E_Port
Fabric “Expansion” Port that attaches to another Interconnect_Port to create an Inter-Switch Link

3.1.30
E_Port Index
index value associated with an E_Port used by the Fabric Shortest Path First Protocol

3.1.31
Error_Detect_Timeout value
E_D_TOV
time constant defined in FC-FS

3.1.32
F_Port
as defined in FC-FS; in this standard an F_Port is assumed to always refer to a port to which non-loop N_Ports are attached to a Fabric, and does not include FL_Ports

3.1.33
Fabric
as defined in FC-FS, 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

3.1.34
Fabric Controller
logical entity responsible for operation of the Fabric identified by the well-known address FFFFFDh

3.1.35
Fabric Element
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

3.1.36
Fabric F_Port
entity at the well-known address FFFFFEh (see FC-FS)

3.1.37
Flood
to cause information to be sent to all Switches within the Fabric

3.1.38
FL_Port
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

3.1.39
Fx_Port
Switch Port capable of operating as an F_Port or FL_Port

3.1.40
Fabric_Stability_Timeout value
F_S_TOV
time constant used to ensure that Fabric stability has been achieved during Fabric Configuration

3.1.41
Fabric Shortest Path First
FSPF
link state protocol used for Path Selection

3.1.42
Fibre Channel Address Domain
set of Domain_IDs associated with an AR

3.1.43
FSPF-Backbone Network
FC-SW-3 defined network backbone connecting different ARs via BSWs using point-to-point links and AR0_SW Switching devices

3.1.44
FSPF-Backbone Protocol
FC-SW-3 defined Switch routing and control protocol that runs over an FSPF-Backbone network

3.1.45
G_Port
generic Fabric Port that may function either as an E_Port, or as an F_Port

3.1.46
GL_Port
generic Fabric Port that may function either as an E_Port, or as an Fx_Port

3.1.47
Hard Zone
zone with the Hard Zone attribute specified

3.1.48
Hard Zoning Enforcement
zoning technique in which the Fabric limits frame exchange by frame-by-frame filtering

3.1.49
Interconnect_Port
generic reference to an E_Port or a B_Port

3.1.50
Intermix
as defined in FC-FS

3.1.51
Inter-Switch Link
ISL
Link directly connecting the E_Port of one Switch to the E_Port of another Switch

3.1.52
Isolated
condition in which it has been determined that no Class N traffic may be transmitted across an ISL (see 7.5)

3.1.53
L_Port
port that contains Arbitrated Loop functions associated with the Arbitrated Loop topology

3.1.54
Link
as defined in FC-FS

3.1.55
Loop Fabric Address
address identifier used to address an FL_Port for purposes of loop management (see FC-FLA)

3.1.56
N_Port
as defined in FC-FS. In this Standard, an N_Port is assumed to always refer to a direct Fabric-attached port, and does not include NL_Ports

3.1.57
N_Port Identifier
address identifier assigned to an N_Port

3.1.58
Name_Identifier
as defined in FC-FS, a 64-bit identifier

3.1.59
NL_Port
L_Port that is able to perform the function of an N_Port, 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

3.1.60
Non-zero Domain_ID_List
Domain_ID_List that contains at least one record (see 7.3)

3.1.61
Nx_Port
Port operating as an N_Port or NL_Port

3.1.62
Path
route through the Fabric between a source and a destination

3.1.63
Path Selection
process whereby paths are selected

3.1.64
Port
generic reference to an N_Port, NL_Port, F_Port, FL_Port, B_Port, or E_Port; the lowest level in a three-level addressing hierarchy

3.1.65
Point-to-Point Link
Fibre Channel link connecting two ports

3.1.66
Port Identifier
bits 7 through 0 of an address identifier

3.1.67
Port Index
three byte value used by FSPF to identify Switch ports

3.1.68
Port Mode
generic reference to E_Port, B_Port, F_Port or FL_Port operation

3.1.69
Preferred Domain_ID
Domain_ID previously granted to a Switch by the Domain Address Manager or through administrative means

3.1.70
Principal ISL
Inter-Switch Link that is used to communicate with the Principal Switch

3.1.71
Principal Switch
Switch that has been selected to perform certain Fabric Configuration duties

3.1.72
Reliable Flood
flooding where all Switches are guaranteed to receive the flooded message

3.1.73
Remote Switch
switch that may be reached via one or more ISLs; a remote Switch may be adjacent to the local Switch, or may be reached via one or more intermediate Switches

3.1.74
Resource_Allocation_Timeout value
R_A_TOV
time constant defined in FC-FS

3.1.75
Router
entity within a Switch responsible for the routing of connectionless frames

3.1.76
Routing
process whereby the appropriate Switch Port(s) to deliver a connectionless frame towards its destination is identified

3.1.77
Soft Zoning Enforcement
zoning technique in which the Fabric enforces membership through name server visibility

3.1.78
Switch
a Fabric Element conforming to this Standard and a member of the Fabric collective

3.1.79
Switch Construct
entity within a Switch responsible for transporting frames between Switch Ports

3.1.80
Switch_Name
Name_Identifier (see FC-FS) that identifies a Switch or a Bridge device for identification purposes; each Switch and Bridge device provides a unique Switch_Name within the Fabric

3.1.81
Switch Port
E_Port, F_Port, or FL_Port

3.1.82
Switch_Priority
value used during Principal Switch selection to cause one Switch to be favored over another

3.1.83
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

3.1.84
Zero Domain_ID_List
Domain_ID_List that is empty (see 7.3)

3.1.85
Zone
group of Zone Members. Members of a Zone are made aware of each other, but not made aware of Zone Members outside the Zone

3.1.86
Zone Definition
parameters that define a Zone

3.1.87
Zone Member
specification of a device to be included in a Zone

3.1.88
Zone Member Definition
parameters that define a Zone Member including the Zone Member Type and Zone Member Information

3.1.89
Zone Name
name assigned to a Zone

3.1.90
Zone Set
set of Zones that are used in combination

3.1.91
Zone Set Database
database that contains the Zone Sets not enforced by the Fabric

3.1.92
Zone Set Name
name assigned to a Zone Set

3.1.93
Zone Set State
state of a Switch Zone Set (*Activated* or *Deactivated*)

3.1.94
Zoning Configuration
set of Zoning data including the Zone Set state, and Zone definitions

3.1.95
Zoning Database
generic term used to indicate both the Active Zone Set and the Zone Set Database

3.2 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.

Area_ID	Area Identifier
AR	Autonomous Region
AR0	Autonomous Region 0
BSW	Border Switch
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
FC-BB-2	Fibre Channel Backbone-2
FC-DA	Fibre Channel- Device Attach
FC-FLA	Fibre Channel - Fabric Loop Attachment
FC-FS	Fibre Channel - Framing and Signaling
FC-GS-4	Fibre Channel - Generic Services-4
FC-MI-2	Fibre Channel - Methodologies for Interconnects-2
F_S_TOV	Fabric_Stability_Timeout value
FDMI	Fabric Device Management Interface
FSPF	Fabric Shortest Path First
ISL	Inter-Switch Link
IU	Information Unit
LFA	Loop Fabric Address
LSR	Link State Record
Port_ID	Port Identifier
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
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

3.3 Editorial conventions

In this Standard, 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, Class). 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 priority relationship between the listed items. Numbered lists (e.g., 1-red, 2-blue, 3-green) show a priority ordering 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 standard, the most significant bit of a binary quantity is shown on the left side. Exceptions to this convention are indicated in the appropriate clauses.

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.

If a field or a control bit in a frame is specified as reserved, the entity that sends the frame shall set the field or control bit to zero, and the entity that receives the frame shall not check that field or control bit.

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.

Unless stated otherwise: 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; and numbers or upper case letters immediately followed by lower-case h (xxh) are hexadecimal values.

The ISO convention of numbering is used (i.e., the thousands and higher multiples are separated by a space and a comma is used as the decimal point). A comparison of the American and ISO conventions are shown in table 1.

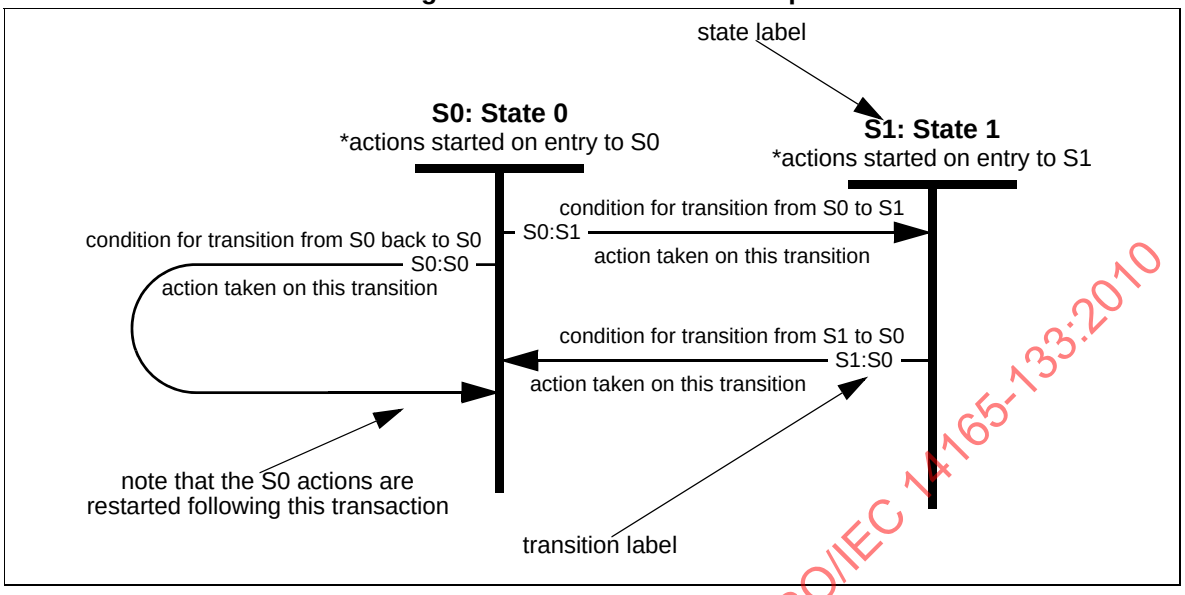
Table 1 – ISO and American Conventions

ISO	American
0,6	0.6
1 000	1,000
1 234 567,8	1,234,567.0

3.4 State machine notation

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

Figure 1 – State Machine Example



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.

3.5 Definition of Compliance Terms

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

3.5.1
Prohibited
if a feature or parameter value is Prohibited, it means that it shall not be used between compliant implementations

3.5.2
Required
if a feature or parameter value is Required, it means that it shall be used between compliant implementations

3.5.3
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
expected
keyword used to describe the behavior of the hardware or software in the design models assumed by this standard. Other hardware and software design models may also be implemented

3.6.2
ignored
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.

3.6.3
invalid
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.

3.6.4
mandatory
keyword indicating an item that is required to be implemented as defined in this standard

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

3.6.6
may not
keyword that indicates flexibility of choice with no implied preference (equivalent to “may or may not”)

3.6.7
obsolete
keyword indicating that an item was defined in prior Fibre Channel standards but has been removed from this standard

3.6.8
optional
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.

3.6.9
reserved
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.

3.6.10
restricted
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.

3.6.11
shall
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

4 Structure and Concepts

4.1 Overview

This FC-SW-3 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 N_Port and F_Port, as defined in FC-FS, 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, and one or more Class 1, 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 and the FSPF-Backbone protocol. FSPF describes how Switches interact to establish path selection for an Intra-Region and the FSPF-Backbone protocol describes Inter-Region routing.

e) Distributed Server communication - The Distributed Server model allows the Well-Known servers 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.

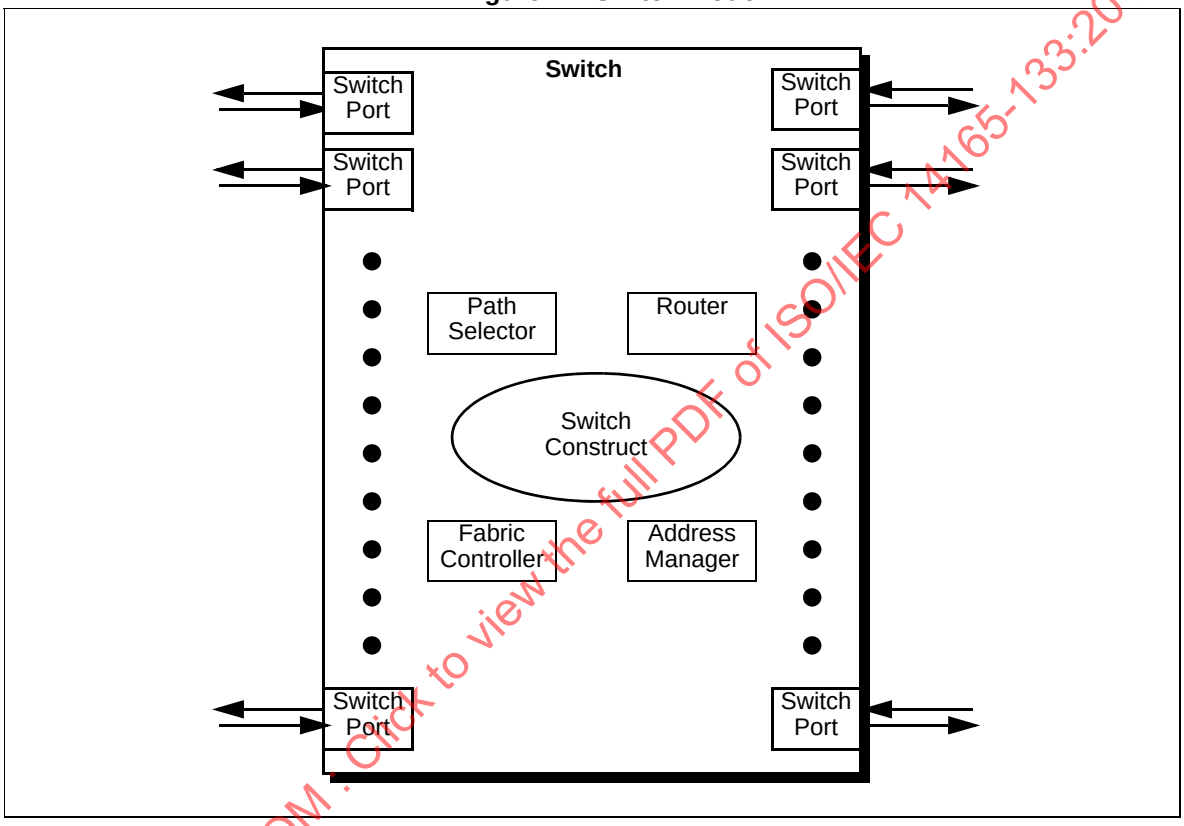
4.4 Fabric Definition

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 Switch.

Figure 2 – 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;

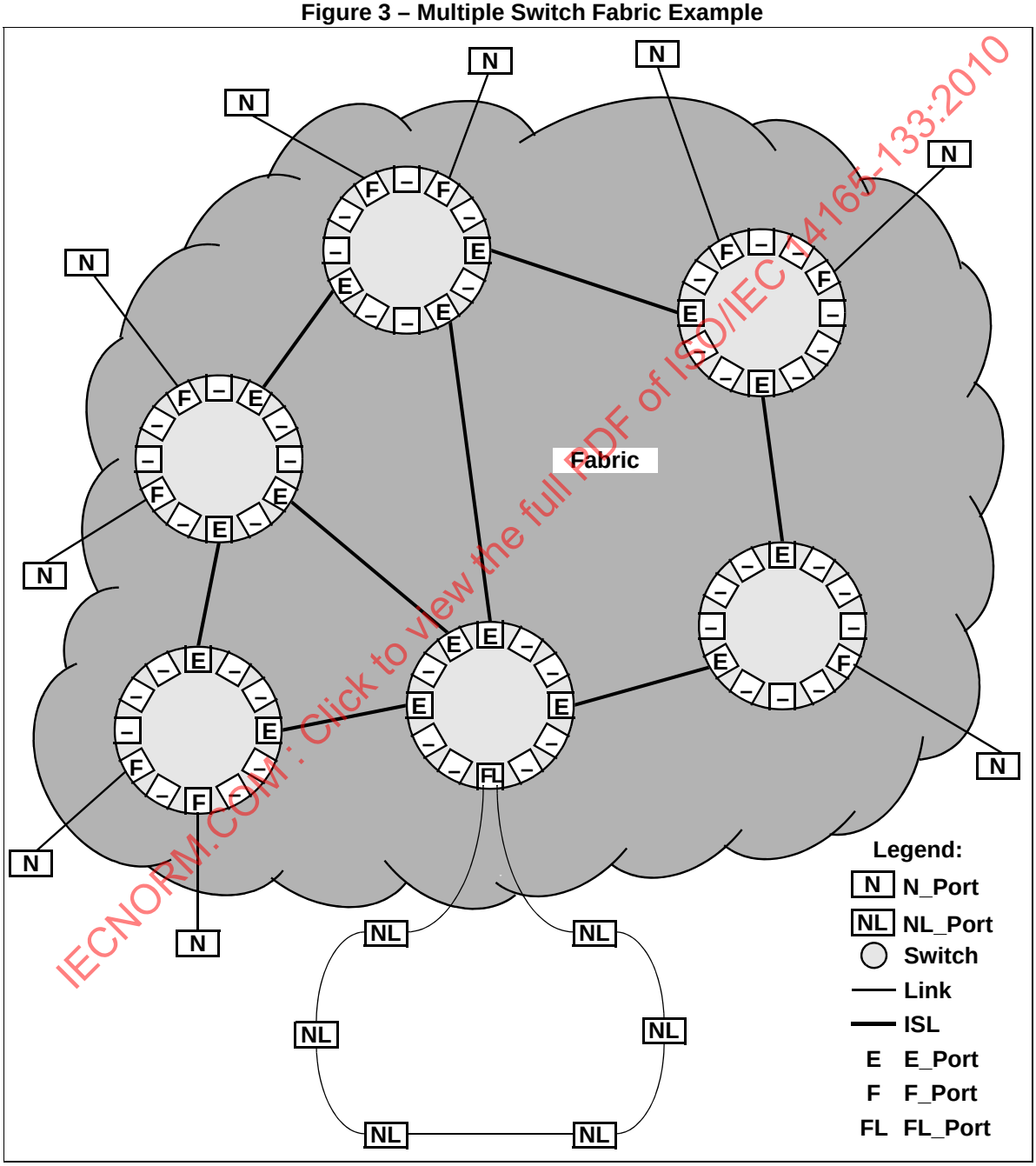
f) and a Fabric Controller.

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 3.

Figure 3 – 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 and 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 may occur synchronously or asynchronously 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. Asynchronous switching does not guarantee retention of word alignment on the outbound fibre of the Switch Port.

A Switch may employ either synchronous or asynchronous switching or a combination of the two (e.g., a Switch may use synchronous switching for Class F, Class 2 and Class 3, and asynchronous switching for Class 1). However, a Switch shall never mix the two within a given Class of Service.

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

4.6.2 Synchronous switching

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. Switches should use synchronous switching in support of Class 2, Class 3 and Class F service.

4.6.3 Asynchronous switching

Asynchronous switching may be performed any time Fill Words are being transmitted. Bit alignment and word alignment may be lost when an asynchronous switching event occurs. A recovery time that allows the attached Port time to regain synchronization shall be inserted before frame transmission resumes for the outbound fibre. Fill Words shall be transmitted during this recovery time. If conditions arise warranting transmission of a Primitive Sequence, then this should take precedence over transmission of Fill Words.

If a Switch or Node Port recognizes that it is linked to a Switch that employs asynchronous switching, and a permissible word realignment event occurs, then the Port may ignore any resulting errors (i.e., not log errors resulting from the realignment event).

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), depending on which of the four Port Modes are supported by the Switch Port. A Bridge Port shall only support B_Port operation.

NOTE 1 The characteristics of a Bridge device are not described in this standard. See FC-BB-2 for a description of Bridge device characteristics.

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.

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.

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.

4.8 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 4.

Figure 4 – Domain, Area, and Port Address Partitioning

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																							

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 2. Any value listed as Reserved is not meaningful within this Standard.

Table 2 – Address Identifier Values

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 FL_Port: Loop Fabric Address[c]
01 - EF	**00 - FF**	**AL_PA**	F_Port: N_Port Identifier FL_Port: N_Port Identifier for Public Loop NL_Port[c]
01 - EF	**00 - FF**	**non-AL_PA**	F_Port: N_Port Identifier FL_Port: Reserved
F0 - FE	**00 - FF**	**00 - FF**	Reserved
FF	**00 - FA**	**00 - FF**	Reserved
FF	**FB**	**00 - FF**	Reserved for Multicast Group_ID
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]
FF	**FF**	**FD**	N_Port Identifier for Fabric Controller[f]
FF	**FF**	**FE**	N_Port Identifier for Fabric F_Port
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-DA 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 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.

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.

4.10 FSPF-Backbone Fabric

The FSPF-Backbone Fabric consists of multiple Autonomous Regions (AR) that are interconnected by a backbone network. The FSPF-Backbone Fabric performs path selection using the FSPF-Backbone routing protocol. All ARs shall have unique Domain_IDs and be equipped with dynamic mechanisms to detect conflicting Domain_IDs. Assigning Domain_IDs within an AR does not necessarily have to be dynamic as in Principal Switch selection and Address assignment, as long as the links of the conflicting Domain_ID are isolated. The AR boundary is administratively defined, with each AR encompassing one or more Fibre Channel Address Domains.

The primary motivation in forming a 2-tier architecture is to allow the Switch Fabric to scale better. In general a layered routing architecture is known to incur shorter propagation delays compared to a flat network. This is particularly important for Fibre Channel networks with a large number of Switches.

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 Transport includes all of the functionality described in FC-FS 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.

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 N_Port is attached to the Fabric. It normally functions as a conduit to the Fabric for frames transmitted by the N_Port, and as a conduit from the Fabric for frames destined for the N_Port.

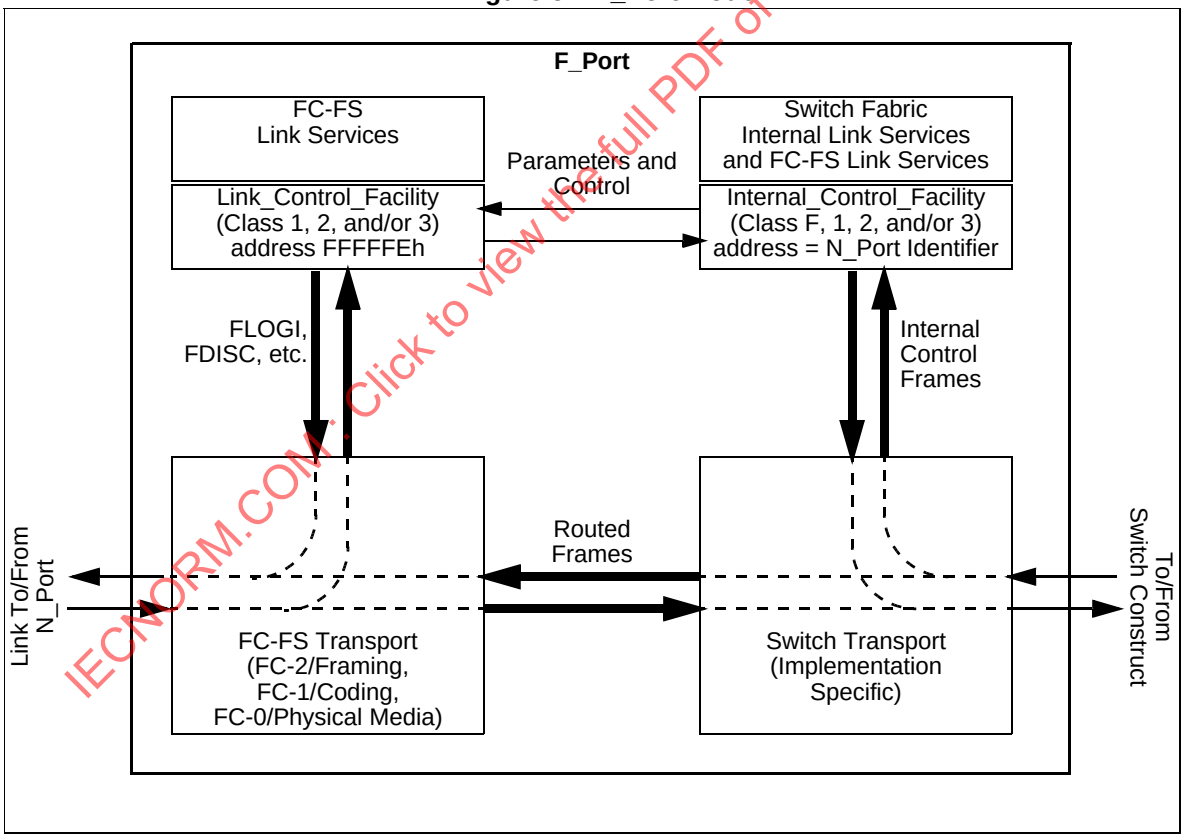
An F_Port shall support one or more of the following classes of service: Class 1 service, Class 2 service, Class 3 service. An F_Port shall not intentionally transmit Class F frames on its outbound fibre. An N_Port that receives a Class F frame shall discard it, as required by FC-FS.

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 2 Primitive Signals and Primitive Sequences are prohibited from entering the Fabric by FC-FS. 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 is shown in figure 5.

Figure 5 – F_Port Model



An F_Port contains an FC-FS Transport element through which passes all frames and Primitives transferred across the Link to and from the N_Port. Frames received from the N_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 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 Transport for transmission to the N_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 N_Port. A Link to an F_Port always connects to exactly one N_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.

5.4 FL_Port Operation

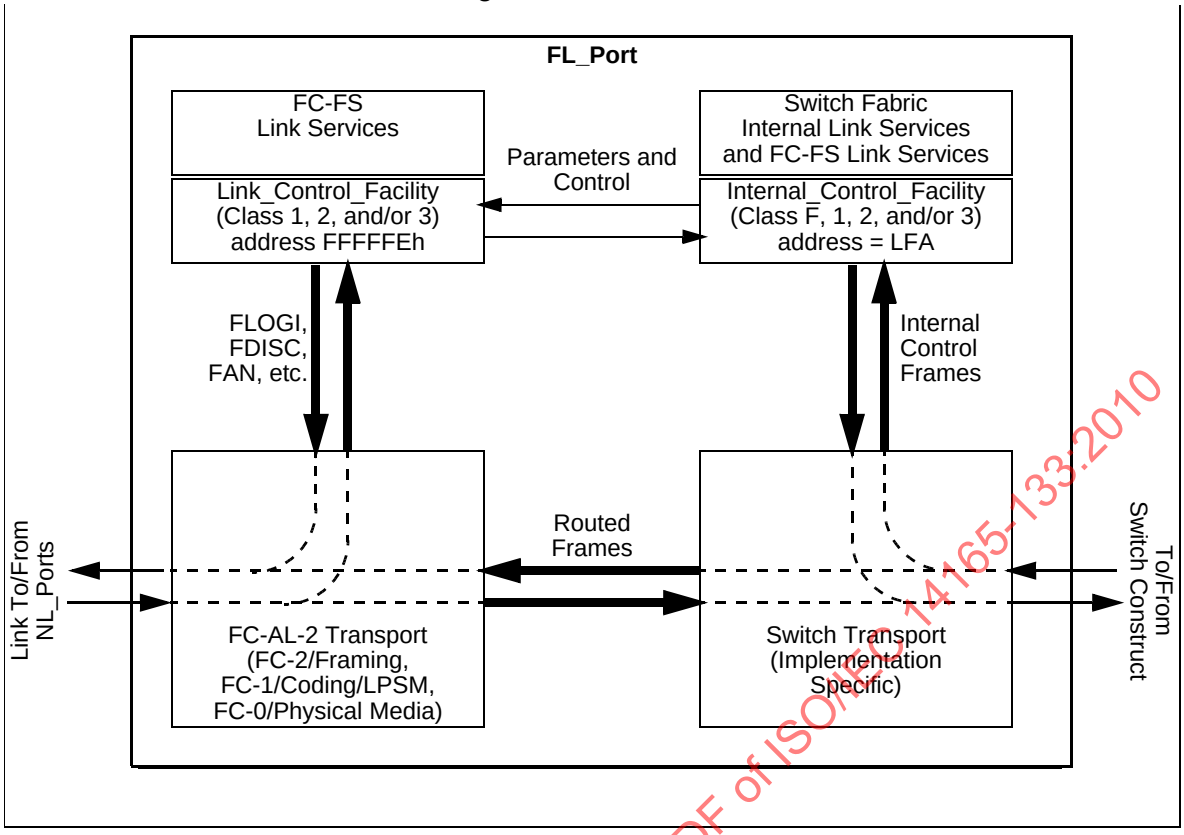
An FL_Port is the point at which one or more external NL_Ports are attached to the Fabric. It normally functions as a conduit to the Fabric for frames transmitted by the attached NL_Ports, and as a conduit from the Fabric for frames destined for the attached NL_Ports.

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 3 It is recommended that an FL_Port that conforms to this Standard also conform to the FL_Port requirements defined in FC-DA.

The model of an FL_Port is shown in figure 6.

Figure 6 – 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-DA), 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, with the additional Arbitrated Loop protocols defined in FC-AL-2.

5.5 E_Port Operation

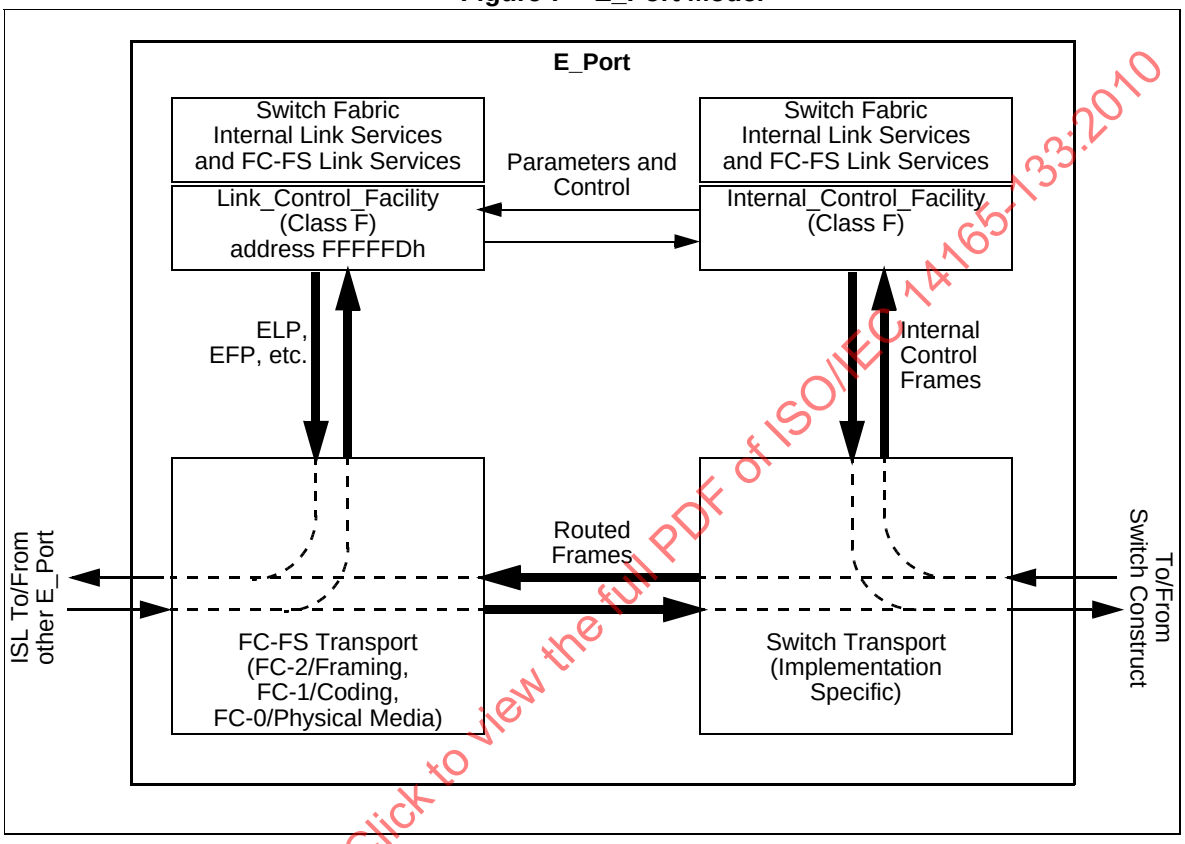
An E_Port is the point at which a Switch is connected to another Switch to create a multi-Switch Fabric. Also, an E_Port is the point at which a Switch is connected to a Bridge device. It normally functions as

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 1 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 is shown in figure 7.

Figure 7 – E_Port Model



An E_Port contains an FC-FS 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 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 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.

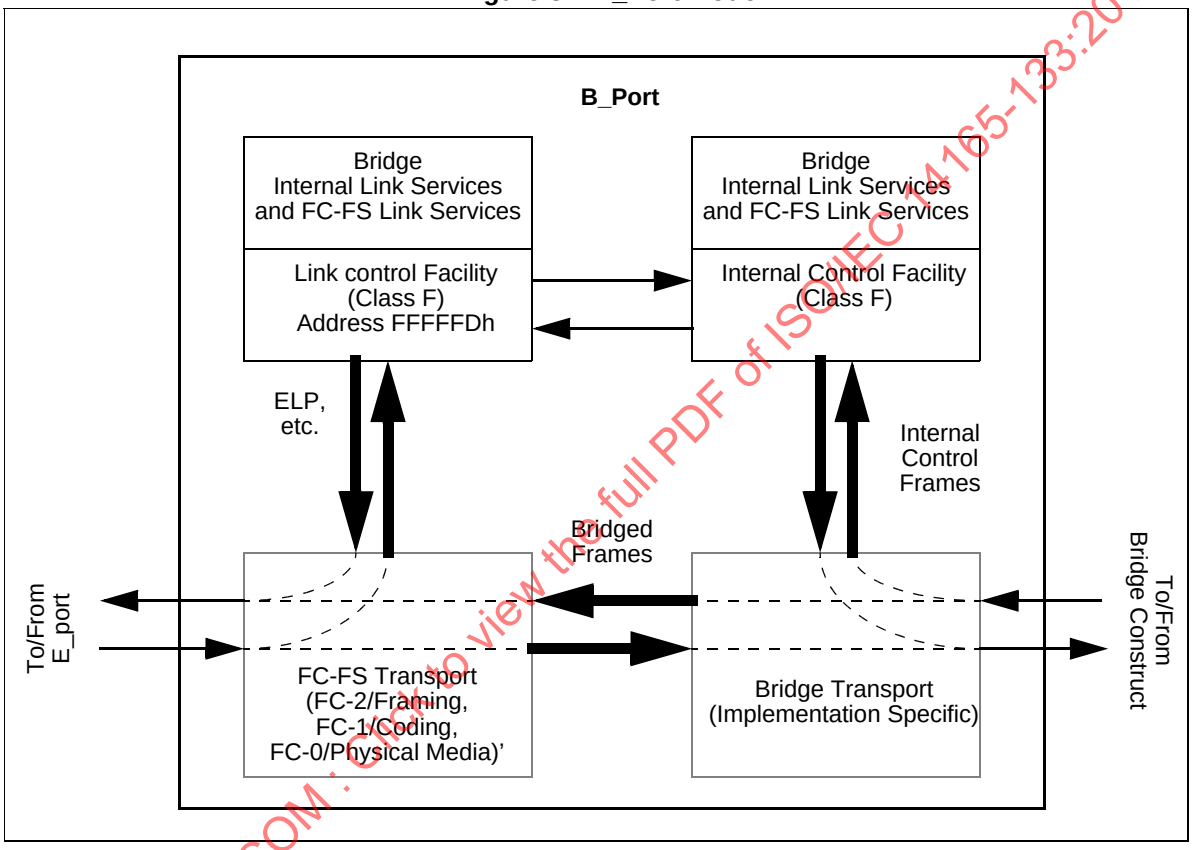
5.6 B_Port operation

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 1 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 8.

Figure 8 – B_Port Model



A B_Port contains an FC-FS 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 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 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 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 follows the FC-0, FC-1, and FC-2 protocols defined for point-to-point Links as defined in FC-FS, 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.

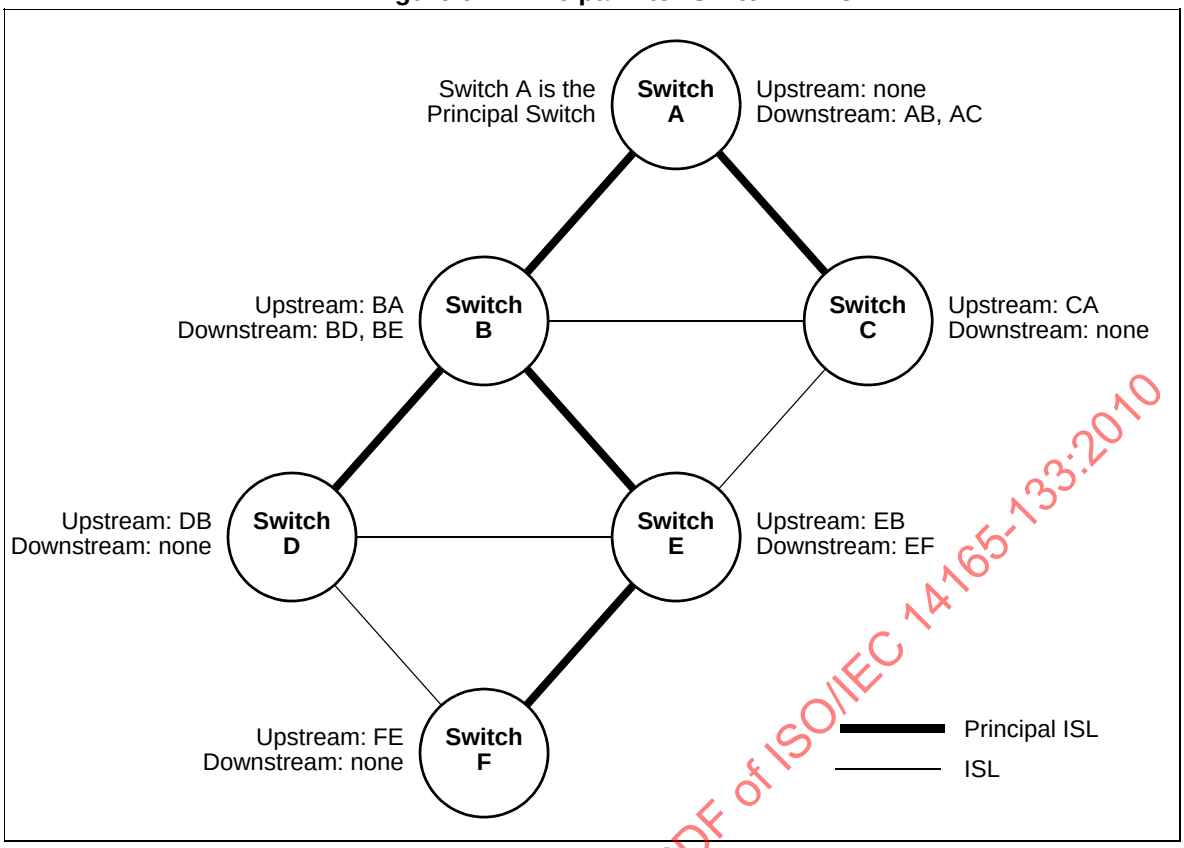
For purposes of defining and maintaining the Fabric Configuration, an ISL may be designated as a Principal ISL. The Principal ISL is a path that is used during configuration and address assignment to route Class F configuration frames, and is therefore a known path between two Switches. If a Principal 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 9.

Figure 9 – 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.

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 F_BSY 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. (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 support insertion of Class F frames onto an established Class 1 Dedicated Connection. However, this insertion shall not cause loss of any Class 1 frames. A Switch may abort (EOFa) or discard an Intermixed Class 2 or Class 3 frame in progress if its transmission of a Class F frame interferes. A Switch shall not abort an Inserted Class F frame. (Class F intermix)

s) An Interconnect_Port shall use R_RDY and FC-FS 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. (Alternate credit models)

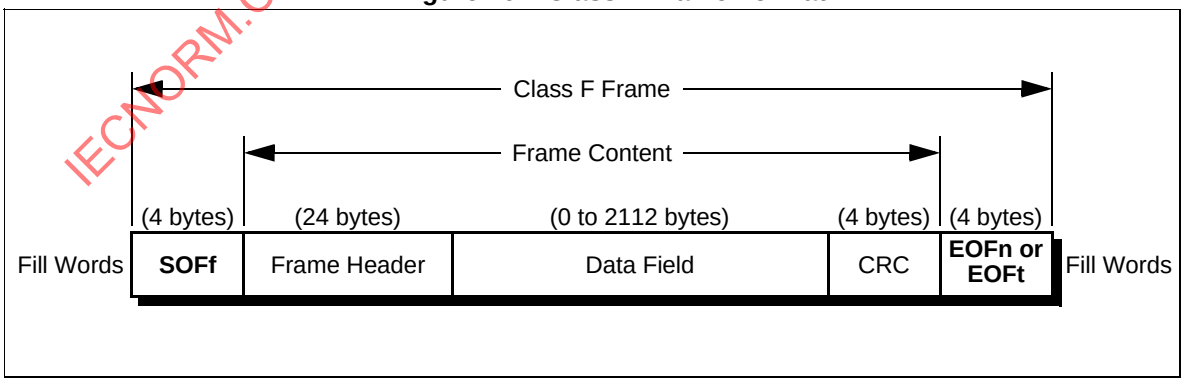
t) A Class F frame shall be forwarded to its destination without checking by an intermediate entity. A Class F frame not destined for the receiving E_Port (or E_Port's domain) shall always be forwarded regardless of whether or not the receiving E_Port recognizes the frame. (Frame forwarding)

5.8.3 Class F Frame Format

Class F frames shall use the Frame_Header defined in FC-FS. The Class F frame format is shown in figure 10. 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. 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. A Class F frame is preceded and followed by Primitive Signals as defined in FC-FS.

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.

Figure 10 – Class F Frame Format



5.8.4 Class F Flow Control

Class F service uses both buffer-to-buffer and end-to-end flow controls. 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.

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.

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 FT_1 frame format via 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. The first word in the payload specifies the Command Code. The Command Codes are summarized in table 3.

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

Encoded Value (hex)	Description	Abbr.
01 00 00 00	Switch Fabric Internal Link Service Reject	SW_RJT
02 xx xx xx	Switch Fabric Internal Link Service Accept	SW_ACC
10 00 00 00	Exchange Link Parameters	ELP
11 xx xx xx	Exchange Fabric Parameters	EFP
12 00 00 00	Domain Identifier Assigned	DIA
13 00 xx xx	Request Domain_ID	RDI
14 00 00 00	Hello	HLO
15 00 00 00	Link State Update	LSU
16 00 00 00	Link State Acknowledgement	LSA
17 00 00 00	Build Fabric	BF
18 00 00 00	Reconfigure Fabric	RCF
1B 00 00 00	Inter-Switch Registered State Change Notification	SW_RSCN
1E 00 00 00	Distribute Registered Link Incident Records	DRLIR
20 00 00 00	Disconnect Class 1 Connection	DSCN
21 00 00 00	Obsoleted in FC-SW-3	LOOPD

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

Encoded Value (hex)	Description	Abbr.
22 00 xx xx	Merge Request	MR
23 00 xx xx	Acquire Change Authorization	ACA
24 00 00 00	Release Change Authorization	RCA
25 xx xx xx	Stage Fabric Configuration	SFC
26 00 00 00	Update Fabric Configuration	UFC
28 xx xx xx	Reserved for FC-BB-2 Use	
30 00 xx xx	Exchange Switch Capabilities	ESC
31 00 00 00	Exchange Switch Support	ESS
32 00 00 00	Reserved for FC-SP Use	
33 00 00 00	Reserved for FC-SP Use	
34 00 00 00	Merge Request Resource Allocation	MRRA
40 xx xx xx	Reserved for FC-SP Use	
41 xx xx xx	Reserved for FC-SP Use	
70 00 00 00 to 7F FF FF FF	Vendor Specific	
90 00 00 00 to 9F FF FF FF	Vendor Specific	
others	Reserved	

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 (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 12.2.

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.

Format: FT–1

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.

Format: FT–1

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 4.

Table 4 – 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 5.

Table 5 – 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 6.

Table 6 – SW_RJT Reason Code Explanation (part 1 of 2)

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
2E	Requested Domain_IDs not available
2F	E_Port is Isolated
31	Authorization Failed[a]
32	Authentication Failed
33	Incompatible Security Attribute
34	Checks in Progress
41	Invalid Data Length[b]
42	Unsupported Command
44	Not Authorized

[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 6 – SW_RJT Reason Code Explanation (part 2 of 2)

Encoded Value (hex)	Description
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
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.

6.1.4 Exchange Link Parameters (ELP)

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

Format: FT–1

Error Detection and Recovery: See table 165.

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 7.

Table 7 – 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
Class F Service Parameters	16
Class 1 Interconnect_Port Parameters	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 protocol. This revision has the value of 3.

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

Bits 14-0 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 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. 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 4 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. 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. Each Switch_Name shall be unique within the Fabric.

Class F Service Parameters: This field contains the E_Port Class F Service Parameters. The format of the Parameters is shown in table 8.

Table 8 – Interconnect_Port 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 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 Interconnect_Port 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 corre-

sponding 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 an FT_1 frame that may be received by the Interconnect_Port supplying the Parameters as a Sequence Recipient for a Class F frame. Values less than 256 or greater than 2 112 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 Interconnect_Port 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. An Interconnect_Port 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 an Interconnect_Port 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 Interconnect_Ports 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 1 Interconnect_Port Parameters: This field contains the Class 1 Interconnect_Port Parameters. The format of the Parameters is shown in table 9.

Table 9 – Class 1 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	IMX	XPS	LKS	Reserved												Receive Data Field Size															

The Class 1 Interconnect_Port Parameters are defined as follows:

a) VAL (Class Valid): This bit is set to one if the Interconnect_Port supports Class 1. If this bit is zero, all other Class 1 Interconnect_Port Parameters shall be invalid.

b) IMX (Intermix): This bit is set to one if the Interconnect_Port may perform Intermix as defined in FC-FS. Intermix shall be functional only if both Interconnect_Ports indicate support for this feature.

c) XPS (Transparent Mode Stacked Connect Request): This bit is set to one if the Interconnect_Port may perform Transparent Mode Stacked Connect Requests as defined in FC-FS. Transparent Mode Stacked Connect Requests shall be functional only if both Interconnect_Ports indicate support for this feature. A Switch shall not indicate support for both XPS and LKS.

d) LKS (Lock-down Mode Stacked Connect Request): This bit is set to one if the Interconnect_Port may perform Lock-down Mode Stacked Connect Requests as defined in FC-FS. Lock-down Mode Stacked Connect Requests shall be functional only if both Interconnect_Ports indicate support for this feature. A Switch shall not indicate support for both XPS and LKS.

e) Receive Data Field Size: This field shall specify the largest Data Field size in bytes for an FT_1 frame that may be received by the Interconnect_Port supplying the Parameters for a Class 1 frame. Values less than 256 or greater than 2 112 are invalid. Values shall be a multiple of four bytes.

Class 2 Interconnect_Port Parameters: This field contains the Class 2 Interconnect_Port Parameters. The format of the Parameters is shown in table 10.

Table 10 – 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) 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 an FT_1 frame that may be received by the Interconnect_Port supplying the Parameters for a Class 2 frame. Values less than 256 or greater than 2 112 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 11.

Table 11 – 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) 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 an FT_1 frame that may be received by the Interconnect_Port supplying the Parameters for a Class 3 frame. Values less than 256 or greater than 2 112 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 12 shows the allowed values for this field.

Table 12 – ISL Flow Control Mode Values

Value (hex)	Usage
0001	Vendor Specific
0002	R_RDY Flow Control
0003 - 1FFF	Vendor Specific
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.

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) 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 13. Values other than 0002h for ISL Flow Control Mode are not required to follow the Flow Control Parameter Field format shown in table 13.

Table 13 – 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 1 connect-request, 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 Class 1 SOFC1 frames, 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 5 Recommended Compatibility Parameter values are described in FC-MI-2.

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 14.

Table 14 – 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
Class F Service Parameters	16
Class 1 Interconnect_Port Parameters	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 14 are the same as defined in table 7.

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

Format: FT–1

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 15.

Table 15 – 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
Multicast_ID_List	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 65 532.

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 16.

Table 16 – 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 17.

Table 17 – 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 18.

Table 18 – Record_Type Field Values

Value (hex)	Description
00	Reserved
01	Domain_ID_List record
02	Multicast_ID_List record
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.

Multicast_ID_List: This field shall contain a list of records that specify the Multicast_Group_number of the Multicast_Group granted by the Principal Switch. The Multicast_ID_List shall contain a record for each value of Multicast_Group_number that has been assigned. If no Multicast_Group_number has been assigned, the Multicast_ID_List shall contain no records. The format of a Multicast_ID_List record is shown in table 19.

Table 19 – Multicast_ID_List Record Format

Item	Size Bytes
Record_Type	1
Multicast_Group_number	1
Reserved	2
Reserved	12

Multicast_Group_number: This field shall specify the Multicast_Group_number assigned 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 20.

Table 20 – 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 20 are the same as defined for table 15 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 an 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

Format: FT–1

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 21.

Table 21 – 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 22.

Table 22 – 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

Format: FT–1

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 23.

Table 23 – 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 24**.**

Table 24 – 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. This field is set to either the Preferred Domain_ID specified in the Request if it is available, another Domain_ID if the Preferred Domain_ID is not available, or the preferred Domain_ID 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 6 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

Format: FT–1

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 25.

Table 25 – 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 164).

Dead_Interval: This field shall specify in seconds, the maximum interval the requesting Switch shall wait for the neighbor to send its Hello. If the interval expires and no Hello has been received, then the detecting Switch shall bring down the Adjacency (see table 164).

NOTE 7 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 26.

Table 26 – FSPF Header

Item	Size Bytes
Command	4
FSPF Version	1
AR Number	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 27.

Table 27 – 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.

AR Number: This field contains the Autonomous Region of the Switch. When using the FSPF-Backbone protocol, this value shall be set to 00h for the FSPF-Backbone, and a non-zero value for all other ARs. When not using the FSPF-Backbone, this value may be set to any value, including 00h.

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

Format: FT–1

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 28.

Table 28 – 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 topology database. The bit map values are listed in table 29.

Table 29 – 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 are two formats for the LSR. They are the Link Descriptor Format and the Summary Descriptor Format. The two formats are shown in table 30 and table 31 respectively. The Summary Descriptor LSR is only used when FSPF-Backbone routing protocol is being used. One or more descriptors may be contained in a single LSR, but each descriptor in the LSR shall be of the same type.

Table 30 – 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 8 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.

Table 31 – Link State Record - Summary Descriptor Format

Item	Size Bytes
Link State Record Header (LSR Type 02h)	24
Reserved	2
Number of Summaries	2
Summary Descriptor #1	8
...	8
...	8
Summary Descriptor #n	8

Number of Summaries: This field specifies the number of summary descriptors contained in the Link State Record.

Summary Descriptor: The format of the Summary Descriptor is described in 6.1.9.5.

6.1.9.3 Link State Header Format

The format of the Link State Header is described in table 32.

Table 32 – 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 33.

Table 33 – Link State Record Type Field Values

Value (hex)	Description
01	Switch Link Record
02	AR Summary Record
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 9 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 10 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 34.

Table 34 – 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 35.

Table 35 – 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.

6.1.9.5 Summary Descriptor Format

This field contains a descriptor that identifies a Domain_ID and its link cost, that may be reached via that ISL. The Summary Descriptor shall only be used when the FSPF-Backbone routing protocol is being used.

Table 36 – Summary Descriptor Format

Item	Size Bytes
Reserved	1
AR Number	1
Domain_ID	4
Link Cost	2

AR Number: Designates the AR whose summary information is being advertised. This information is used to build the associations between ARs and Domain_IDs. The AR designated by the AR number is located behind the BSW originating the Summary Record.

Domain_ID: The Domain_ID of the remote Switch for which link data is being summarized. The format of Domain_ID field shall be set to 000000hllDomain_ID'.

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

Format: FT–1

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 11 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 a loss 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

Format: FT–1

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 12 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

Format: FT–1

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) unused nibble in the address format field is set to 1 if the port is online, or 2 if the port is offline. This 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

Format: FT_1

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) 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 13 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

Format: FT_1

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-FS.

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 Disconnect Class 1 Connection (DSCN)

The Disconnect Class 1 Connection Switch Fabric Internal Link Service requests that the receiving E_Port abort an existing Class 1 Connection. This SW_ILS is used only if Link Failure or Link Reset is detected in the connection path. An F_Port that receives this SW_ILS shall perform a Link Reset to abort the connection with the attached N_Port. An FL_Port shall perform a LIP when DSCN is received.

NOTE 14 Normal disconnect should be performed by detecting EOFdt.

Protocol:

Disconnect Class 1 Connection (DSCN) request Sequence
Accept (SW_ACC) Reply Sequence

Format: FT–1

Addressing: The S_ID field shall be set to the address identifier of the sending E_Port. The D_ID field shall be set to the address identifier of the destination E_Port or F_Port.

Payload: The format of the DSCN request Payload is shown in table 47.

Table 47 – DSCN Request Payload

Item	Size Bytes
20000000h	4
Reserved	3
Reason code for disconnect	1

Reason code for disconnect: This field specifies the reason for the disconnect, summarized in table 48.

Table 48 – DSCN Reason Codes

Encoded Value (Bits 7-0)	Description
0000 0001	Link Failure or Link Reset occurred
others	Reserved
1111 1111	Vendor Specific error

Reply Switch Fabric Internal Link Service Sequence:

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

Payload: The format of the DSCN accept Payload is shown in table 49.

Table 49 – DSCN Accept Payload

Item	Size Bytes
02000000h	4

6.1.16 Merge Request (MR)

6.1.16.1 MR overview

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 162. 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, otherwise 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

Format: FT_1

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 6.1.16.2.

6.1.16.2 Merge Request Payload

6.1.16.2.1 Merge Request Payload overview

The format of the Merge Zone request payload is depicted in table 50.

Table 50 – 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 51 depicts the defined values.

Table 51 – Protocol Version Values

Value	Meaning
00	Basic Zoning
01	Enhanced Zoning
others	Reserved

6.1.16.2.2 Merge Request Payload in Basic Zoning

The format of the Version Specific payload for Protocol Version = 00h is depicted in table 52.

Table 52 – 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[a]	y

[a] 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.

6.1.16.2.3 Merge Request Payload in Enhanced Zoning

The format of the Version Specific payload for Protocol Version = 01h is depicted in table 53.

Table 53 – 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').

6.1.16.3 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 54.

Table 54 – Merge Request Accept Payload

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

6.1.17 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 request 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

Format: FT_1

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 55.

Table 55 – 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 56.

Table 56 – Acquire Change Authorization Accept Payload

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

NOTE 15 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.

6.1.18 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

Format: FT_1

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 57.

Table 57 – 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 58.

Table 58 – Release Change Authorization Accept Payload

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

6.1.19 Stage Fabric Configuration Update (SFC) Request

6.1.19.1 SFC overview

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

Format: FT_1

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 59.

Table 59 – SFC Request Payload

Item	Size (Bytes)
25h	1
Operation Request (see table 60)	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 60 depicts the currently defined Operation Request values.

.

Table 60 – Operation Request Value

Value (hex)	Description
00-02	Reserved
03	Activate Zone Set
04	Deactivate Zone Set
05-07	Reserved for FC-SP Use
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
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 61.

Table 61 – Stage Fabric Configuration Update Accept Payload

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

6.1.19.2 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 62 depicts the payload structure for them.

Table 62 – 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 6.1.16.2 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

6.1.19.3 SFC in Enhanced Zoning

6.1.19.3.1 Overview

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.

6.1.19.3.2 Operation Request 'Activate Zone Set Enhanced'

6.1.19.3.2.1 Overview

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 63 depicts the payload format.

Table 63 – 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

6.1.19.3.2.2 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. If set to zero, then the Zone Set Database is not included in the payload.

6.1.19.3.2.3 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.

6.1.19.3.3 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 64 depicts the payload format.

Table 64 – Payload for Operation Request 'Deactivate Zone Set Enhanced'

Item	Size (Bytes)
Reserved	2

6.1.19.3.4 Operation Request ‘Distribute Zone Set Database’

6.1.19.3.4.1 Overview

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 65 defines its payload.

Table 65 – Payload for Operation Request ‘Distribute Zone Set Database’

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

6.1.19.3.4.2 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.

6.1.19.3.4.3 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.

6.1.19.3.5 Operation Request ‘Activate Zone Set by Name’

6.1.19.3.5.1 Overview

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 66 depicts the payload format.

Table 66 – Payload for Operation Request ‘Activate Zone Set by Name’

Item	Size (Bytes)
Reserved	2
Zone Set Name	a

6.1.19.3.5.2 Zone Set Name

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

6.1.19.3.6 Operation Request 'Set Zoning Policies'

6.1.19.3.6.1 Overview

Operation Request 'Set Zoning Policies' is used in Enhanced Zoning to establish the Fabric Zoning Policies. Table 67 depicts the payload format.

Table 67 – Payload for Operation Request 'Set Zoning Policies'

Item	Size (Bytes)
Reserved	2
Enhanced Zoning Flags	4

6.1.19.3.6.2 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.

6.1.20 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

Format: FT_1

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 68.

Table 68 – 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 69.

Table 69 – Update Fabric Configuration Accept Payload

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

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

Format: FT–1

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 70.

Table 70 – ESC Request Payload

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

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 unique identifier of the requesting Switch. This field shall contain either an identifier for the Switch manufacturer, or an OEM identifier.

NOTE 16 Unique Vendor ID Strings are maintained by ANSI T10. A list of current Vendor ID Strings and instructions for requesting a new Vendor ID String may be found on the T10 web site at 'www.t10.org'.

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 71.

Table 71 – 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 unique vendor identifier associated with the Protocol ID field. It is the intention that this field contain the Vendor ID String of the original Switch manufacturer that designed the protocol being described.

NOTE 17 Unique Vendor ID Strings are maintained by ANSI T10. A list of current Vendor ID Strings and instructions for requesting a new Vendor ID String may be found on the T10 web site at 'www.t10.org'.

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 spe-

cific 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 72.

NOTE 18 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 72 – Protocol ID Values

Value	Use
0000h	Reserved
0001h	FSPF-Backbone Protocol
0002h	FSPF Protocol
0003h - 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 73.

Table 73 – ESC Accept Payload

Item	Size Bytes
Command Code = 02h	1
Reserved	3
Vendor ID String	8
Accepted Protocol Descriptor	12

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

NOTE 19 Unique Vendor ID Strings are maintained by ANSI T10. A list of current Vendor ID Strings and instructions for requesting a new Vendor ID String may be found on the T10 web site at ‘www.t10.org’.

Accepted Protocol Descriptor: This field shall contain the 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 71.

6.1.22 Exchange Switch Support (ESS)

6.1.22.1 ESS overview

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

Format: FT_1

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.

6.1.22.2 ESS Request Payload

The format of the ESS request payload is shown in table 74.

Table 74 – 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.

6.1.22.3 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-4.

6.1.22.4 Capability Object

The capability object is used to convey levels of support for FC-SW-3 and FC-GS-4 functionality. The format of the Capability Object is shown in table 75.

Table 75 – 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-4.

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.

6.1.22.5 Service Specific Capability Formats

6.1.22.5.1 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 76 defines the bit definition for identifying specific support for the Name server subtype of the Directory server.

Table 76 – 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 + FC-4 Descriptor 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 + FC-4 Descriptor name server objects.

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.

6.1.22.5.2 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 77 defines the bit definition for identifying specific support for the Fabric Controller.

Table 77 – 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.

6.1.22.5.3 ESS Fabric Configuration Server Capability Object

The WKA Type shall be set to FAh and the WKA Subtype shall be set to 01h.

Table 78 depicts the bit definitions for identifying specific support for the Fabric Configuration server subtype of the Management server.

Table 78 – 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. 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. 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. 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. When this bit is zero, the Switch does not support commands that are members of the Enhanced Configuration Service class.

Bits 4-31 reserved.

6.1.22.5.4 ESS Enhanced Zone Server Capability Object

The WKA Type shall be set to FAh and the WKA Subtype shall be set to 03h.

Table 79 depicts the bit definitions for identifying specific support for the Zone server subtype of the Management server.

Table 79 – 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 merge-able 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 permits 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.

Bits 8-31 reserved.

6.1.22.5.5 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 80 – Vendor Specific Capability Object

Item	Size Bytes
Well-Known Address Type	1
Well-Known Address Subtype	1
Reserved	1
Length (n)	1
T10 Vendor Identifier	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 double-words of vendor specific information contained in the capability object. The T10 Vendor Identifier field is included in the doubleword count.

T10 Vendor Identifier: The T10 Vendor Identifier field shall contain the vendor's eight byte T10 administered vendor identifier.

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.

6.1.22.6 ESS Accept Payload

The format of the accept payload is shown in table 81.

Table 81 – 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

Format: FT_1

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 82.

Table 82 – MRRA Request Payload

Item	Size (Bytes)
32000000h	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 83.

Table 83 – Vendor Specific Field

Item	Size (Bytes)
Vendor Identifier	8
Vendor Specific Information	8

Vendor Identifier: This field contains the eight byte T10 administered vendor identifier.

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 84.

Table 84 – MRRA Response Payload

Item	Size (Bytes)
02000000h	4
Vendor Identifier	8
MRRA Response	4
Maximum Resources Available	4
Waiting Period	4

Vendor Identifier: This field contains the eight byte T10 administered vendor identifier. This allows the vendor specific fields in the Merge Response field to be used.

MRRA Response: This field shall specify the response to the MRRA request. The values are defined in table 85.

Table 85 – 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.

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 86.

Table 86 – 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.

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 11 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 and FC-AL-2. This state machine is also applicable to B_Port operation.

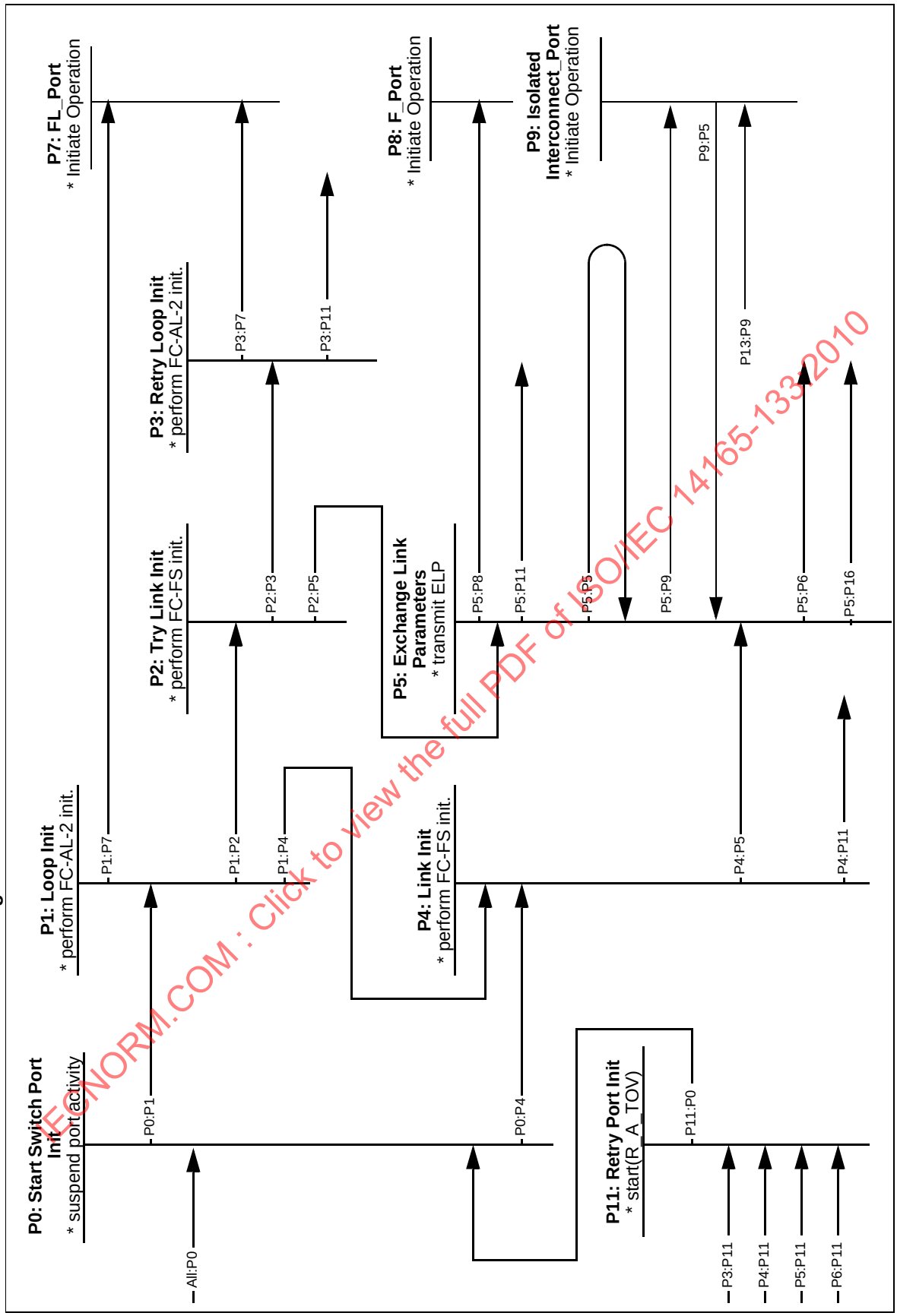


Figure 11 – Switch Port Mode Initialization State Machine

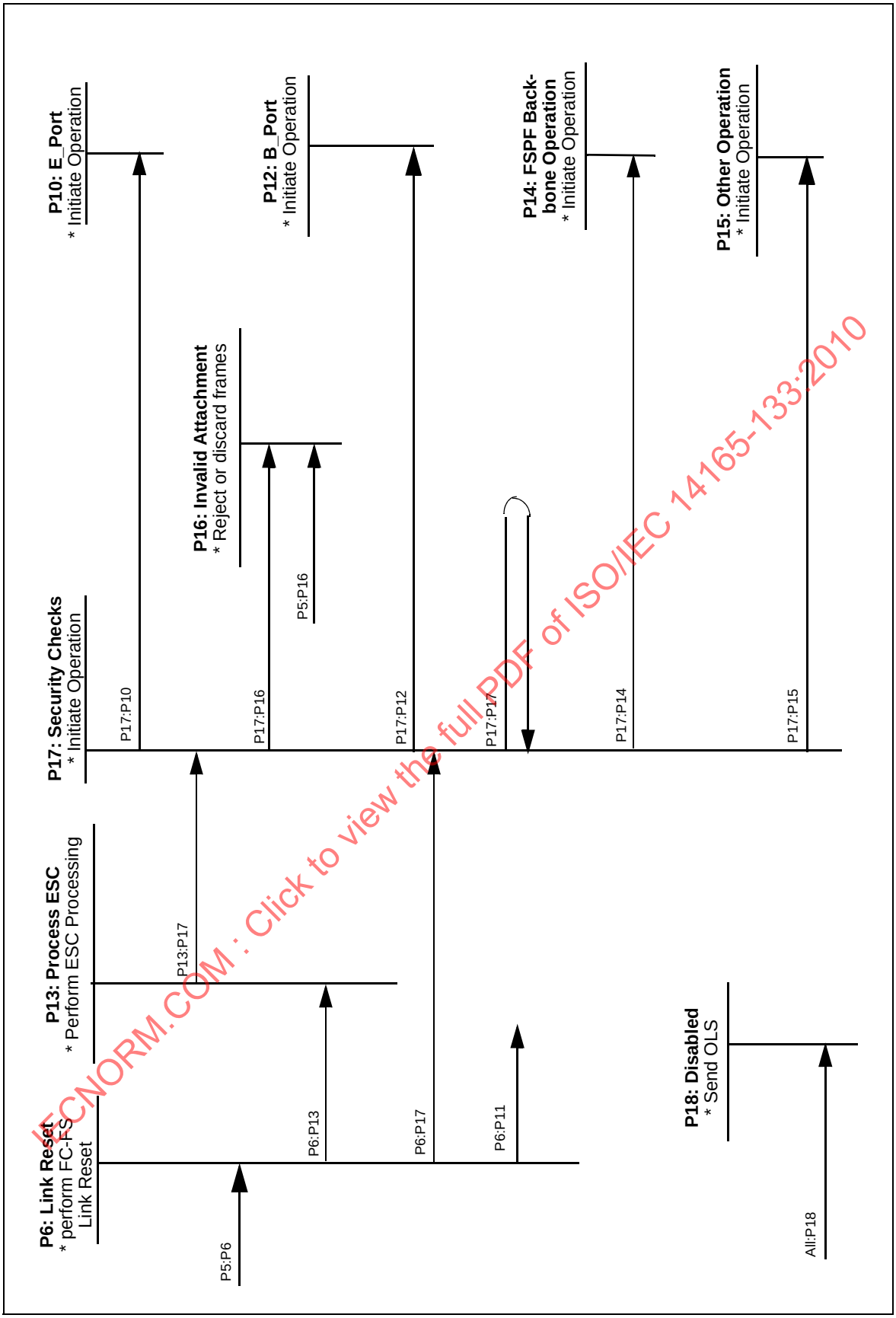


Figure 12 – Switch Port Mode Initialization State Machine - Continued

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; 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 20 LR is not considered an Initialization Event, but shall operate as specified in FC-FS.

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 a minimum of 2xAL_TIME (see annex A), and then complete Link Initialization as defined in FC-FS. 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.

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 87 below defines the responses and actions to an ELP request for the originating Interconnect_Port**.**

Table 87 – 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)
7. ELP (rcvd Switch_Name < own Switch_Name)	Both Interconnect_Ports sent ELP at the same time	Send SW_RJT[b], Transition (P5:P5) (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.

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 87 – Responses to ELP Request for Originating Interconnect_Port (part 2 of 2)

Response to ELP	Indication	Originating Interconnect_Port Action
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 an N_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.

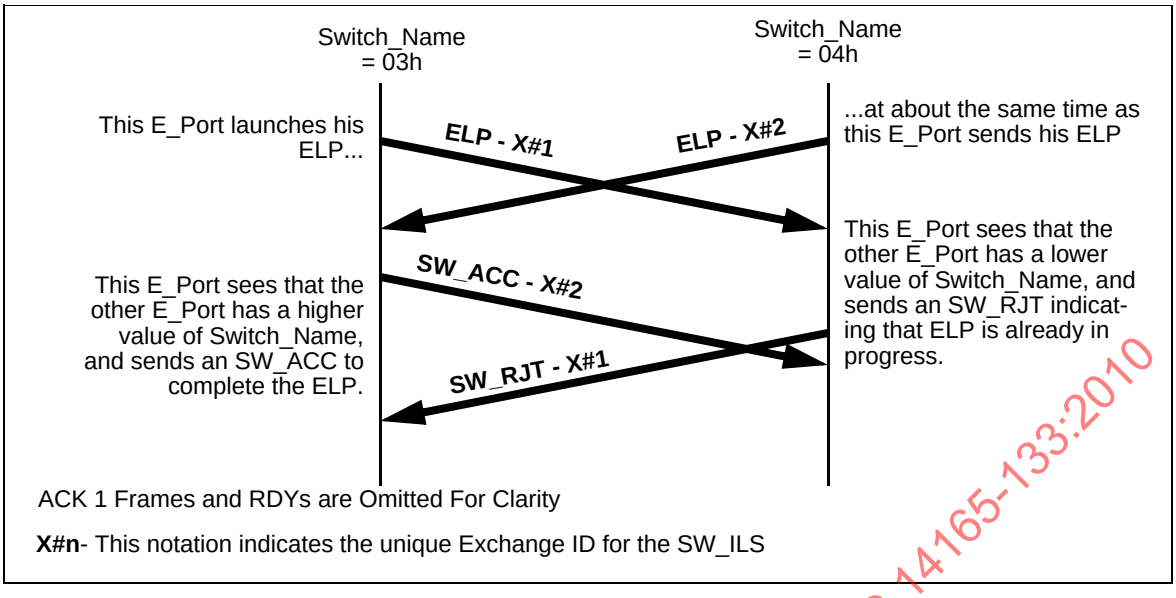
[d] An SW_ACC is sent for the other ELP Exchange in progress, as described in Response #6, if the received ELP parameters are acceptable. If the received ELP parameters are not acceptable, then an SW_RJT is sent. See figure 13.

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 – Example of Simultaneous ELP Processing- Parameters Acceptable to Both Switches



Transition P5:P5. This transition occurs if the responding Interconnect_Port does not agree that the parameters are acceptable, it shall return an SW_RJT reply Sequence indicating the reason for the disagreement, and wait for the originating Interconnect_Port to initiate another ELP request Sequence. This transition may also occur 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 7.5).

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 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.

NOTE 21 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 for establishing a known credit state.

Transition P6:P11. This transition occurs if the Link Reset fails.

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.

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 N_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 Switch Port has completed the exchange of Link Parameters with another Interconnect_Port. Because the Link Parameters exchanged were not acceptable, then the Interconnect_Port shall become Isolated and not continue with Fabric Configuration, as described in 7.5. 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 7.5).

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 P14: FSPF-Backbone Operation. The port operates in FSPF-Backbone mode.

State P15: Other Operation. The Port operates in a mode other than FSPF or FSPF-Backbone.

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). 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 a 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 process. The order and protocol of the security checks is defined in Fibre Channel Security Protocols (FC-SP).

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:P12. This transition occurs if all required security checks are successful and the port is to operate as a B_Port.

Transition P17:P14. This transition occurs if all required security checks are successful and FSPF-Backbone is the agreed upon path selection mechanism per the prior ESC exchange.

Transition P17:P15. This transition occurs if a routing protocol other than FSPF or FSPF-Backbone is agreed to in the prior ESC exchange.

Transition P17:P16 This transition occurs when a required security check 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.

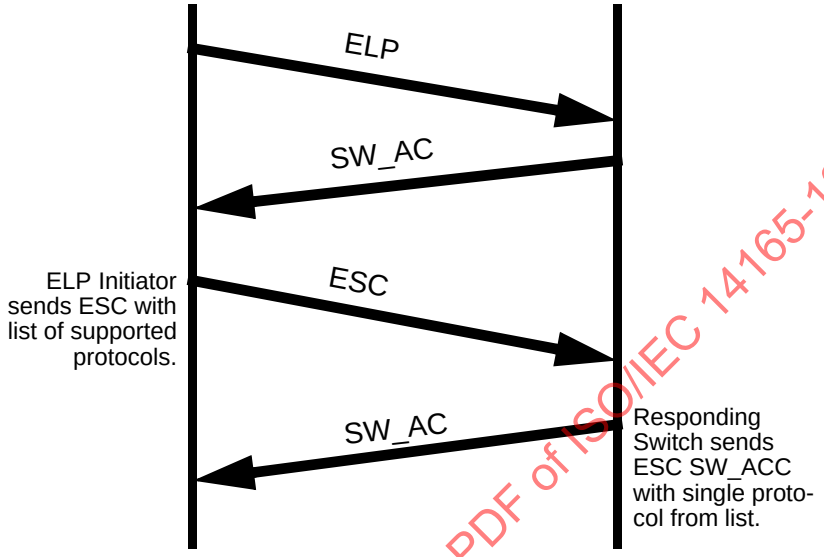
Transition All:P18. The transition to this state occurs when the Switch determines that a model dependent threshold has been exceeded.

When an Inter-Switch Link is established the Switch shall request EFP and enter state F2.

7.2.2 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.

7.3 Principal Switch Selection

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 designated 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 88.

Table 88 – 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 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**

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.

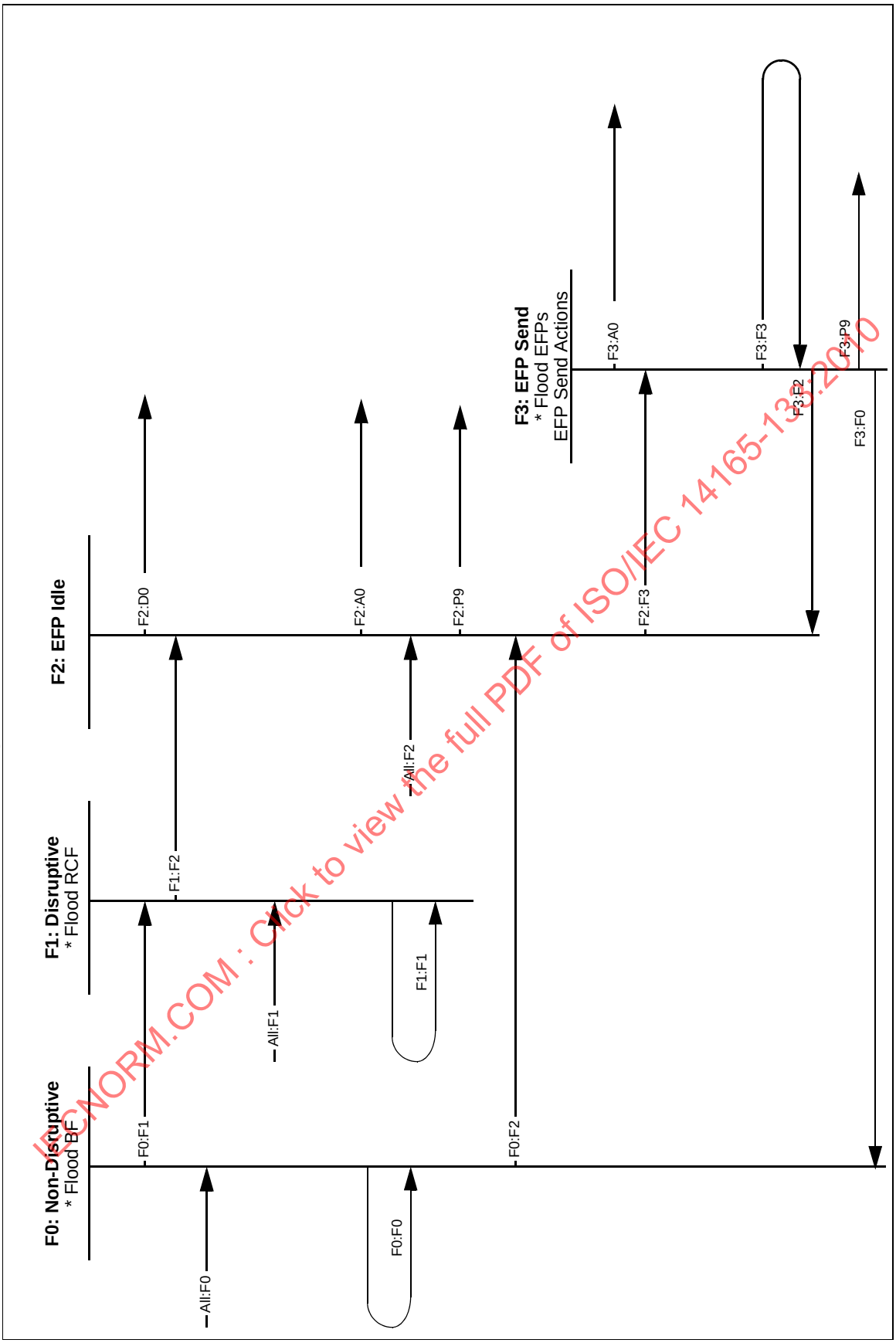


Figure 15 – Principal Switch Selection State Machine

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, and DSCN 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.

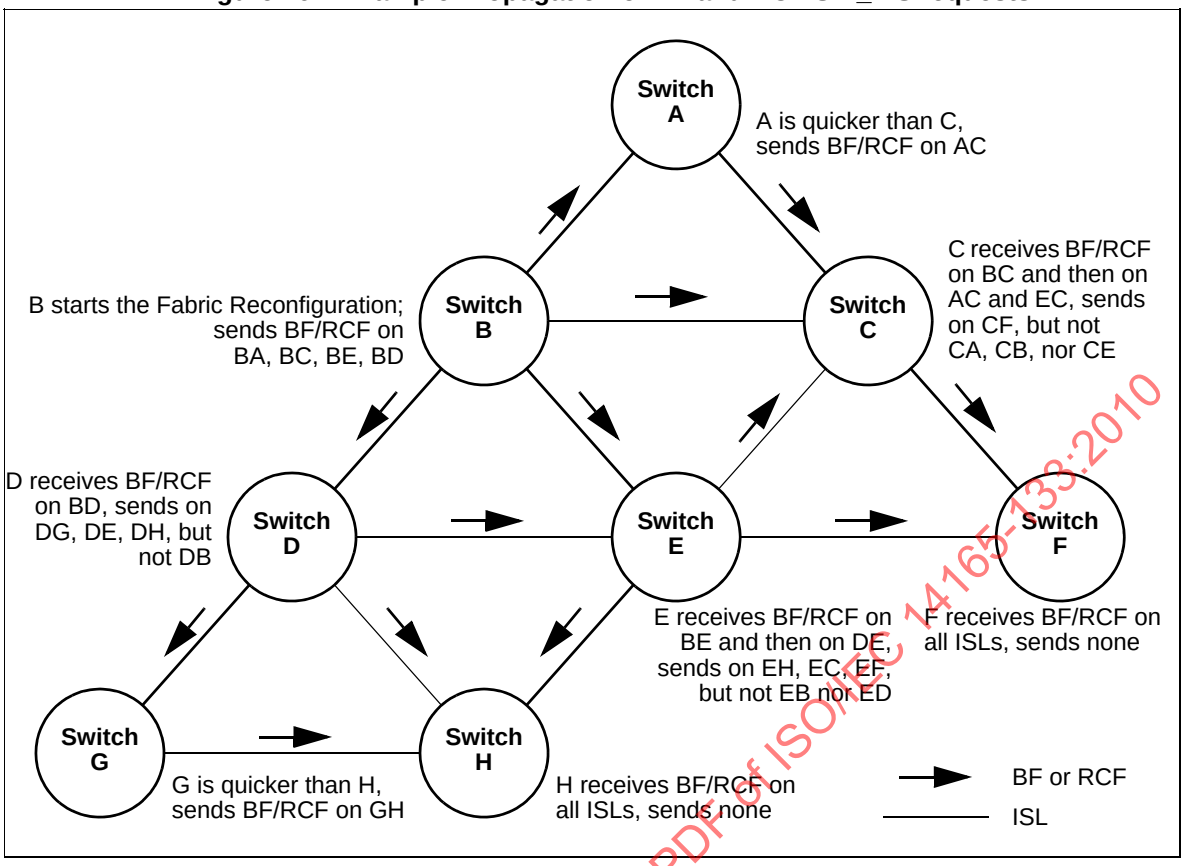
Transition All:F0. This transition enters the state machine performing a non-disruptive Fabric Reconfiguration. This transition occurs when the Switch originates a BF, or when it receives a BF, or when the Switch receives a EFP where the received Domain_ID_List is non-zero, the retained Domain_ID_list 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, RDI, or DSCN 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 Topology database and the associated initial message number counter shall be cleared.

NOTE 22 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 7.5.

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 7.5.

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

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.

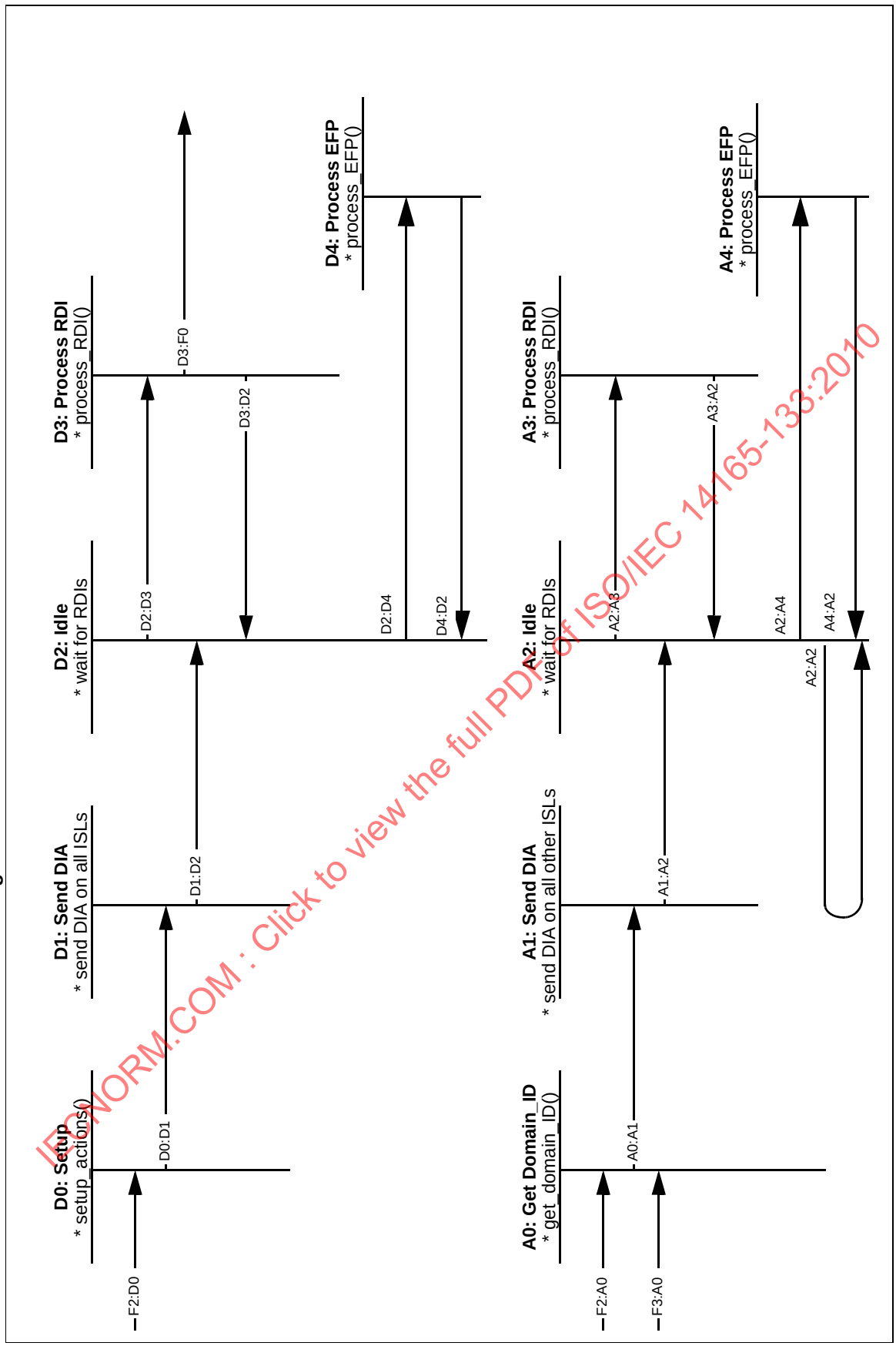


Figure 17 – Address Distribution State Machines

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.

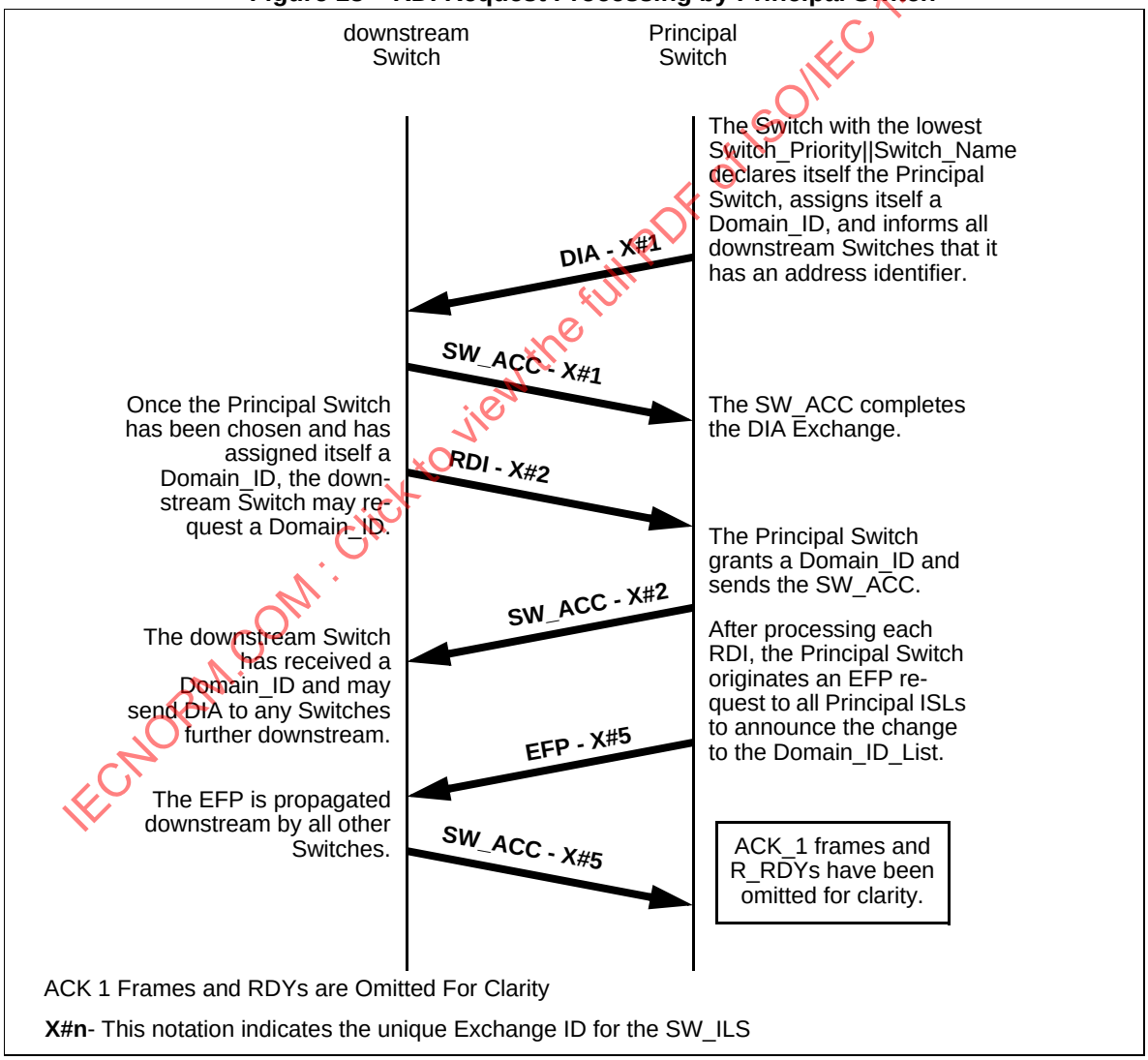
d) If a Switch that has already been granted a Domain_ID transmits a request to the Principal Switch for a different Domain_ID, the Principal Switch shall perform a Fabric Reconfiguration (see 7.3).

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 23 The change of Domain_ID may make out-of-date the local copies of some databases related to Distributed Services, or the FSPF Topology 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.

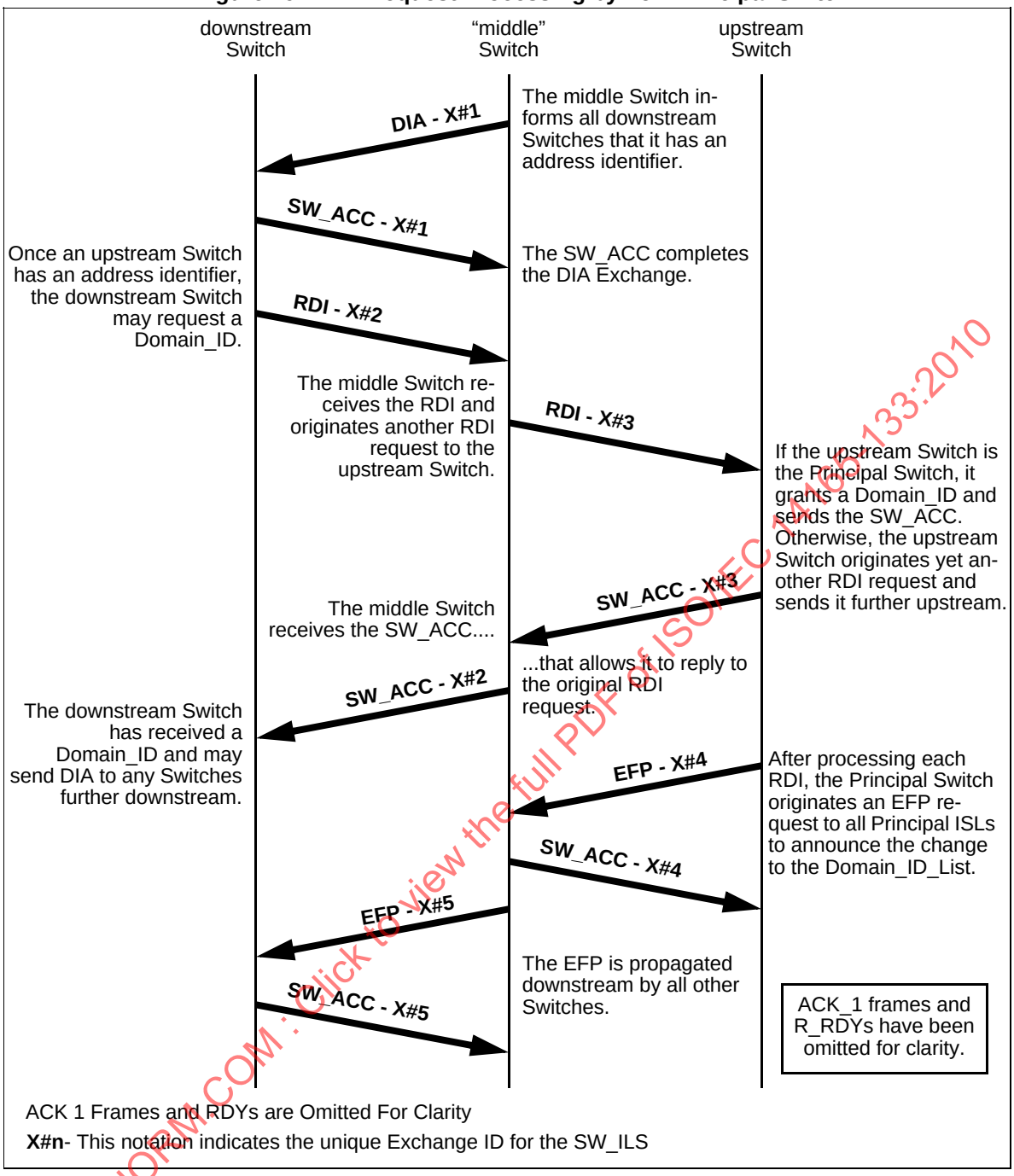
Transition A2:A2. This transition occurs when the Switch receives an EFP request from an upstream ISL and an EFP response is sent, and the EFP request is forwarded on all other ISLs.

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 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) The two E_Ports are a new Link between two existing Fabrics, and the Domain_ID allocations in each Fabric overlap. For example, if each existing Fabric had allocated Domain_ID 44h to a Switch, one Switch would have to give up its Preferred Domain_ID to reconfigure; this may cause a major disruption to current traffic;

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; or

h) An SW_RJT with reason code explanation of “E_Port is Isolated” is received.

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; or

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.

7.6 B_Port Operation

7.6.1 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 only ELP and are transparent to any other messages (see 5.6).

7.6.2 B_Port Internal Link Services

The B_Port shall generate a subset of the Internal Link Services defined in this standard. Table 89 details the ILS support as either being propagated or generated by the B_Port.

Table 89 – B_Port - ILS Support

FC-SW-3 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
Disconnect Class 1 Connection (DSCN)	Prohibited	SW_RJT	Prohibited
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
Exchange Switch Support (ESS)	Prohibited	Propagate	Allowed
Merge Request Resource Allocation (MRRA)	Prohibited	Propagate	Allowed

7.6.3 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 90

Table 90 – Bridge Port Initialization Summary

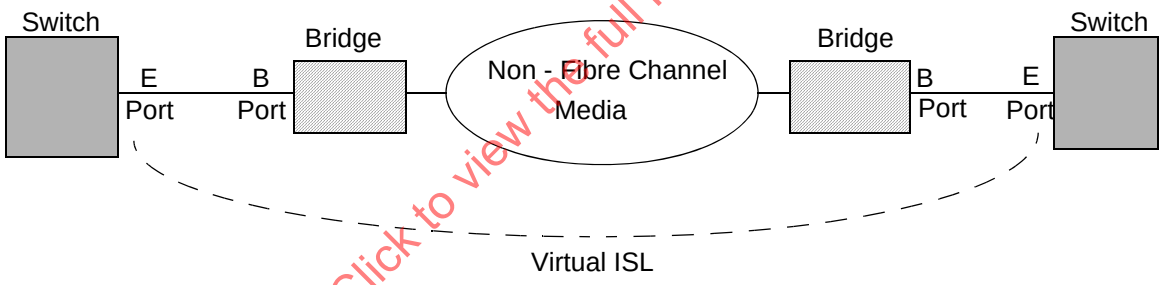
Operation	Starting Condition	Process	Ending Condition
Establish Link Parameters and Switch Port operating mode	Bridge Port has achieved word synchronization.	ELPs are exchanged and the B_Port determines that it is attached to an E_Port.	Link Parameters have been exchanged and Credit has been initialized, and it is known if the attached port is an E_Port.

7.6.4 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 24 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 topology database (or 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 topology 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 topology 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 25 One possible candidate is Dijkstra's algorithm, but this is not a requirement for interoperability.

The routing table update is not covered in this standard, 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 topology 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 Hierarchical Path Selection

FSPF allows the subdivision of the Switches in the Fabric into different Autonomous Regions. The interconnection of these Autonomous Regions may then be used to form a higher level path selection structure. In this case the Fabric path selection process includes both an intra-AR component and an inter-AR component and the overall routing in the Fabric constitutes a 2-level hierarchy (see 8.8).

8.1.6 FSPF Path Selection Summary

Table 91 summarizes path selection via FSPF.

Table 91 – 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) Autonomous Region: The Autonomous Region to which the message is transmitted. For a Fabric not implementing an FSPF-Backbone this value may be set to any value, including zero. For a Fabric implementing an FSPF-Backbone, the value zero is reserved for the FSPF-Backbone itself, and non-zero values are used for all other ARs;

d) Authentication Type: No authentication is specified at this time. This field shall be set to 00h;

e) 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; and

f) 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 Autonomous Region to which the message is transmitted shall be equal to the Autonomous Region number of the receiving Switch;

c) The Authentication Type shall be 00h;

d) The Originating Domain_ID shall be checked for a valid value as specified in 6.1.8.2; and

e) 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.

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 164.

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 Topology Database

The topology 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 two types of LSRs. They are:

a) Switch Link Record;

b) AR Summary 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 topology 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 topology 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 topology 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 26 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 27 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 min.

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 topology 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 28 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 29 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 92:

Table 92 – 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 30 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 31 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 Topology Database Synchronization

8.6.1 Synchronization overview

The topology 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 topology databases have been synchronized. Once the topology 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 topology 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 topology 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 recent ver-

sion 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 topology 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 topology 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 LSUs. 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 Topology 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; or

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 topology 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 than 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 topology database is empty because it has not recognized any neighbors yet. Topology 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 topology 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 32 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 topology database.

NOTE 33 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 data-

base. 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 topology 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; and

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 topology 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 topology 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 topology 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 93 – 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.
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 topology 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.

Table 93 – Neighbor Finite State Machine (part 2 of 3)

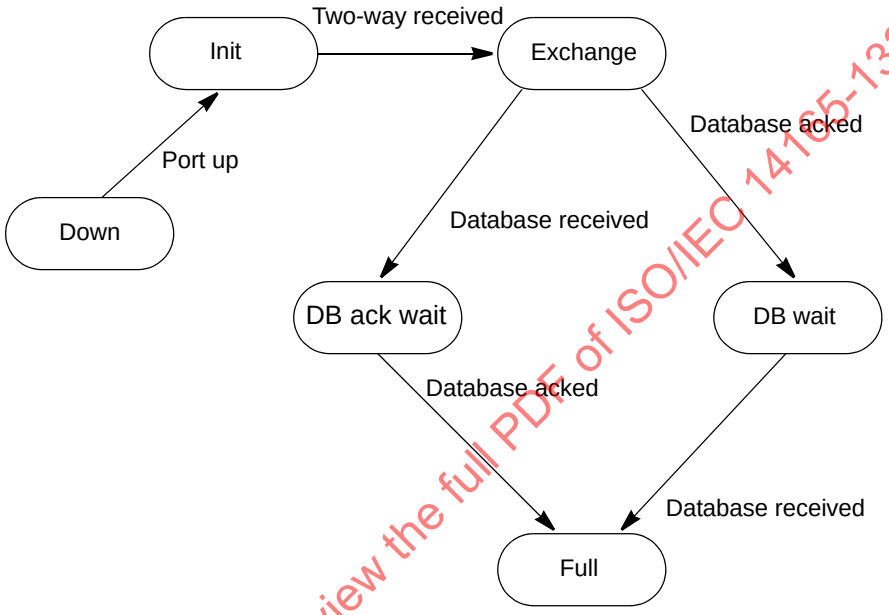
State	Input	Next State	Actions
Database Exchange	Database Sent	Database Exchange	This input indicates that all the Link State Records that describe the topology database have been sent to the neighbor. Send an LSU to the neighbor with no LSRs and the Database Complete flag set.
Database Exchange	Database 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 topology 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.
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. The port is still in E_Port state, but the local Switch has not received Hello messages from the neighbor. 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.

Table 93 – Neighbor Finite State Machine (part 3 of 3)

State	Input	Next State	Actions
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.

Figure 21 is a pictorial representation of the FSM where only the major state transitions are represented.

Figure 21 – Neighbor Finite State Machine



8.8 FSPF-Backbone

8.8.1 FSPF-Backbone overview

The FSPF-Backbone runs the FSPF-Backbone routing protocol. The FSPF-Backbone routing protocol is the same as the FSPF protocol, as defined in 8.1 through 8.7, but exchanges Summary Descriptor LSRs. Subclause 6.1 describes the SW_ILS used with the FSPF protocol with additional support for the FSPF-Backbone routing. A Border Switch (BSW) provides the connectivity between an AR and the FSPF-Backbone network. The BSW originates a Summary Descriptor LSR for its associated non-zero AR and floods this summary on the FSPF-Backbone. A BSW receiving this Summary Descriptor LSR originates Link Descriptor LSRs extracted from the Summary LSR to its associated AR. BSWs are the only Switches that originate or receive Summary Descriptor LSRs.

Figure 22 illustrates the architecture of the FSPF-Backbone Fabric and the corresponding backbone for a small number of ARs. The FSPF-Backbone belongs to a special Autonomous Region called AR0, that utilizes the FSPF-Backbone routing protocol. The FSPF-Backbone consists of a physically contiguous arbitrary network topology of one or more Border Switches (BSW) and possibly Intra-AR0 (ISW-0)

Switches inter-connected by E_Ports. A Switch that may support at least one E_Port with FSPF-Backbone routing capability is capable of attaching to the FSPF-Backbone network. The FSPF-Backbone architecture defines three types of Switches:

a) Border Switches (BSW) - those Switches that connect an AR to the FSPF-Backbone;

b) Internal AR Switches (ISW) - those Switches that are internal to an AR;

c) ISW-0 Switch is an AR0 Internal Switch that connects to BSWs, or to other ISW-0s. This Switch shall only communicate path information using the Summary Record format of the Link-State Record.

Figure 22 – FSPF-Backbone Architecture

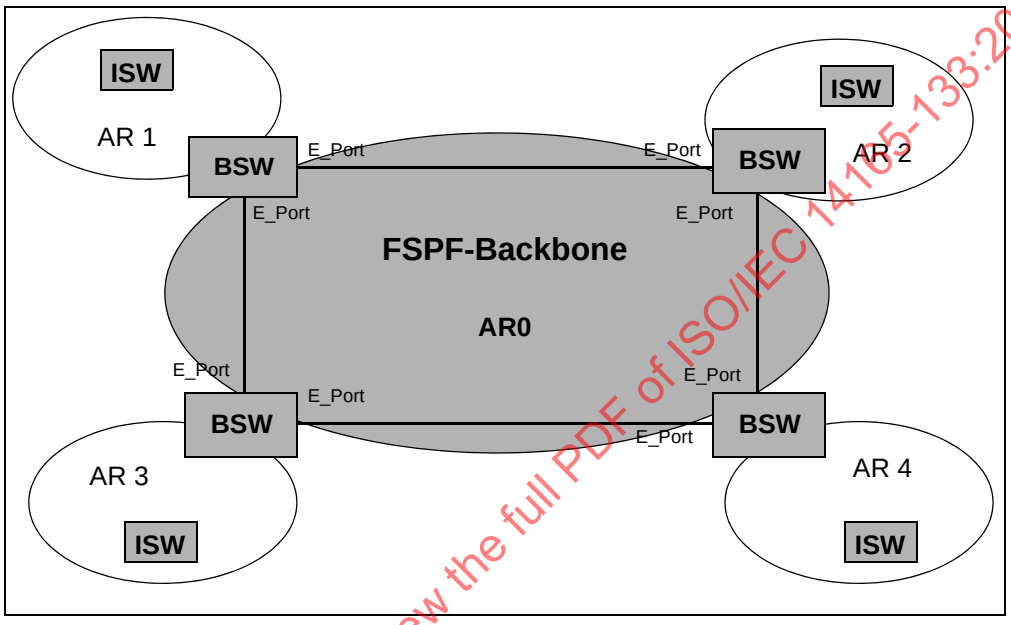


Figure 23 shows the FSPF-Backbone consisting of point-to-point links and FC-BB-2 WAN interfaces that serve to extend the point-to-point connectivity between two BSWs across a non-Fibre Channel network.

Figure 23 – Point-to-point FSPF-Backbone Links

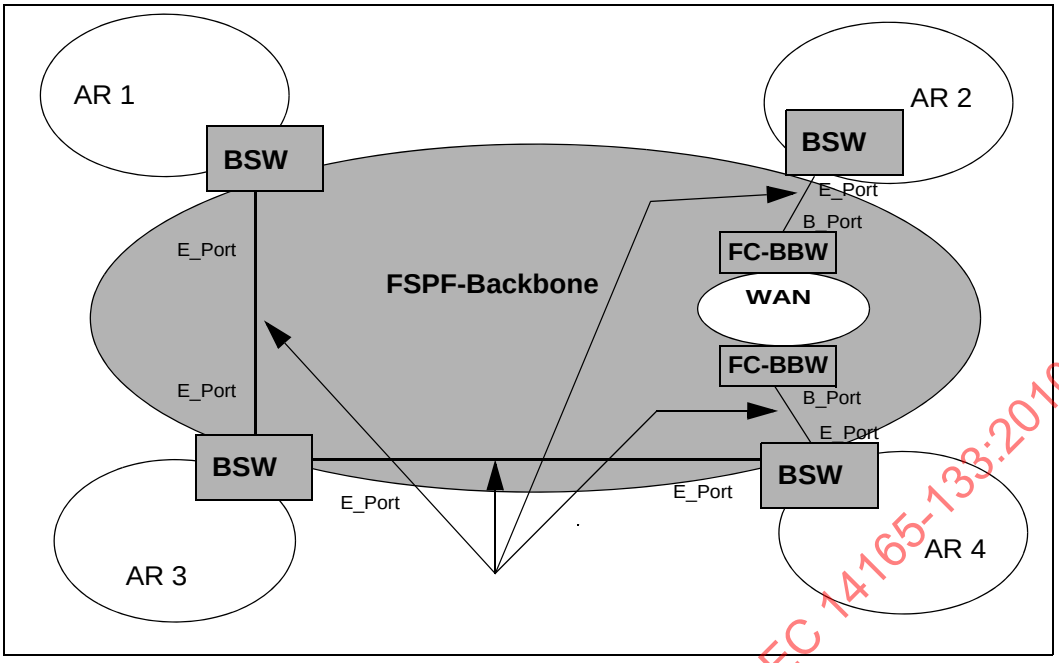
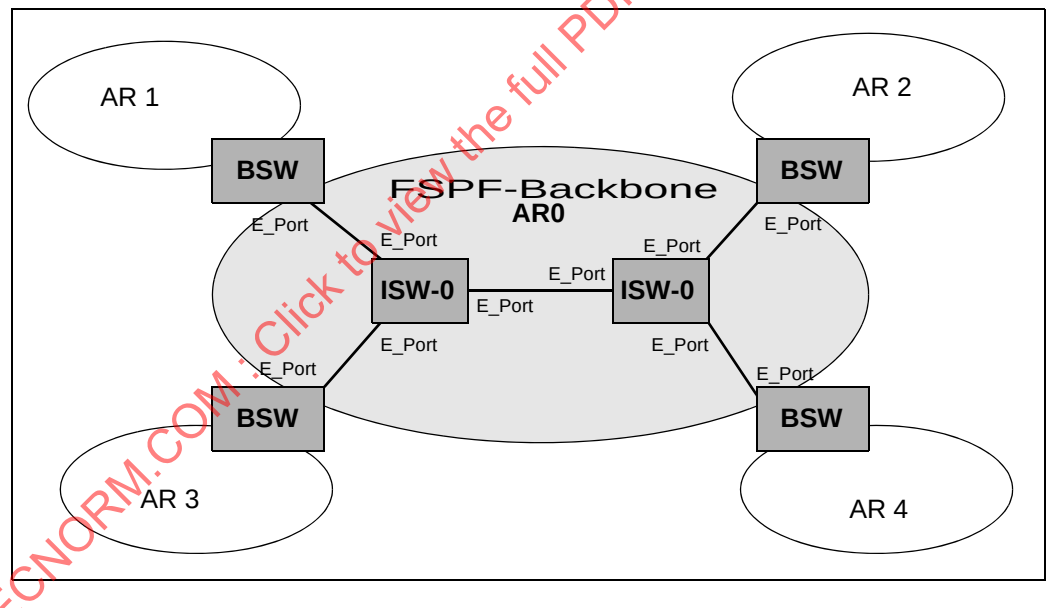


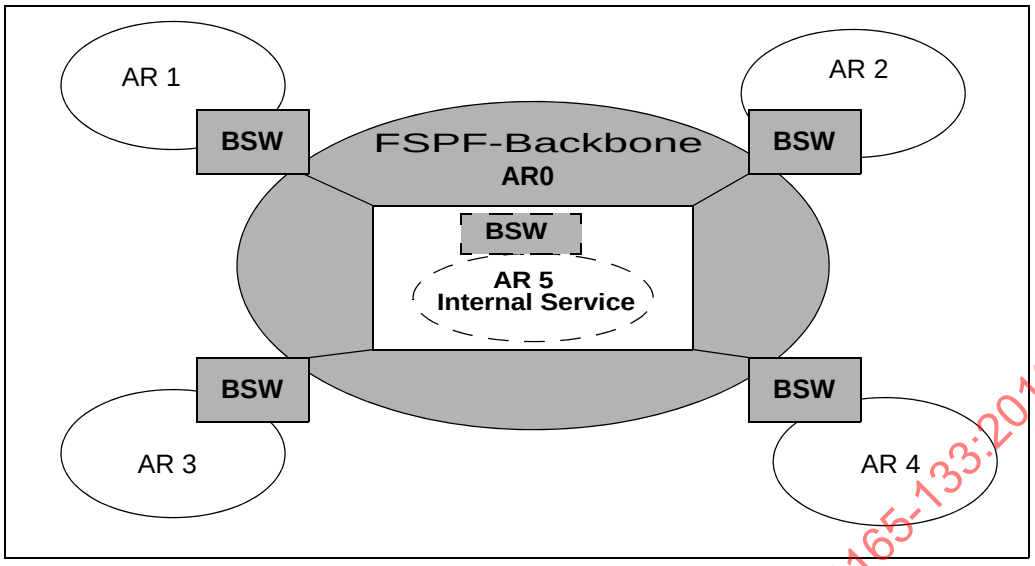
Figure 24 shows two ISW-0 Switches connecting BSW Switches.

Figure 24 – AR0 Consisting of Two ISW-0 Switching Devices



A Switch that connects BSWs and supports any internal service (e.g., Extended Copy) that requires a Domain_ID is a BSW Switch. Figure 25 shows an example of such a configuration. Note that a well-known service such as a Name Server, does not constitute an internal service and such services may be supported in the ISW-0. It is expected that the total number of Switches (BSWs and ISW-0s) should be small in any real implementation. Therefore assignment of Domain_IDs to the AR0 Switches may be manually configured.

Figure 25 – Internal Service Supported By a BSW



The Summary LSR provides information on the Fibre Channel Address Domains supported by an AR. A Domain_ID conflict occurs whenever the same Domain Address is reported in two (or more) LSAs originating from different ARs. All Border Switches (BSW) and ISW-0 Switches shall be required to run a check against such a conflict every time they receive an LSA. All BSWs shall refrain from routing data frames for the conflicting Domain_IDs, until such time that a new LSA is issued without a conflict.

8.8.2 Multiple Switch Connections

Multiple BSWs from the same AR may connect to the FSPF-Backbone, with the requirement that physical contiguity be always maintained with the FSPF-Backbone. Figure 26 shows a dual BSW connection to the FSPF-Backbone. Illustrative examples of allowed and disallowed dual BSW connectivity are shown in figure 27 and figure 28. The example in figure 28 is disallowed because the FSPF-Backbone is physically discontiguous resulting in two disjoint FSPF-Backbones. In figure 27 and figure 28 note that an internal connection between two BSWs is allowed but not necessarily part of the FSPF-Backbone. Two E_Ports supporting FSPF-Backbone on two separate BSWs belonging to the same AR, may be connected by one or more links.

Figure 26 – Dual BSW connectivity

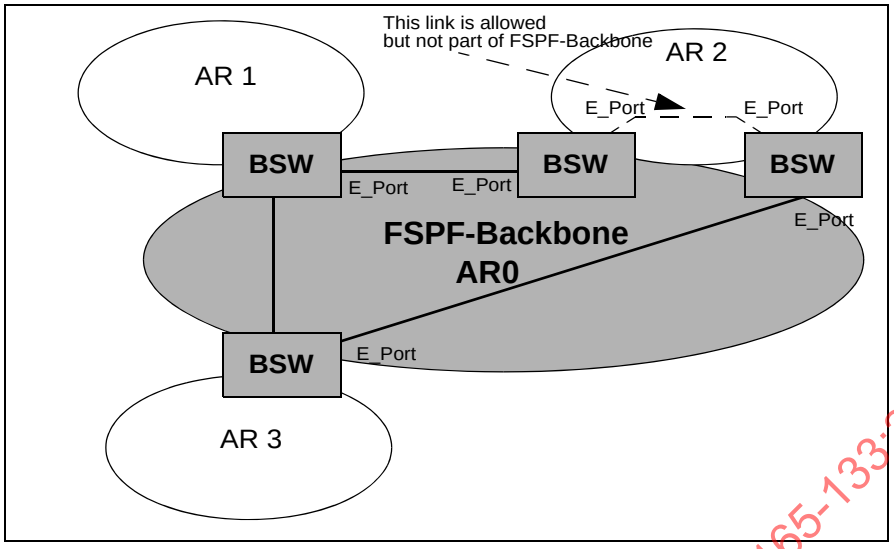


Figure 27 – Physically Contiguous FSPF-Backbone with dual BSWs (Allowed)

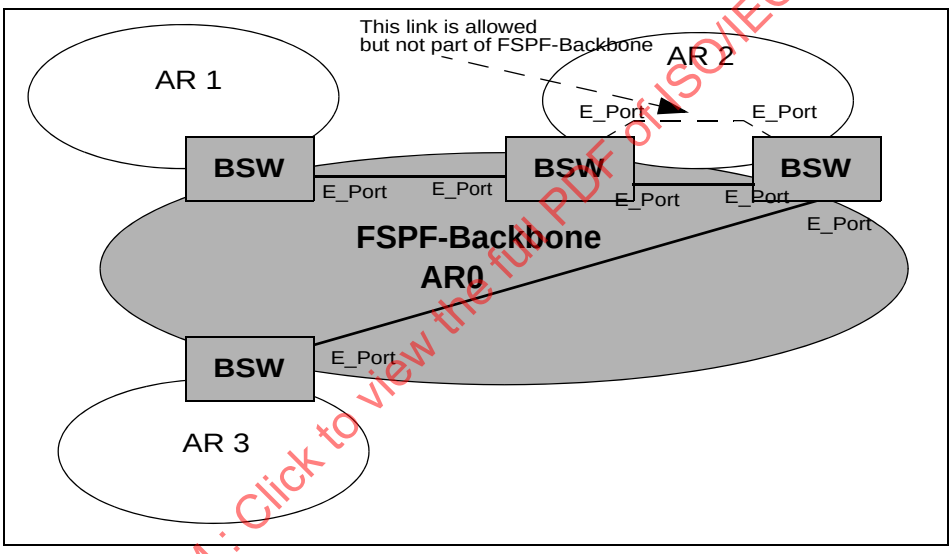
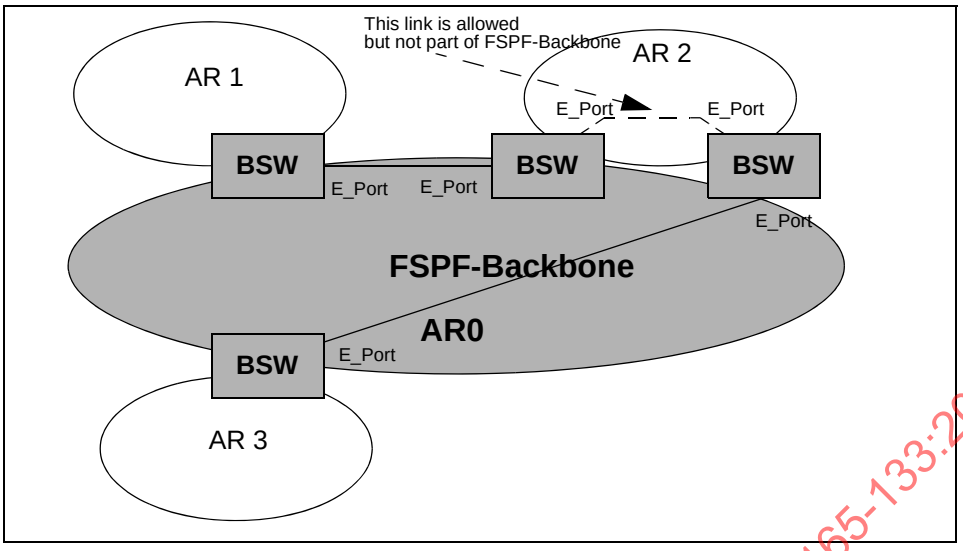


Figure 28 – Physically Non-Contiguous FSPF-Backbone with dual BSWs (Disallowed)



8.8.3 FSPF-Backbone Point-to-point Links

The FSPF-Backbone is a physically contiguous arbitrary network topology of Switches that inter-connect different ARs by one or more of the following methods:

a) a point-to-point link that directly connects two BSW Switches via E_Ports; or

b) a point-to-point link that connects exactly two remote BSW Switches over a WAN link using the FC-BB-2 defined specifications (e.g., FC-BB-2_ATM, FC-BB-2_SONET).

The E_Port is the point at which frames between two BSW Switches pass directly on a point-to-point FSPF-Backbone link. Frames that enter a Switch via an E_Port connected to the FSPF-Backbone are forwarded to the local AR, or forwarded towards their ultimate destination via another E_Port connected to the FSPF-Backbone.

Inter-AR frames arriving on a BSW Switch are forwarded to a FC-BBW (see FC-BB-2) on a B_Port and eventually on a WAN Interface towards their remote destination. Frames that are received from the WAN enter the FC-BBW and are forwarded to the BSW Switch.

8.8.4 FSPF-Backbone Routing Protocol

Each AR supports one or more Fibre Channel Address Domains. The FSPF routing protocol exchanges Link Descriptor LSRs in each AR and Summary Descriptor LSRs in the FSPF-Backbone. If there is a single AR, the FSPF routing protocol architecture does not have to provide support for Inter-AR routing across the FSPF-Backbone. The FSPF-Backbone routing protocol requires each BSW to advertise summarized Fibre Channel Address Domain reachability information, but not the actual routes within its attached AR. This summary is sent across the backbone to all BSWs. Each BSW also advertises all reachable Domains and ARs through the BSW to all its attached AR ISWs using Link Descriptor LSRs. Summarized advertisements help the architecture to scale as a function of the number of ARs. Additionally, each BSW participates in advertising its neighbor connections within its attached ARs, including AR0.

Figure 29 shows a FSPF-backbone connection with internal Switch connections. In this diagram, the following LSRs are distributed:

a) BSW 1 sends a Summary LSR to BSW 2 that contains the path costs to reach Domain 1, Domain 2, and Domain 3. This path cost comes from BSW 1 running its path selection algorithm on the LSRs that it received from its neighbor Switches.

b) BSW 2 sends a Summary LSR to BSW 1 that contains the path costs to reach Domain 4, Domain 5, Domain 6, and the Summary LSR from BSW 3. The path cost for Domain 4, Domain 5, and Domain 6 comes from BSW 2 running its path selection algorithm on the LSRs that it received from its neighbor Switch. The Summary LSR forwarded from BSW 3 shall incorporate the path cost between BSW 3 and BSW 2.

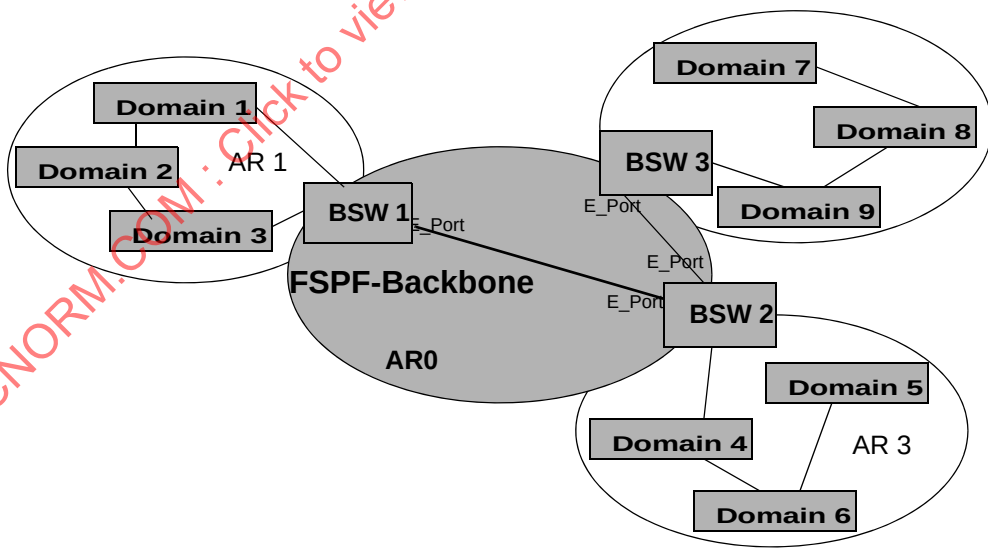
c) BSW 3 sends a Summary LSR to BSW 2 that contains the path costs to reach Domain 7, Domain 8, and Domain 9. This path cost comes from BSW 3 running its path selection algorithm on the LSRs that it received from its neighbor Switch.

d) BSW 1 sends Link Descriptor LSRs to neighbor Switches Domain 1 and Domain 3, that describe virtual connections to Domain 4, Domain 5, Domain 6, Domain 7, Domain 8, and Domain 9. These LSRs contain the path cost that came in the Summary LSR from BSW 2 plus the cost of the link from BSW 1 to BSW 2.

e) BSW 2 sends Link Descriptor LSRs to neighbor Switch Domain 4, that describe virtual connections to Domain 1, Domain 2, Domain 3, Domain 7, Domain 8, and Domain 9. These LSRs contain the path cost that came in the Summary LSRs from BSW 1 and BSW 3 plus the cost of the link from BSW 2 to BSW 1.

If an Autonomous Region has 2 BSWs, they shall not forward Summary LSRs that came from the local Autonomous region, and frames destined to the local Autonomous Region shall not traverse the FSPF-Backbone.

Figure 29 – FSPF-Backbone Routing Protocol overview.



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 three-fold:

a) define a consistent method for distributing services across Switches in a Fabric;

b) define a distribution method that is topology independent; and

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; and

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-4.

All CT frames shall be transmitted using the FT_1 frame format via 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.

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: 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 34 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 its well known address and Nx_Ports shall not send Distributed Services requests to Domain Controllers. In addition, Distributed Services requests and responses are transported only between Switches and not between a Switch and an Nx_Port.

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 does not have the requested data, then it shall respond with an LS_RJT (reason code 09h “Unable to perform command request”, reason code explanation 2Ah “Unable to supply requested data”). 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 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:

   A) one of the Switches a request is directed to is busy and unable to respond within the time-out period; or

   B) 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;

f) Data Merge: Describes how data from multiple responses is consolidated; and

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; and

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 as follows:

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; or

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-4.

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 the 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-4 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-4) 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 dNSs 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 dNSs to acquire the requested information is transparent to the original requesting client; and

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 94. Codes 0100h to 0300h shall be as defined in FC-GS-4. 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 94 – 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-4	
0101	Get Identifiers According to Scope	GID_A	1-to-1 or 1-to-Many	FC-GS-4	a
0112	Get Port_Name, based on Port Identifier	GPN_ID	1-to-1	FC-GS-4	
0113	Get Node_Name, based on Port Identifier	GNN_ID	1-to-1	FC-GS-4	
0114	Get Class of Service, based on Port Identifier	GCS_ID	1-to-1	FC-GS-4	
0117	Get FC-4 TYPEs, based on Port Identifier	GFT_ID	1-to-1	FC-GS-4	
0118	Get Symbolic Port_Name, based on Port Identifier	GSPN_ID	1-to-1	FC-GS-4	
011A	Get Port Type, based on Port Identifier	GPT_ID	1-to-1	FC-GS-4	
011B	Get Port IP Address, based on Port Identifier	GIPP_ID	1-to-1	FC-GS-4	
011C	Get Fabric Port_Name, based on Port Identifier	GFPN_ID	1-to-1	FC-GS-4	
011D	Get Hard Address, based on Port Identifier	GHA_ID	1-to-1	FC-GS-4	

[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 standard.

[c] De-registration requests may be distributed if performed by a third party. Otherwise, they are local and outside the scope of this standard.

[d] Work Categories for Name Server Entry Object requests are at the discretion of the originating Switch.

Table 94 – FC-CT Command Codes for dNS (part 2 of 4)

Encoded Value (hex)	Description	Mnemonic	Work Category	Payload Defined in	Notes
011E	Get FC-4 Descriptors - Port Identifier	GFD_ID	1-to-1	FC-GS-4	
011F	Get FC-4 Features - Port Identifier	GFF_ID	1-to-1	FC-GS-4	
0121	Get Port Identifier, based on Port_Name	GID_PN	1-to-All	FC-GS-4	
012B	Get Port IP Address, based on Port_Name	GIPP_PN	1-to-All	FC-GS-4	
0131	Get Port Identifier, based on Node_Name	GID_NN	1-to-All	FC-GS-4	
0132	Get Port_Names based on Node_Name	GPN_NN	1-to-All	FC-GS-4	
0135	Get IP address, based on Node_Name	GIP_NN	1-to-All	FC-GS-4	
0136	Get Initial Process Associator, based on Node_Name	GIPA_NN	1-to-All	FC-GS-4	
0139	Get Symbolic Node_Name, based on Node_Name	GSNN_NN	1-to-All	FC-GS-4	
0153	Get Node_Name, based on IP address	GNN_IP	1-to-All	FC-GS-4	
0156	Get Initial Process Associator, based on IP address	GIPA_IP	1-to-All	FC-GS-4	
0171	Get Port Identifiers, based on FC-4 TYPE	GID_FT	1-to-All	FC-GS-4	
0172	Get Port_Names, based on FC-4 TYPE	GPN_FT	1-to-All	FC-GS-4	
0173	Get Node_Names, based on FC-4 TYPE	GNN_FT	1-to-All	FC-GS-4	
01A1	Get Port Identifiers, based on Port Type	GID_PT	1-to-All	FC-GS-4	
01B1	Get Port Identifiers, based on Port IP Address	GID_IPP	1-to-All	FC-GS-4	
01B2	Get Port_Name, based on Port IP Address	GPN_IPP	1-to-All	FC-GS-4	
01C1	Get Port Identifiers, based on Fabric Port_Name	GID_FPN	1-to-All	FC-GS-4	
01D1	Get Permanent Port_Name	GPPN_ID	1-to-All	FC-GS-4	
01F1	Get Port Identfiers, based on FC-4 Features	GID_FF	1-to-All	FC-GS-4	

[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 standard.

[c] De-registration requests may be distributed if performed by a third party. Otherwise, they are local and outside the scope of this standard.

[d] Work Categories for Name Server Entry Object requests are at the discretion of the originating Switch.

Table 94 – FC-CT Command Codes for dNS (part 3 of 4)

Encoded Value (hex)	Description	Mnemonic	Work Category	Payload Defined in	Notes
0212	Register Port_Name	RPN_ID	1-to-1	FC-GS-4	b
0213	Register Node_Name	RNN_ID	1-to-1	FC-GS-4	b
0214	Register Class of Service	RCS_ID	1-to-1	FC-GS-4	b
0217	Register FC-4 TYPEs	RFT_ID	1-to-1	FC-GS-4	b
0218	Register Symbolic Port_Name	RSPN_ID	1-to-1	FC-GS-4	b
021A	Register Port Type	RPT_ID	1-to-1	FC-GS-4	b
021B	Register IP Address (Port) - Port Identifier	RIPP_ID	1-to-1	FC-GS-4	b
021D	Register Hard Address - Port Identifier	RHA_ID	1-to-1	FC-GS-4	b
021E	Register FC-4 Descriptors - Port Identifier	RFD_ID	1-to-1	FC-GS-4	b
021F	Register FC-4 Features - Port Identifier	RFF_ID	1-to-1	FC-GS-4	b
0235	Register IP Address (Node)	RIP_NN	1-to-All	FC-GS-4	b
0236	Register Initial Process Associator	RIPA_NN	1-to-All	FC-GS-4	b
0239	Register Symbolic Node_Name	RSNN_NN	1-to-All	FC-GS-4	b
0300	De-register all	DA_ID	1-to-1	FC-GS-4	c
0410	Get Entry, based on Port Identifier	GE_ID	Any	FC-SW-3	d
0420	Get Entry, based on Port_Name	GE_PN	Any	FC-SW-3	d
0430	Get Entries, based on Node_Name	GE_NN	Any	FC-SW-3	d
0450	Get Entries, based on IP address	GE_IP	Any	FC-SW-3	d
0470	Get Entries, based on FC-4 TYPE	GE_FT	Any	FC-SW-3	d
04A0	Get Entries, based on Port Type	GE_PT	Any	FC-SW-3	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 standard.

[c] De-registration requests may be distributed if performed by a third party. Otherwise, they are local and outside the scope of this standard.

[d] Work Categories for Name Server Entry Object requests are at the discretion of the originating Switch.

Table 94 – FC-CT Command Codes for dNS (part 4 of 4)

Encoded Value (hex)	Description	Mnemonic	Work Category	Payload Defined in	Notes
04B0	Get Entries, based on Zone Member	GE_ZM	Any	FC-SW-3	
04C0	Get Entries, Based on Zone Name	GE_ZN	Any	FC-SW-3	
04D0	Get Entries, Based on Port IP Address	GE_IPP	Any	FC-SW-3	
04E0	Get Entries, Based on FC-4 Features	GE_FF	Any	FC-SW-3	
04F0	Get Entries, Based on Fabric Port_Name	GE_FPN	Any	FC-SW-3	
0500	Remove All	RA	1-to-1	FC-SW-3	
8001	Reject CT_IU	CT_RJT	1-to-1	FC-GS-4	
8002	Accept CT_IU	CT_ACC	1-to-1	FC-GS-4	

[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 standard.

[c] De-registration requests may be distributed if performed by a third party. Otherwise, they are local and outside the scope of this standard.

[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 94.

9.3.3 Name Server Objects

The Name Server Objects communicated between distributed Name Servers using FC-CT are as defined in FC-GS-4 with no modification, but with several additions. The format of a Name Server Entry Object is as shown in table 95.

Table 95 – Name Server Entry Object (part 1 of 2)

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	IP address (Node)	16	
Yes	Class of Service	4	
Yes	FC-4 TYPEs	32	
Yes	IP Address (Port)	16	
Yes	F_Port_Name	8	

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.

c See table 96 for the description of the FC-4 Descriptor as it relates to the Name Server Entry object.

Table 95 – Name Server Entry Object (part 2 of 2)

Mandatory	Item	Size Bytes	Comments
Yes	Reserved	1	
Yes	Hard Address	3	
No	FC-4 Features	128	[b]
No	FC-4 Descriptor	260	b and c

[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.

[c] See table 96 for the description of the FC-4 Descriptor as it relates to the Name Server Entry object.

All fields shall be a fixed length as indicated in table 95. 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-4.

The format of the FC-4 Descriptor as it relates to the Name Server Entry object is depicted in table 96 below:

Table 96 – FC-4 Descriptor Format for Name Server Object

Item	Size Bytes	Notes
FC-4 Type	1	
Number of FC-4 Descriptor Types Registered	1	
Reserved	2	
FC-4 Descriptor length	1	
FC-4 Descriptor	255	[a]

[a] For use in the Name Server Object, this field is always a fixed length of 255 bytes, even though the contents may have a different length.

FC-4 Type: This field indicates the FC-4 Type for which the FC-4 Descriptor is associated.

Number of FC-4 Descriptor Types Registered: This field indicates the total number of FC-4 Type Descriptors registered.

FC-4 Descriptor Length and FC-4 Descriptor: Follows the format of the FC-4 Descriptor described in FC-GS-4.

The rules associated with the FC-4 Descriptor are listed below:

a) if more than one FC-4 Descriptor is registered per port being returned, then only a single FC-4 Descriptor shall be returned in the Name Server Object in the response;

b) if more than one FC-4 Descriptor is registered per port, the determination of which single FC-4 Descriptor is returned in the Name Server Object is outside the scope of this standard; and

c) if more than one FC-4 Descriptor is registered for the responding port, the remaining FC-4 Descriptors may be accessed via Name Server to Name Server requests specific to FC-4 Descriptors (e.g., GFD_ID).

The Entry Object Format Indicator is depicted in table 97.

Table 97 – 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 and FC-4 Descriptor fields are Included in the Entry Object.
2-7	Reserved

The sizes of the Name Server Entry Object is depicted in table 98.

Table 98 – 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 + FC-4 Descriptor
03	500	Small Name Server Entry Object + FC-4 Features + FC-4 Descriptor

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 an FC-4 Descriptor or 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, Node Symbolic Name, or FC-4 Descriptor 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 an FC-4 Descriptor or 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 an FC-4 Descriptor or 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 Remove All

The Remove All shall be used to delete all locally copied Entries in the database of another dNS for a given Port Identifier. A dNS should issue an RA request only if the associated Port Identifier is removed or has disappeared from the Fabric, or if the Port Identifier has been reused.

The dNS shall accept RA requests received from any valid Domain Controller Identifier. The dNS may reject an RA request from any other source.

The format of the RA request is shown in table 99.

Table 99 – RA 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-4. A dNS shall not reject an RA request if it has no locally copied Entry associated with the Port Identifier.

The format of the RA reply is shown in table 100.

Table 100 – RA Accept payload

Item	Size Bytes
FC-CT Header	16

9.3.4.2 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 101.

Table 101 – 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-4. The dNS may reject a GE_ID request for reasons not specified in this standard.

The format of the reply payload to a GE_ID request is shown in table 102.

Table 102 – 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.

9.3.4.3 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 103.

Table 103 – 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-4. The dNS may reject a GE_PN request for reasons not specified in this standard.

The format of the reply payload to a GE_PN request is shown in table 104.

Table 104 – 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.

9.3.4.4 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 105.

Table 105 – 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-4. The dNS may reject a GE_NN request for reasons not specified in this standard.

The format of the reply payload to a GE_NN request is shown in table 106.

Table 106 – 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.

9.3.4.5 Get Entries based on IP address (Node)

The dNS shall, when it receives a GE_IP request, return the Entry object for the specified IP address. The format of the GE_IP request is shown in table 107.

Table 107 – GE_IP request payload

Item	Size Bytes
FC-CT Header	16
IP address (Node)	16

The Node IP address format shall be as defined in FC-GS-4. The dNS may reject a GE_IP request for reasons not specified in this standard.

The format of the reply payload to a GE_IP request is shown in table 108.

Table 108 – GE_IP 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 IP address.

9.3.4.6 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 and more than one FC-4 TYPE may be specified. The format of the GE_FT request is shown in table 109.

Table 109 – 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-4. The dNS may reject a GE_FT request for reasons not specified in this standard.

The format of the reply payload to a GE_FT request is shown in table 110.

Table 110 – 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.

9.3.4.7 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 111.

Table 111 – 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-4. The dNS may reject a GE_PT request for reasons not specified in this standard.

The format of the reply payload to a GE_PT request is shown in table 112.

Table 112 – 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.

9.3.4.8 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 113.

Table 113 – GE_ZM request payload

Item	Size Bytes
FC-CT Header	16
Zone Member	n

The Zone Member format shall be as defined in 10.4.4.6.2.

The format of the reply payload to a GE_ZM request is shown in table 114.

Table 114 – 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.

9.3.4.9 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 115.

Table 115 – 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 116.

Table 116 – 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.

9.3.4.10 Get Entries based on Port IP Address

The dNS shall, when it receives a GE_IPP request, return the Entry objects for the Port IP Address specified in the GE_IPP request. The format of the GE_IPP request is shown in table 117.

Table 117 – GE_IPP request payload

Item	Size Bytes
FC-CT Header	16
Port IP Address	16

The format of the Port IP Address is as specified in FC-GS-4. The dNS may reject a GE_IPP for reasons not specified in this standard.

The format of the reply payload to a GE_IPP request is shown in table 118.

Table 118 – GE_IPP 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 IP address.

9.3.4.11 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 119.

Table 119 – 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-4. 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 120.

Table 120 – 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.

9.3.4.12 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 121.

Table 121 – 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-4. The dNS may reject a GE_FPN request for reasons not specified in this standard. The format of the reply payload to a GE_FPN request is shown in table 122.

Table 122 – 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 Server

9.4.1 General Behavior

The distributed Management Server is provided as follows:

a) Each Switch contains its own resident Management Server, 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-4) to the Entry Switch via the well-known address;

d) The dMS within the Entry Switch services the request by making any needed requests of other dMS 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 dMS to acquire the requested information is transparent to the original requesting client;

g) Partial responses for some dMS Services are allowed;

h) Management Server responses returned to a client are not subject to zoning.

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 Services;
02h - Unzoned Name Service;
10h - Fabric Device Management Server
others - Reserved.

Command Code: see tables 94, 123, 127, 132, and 133.

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 123.

Table 123 – Fabric Configuration Service Command Codes for dMS (part 1 of 3)

Encoded Value (hex)	Description	Mnemonic	Work Category	Payload Defined in	Capability See 6.1.22.5.3
0100	Get Topology Information	GTIN	1-to-1	FC-GS-4	Topology
0101	Get Interconnect Element (IE) List[a]	GIEL	Local	FC-GS-4	Basic
0111	Get Interconnect Element Type	GIET	Local or 1-to-1	FC-GS-4	Basic
0112	Get Domain Identifier[a]	GDID	Local	FC-GS-4	Basic
0113	Get Management Identifier	GMID	Local or 1-1	FC-GS-4	Basic
0114	Get Fabric Name[a]	GFN	Local	FC-GS-4	Basic
0115	Get Interconnect Element Logical Name	GIELN	Local or 1-to-1	FC-GS-4	Basic
0116	Get Interconnect Element Management Address List	GMAL	Local or 1-to-1	FC-GS-4	Basic
0117	Get Interconnect Element Information List	GIEIL	Local or 1-to-1	FC-GS-4	Basic
0118	Get Port List	GPL	Local or 1-to-1	FC-GS-4	Basic
0121	Get Port Type	GPT	Local or 1-to-All	FC-GS-4	Basic
0122	Get Physical Port Number	GPPN	Local or 1-to-All	FC-GS-4	Basic
0124	Get Attached Port_Name List	GAPNL	Local or 1-to-All	FC-GS-4	Basic
0126	Get Port State	GPS	Local or 1-to-All	FC-GS-4	Basic

[a] These requests are handled by the Entry Switch with no assistance from other Switches.

Table 123 – Fabric Configuration Service Command Codes for dMS (part 2 of 3)

Encoded Value (hex)	Description	Mnemonic	Work Category	Payload Defined in	Capability See 6.1.22.5.3
0127	Get Port Speed Capabilities	GPSC	Local or 1-to-All	FC-GS-4	Basic
0128	Get Attached Topology Information	GATIN	Local or 1-to-All	FC-GS-4	Topology
0130	Get Switch Enforcement Status	GSES	Local or 1-to1	FC-GS-4	Enhanced
0191	Get Platform Node_Name List	GPLNL	Local or 1-to-All	FC-GS-4	Platform
0192	Get Platform Type	GPLT	Local or 1-to-All	FC-GS-4	Platform
0193	Get Platform Management Address List	GPLML	Local or 1-to-All	FC-GS-4	Platform
0197	Get Platform Attribute Block	GPAB	Local or 1-to-All	FC-GS-4	Platform
01A1	Get Platform Name - Node_Name	GNPL	Local or 1-to-All	FC-GS-4	Platform
01A2	Get Platform List	GPNL	Local or 1-to-All	FC-GS-4	Platform
01A4	Get Platform FCP Type	GPFCP	Local or 1-to-All	FC-GS-4	Platform
01A5	Get Platform OS LUN Mappings	GPLI	Local or 1-to-All	FC-GS-4	Platform
01B1	Get Node Identification Data - Node_Name	GNID	Local or 1-to-All	FC-GS-4	Platform
215	Register Interconnect Element Logical Name	RIELN	Local or 1-to-1	FC-GS-4	Basic
0280	Register Platform	RPL	Local or 1-to-All	FC-GS-4	Platform
0291	Register Platform Node_Name	RPLN	Local or 1-to-All	FC-GS-4	Platform

[a] These requests are handled by the Entry Switch with no assistance from other Switches.

Table 123 – Fabric Configuration Service Command Codes for dMS (part 3 of 3)

Encoded Value (hex)	Description	Mnemonic	Work Category	Payload Defined in	Capability See 6.1.22.5.3
0292	Register Platform Type	RPLT	Local or 1-to-All	FC-GS-4	Platform
0293	Register Platform Management Address	RPLM	Local or 1-to-All	FC-GS-4	Platform
0298	Register Platform Attribute Block	RPAB	Local or 1-to-All	FC-GS-4	Platform
029A	Register Platform FCP Type	RPFCP	Local or 1-to-All	FC-GS-4	Platform
029B	Register Platform OS LUN Mappings	RPLI	Local or 1-to-All	FC-GS-4	Platform
0380	Deregister Platform	DPL	Local or 1-to-All	FC-GS-4	Platform
0391	Deregister Platform Node_Name	DPLN	Local or 1-to-All	FC-GS-4	Platform
0392	Deregister Platform Management Address	DPLM	Local or 1-to-All	FC-GS-4	Platform
0393	Deregister Platform Management Address List	DPLML	Local or 1-to-All	FC-GS-4	Platform
0394	Deregister Platform OS LUN Mappings	DPLI	Local or 1-to-All	FC-GS-4	Platform
0395	Deregister Platform Attribute Block	DPAB	Local or 1-to-All	FC-GS-4	Platform
039F	De-Register All Platform Information	DPALL	Local or 1-to-All	FC-GS-4	Platform

[a] These requests are handled by the Entry Switch with no assistance from other Switches.

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 HBAs 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-4. 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 in table 124.

Table 124 – FDMI Inter-Switch Message

Item	Size Bytes
FC-CT Header	See FC-GS-4
FDMI Header	28
Payload	n

9.4.6.3.2 FC-CT Header

The FC-CT header follows the format specified in FC-GS-4. 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 125 – 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 126.

Table 126 – Vendor Specified

Item	Size (Bytes)
Vendor Identifier	8
Vendor Specified Information	8

Vendor Identifier: Contains the eight byte T10 administered vendor identifier.

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 127. 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 127 – 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
E103	FDRF	De-Registration Forward	FDMI Header, GS Client Payload	Null

Table 127 – FDMI Fabric Internal Command Codes (part 2 of 2)

Code	Mnemonic	Description	Request Attributes	Accept Attributes
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 128 – 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 129 – 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 130 below.

Table 130 – 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 131 below.

Table 131 – 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 132.

Table 132 – 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-4
0101	Get HBA Attributes	GHAT	Local or 1-to-1	FC-GS-4
0102	Get Registered Port List	GRPL	Local or 1-to-1	FC-GS-4
0110	Get Port Attributes	GPAT	Local or 1-to-1	FC-GS-4
0200	Register HBA	RHBA	Local	FC-GS-4
0201	Register HBA Attributes	RHAT	Local	FC-GS-4
0210	Register Port	RPRT	Local	FC-GS-4
0211	Register Port Attributes	RPA	Local	FC-GS-4
0300	De-Register HBA	DHBA	Local	FC-GS-4
0301	De-Register HBA Attributes	DHAT	Local	FC-GS-4
0310	De-Register Port	DPRT	Local	FC-GS-4
0311	De-Register Port Attributes	DPA	Local	FC-GS-4

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 133.

Table 133 – 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

9.4.7.2.1 Overview

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 134 – GCAP Request Payload

Item	Size (Bytes)
Null	0

Table 135 – 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

9.4.7.2.2 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 136 – 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.

9.4.7.2.3 Subtype Capability Bit Masks

The Capability Bit Masks currently defined by this standard are listed below.

Table 137 – Fabric Configuration Server (CT_Subtype 01h)

Option Bit(s)	Description (see 6.1.22.5.3)
0	Basic Configuration Services
1	Platform Configuration Services
2	Topology Discovery Configuration Services
3	Enhanced Configuration Services
4-31	Reserved

Table 138 – Unzoned Name Server (CT_Subtype 02h)

Option Bit(s)	Description (see 9.4.4)
0	Basic Unzoned Name Services
1-31	Reserved

10 Switch Zone Exchange and Merge

10.1 Overview

This clause describes a mechanism for Switches to exchange zoning data. FC-GS-4 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 35 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 6.1.22.5.4) 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 6.1.16).

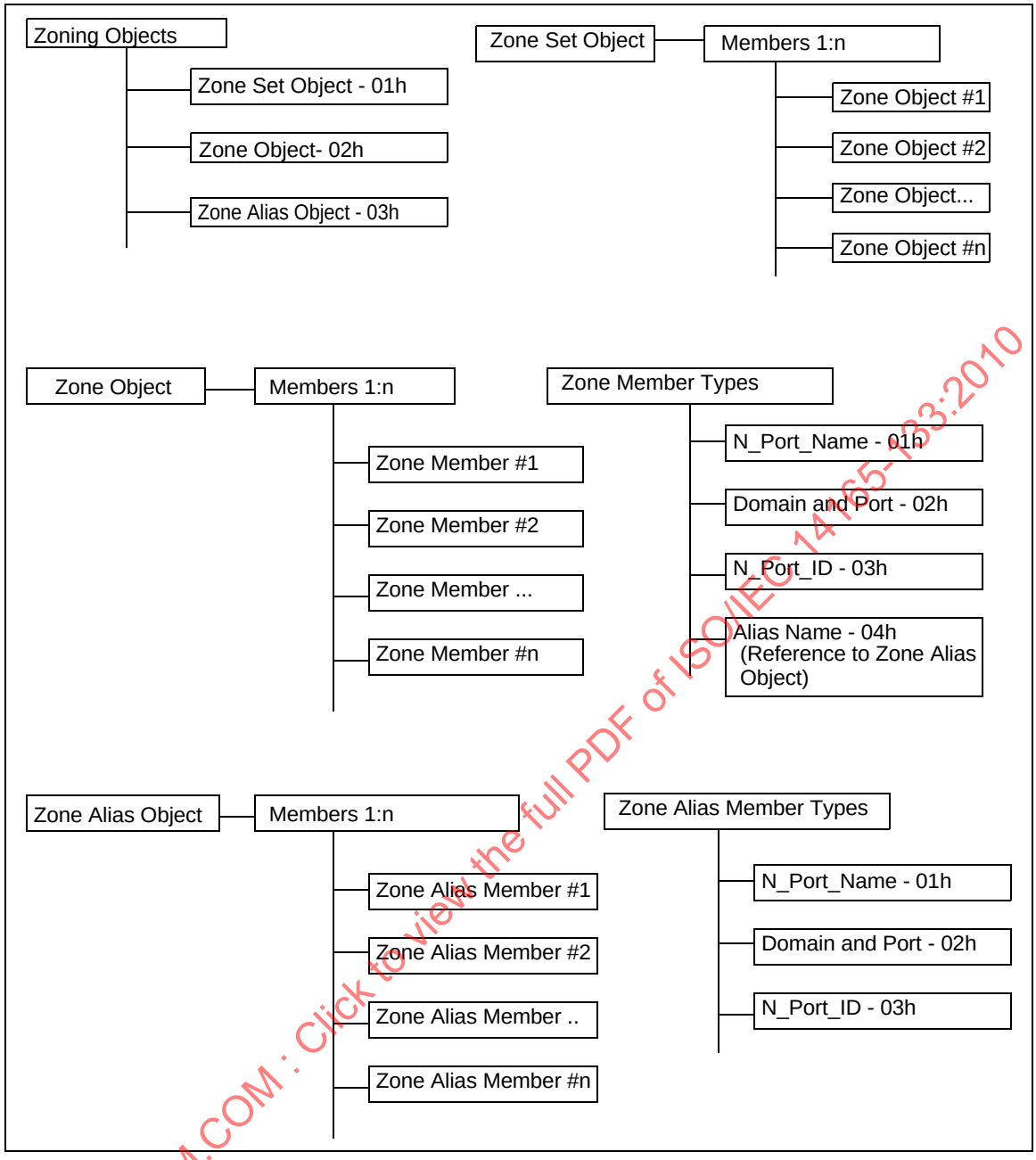
If all the Switches in the Fabric support the Zone Set Database, then the Zone Set Database supported bit (bit 4) of the Fabric Enhanced Zoning support flags of the GFEZ Accept shall be set to one, otherwise it shall be set to 0.

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 30 – Basic Zoning Framework



Zoning Objects: Three types of Zoning Objects are defined. They are:

a) Zone Set;

b) Zone; and

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; and

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; and

d) Zone Alias Name.

The Zone Object has three attributes:

a) Name;

b) Protocol Type; and

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; and

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; and

b) Number of Members.