

# Cisco MDS 9000 Series IP Services Configuration Guide, Release 9.x

**First Published:** 2021-08-25

**Last Modified:** 2026-03-04

**Americas Headquarters**
Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134-1706
USA
http://www.cisco.com
Tel: 408 526-4000
800 553-NETS (6387)
Fax: 408 527-0883

THE SPECIFICATIONS AND INFORMATION REGARDING THE PRODUCTS IN THIS MANUAL ARE SUBJECT TO CHANGE WITHOUT NOTICE. ALL STATEMENTS, INFORMATION, AND RECOMMENDATIONS IN THIS MANUAL ARE BELIEVED TO BE ACCURATE BUT ARE PRESENTED WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED. USERS MUST TAKE FULL RESPONSIBILITY FOR THEIR APPLICATION OF ANY PRODUCTS.

THE SOFTWARE LICENSE AND LIMITED WARRANTY FOR THE ACCOMPANYING PRODUCT ARE SET FORTH IN THE INFORMATION PACKET THAT SHIPPED WITH THE PRODUCT AND ARE INCORPORATED HEREIN BY THIS REFERENCE. IF YOU ARE UNABLE TO LOCATE THE SOFTWARE LICENSE OR LIMITED WARRANTY, CONTACT YOUR CISCO REPRESENTATIVE FOR A COPY.

The Cisco implementation of TCP header compression is an adaptation of a program developed by the University of California, Berkeley (UCB) as part of UCB's public domain version of the UNIX operating system. All rights reserved. Copyright © 1981, Regents of the University of California.

NOTWITHSTANDING ANY OTHER WARRANTY HEREIN, ALL DOCUMENT FILES AND SOFTWARE OF THESE SUPPLIERS ARE PROVIDED “AS IS" WITH ALL FAULTS. CISCO AND THE ABOVE-NAMED SUPPLIERS DISCLAIM ALL WARRANTIES, EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, THOSE OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OR ARISING FROM A COURSE OF DEALING, USAGE, OR TRADE PRACTICE.

IN NO EVENT SHALL CISCO OR ITS SUPPLIERS BE LIABLE FOR ANY INDIRECT, SPECIAL, CONSEQUENTIAL, OR INCIDENTAL DAMAGES, INCLUDING, WITHOUT LIMITATION, LOST PROFITS OR LOSS OR DAMAGE TO DATA ARISING OUT OF THE USE OR INABILITY TO USE THIS MANUAL, EVEN IF CISCO OR ITS SUPPLIERS HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

Any Internet Protocol (IP) addresses and phone numbers used in this document are not intended to be actual addresses and phone numbers. Any examples, command display output, network topology diagrams, and other figures included in the document are shown for illustrative purposes only. Any use of actual IP addresses or phone numbers in illustrative content is unintentional and coincidental.

All printed copies and duplicate soft copies of this document are considered uncontrolled. See the current online version for the latest version.

Cisco has more than 200 offices worldwide. Addresses and phone numbers are listed on the Cisco website at www.cisco.com/go/offices.

Cisco and the Cisco logo are trademarks or registered trademarks of Cisco and/or its affiliates in the U.S. and other countries. To view a list of Cisco trademarks, go to this URL: https://www.cisco.com/c/en/us/about/legal/trademarks.html. Third-party trademarks mentioned are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (1721R)

© 2021–2026 Cisco Systems, Inc. All rights reserved.



# CONTENTS


**Full Cisco Trademarks with Software License** ?

PREFACE

**Preface** xiii

Audience xiii

Document Conventions xiii

Documentation Feedback xv

Related Documentation xv

Communications, Services, and Additional Information xv

CHAPTER 1

**New and Changed Information** 1

New and Changed Information 1

CHAPTER 2

**Overview** 3

Overview 3

CHAPTER 3

**Configuring FCIP** 7

Feature Information 7

Overview of FCIP 8

FCIP Concepts 8

FCIP High-Availability Solutions 10

Fibre Channel Port Channels 10

FSPF 11

Configuring FCIP 12

Enabling FCIP 12

Basic FCIP Configuration 13

Creating FCIP Profiles 13

Displaying FCIP Profile Information 14
Creating FCIP Links 15
Displaying FCIP MAC Statistics 17
Advanced FCIP Profile Configuration 18
Configuring TCP Listener Ports 18
Configuring TCP Parameters 18
Minimum Retransmit Timeout 19
Keepalive Timeout 19
Maximum Retransmissions 20
Maximum Retransmission Threshold for OBFL Logging 20
Path MTUs 21
Selective Acknowledgments 22
Window Management 22
Monitoring Congestion 23
Displaying FCIP Profile Information 24
Advanced FCIP Interface Configuration 25
Configuring Peers 25
Assigning Peer IP Address (IPv4) 25
Assigning Peer IP Address (IPv6) 26
Configuring Number of TCP Connections 26
Configuring Active Connections 27
Enabling Time Stamp Control 28
Quality of Service 29
Configuring E Ports 29
Displaying FCIP Interface Information 30
Advanced FCIP Features 36
FCIP Write Acceleration 36
Configuring FCIP Write Acceleration 39
Displaying Write Acceleration Activity Information 39
FCIP Tape Acceleration 40
Tape Library LUN Mapping for FCIP Tape Acceleration 43
Configuring FCIP Tape Acceleration 45
Displaying Tape Acceleration Activity Information 46
FCIP Compression 47

Configuring FCIP Compression 48
Displaying FCIP Compression Information 48
Configuring FCIP Links for Maximum Performance 49
Configuring FCIP Links for Maximum Performance on a Cisco MDS 9250i Switch 50
Configuring FCIP performance in 1 Gbps mode 50
Configuring FCIP performance in 10 Gbps mode 52
Configuring FCIP Links for Maximum Performance on Cisco MDS 24/10 port SAN Extension Module and M9220i switch 53
Configuring FCIP performance in 1 Gbps mode 53
Configuring FCIP performance in 10 Gbps mode 54
Configuring FCIP performance in 25 Gbps mode 56
Configuring FCIP performance in 40 Gbps mode 57
Verifying FCIP Configuration 59
Default Settings for FCIP Parameters 59

CHAPTER 4 **Configuring the SAN Extension Tuner 61**
Overview of SAN Extension Tuner 61
SAN Extension Tuner Setup 63
Data Pattern 63
License Prerequisites 64
Configuring the SAN Extension Tuner 64
Tuning the FCIP Link 64
Enabling the Tuner 64
Configuring nWWN 65
Configuring the Virtual N Port 65
Generating SCSI Disk Read/Write IO 65
Generating SCSI Disk Read/Write IO One Time 66
Generating SCSI Disk Read/Write IO Continuously 66
Transferring Ready Size for SCSI 67
Generating SCSI Tape Read/Write IO 67
Generating SCSI Tape Read/Write IO One Time 68
Generating SCSI Tape Read/Write IO Continuously 68
Transferring Ready Size for SCSI 69
Configuring a Data Pattern 69

Verifying the SAN Extension Tuner Configuration 70
Default Settings for Tuning Parameters 72

CHAPTER 5 **Configuring iSCSI 73**
Overview of iSCSI 73
iSCSI Configuration Limits 76
Configuring iSCSI 76
Enabling iSCSI 77
Presenting iSCSI Hosts as Virtual Fibre Channel Hosts 77
Creating iSCSI Interfaces 77
Presenting Fibre Channel Targets as iSCSI Targets 83
Dynamic Mapping 84
Static Mapping 85
Configuing an Interface to Advertise the iSCSI Virtual Target 85
Presenting iSCSI Hosts as Virtual Fibre Channel Hosts 86
Specifying the Initiator Identification Mode 86
iSCSI Virtual Target Configuration Examples 86
Initiator Presentation Modes 88
Configuring Initiator Idle Timeout 90
WWN Assignment for iSCSI Initiators 90
Configuring Dynamic Mapping Using the Name Option for an iSCSI Initiator 92
Configuring Static Mapping Using the Name Option for an iSCSI Initiator 92
Configuring Static Mapping Using the IP-Address Option for an iSCSI Initiator 92
Assigning the WWN for an iSCSI Initiator 93
Saving the Automatically Assigned nWWN/pWWN Mapping 94
Checking and Removing WWN Conflicts 94
Proxy Initiator Mode 95
Configuring Proxy Initiator 96
VSAN Membership for iSCSI 97
Assigning VSAN Membership for iSCSI Hosts 97
Configuring Default Port VSAN for iSCSI Interfaces 98
VSAN Membership for iSCSI Devices 98
iSCSI Access Control 99
Configuring AAA Authentication for iSCSI User 104

Configuring Authentication Mechanism for iSCSI 104
Configuring Authentication Mechanism for iSCSI sessions on an Interface 104
Configuring Local Authentication 105
Restricting iSCSI Initiator Authentication 105
Configuring Mutual CHAP Authentication 106
Configuring Per-Initiator iSCSI Target User Name and Password for Authentication 107
iSCSI Immediate Data and Unsolicited Data Features 108
Advanced Features for iSCSI Interfaces for Cisco MDS Switches 108
Setting QoS Values 111
Displaying iSCSI Information 111
Configuring iSLB 120
iSLB Configuration Limits 121
iSLB Configuration Prerequisites 121
iSLB Initiators 122
Configuring iSLB Initiators 122
Configuring iSLB Initiator Names 122
Configuring iSLB Initiator IP Address 123
Assigning WWNs to iSLB Initiators 124
Assigning VSAN Membership for iSLB Initiators 126
Configuring Metrics for Load Balancing 127
Verifying iSLB Initiator Configuration 127
Configuring iSLB Initiator Targets 128
Configuring and Activating Zones for iSLB Initiators and Initiator Targets 129
Verifying iSLB Zoning Configuration 130
Configuring iSLB Session Authentication 130
Restricting iSLB Initiator Authentication 131
Mutual CHAP Authentication 131
Verifying iSLB Authentication Configuration 132
Load Balancing Using VRRP 132
VRRP Load Balancing Algorithm for Selecting Gigabit Ethernet Interfaces 134
Configuring Load Balancing Using VRRP 137
Enabling VRRP for Load Balancing 137
Verifying iSLB VRRP Load Balancing Configuration 137
Displaying iSLB VRRP Information 138

iSLB Configuration Distribution Using CFS 138
Distributing iSLB Configuration Using CFS 139
Enabling iSLB Configuration Distribution 139
Locking the Fabric 139
Committing Changes to the Fabric 140
Discarding Pending Changes 140
Clearing a Fabric Lock 141
CFS Merge Process 141
Displaying Pending iSLB Configuration Changes 141
Displaying iSLB CFS Status 142
iSLB CFS Merge Status Conflicts 142
iSCSI High Availability 143
Transparent Target Failover 143
Createing a Static iSCSI Virtual Target 145
Enabling Trespass Feature for a Static iSCSI Virtual Target 146
Multiple IPS Ports Connected to the Same IP Network 147
VRRP-Based High Availability 147
Ethernet Port Channel-Based High Availability 148
iSCSI Authentication Setup Guidelines and Scenarios 149
Configuring No Authentication 149
Configuring CHAP with Local Password Database 149
Configuring CHAP with External RADIUS Server 150
iSCSI Transparent Mode Initiator 151
Configuring iSCSI Scenario 1 152
Configuring iSCSI Scenario 2 156
Configuring iSCSI Scenario 1 160
Configuring iSCSI Scenario 2 164
Overview of Internet Storage Name Service 169
Overview of iSNS Client Functionality 170
Creating an iSNS Client Profile 170
Tagging a Profile to an Interface 171
Verifying iSNS Client Configuration 171
iSNS Server Functionality 173
Configuring an iSNS Server 174

Enabling an iSNS Server 175
iSNS Configuration Distribution 175
Configuring the ESI Retry Count 175
Configuring a Registration Period 176
iSNS Client Registration and Deregistration 176
Verifying the iSNS Server Configuration 177
iSNS Cloud Discovery 183
Cloud Discovery 183
Configuring iSNS Cloud Discovery 184
Default Settings 186

CHAPTER 6 **Configuring IP Services** 189
Traffic Management Services 190
Management Interface Configuration 190
Configuring the mgmt0 Ethernet interface for IPv4 191
Configuring the mgmt0 Ethernet interface for IPv6 191
Default Gateway 192
Configuring the Default Gateway 192
Verifying the Default Gateway Configuration 192
IPv4 Default Network Configuration 193
IP over Fibre Channel 194
IPFC Configuration 194
Configuring an IPv4 Address in a VSAN 195
Verifying the VSAN Interface Configuration 195
Enabling IPv4 Routing 195
IPFC Configuration Example 196
IPv4 Static Routes 198
Configuring IPv4 Static Routes 199
Verifying IPv4 Static Route Information 199
Displaying and Clearing ARPs 200
Overlay VSANs 200
Configuring Overlay VSANs 200
Configuring Multiple VSANs 202
DNS Configuration 204

Displaying DNS Host Information 206
Default Settings for DNS Features 206

CHAPTER 7 **Configuring IP Storage Services** 207
Feature Information 208
Supported Hardware 208
Upgrading FCIP and iSCSI Interfaces 208
Configuring IPS Ports on Cisco MDS 9220i Switch 208
Configuring 40 Gbps Speed on Cisco MDS 24/10 port SAN Extension Module 209
Configuring IPStorage Interfaces for IPv4 210
Basic IPStorage Configuration 210
Fibre Channel Module with IPS Ports Core Dumps 211
About VLANs for IPStorage 212
Interface Subnet Requirements 213
Verifying IPStorage Connectivity 213
IPStorage IPv4-ACL Guidelines 214
Configuring IPStorage High Availability 214
VRRP for iSCSI and FCIP Services 215
Configuring VRRP for IPStorage Interfaces 215
Configuring CDP 217
Changing Link Speed on IPStorage Interfaces 217
Changing Link Speed on Cisco MDS 9250i Multiservice Fabric Switch 217
Configuring 1 Gbps Link Speed on an IPStorage Interface 217
Configuring 10 Gbps Link Speed on an IPStorage Interface 218
Changing Link Speed on Cisco MDS 9220i Switch 219
Configuring 1 Gbps Link Speed on an IPStorage Interface 219
Configuring 10 Gbps Link Speed on an IPStorage Interface 220
Configuring 25 Gbps Link Speed on an IPStorage Interface 220
Configuring 40 Gbps Link Speed on an IPStorage Interface 221
Changing Link Speed on Cisco MDS 24/10 port SAN Extension Module 221
Configuring 1 Gbps Link Speed on an IPStorage Interface 221
Configuring 10 Gbps Link Speed on an IPStorage Interface 222
Configuring 40 Gbps Link Speed on an IPStorage Interface 222
Displaying Statistics 223

Displaying Ethernet MAC Statistics 224
Displaying TCP Statistics 225
Default Settings for IPStorage Services Parameters 226

CHAPTER 8 Configuring IPv4 for Gigabit Ethernet Interfaces 229
Overview of IPv4 229
Basic Gigabit Ethernet Configuration for IPv4 230
Configuring Autonegotiation 231
Configuring the MTU Frame Size 231
Configuring Promiscuous Mode 232
Verifying Gigabit Ethernet Connectivity 233
VLANs Support in Cisco MDS NX-OS 233
VLANs for Gigabit Ethernet 233
Configuring a VLAN Subinterface 234
Interface Subnet Requirements 234
Configuring Static IPv4 Routing 235
Displaying the IPv4 Route Table 235
IPv4-Access Control Lists 235
Gigabit Ethernet Guidelines 236
Applying on Gigabit Ethernet Interfaces 236
Address Resolution Protocol Cache 237
Displaying ARP Cache 237
Clearing ARP Cache 237
Displaying IPv4 Statistics 238
Default Settings for IPv4 Parameters 238

CHAPTER 9 Configuring IPv6 239
Overview of IPv6 239
Extended IPv6 Address Space for Unique Addresses 240
IPv6 Address Formats 240
IPv6 Address Prefix Format 241
IPv6 Address Type-Unicast 241
Global Addresses 241
Link-Local Address 242

IPv6 Address Type-Multicast 242
ICMP for IPv6 243
Path MTU Discovery for IPv6 244
IPv6 Neighbor Discovery 244
IPv6 Neighbor Solicitation and Advertisement Messages 245
Router Discovery 246
IPv6 Stateless Autoconfiguration 246
Dual IPv4 and IPv6 Protocol Stacks 247
Configuring Basic Connectivity for IPv6 248
Configuring IPv6 Addressing and Enabling IPv6 Routing 248
Configuring IPv4 and IPv6 Protocol Addresses 249
Verifying Basic IPv6 Connectivity Configuration and Operation 250
Verifying Neighbor Discovery Parameter Configuration 252
Duplicate Address Detection Attempts 252
Reachability Time 252
Retransmission Time 253
Verifying Neighbor Discovery Parameter Configuration 253
IPv6 Static Routes 254
Configuring an IPv6 Static Route 254
Verifying IPv6 Static Route Configuration and Operation 254
Gigabit Ethernet IPv6-ACL Guidelines 255
Transitioning from IPv4 to IPv6 256
Displaying IPv6 256
Default Settings 257



# Preface

This preface describes the audience, organization of, and conventions used in the Cisco MDS 9000 Series Configuration Guides. It also provides information on how to obtain related documentation, and contains the following sections:


- Audience, on page xiii
- Document Conventions, on page xiii
- Documentation Feedback, on page xv
- Related Documentation, on page xv
- Communications, Services, and Additional Information, on page xv


## Audience

This publication is for network administrators who install, configure, and maintain Cisco MDS 9000 Series Switches.

## Document Conventions

Command descriptions use these conventions:

| Convention | Description |
|---|---|
| **bold** | Bold text indicates the commands and keywords that you enter literally as shown. |
| *Italic* | Italic text indicates arguments for which the user supplies the values. |
| [x] | Square brackets enclose an optional element (keyword or argument). |
| [x \| y] | Square brackets enclosing keywords or arguments separated by a vertical bar indicate an optional choice. |
| {x \| y} | Braces enclosing keywords or arguments separated by a vertical bar indicate a required choice. |

| Convention | Description |
|---|---|
| [x {y \| z}] | Nested set of square brackets or braces indicate optional or required choices within optional or required elements. Braces and a vertical bar within square brackets indicate a required choice within an optional element. |
| `variable` | Indicates a variable for which you supply values, in context where italics cannot be used. |
| string | A nonquoted set of characters. Do not use quotation marks around the string or the string will include the quotation marks. |

Examples use these conventions:

| Convention | Description |
|---|---|
| `screen font` | Terminal sessions and information the switch displays are in screen font. |
| **`boldface screen font`** | Information you must enter is in boldface screen font. |
| *italic screen font* | Arguments for which you supply values are in italic screen font. |
| < > | Nonprinting characters, such as passwords, are in angle brackets. |
| [ ] | Default responses to system prompts are in square brackets. |
| !, # | An exclamation point (!) or a pound sign (#) at the beginning of a line of code indicates a comment line. |

This document uses the following conventions:



**Note** Means *reader take note*. Notes contain helpful suggestions or references to material not covered in the manual.



**Caution** Means *reader be careful*. In this situation, you might do something that could result in equipment damage or loss of data.



**Warning** IMPORTANT SAFETY INSTRUCTIONS

This warning symbol means danger. You are in a situation that could cause bodily injury. Before you work on any equipment, be aware of the hazards involved with electrical circuitry and be familiar with standard practices for preventing accidents. Use the statement number provided at the end of each warning to locate its translation in the translated safety warnings that accompanied this device.

SAVE THESE INSTRUCTIONS

# Documentation Feedback

To provide technical feedback on this document, or to report an error or omission, please send your comments to mds-docfeedback@cisco.com. We appreciate your feedback.

# Related Documentation

The entire Cisco MDS 9000 Series switches documentation set is available at the following URL:

https://www.cisco.com/c/en/us/support/storage-networking/mds-9000-nx-os-san-os-software/series.html

Documentation Roadmap

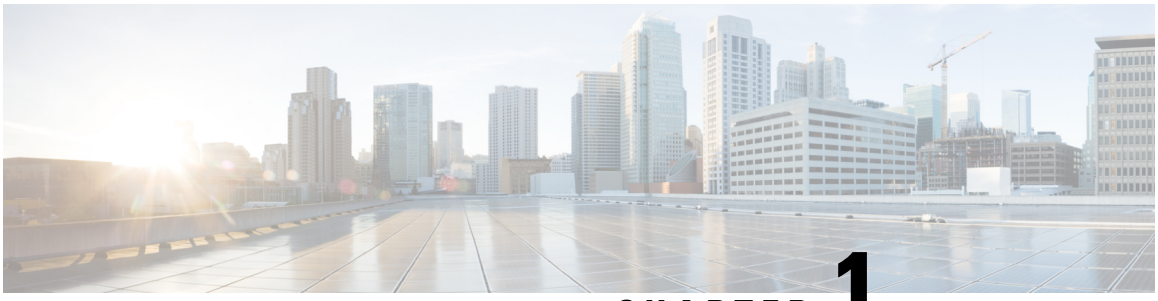
https://www.cisco.com/c/en/us/td/docs/storage/san_switches/mds9000/roadmaps/rel90.html

# Communications, Services, and Additional Information

- To receive timely, relevant information from Cisco, sign up at Cisco Profile Manager.
- To get the business results you're looking for with the technologies that matter, visit Cisco Services.
- To submit a service request, visit Cisco Support.
- To discover and browse secure, validated enterprise-class apps, products, solutions and services, visit Cisco DevNet.
- To obtain general networking, training, and certification titles, visit Cisco Press.
- To find warranty information for a specific product or product family, access Cisco Warranty Finder.

### Cisco Bug Search Tool

Cisco Bug Search Tool (BST) is a web-based tool that acts as a gateway to the Cisco bug tracking system that maintains a comprehensive list of defects and vulnerabilities in Cisco products and software. BST provides you with detailed defect information about your products and software.



CHAPTER 1

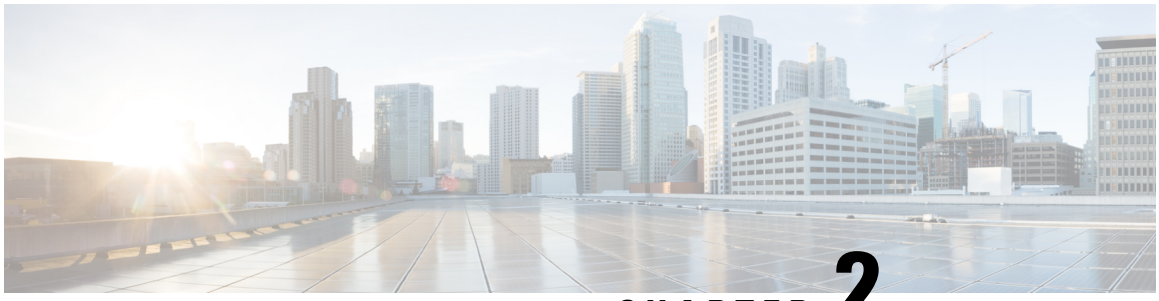
# New and Changed Information


• New and Changed Information, on page 1


## New and Changed Information

| Feature Name | Release | Description | Where Documented |
|---|---|---|---|
| FCIP MAC statistics | 9.4(3) | Added the Ethernet MAC statistics for an IPStorage interface used for FCIP | Configuring FCIP Links |
| **tcp min-retransmit-time** *milliseconds* | 9.4(2) | The default TCP minimum retransmit time is changed to 50 from 200 milliseconds.<br>The TCP minimum retransmit time range is changed to 50 to 5000 milliseconds from 200 to 5000 milliseconds. | Configuring FCIP Links |
| Maximum throughput on a single FCIP tunnel. | 9.3(1) | Maximum throughput on a single FCIP tunnel<br>• on Cisco MDS 9220i is 10Gbps.<br>• the 40 Gbps IPS port, the maximum throughput is 34.6 Gbps when 4 FCIP tunnels are created on separate VLAN subinterfaces.<br>• is 7 Gbps. For maximum performance, create two fcip tunnels on 10 Gbps IPS ports. | Configuring FCIP Links for Maximum Performance on Cisco MDS 24/10 port SAN Extension Module and M9220i switch |
| Introduced support for 25-Gbps IPS Port Speed for Cisco MDS 9220i. | 9.3(1) | 25-Gbps speed is supported from Cisco MDS NX-OS Release 9.3(1). | Configuring IPS Ports on Cisco MDS 9220i Switch |

New and Changed Information



CHAPTER 2

# Overview

The Cisco MDS 9000 NX-OS software provides features such as FCIP, SAN Extension Tuner, iSCSI, IP storage, IPv4, and IPv6 in a single platform. These IP services simplify SAN provisioning by automatically distributing configuration information to all the switches in a storage network. The Virtual Routing Redundancy Protocol (VRRP) increases the IP network availability for iSCSI and FCIP connections by allowing failover of connections from one port to another. The increased IP network availability facilitates the failover of an iSCSI volume from one IP services port to any other IP services port, either locally or on another Cisco MDS 9000 switch.


- Overview, on page 3


# Overview

This chapter includes the following sections:


- Fibre Channel over IP Protocol, on page 3
- SAN Extension Tuner, on page 4
- Internet Small Computer Systems Interface, on page 4
- IP Services, on page 4
- IP Storage, on page 4
- IPv4 and IPv6, on page 4


### Fibre Channel over IP Protocol

Fibre Channel over IP Protocol (FCIP) transparently connects a remote Fibre Channel storage area network (SAN island) by transporting Fibre Channel data from a local SAN to a remote SAN using IP networks. IP network availability for the FCIP connections can be increased by using features such as Virtual Routing Redundancy Protocol (VRRP) and quality of service (QoS). FCIP can be optimized for wire performance through enhancements that address out-of-order delivery issues, support jumbo frames, provide traffic shaping, and perform TCP optimization.

For more information on configuring FCIP, see Configuring FCIP, on page 7.

### SAN Extension Tuner

The SAN Extension Tuner (SET) feature helps you optimize FCIP performance by generating Small Computer System Interface (SCSI) I/O commands and directing the traffic to a specific virtual target. SET reports the I/Os per second and I/O latency results, which helps you to determine the number of concurrent I/Os needed to maximize the FCIP throughput.

For information on configuring the SAN Extension Tuner, see Configuring the SAN Extension Tuner, on page 61

### Internet Small Computer Systems Interface

The Internet Small Computer Systems Interface (iSCSI) feature allows an IP host to access Fibre Channel storage. This feature enables routing iSCSI requests and responses between iSCSI hosts in an IP network and Fibre Channel storage devices in the Fibre Channel SAN. The Fibre Channel storage devices are accessible from any Fibre Channel interface of the Cisco MDS 9000 Family switch.

For information on configuring iSCSI, see Configuring iSCSI, on page 73.

### IP Services

The IP Services modules allow you to extend storage networks using the Ethernet infrastructure. The Cisco MDS 9000 Family switches route IP traffic between Ethernet and Fibre Channel interfaces. The IP static routing feature is used to route the traffic between VSANs. An IP route using Fabric Manager and Device Manager. From NX-OS release 4.2(1) and later, CPP interfaces are also available for selection while creating a new IP route.

For information on configuring IP services, see Configuring IP Services, on page 189

### IP Storage

The IP Storage (IPS) Service module allows you to use the open-standard FCIP protocol to enable interconnection of SAN islands over extended distances. The Fibre Channel module with IPS ports and the MSM-18/4 module allow you to use FCIP and iSCSI features. Both modules integrate seamlessly into the Cisco MDS 9000 Family, and support the full range of features that are available on other switching modules, including VSANs, security, and traffic management.

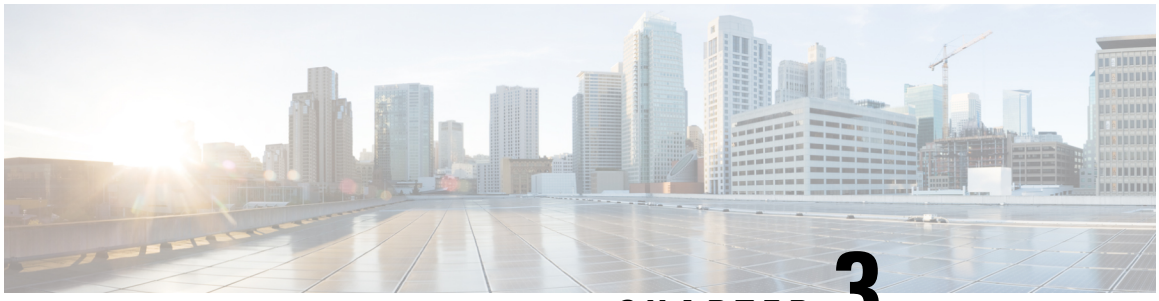
For information on configuring IP Storage, see Configuring IP Storage Services, on page 207

### IPv4 and IPv6

The Cisco MDS 9000 NX-OS software supports the IP version 4 (IPv4) and version 6 (IPv6) protocols on Gigabit Ethernet interfaces. The architecture of IPv6 has been designed to allow existing IPv4 users to transition easily to IPv6, while providing services such as end-to-end security, quality of service (QoS), and globally unique addresses. The dual stack approach for IPv4 and IPv6 allows Cisco MDS 9000 Family switches to connect to older IP networks, transitional networks of both versions, and IPv6 data networks.

The following table lists the Cisco MDS 9250i Multiservice Fabric Switch IPS features with the previous platforms (SSN16 and 18+4).

| Line Card | Physical Port Speed | Number of Physical Ports | Number of FCIP Tunnels | Number of FCIP Tunnels Bound to Each IPS/Gigabit Ethernet Interface | Number of iSCSI Ports |
|---|---|---|---|---|---|
| Cisco MDS 24/10 port SAN Extension Module | 1/10 Gbps | 8 | 24 | 3 | NA |
| Cisco MDS 9250i Multiservice Fabric Switch | 1/10 Gbps | 2 | 12 | 6 | 2 |
| SSN-16 | 1 Gbps | 16 | 48 | 3 | 16 |
| 18+4 | 1 Gbps | 4 | 12 | 3 | 4 |



CHAPTER 3

# Configuring FCIP

Cisco MDS 9000 Family IP Storage (IPS) services extend the reach of Fibre Channel SANs by using open-standard, IP-based technology. The switch can connect separated SAN islands using Fibre Channel over IP (FCIP).

**Note** FCIP is supported on Fibre Channel module with IPS ports on Cisco 24/10 port SAN Extension Module on Cisco MDS 9700 Series switches, MDS 9250i Multiservice Fabric Switch, and MDS 9220i Fabric Switch.

This chapter includes the following sections:


- Feature Information, on page 7
- Overview of FCIP, on page 8
- Configuring FCIP, on page 12
- Advanced FCIP Profile Configuration, on page 18
- Advanced FCIP Interface Configuration, on page 25
- Advanced FCIP Features, on page 36
- Configuring FCIP Links for Maximum Performance, on page 49
- Verifying FCIP Configuration, on page 59
- Default Settings for FCIP Parameters, on page 59


# Feature Information

This section briefly describes the new and updated features for releases.

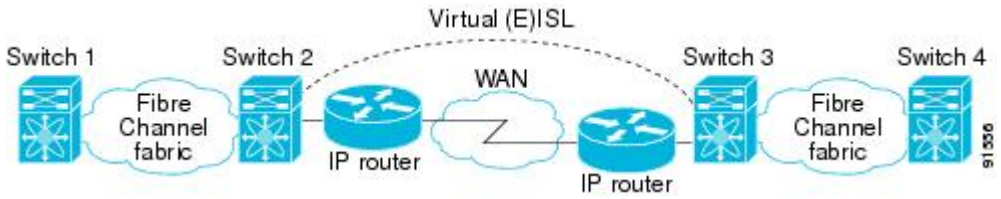
***Table 1: Feature Information Table***

| Feature | Release | Description |
|---|---|---|
| FCIP MAC statistics | 9.4(3) | Added the Ethernet MAC statistics for an IPStorage interface used for FCIP |
| **tcp min-retransmit-time** *milliseconds* | 9.4(2) | The default TCP minimum retransmit time is changed to 50 from 200 milliseconds.<br>The TCP minimum retransmit time range is changed to 50 to 5000 milliseconds from 200 to 5000 milliseconds. |

# Overview of FCIP

The Fibre Channel over IP Protocol (FCIP) is a tunneling protocol that connects geographically distributed Fibre Channel storage area networks (SAN islands) transparently over IP local area networks (LANs), metropolitan area networks (MANs), and wide area networks (WANs). The switch can connect separated SAN islands using Fibre Channel over IP (FCIP). (See the following figure).

***Figure 1: Fibre Channel SANs Connected by FCIP***




FCIP uses TCP as a network layer transport. The DF bit is set in the TCP header.

For more information about FCIP protocols, refer to the IETF standards for IPStorage at http://www.ietf.org. Also refer to the Fibre Channel standards for switch backbone connection at http://www.t11.org (see FC-BB-2).

This section includes the following topics:

# FCIP Concepts

To configure Fibre Channel module with IPS ports for FCIP, you should have a basic understanding of the following concepts:


- FCIP and VE Ports, on page 8
- FCIP Links, on page 9
- FCIP Profiles, on page 10
- FCIP Interfaces, on page 10

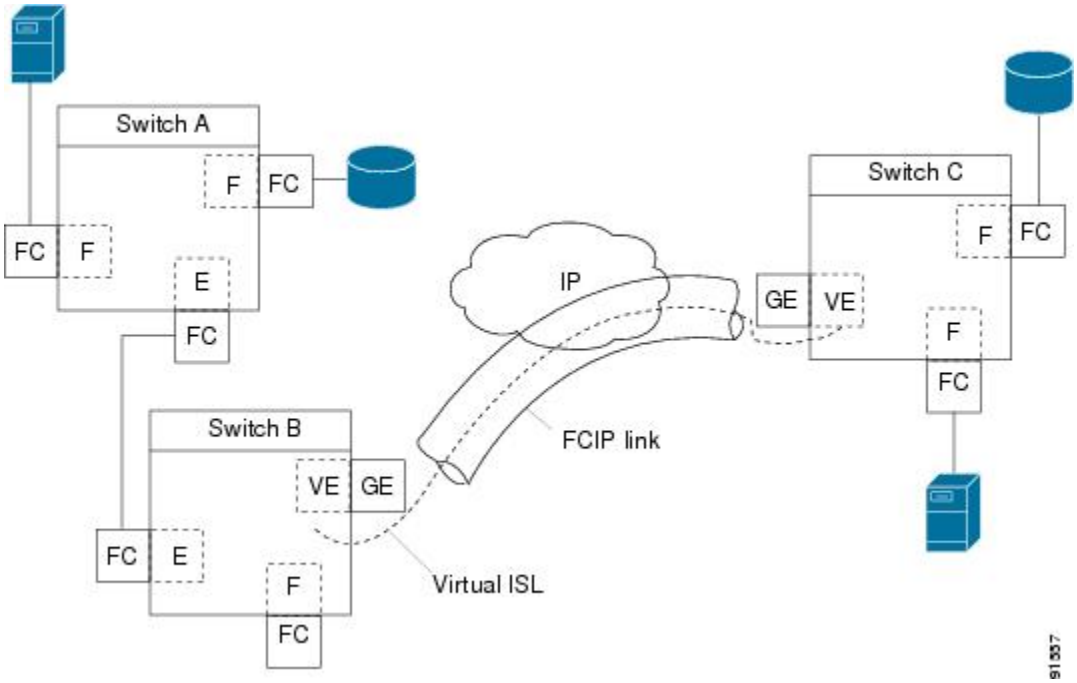

### FCIP and VE Ports

The below figure shows the internal model of FCIP in relation to Fibre Channel Inter-Switch Links (ISLs) and Cisco's extended ISLs (EISLs).

FCIP virtual E (VE) ports operate exactly like standard Fibre Channel E ports, except that the transport in this case is FCIP instead of Fibre Channel. The only requirement is for the other end of the VE port to be another VE port.

A virtual ISL is established over an FCIP link and transports Fibre Channel traffic. Each associated virtual ISL looks like a Fibre Channel ISL with either an E port or a TE port at each end (see the below figure).

*Figure 2: FCIP Links and Virtual ISLs*




See the Configuring E Ports, on page 29 for more information.

### FCIP Links

**Note** In-order delivery (IOD) is not supported on FCIP ISLs where WA and TA are enabled.

FCIP links consist of one or more TCP connections between two FCIP link endpoints. Each link carries encapsulated Fibre Channel frames.

When the FCIP link comes up, the VE ports at both ends of the FCIP link create a virtual Fibre Channel (E)ISL and initiate the E port protocol to bring up the (E)ISL.

By default, the FCIP feature on any Cisco MDS 9000 Family switch creates two TCP connections for each FCIP link:

- One connection is used for data frames.
- The other connection is used only for Fibre Channel control frames, that is, switch-to-switch protocol frames (all Class F). This arrangement provides low latency for all control frames.

To enable FCIP on the Fibre Channel module with IPS ports, an FCIP profile and FCIP interface (interface FCIP) must be configured.

The FCIP link is established between two peers, the VE port initialization operation is identical to a normal E port. This operation is independent of the link being FCIP or pure Fibre Channel, and is based on the E port discovery process (ELP, ESC).

Once the FCIP link is established, the VE port operation is identical to E port operation for all inter-switch communication (including domain management, zones, and VSANs). At the Fibre Channel layer, all VE and E port operations are identical.

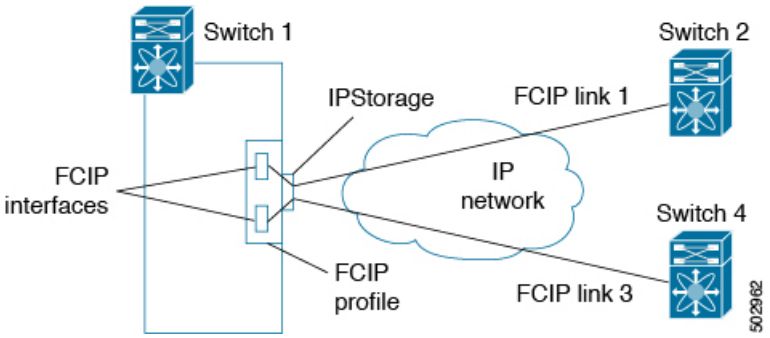
### FCIP Profiles

The FCIP profile contains information about the local IP address and TCP parameters. The profile defines the following information:

- The local connection points (IP address and TCP port number)
- The operation of the underlying TCP connections for all FCIP links that use this profile

The FCIP profile's local IP address determines the IPStorage port where the FCIP links terminate (see the below figure).

***Figure 3: FCIP Profile and FCIP Link***




### FCIP Interfaces

The FCIP interface is the local endpoint of the FCIP link and a VE port interface. All the FCIP and E port parameters are configured in context to the FCIP interface.

The FCIP parameters consist of the following:

- The FCIP profile determines which IPStorage port initiates the FCIP links and defines the TCP connection operation.
- Peer information.
- Number of TCP connections for the FCIP link.
- E port parameters—trunking mode and trunk-allowed VSAN list.

# FCIP High-Availability Solutions

The following high-availability solutions are available for FCIP configurations:

## Fibre Channel Port Channels

Port channels comprised of FCIP interfaces behave the same way as Fibre Channel port channels. They offer the same benefits of link redundancy between Fibre Channel switches as native FC port channels. Beneath

the FCIP level, an FCIP link can run on top of a IPStorage port. This link is totally transparent to the Fibre Channel layer.

The Fibre Channel Port Channel (to which FCIP link can be a part of) does not have a restriction on which (E)ISL links can be combined in a Fibre Channel Port Channel as long as it passes the compatibility check.

**Note** When port channel is configured with members on Cisco MDS 48-Port 64-Gbps Fibre Channel Switching Module (DS-X9748-3072K9), packet drops may be expected when IOD is configured and a member of the port channel is added, removed, or flaps.

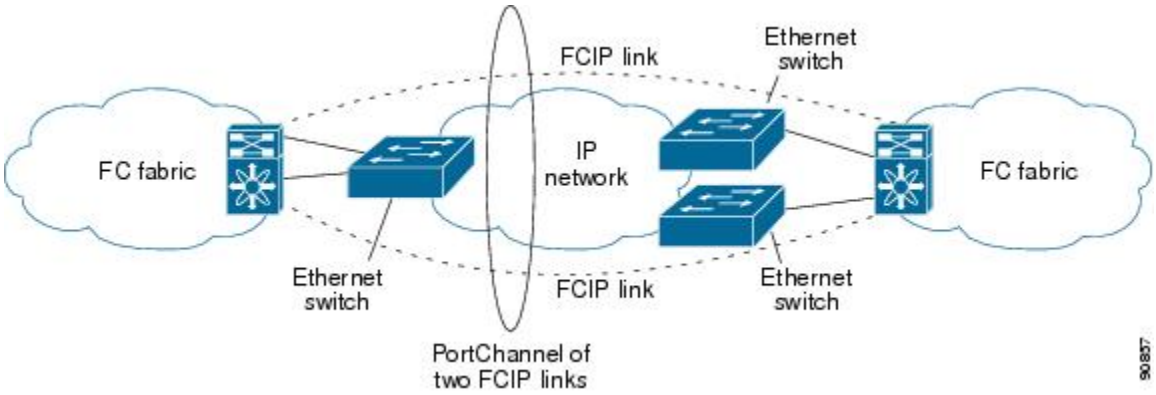
For information, see the Cisco Fabric Manager Interfaces Configuration Guide and Cisco MDS 9000 Series NX-OS Interfaces Configuration Guide.

The maximum number of Fibre Channel ports that can be put into a Fibre Channel Port Channel is 16.

To configure Fibre Channel Port Channels, see the Cisco MDS 9000 Series NX-OS Interfaces Configuration Guide and Cisco Fabric Manager Interfaces Configuration Guide.

The below figure provides an example of a port channel-based load-balancing configuration. To perform this configuration, you need two IP addresses on each SAN island. This solution addresses link failures.

***Figure 4: Port Channel-Based Load Balancing***



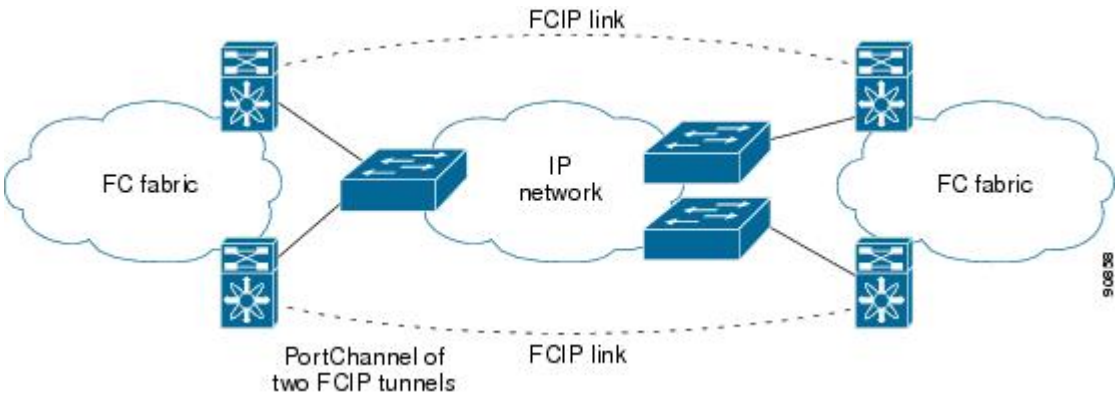

The following characteristics set Fibre Channel port channel solutions apart from other solutions:

- The entire bundle is one logical (E)ISL link.
- All FCIP links in the port channel should terminate on the same two switches.
- The Fibre Channel traffic is load balanced across the FCIP links in the port channel.

## FSPF

The below figure displays a FPSF-based load balancing configuration example. This configuration requires two IP addresses on each SAN island, and addresses IP and FCIP link failures.

*Figure 5: FSPF-Based Load Balancing*




The following characteristics set FSPF solutions apart from other solutions:

- Each FCIP link is a separate (E)ISL.
- The FCIP links can connect to different switches across two SAN islands.
- The Fibre Channel traffic is load balanced across the FCIP links.

# Configuring FCIP

This section describes how to configure FCIP and includes the following topics:

## Enabling FCIP

The Fibre Channel over IP Protocol (FCIP) is a tunneling protocol that connects geographically distributed Fibre Channel storage area networks (SAN islands) transparently over IP local area networks (LANs), metropolitan area networks (MANs), and wide area networks (WANs).

To begin configuring the FCIP feature, you must explicitly enable FCIP on the required switches in the fabric. By default, this feature is disabled in all switches in the Cisco MDS 9000 Family.

The configuration and verification operations commands for the FCIP feature are only available when FCIP is enabled on a switch. When you disable this feature, all related configurations are automatically discarded.

By default, the Cisco MDS 9700 series switches, MDS 9250i switches, and MDS 9220i switches are shipped with the SAN extension over IP package license.

To enable FCIP on any participating switch, follow these steps:

**Procedure**

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | switch# **configure terminal** | Enters configuration mode. |
| **Step 2** | switch(config)# **feature fcip** | Enables FCIP on that switch. |

| | Command or Action | Purpose |
|---|---|---|
| **Step 3** | (Optional) switch(config)# **no feature fcip** | Disables (default) FCIP on that switch. |

# Basic FCIP Configuration

Once you have created FCIP links using the FCIP wizard, you may need to modify parameters for these links. This includes modifying the FCIP profiles as well as the FCIP link parameters.

- For Cisco MDS 9220i Switch, each 1 Gbps or 10 Gbps IPStorage interface can have three FCIP links configured at a time and the 40 Gbps IPStorage interface can have four FCIP links configured at a time.
- For Cisco MDS 9250i Switch, each IPStorage interface can have six FCIP links configured at a time.
- For Cisco MDS 24/10-Port SAN Extension Module, each 1 Gbps or 10 Gbps IPStorage interface can have three FCIP links configured at a time and the 40 Gbps IPStorage interface can have four FCIP links configured at a time.



**Note** When using IPsec and IKE, each IPStorage interface on the Fibre Channel module with IPS ports must be configured in its own IP subnet. If there are multiple IPStorage interfaces configured with IP address or network-mask in the same IP subnet, IKE packets might not be sent out to the correct IPS port and the IPsec link will not come up.



**Note** You can configure FCIP with IPsec and NAT on Cisco MDS 9220i switches.

To configure an FCIP link, follow these steps on both switches:

**Procedure**

**Step 1** Configure the IPStorage interface.

See the *Cisco MDS 9000 Family NX-OS IP Services Configuration Guide*.

**Step 2** Create an FCIP profile and then assign the IPStorage interface's IP address to the profile.

**Step 3** Create an FCIP interface and then assign the profile to the interface.

**Step 4** Configure the peer IP address for the FCIP interface.

**Step 5** Enable the interface.

# Creating FCIP Profiles

You must assign a local IP address of an IPStorage interface or subinterface to the FCIP profile to create an FCIP profile. You can assign IPv4 or IPv6 addresses to the interfaces. The below figure shows an example configuration.

*Figure 6: Assigning Profiles to Each IPStorage Interface*




### Procedure

**Step 1** To create an FCIP profile in switch 1 in the above figure, follow these steps:

a) switch1# **configure terminal**

Enters configuration mode.

b) switch1(config)# **fcip profile 10**

Creates a profile for the FCIP connection. The valid range is from 1 to 255.

c) switch1(config-profile)# **ip address 10.100.1.25**

Associates the profile (10) with the local IPv4 address of the IPStorage interface (3/1).

d) switch1# **show fcip profile**

Displays FCIP profile information.

**Step 2** To assign an FCIP profile in switch 2 in the above figure, follow these steps:

a) switch2# **configure terminal**

Enters configuration mode.

b) switch2(config)# **fcip profile 20**

Creates a profile for the FCIP connection.

c) switch2(config-profile)# **ip address 10.1.1.1**

Associates the profile (20) with the local IPv4 address of the IPStorage interface.

d) switch2# **show fcip profile**

Displays FCIP profile information.

# Displaying FCIP Profile Information

### Displaying Summary of FCIP Profiles

```
switch# show fcip profile

-------------------------------------------------------------------------------
```

```
ProfileId Ipaddr   TcpPort
------------------------------------------------------------------------------
1         20.1.1.1 3225
2         20.1.1.1 2000
3         20.1.1.1 3000
4         20.1.1.1 4000
5         20.1.1.1 5000
6         20.1.1.1 6000
7         30.1.1.1 3225
8         31.1.1.1 3225
9         32.1.1.1 3225
10        33.1.1.1 3225
11        34.1.1.1 3225
12        35.1.1.1 3225
```

### Displaying Detailed FCIP Profile Information

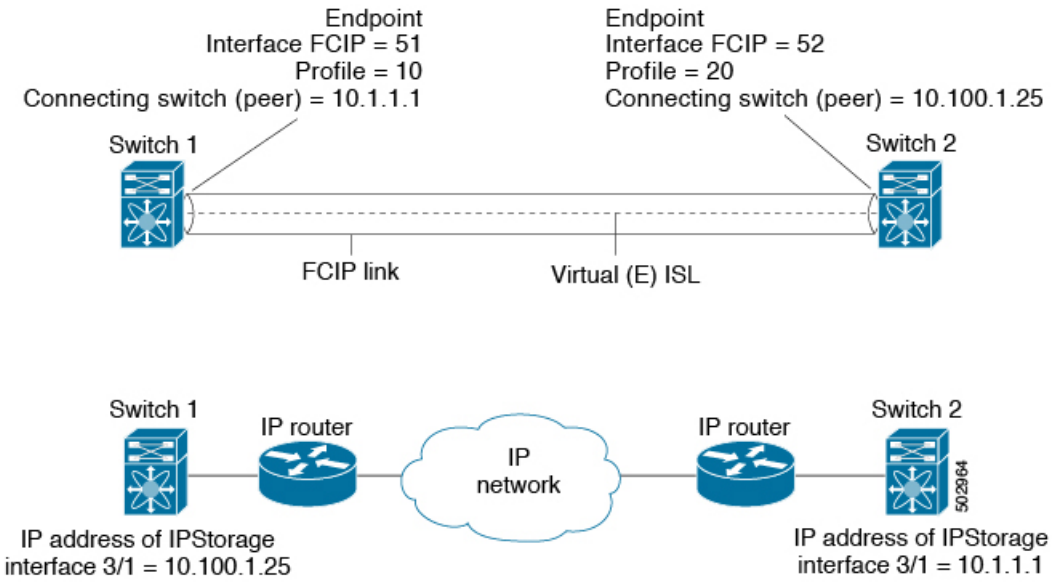
```
switch# show fcip profile 1
FCIP Profile 1
Internet Address is 20.1.1.1 (interface IPStorage1/1)
Tunnels Using this Profile: fcip1
Listen Port is 3225
TCP parameters
SACK is enabled
PMTU discovery is enabled, reset timeout is 3600 sec
Keep alive is 60 sec
Minimum retransmission timeout is 200 ms
Maximum number of re-transmissions is 4
Send buffer size is 16384 KB
Maximum allowed bandwidth is 5000000 kbps
Minimum available bandwidth is 4000000 kbps
Configured round trip time is 1000 usec
Congestion window monitoring is enabled, burst size is 50 KB
Auto jitter detection is enabled
```

# Creating FCIP Links

When two FCIP link endpoints are created, an FCIP link is established between the two IPS ports. The peer IP address specifies the address of the remote FCIP endpoint. This allows the creation of an FCIP link to that peer switch once the FCIP interface is enabled.

The below figure shows an example configuration of an FCIP link.

*Figure 7: Assigning Profiles to Each IPStorage Interface*




### Procedure

**Step 1** To create an FCIP link endpoint in switch 1, follow these steps:

a) switch1# **configure terminal**

Enters configuration mode.

b) switch1(config)# **interface fcip 51**

Creates an FCIP interface (51).

c) switch1(config-if)# **use-profile 10**

Assigns the profile (10) to the FCIP interface.

d) switch1(config-if)# **peer-info ipaddr 10.1.1.1**

Assigns the peer IPv4 address information (10.1.1.1 for switch 2) to the FCIP interface.

e) switch1(config-if)# **no shutdown**

Enables the interface.

**Step 2** To create an FCIP link endpoint in switch 2, follow these steps:

a) switch2# **configure terminal**

Enters configuration mode.

b) switch2(config)# **interface fcip 52**

Creates an FCIP interface (52).

c) switch2(config-if)# **use-profile 20**

Assigns the profile (20) to the FCIP interface.

d) switch2(config-if)# **peer-info ipaddr 10.100.1.25**

Assigns the peer IPv4 address information (10.100.1.25 for switch 1) to the FCIP interface.

e) switch2(config-if)# **no shutdown**

Enables the interface.

# Displaying FCIP MAC Statistics

The following example shows how to display Ethernet MAC statistics for an IPStorage interface used for FCIP from Cisco MDS 9000 NX-OS Release 9.4(3):

```
switch# show interface IPStorage 1/1 counters detailed
IPStorage1/1
Rate Stats:
Rx 5 minutes input rate bits/sec: 0
Rx 5 minutes input rate bytes/sec: 0
Rx 5 minutes input rate packets/sec: 0
Tx 5 minutes input rate bits/sec: 0
Tx 5 minutes input rate bytes/sec: 0
Tx 5 minutes input rate packets/sec: 0
Total Stats:

Rx packets input: 0
Rx bytes: 0
Rx multicast frames: 0
Rx broadcast frames: 0
Rx errors: 0
Rx queue drops: 0
Rx if-down drops: 0
Rx RED drops: 0
Rx bad ether type drops: 0
Rx bad protocol drops: 0
Tx packets output: 0
Tx bytes: 0
Tx underruns: 0
Tx multicast frames: 0
Tx broadcast frames: 0
Tx errors: 0
Tx collisions: 0
Tx ARP drops: 0
Tx if-down drops: 0
MAC Error Stats
Rx packets with FCS or align errors: 0
Rx packets with length < min: 0
Rx packets with length < min & FCS error: 0
Rx packets with length > max: 0
Rx packets with length > max & FCS error: 0
Rx packets with PCI error: 0
Rx packets with GMX/SPX/PCI error: 0
Rx packets dropped by RED buffer exhaustion: 0
Rx free buffers went below RED threshold count: 0
Rx L2 multicast packets: 0
Rx L2 broadcast packets: 0
Rx packets dropped by dest MAC filter: 0
Rx packets dropped due to a full receive FIFO: 0
```

```
Rx total MAC errors: 0
Tx packets dropped in PKO queue: 0
Tx packets sent with single collision: 0
Tx packets sent with multiple collisions: 0
Tx packets dropped due to excessive collisions: 0
Tx packets delayed count: 0
Tx total MAC errors: 0
```

# Advanced FCIP Profile Configuration

A basic FCIP configuration uses the local IP address to configure the FCIP profile. In addition to the local IP address and the local port, you can specify other TCP parameters as part of the FCIP profile configuration.

This sections includes the following topics:



**Note** FCIP configuration options can be accessed from the `switch (config-` profile`)#` submode prompt.

## Configuring TCP Listener Ports

**Procedure**

**Step 1** To configure TCP listener ports, follow these steps:

a) switch# **configure terminal**

Enters configuration mode.

b) switch(config)# **fcip profile 20**

Creates the profile (if it does not already exist) and enters profile configuration submode. The valid range is from 1 to 511.

The default TCP port for FCIP is 3225. You can change this port by using the **port** command.

**Step 2** To change the default FCIP port number (3225), follow these steps:

a) switch(config-profile)# **port 5000**

Associates the profile with the local port number (5000).

b) switch(config-profile)# **no port**

Reverts to the default 3225 port.

## Configuring TCP Parameters

You can control TCP behavior in a switch by configuring the TCP parameters that are described in this section.

**Note** When FCIP is sent over a WAN link, the default TCP settings may not be appropriate. In such cases, we recommend that you tune the FCIP WAN link by modifying the TCP parameters (specifically bandwidth, round-trip times, and CWM burst size).

This section includes the following topics:

## Minimum Retransmit Timeout

You can control the minimum amount of time TCP waits before retransmitting. By default, from Cisco MDS NX-OS Release 9.4(2) and later releases the value is 50 milliseconds and prior to Cisco MDS NX-OS Release 9.4(2) the value is 200 milliseconds.

To configure the minimum retransmit time, follow these steps:

**Procedure**

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | switch(config-profile)# **tcp min-retransmit-time 500** | Specifies the minimum TCP retransmit time for the TCP connection to be 500 milliseconds.<br>From Cisco MDS NX-OS Release 9.4(2) and later releases, the default is 50 milliseconds and the range is from 50 to 5000 milliseconds.<br>Prior to Cisco MDS NX-OS Release 9.4(2), the default value is 200 milliseconds and the range is from 200 to 5000 milliseconds. |
| **Step 2** | (Optional) switch(config-profile)# **no tcp min-retransmit-time 500** | Reverts the minimum TCP retransmit time to the factory default.<br>From Cisco MDS NX-OS Release 9.4(2) and later releases, the factory default is 50 milliseconds.<br>Prior to Cisco MDS NX-OS Release 9.4(2), the factory default value is 200 milliseconds. |

## Keepalive Timeout

You can configure the interval that the TCP connection uses to verify that the FCIP link is functioning. This ensures that an FCIP link failure is detected quickly even when there is no traffic.

If the TCP connection is idle for more than the specified time, then keepalive timeout packets are sent to ensure that the connection is active. The keepalive timeout feature can be used to tune the time taken to detect FCIP link failures.

You can configure the first interval during which the connection is idle (the default is 60 seconds). When the connection is idle for the configured interval, eight keepalive probes are sent at 1-second interval. If no response is received for these eight probes and the connection remains idle throughout, then the FCIP link is automatically closed.

**Note** Only the first interval (during which the connection is idle) can be changed.

To configure the first keepalive timeout interval, follow these steps:

**Procedure**

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | switch(config-profile)# **tcp keepalive-timeout 120** | Specifies the keepalive timeout interval for the TCP connection in seconds (120). The range is from 1 to 7200 seconds. |
| **Step 2** | (Optional) switch(config-profile)# **no tcp keepalive-timeout 120** | Reverts the keepalive timeout interval to the default 60 seconds. |

## Maximum Retransmissions

You can specify the maximum number of times a packet is retransmitted before TCP decides to close the connection.

To configure maximum retransmissions, follow these steps:

**Procedure**

**Step 1** switch(config-profile)# **tcp max-retransmissions 6**

Specifies the maximum number of retransmissions (6). The range is from 1 to 8 retransmissions.

**Step 2** (Optional) switch(config-profile)# **no tcp max-retransmissions 6**

Reverts to the default of 4 retransmissions.

## Maximum Retransmission Threshold for OBFL Logging

FCIP links utilize peer to peer TCP sessions. The intermediate network is often Ethernet which is not lossless and may drop frames. These are automatically retransmitted by the TCP sender. End applications should be able to tolerate some degree of delay caused by retransmissions. However, some applications may have a lower tolerance for retransmissions than others. Excessive TCP retransmission events are logged to OBFL to assist in troubleshooting. The threshold retransmission rate that will trigger logging can be configured. This retransmission rate is measured per FCIP profile.

To configure the OBFL retransmission rate per FCIP profile, follow these steps:

**Procedure**

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | switch# **configure terminal** | Enters the global configuration mode. |
| **Step 2** | switch(config)# **fcip profile** *profile_id* | Creates a profile (if it does not already exist) and enters profile configuration submode. The range is from 1 to 255. |
| **Step 3** | • `switch(config-profile)# tcp obfl max-retransmissions_pkt_thresh` *value*<br>• `switch(config-profile)# tcp logging onboard tcp-retransmission-threshold` *value* | Specifies the threshold for logging the TCP retransmission rate to OBFL, that is, the number of packets retransmitted per 10000 packets measured every minute. The range is from 5 to 10. The default value is 5.<br>Starting from Cisco MDS Release 8.4(1), the **tcp obfl max-retransmissions_pkt_thresh** *value* command is deprecated. Use the **tcp logging onboard tcp-retransmission-threshold** *value* command to configure the TCP retransmission rate to OBFL. The range is from 0.01 to 0.10. The default value is 0.05%. |
| **Step 4** | switch# **show logging onboard error-stats** | Displays the excessive FCIP retransmission logs. |

## Path MTUs

Path MTU (PMTU) is the minimum MTU on the IP network between the two endpoints of the FCIP link. PMTU discovery is a mechanism by which TCP learns of the PMTU dynamically and adjusts the maximum TCP segment accordingly (RFC 1191).

By default, PMTU discovery is enabled on all switches with a timeout of 3600 seconds. If TCP reduces the size of the maximum segment because of PMTU change, the reset-timeout specifies the time after which TCP tries the original MTU.

To configure PMTU, follow these steps:

**Procedure**

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | switch(config-profile)# **no tcp pmtu-enable** | Disables PMTU discovery. |
| **Step 2** | switch(config-profile)# **tcp pmtu-enable** | Enables (default) PMTU discovery with the default value of 3600 seconds. |
| **Step 3** | switch(config-profile)# **tcp pmtu-enable reset-timeout 90** | Specifies the PMTU reset timeout to 90 seconds. The range is 60 to 3600 seconds. |
| **Step 4** | switch(config-profile)# **no tcp pmtu-enable reset-timeout 600** | Leaves PMTU discovery enabled but reverts the timeout to the default of 3600 seconds. |

## Selective Acknowledgments

TCP may experience poor performance when multiple packets are lost within one window. With the limited information available from cumulative acknowledgments, a TCP sender can only learn about a single lost packet per round trip. A selective acknowledgment (SACK) mechanism helps overcome the limitations of multiple lost packets during a TCP transmission.

The receiving TCP sends back SACK advertisements to the sender. The sender can then retransmit only the missing data segments. By default, SACK is enabled on Cisco MDS 9000 Family switches.

To configure SACK, follow these steps:

**Procedure**

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | switch(config-profile)# **no tcp sack-enable** | Disables SACK. |
| **Step 2** | switch(config-profile)# **tcp sack-enable** | Enables SACK (default). |

## Window Management

The optimal TCP window size is automatically calculated using the maximum bandwidth parameter, the minimum available bandwidth parameter, and the dynamically measured round-trip time (RTT).

**Note** The configured round-trip-time parameter determines the window scaling factor of the TCP connection. This parameter is only an approximation. The measured RTT value overrides the round trip time parameter for window management. If the configured round-trip-time is too small compared to the measured RTT, then the link may not be fully utilized due to the window scaling factor being too small.

The **min-available-bandwidth** parameter and the measured RTT together determine the threshold below which TCP aggressively maintains a window size sufficient to transmit at minimum available bandwidth.

The **max-bandwidth-mbps** parameter and the measured RTT together determine the maximum window size.

**Note** Set the maximum bandwidth to match the worst-case bandwidth available on the physical link, considering other traffic that might be going across this link (for example, other FCIP links, WAN limitations). Maximum bandwidth should be the total bandwidth minus all other traffic going across that link.

**Note** In Cisco MDS 9250i Multiservice Fabric Switch, you can configure the TCP maximum bandwidth up to 5 Gbps. We recommend that the minimum available bandwidth is 80% of the maximum bandwidth.

To configure window management, follow these steps:

**Procedure**

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | switch(config-profile)# **tcp max-bandwidth-mbps 900 min-available-bandwidth-mbps 300 round-trip-time-ms 10** | Configures the maximum available bandwidth at 900 Mbps, the minimum slow start threshold at 300 Mbps, and the RTT at 10 milliseconds. |
| **Step 2** | (Optional) switch(config-profile)# **no tcp max-bandwidth-mbps 900 min-available-bandwidth-mbps 300 round-trip-time-ms 10** | Reverts to the factory defaults. The FCIP defaults are maximum bandwidth at 1 Gbps, minimum available bandwidth at 500 Mbps, and RTT at 1 milliseconds. |
| **Step 3** | switch(config-profile)# **tcp max-bandwidth-kbps 2000 min-available-bandwidth-kbps 2000 round-trip-time-us 200** | Configures the maximum available bandwidth at 2000 Kbps, the minimum available bandwidth at 2000 Kbps, and the RTT at 200 milliseconds. |

## Monitoring Congestion

By enabling the congestion window monitoring (CWM) parameter, you allow TCP to monitor congestion after each idle period. The CWM parameter also determines the maximum burst size allowed after an idle period. By default, this parameter is enabled and the default burst size is 50 KB.

The interaction of bandwidth parameters and CWM and the resulting TCP behavior is outlined as follows:

- If the average rate of the Fibre Channel traffic over the preceding RTT is less than the min-available-bandwidth multiplied by the RTT, the entire burst is sent immediately at the min-available-bandwidth rate, provided no TCP drops occur.
- If the average rate of the Fibre Channel traffic is greater than min-available-bandwidth multiplied by the RTT, but less than max-bandwidth multiplied by the RTT, then if the Fibre Channel traffic is transmitted in burst sizes smaller than the configured CWM value the entire burst is sent immediately by FCIP at the max-bandwidth rate.
- If the average rate of the Fibre Channel traffic is larger than the min-available-bandwidth multiplied by the RTT and the burst size is greater than the CWM value, then only a part of the burst is sent immediately. The remainder is sent with the next RTT.

The software uses standard TCP rules to increase the window beyond the one required to maintain the min-available-bandwidth to reach the max-bandwidth.



**Tip** We recommend that this feature remains enabled to realize optimal performance. Increasing the CWM burst size can result in more packet drops in the IP network, impacting TCP performance. Only if the IP network has sufficient buffering, try increasing the CWM burst size beyond the default to achieve lower transmit latency.

To change the CWM defaults, follow these steps:

**Procedure**

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | switch(config-profile)# **no tcp cwm** | Disables congestion monitoring. |
| **Step 2** | switch(config-profile)# **tcp cwm** | Enables congestion monitoring and sets the burst size to its default size. The default burst size is 50 KB. |
| **Step 3** | switch(config-profile)# **tcp cwm burstsize 30** | Changes the burst size to 30 KB. The valid range is from 10 to 100 KB. |
| **Step 4** | (Optional) switch(config-profile)# **no tcp cwm burstsize 25** | Leaves the CWM feature in an enabled state but changes the burst size to its factory default. |

## Displaying FCIP Profile Information

Use the **show fcip profile** command to display FCIP profile information for the Cisco MDS 9250i Multiservice Fabric Switch:

```
switch# show fcip profile 1
Internet address is 209.165.200.226 (interface IPStorage5/4.101)
Listen Port is 3225
TCP parameters
SACK is enabled
PMTU discovery is enabled, reset timeout is 3600 sec
Keep alive is 60 sec
Minimum retransmission timeout is 200 ms
Maximum number of re-transmissions is 4
Maximum number of obfl re-transmission Thresh is 5
Maximum allowed bandwidth is 5000000 kbps
Minimum available bandwidth is 4000000 kbps
Configured round trip time is 1000 usec
Congestion window monitoring is enabled, burst size is 50 KB
Auto jitter detection is enabled
```

Use the **show fcip profile** command to display FCIP profile information for the 24/10 port SAN Extension module:

```
switch# show fcip profile 41
FCIP Profile 41
Internet Address is 209.165.200.225 (interface IPStorage5/4.101)
Listen Port is 3225
TCP parameters
SACK is enabled
PMTU discovery is enabled, reset timeout is 3600 sec
Keep alive is 60 sec
Minimum retransmission timeout is 200 ms
Maximum number of re-transmissions is 4
Maximum number of obfl re-transmission Thresh is 6
Maximum allowed bandwidth is 10000000 kbps
Minimum available bandwidth is 8000000 kbps
Configured round trip time is 1000 usec
Congestion window monitoring is enabled, burst size is 50 KB
Auto jitter detection is enabled
```

# Advanced FCIP Interface Configuration

This section describes the options you can configure on an FCIP interface to establish connection to a peer and includes the following topics:

## Configuring Peers

All the FCIP and E port parameters are configured in context to the FCIP interface. To create an FCIP link, assign a profile to the FCIP interface and configure the peer information on the two switches at the ends of the FCIP link. The peer IP switch information causes the switch to initiate an FCIP link to that peer switch. The basic FCIP configuration uses the peer's IP address to configure the peer information. You can establish an FCIP link with the peer using the Peer IP address option. This option configures both ends of the FCIP link. Optionally, you can also use the peer TCP port along with the IP address.

To enter the `config-if` submode, follow these steps:

### Before you begin

To establish a peer connection, you must first create the FCIP interface and enter the `config-if` submode.

### Procedure

**Step 1** switch# **configure terminal**

Enters configuration mode.

**Step 2** switch(config)# **interface fcip 100**

Creates an FCIP interface (100).

**Note**
Each IPStorage interface can have three FCIP links configured at a time. For Cisco MDS 9250i, each IPStorage port can have six FCIP links configured at a time. For Cisco MDS 24/10-Port SAN Extension Module, each IPStorage port can have three FCIP links configured at a time.

## Assigning Peer IP Address (IPv4)

To assign the peer information based on the IPv4 address and port number, follow these steps:

### Procedure

**Step 1** switch(config-if)# **peer-info ipaddr 10.1.1.1**

Assigns an IPv4 address to configure the peer information. Because no port is specified, the default port number (3225) is used.

**Step 2** switch(config-if)# **peer-info ipaddr 10.1.1.1 port 3000**

Assigns the IPv4 address and sets the peer TCP port to 3000. The valid port number range is 0 to 65535.

**Step 3** switch(config-if)# **no shutdown**

Enables the interface.

# Assigning Peer IP Address (IPv6)

To assign the peer information based on the IPv6 address and port number, follow these steps:

**Procedure**

**Step 1** switch(config-if)# **peer-info ipaddr**

Assigns an IPv6 address to configure the peer information. Because no port is specified, the default port number (3225) is used.

**Step 2** switch(config-if)# **peer-info ipaddr 2001:0db8:800:200c::417a port 3000**

Assigns the IPv6 address and sets the peer TCP port to 3000. The valid port number range is 0 to 65535.

**Step 3** switch(config-if)# **no shutdown**

Enables the interface.

**Step 4** switch(config-if)# **interface IPStorage slot-number/plot-number**

Enters IPStorage interface configuration mode.

**Step 5** switch(config-if)# **shutdown**

Disable the interface.

**Step 6** switch(config-if)# **ipv6 enable**

Enables IPv6 processing on the interface.

**Step 7** switch(config-if)# **no shutdown**

Enables the interface.

# Configuring Number of TCP Connections

You can specify the number of TCP connections used for an FCIP link to be either two or five connections. By default, FCIP uses two connections for each link. Connection 0 is the FCIP control connection. The remaining one or four TCP connections are used for data.

**Note** Make sure that the peer switch FCIP link is also configured with the same number of TCP connections, otherwise FCIP link will not come up.

**Note** On the Cisco MDS platform, 10 Gb IPStorage ports have different performance characteristics than 1 Gb Ethernet ports. To achieve maximum throughput on FCIP links utilizing MDS 10 Gb IPStorage ports, set the number of TCP connections to 5 on these links.

To specify the TCP connection attempts, follow these steps:

**Procedure**

**Step 1** switch# **configure terminal**

Enters global configuration mode.

**Step 2** switch(config)# **interface fcip 4**

Enters FCIP interface configuration mode.

**Step 3** switch(config-if)# **shutdown**

Disables the interface.

**Step 4** switch(config-if)# **tcp-connection 5**

Specifies the number of TCP connections. Valid values are 2 or 5.

**Note**
To change the number of TCP connections ensure that the FCIP interface is shut down first.

**Step 5** (Optional) switch(config-if)# **no tcp-connection**

Reverts to the factory set default of two TCP sessions per FCIP interface.

**Step 6** switch(config-if)# **no shutdown**

Enables the interface.

# Configuring Active Connections

You can configure the mode for initiating a TCP connection. By default, the active mode is enabled to actively attempt an IP connection. If you enable the passive mode, the switch does not initiate a TCP connection but waits for the peer to connect to it. By default, the switch tries two TCP connections for each FCIP link.

**Note** Ensure that both ends of the FCIP link are not configured as passive mode. If both ends are configured as passive, the connection is not initiated.

To configure the passive mode, follow these steps:

**Procedure**

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | switch(config-if)# **passive-mode** | Enables passive mode while attempting a TCP connection. |
| **Step 2** | switch(config-if)# **no passive-mode** | Reverts to the factory set default of using the active mode while attempting the TCP connection. |
| **Step 3** | switch(config-if)# **no shutdown** | Enables the interface. |

# Enabling Time Stamp Control

You can configure the switch to discard packets that are outside a specified time range. When enabled, this feature specifies the time range within which packets can be accepted. If the packet arrived within the range specified by this option, the packet is accepted. Otherwise, it is dropped.

By default, time stamp control is disabled in all switches in the Cisco MDS 9000 Family. When enabled, if a packet arrives within a 2000 millisecond interval (+ or –2000 milliseconds) from the network time that packet is accepted.

**Note** If the **time-stamp** option is enabled, ensure to configure NTP on both switches (see the Cisco NX-OS Fundamentals Configuration Guide for more information).

**Tip** Do not enable time stamp control on an FCIP interface that has tape acceleration or Write Acceleration configured.

To enable or disable the time stamp control, follow these steps:

**Procedure**

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | switch(config-if)# **time-stamp**`Please enable NTP with a common time source on both MDS Switches that are on either side of the FCIP link` | Enables time stamp checking for received packets with a default acceptable time difference of 2000 milliseconds. |
| **Step 2** | (Optional) switch(config-if)# **no time-stamp** | Disables (default) time stamps. |

| | Command or Action | Purpose |
|---|---|---|
| **Step 3** | switch(config-if)# **time-stamp acceptable-diff 4000** | Configures the packet acceptance time. The valid range is from 500 to 10,000 millisecond. |
| **Step 4** | (Optional) switch(config-if)# **no time-stamp acceptable-diff 500** | Deletes the configured time difference and reverts the difference to factory defaults. The default difference is a 2000-millisecond interval from the network time. |
| **Step 5** | switch(config-if)# **no shutdown** | Enables the interface. |

# Quality of Service

The quality of service (QoS) parameter specifies the differentiated services code point (DSCP) value to mark all IP packets (type of service—TOS field in the IP header).

- The control DSCP value applies to all FCIP frames in the control TCP connection.
- The data DSCP value applies to all FCIP frames in the data connection.

If the FCIP link has only one TCP connection, that data DSCP value is applied to all packets in that connection.

To set the QoS values on FCIP interfaces, follow these steps:

### Procedure

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | switch(config-if)# **qos control 24 data 26** | Configures the control TCP connection and data connection to mark all packets on that DSCP value. The control and data value ranges from 0 to 63. |
| **Step 2** | switch(config-if)# **no qos control 24 data 26** | Reverts the switch to its factory default (marks all control and data packets with DCSP value 0). |

# Configuring E Ports

You configure FCIP interfaces in the same way as you configure FC (T)E interfaces. Specifically, the following features are available for FCIP interfaces:

- An FCIP interface can be a member of any VSAN.

  For more information, see the Cisco MDS 9000 Series Fabric Configuration Guide.

- Trunk mode and trunk allowed VSANs

  For more information, see the Cisco MDS 9000 Series Interfaces Configuration Guide.

- Port Channels
  - Multiple FCIP links can be bundled into a Fibre Channel Port Channel.
  - FCIP links and Fibre Channel links cannot be combined in one Port Channel.

For more information, see the Cisco MDS 9000 Series Security Configuration Guide.

- FSPF

  For more information, see the Cisco MDS 9000 Series Fabric Configuration Guide.

- Importing and exporting the zone database from the adjacent switch

  For more information, see the Cisco MDS 9000 Series System Management Configuration Guide.

# Displaying FCIP Interface Information

Use the **show interface** commands to view the summary, counter, description, and status of the FCIP link. Use the output of these commands to verify the administration mode, the interface status, the operational mode, the related VSAN ID, and the profile used.

### Displaying the FCIP Summary

```
switch# show fcip summary
--------------------------------------------------------------------------------
Tun prof IPS-if peer-ip Status T W T Enc Comp Bandwidth rtt
E A A max/min (us)
--------------------------------------------------------------------------------
1 1 IPS1/1 20.1.1.2 TRNK Y N N N A 5000M/4000M 1000
2 2 IPS1/1 20.1.1.2 TRNK Y N N N A 1000M/800M 1000
3 3 IPS1/1 20.1.1.2 DOWN N N N N N 1000M/800M 1000
4 4 IPS1/1 20.1.1.2 DOWN N N N N N 1000M/800M 1000
5 5 IPS1/1 20.1.1.2 DOWN N N N N N 1000M/800M 1000
6 6 IPS1/1 20.1.1.2 DOWN N N N N N 1000M/800M 1000
7 7 IPS1/2.1 30.1.1.2 TRNK Y N N N M2 1000M/800M 1000
8 8 IPS1/2.2 31.1.1.2 TRNK Y N N N M2 1000M/800M 1000
9 9 IPS1/2.3 32.1.1.2 DOWN N N N N N 1000M/800M 1000
10 10 IPS1/2.4 33.1.1.2 DOWN N N N N N 1000M/800M 1000
11 11 IPS1/2.5 34.1.1.2 DOWN N N N N N 1000M/800M 1000
12 12 IPS1/2.6 35.1.1.2 DOWN N N N N N 1000M/800M 1000
```

### Displaying the FCIP Interface Summary of Counters for a Specified Interface

```
switch# show interface fcip 1
fcip1 is trunking
Hardware is IPStorage
Port WWN is 20:2b:00:2a:6a:1b:60:70
Peer port WWN is 20:62:8c:60:4f:73:d7:80
Admin port mode is auto, trunk mode is on
snmp link state traps are enabled
Port mode is TE
Port vsan is 1
Operating Speed is 5 Gbps
Trunk vsans (admin allowed and active) (1)
Trunk vsans (up) (1)
Trunk vsans (isolated) ()
Trunk vsans (initializing) ()
Interface last changed at Wed Nov 27 15:12:25 2019

Using Profile id 1 (interface IPStorage1/1)
Peer Information
Peer Internet address is 10.1.1.6 and port is 3225
Write acceleration mode is configured off
```

```
Tape acceleration mode is configured off
Tape Accelerator flow control buffer size is automatic
FICON XRC Accelerator is configured off
Ficon Load Balancer configured off for all vsans
Ficon Tape acceleration configured off for all vsans
IP Compression is enabled and set for mode2
Maximum number of TCP connections is 5
QOS control code point is 0
QOS data code point is 0
TCP Connection Information
5 Active TCP connections
27 Attempts for active connections, 2 close of connections
Path MTU 1500 bytes
Current retransmission timeout is 6400 ms
Current Send Buffer Size: 87080 KB, Requested Send Buffer Size: 62500 KB
CWM Burst Size: 50 KB
CONN<0>
Data connection: Local 10.1.1.12:65489, Remote 10.1.1.6:3225
TCP Parameters
Advertized window: Current: 24580 KB, Maximum: 24580 KB, Scale: 6
Peer receive window: Current: 8191 KB, Maximum: 8191 KB, Scale: 7
Congestion window: Current: 7372 KB, Slow start threshold: 8094 KB
Measured RTT : 500000 us Min RTT: 500000 us Max RTT: 0 us
Round trip time: Smoothed 0 ms, Variance: 100 Jitter: 150 us
TCP Connection Rate
Input Bytes: 0.00 MB/sec, Output Bytes: 0.00 MB/sec
Input Frames: 0/sec, Output Frames: 0/sec
CONN<1>
Data connection: Local 10.1.1.12:65487, Remote 10.1.1.6:3225
TCP Parameters
Advertized window: Current: 487 KB, Maximum: 24580 KB, Scale: 6
Peer receive window: Current: 8191 KB, Maximum: 8191 KB, Scale: 7
Congestion window: Current: 870 KB, Slow start threshold: 8094 KB
Measured RTT : 500000 us Min RTT: 7573 us Max RTT: 0 us
Round trip time: Smoothed 8 ms, Variance: 4 Jitter: 150 us
TCP Connection Rate
Input Bytes: 0.00 MB/sec, Output Bytes: 0.00 MB/sec
Input Frames: 0/sec, Output Frames: 0/sec
CONN<2>
Data connection: Local 10.1.1.12:65485, Remote 10.1.1.6:3225
TCP Parameters
Advertized window: Current: 477 KB, Maximum: 24580 KB, Scale: 6
Peer receive window: Current: 8191 KB, Maximum: 8191 KB, Scale: 7
Congestion window: Current: 853 KB, Slow start threshold: 8094 KB
Measured RTT : 500000 us Min RTT: 7419 us Max RTT: 0 us
Round trip time: Smoothed 8 ms, Variance: 4 Jitter: 150 us
TCP Connection Rate
Input Bytes: 0.00 MB/sec, Output Bytes: 0.00 MB/sec
Input Frames: 0/sec, Output Frames: 0/sec
CONN<3>
Data connection: Local 10.1.1.12:65483, Remote 10.1.1.6:3225
TCP Parameters
Advertized window: Current: 488 KB, Maximum: 24580 KB, Scale: 6
Peer receive window: Current: 8191 KB, Maximum: 8191 KB, Scale: 7
Congestion window: Current: 872 KB, Slow start threshold: 8094 KB
Measured RTT : 500000 us Min RTT: 7590 us Max RTT: 0 us
Round trip time: Smoothed 8 ms, Variance: 4 Jitter: 150 us
TCP Connection Rate
Input Bytes: 0.00 MB/sec, Output Bytes: 0.00 MB/sec
Input Frames: 0/sec, Output Frames: 0/sec
CONN<4>
Control connection: Local 10.1.1.12:65481, Remote 10.1.1.6:3225
TCP Parameters
Advertized window: Current: 335 KB, Maximum: 24580 KB, Scale: 6
```

```
Peer receive window: Current: 8125 KB, Maximum: 8125 KB, Scale: 7
Congestion window: Current: 50 KB, Slow start threshold: 8069 KB
Measured RTT : 18 us Min RTT: 18 us Max RTT: 23 us
Round trip time: Smoothed 1 ms, Variance: 1 Jitter: 150 us
TCP Connection Rate
Input Bytes: 0.00 MB/sec, Output Bytes: 0.00 MB/sec
Input Frames: 0/sec, Output Frames: 0/sec
5 minutes input rate 208 bits/sec, 26 bytes/sec, 0 frames/sec
5 minutes output rate 272 bits/sec, 34 bytes/sec, 0 frames/sec
1658 frames input, 174756 bytes
1658 Class F frames input, 174756 bytes
0 Class 2/3 frames input, 0 bytes
0 Reass frames
0 Error frames timestamp error 0
1663 frames output, 202836 bytes
1663 Class F frames output, 202836 bytes
0 Class 2/3 frames output, 0 bytes
0 Error frames
```

### Displaying the FCIP Interface Summary of Counters for a Specified Interface

```
switch# show interface fcip 1
fcip1 is trunking
Hardware is IPStorage
Port WWN is 20:62:8c:60:4f:73:d7:80
Peer port WWN is 20:2b:00:2a:6a:1b:60:70
Admin port mode is auto, trunk mode is on
snmp link state traps are enabled
Port mode is TE
Port vsan is 1
Operating Speed is 10000 Mbps
Trunk vsans (admin allowed and active) (1)
Trunk vsans (up) (1)
Trunk vsans (isolated) ()
Trunk vsans (initializing) ()
Interface last changed at Tue Nov 26 15:16:55 2019

Using Profile id 1 (interface IPStorage2/3)
Peer Information
Peer Internet address is 10.1.1.12 and port is 3225
Write acceleration mode is configured off
Tape acceleration mode is configured off
Tape Accelerator flow control buffer size is automatic
FICON XRC Accelerator is configured off
Ficon Load Balancer configured off for all vsans
Ficon Tape acceleration configured off for all vsans
IP Compression is enabled and set for mode2
Maximum number of TCP connections is 5
QOS control code point is 0
QOS data code point is 0
TCP Connection Information
5 Active TCP connections
23 Attempts for active connections, 5 close of connections
Path MTU 1500 bytes
Current retransmission timeout is 200 ms
Current Send Buffer Size: 149580 KB, Requested Send Buffer Size: 125000 KB
CWM Burst Size: 50 KB
CONN<0>
Data connection: Local 10.1.1.6:3225, Remote 10.1.1.12:65489
TCP Parameters
Advertized window: Current: 24580 KB, Maximum: 24580 KB, Scale: 7
Peer receive window: Current: 4095 KB, Maximum: 4095 KB, Scale: 6
```

```
Congestion window: Current: 3686 KB, Slow start threshold: 3998 KB
Measured RTT : 500000 us Min RTT: 500000 us Max RTT: 0 us
Round trip time: Smoothed 24 ms, Variance: 12 Jitter: 150 us
TCP Connection Rate
Input Bytes: 0.00 MB/sec, Output Bytes: 0.00 MB/sec
Input Frames: 0/sec, Output Frames: 0/sec
CONN<1>
Data connection: Local 10.1.1.6:3225, Remote 10.1.1.12:65487
TCP Parameters
Advertized window: Current: 24580 KB, Maximum: 24580 KB, Scale: 7
Peer receive window: Current: 487 KB, Maximum: 487 KB, Scale: 6
Congestion window: Current: 438 KB, Slow start threshold: 462 KB
Measured RTT : 500000 us Min RTT: 500000 us Max RTT: 0 us
Round trip time: Smoothed 24 ms, Variance: 12 Jitter: 150 us
TCP Connection Rate
Input Bytes: 0.00 MB/sec, Output Bytes: 0.00 MB/sec
Input Frames: 0/sec, Output Frames: 0/sec
CONN<2>
Data connection: Local 10.1.1.6:3225, Remote 10.1.1.12:65485
TCP Parameters
Advertized window: Current: 24580 KB, Maximum: 24580 KB, Scale: 7
Peer receive window: Current: 477 KB, Maximum: 477 KB, Scale: 6
Congestion window: Current: 429 KB, Slow start threshold: 453 KB
Measured RTT : 500000 us Min RTT: 500000 us Max RTT: 0 us
Round trip time: Smoothed 24 ms, Variance: 12 Jitter: 150 us
TCP Connection Rate
Input Bytes: 0.00 MB/sec, Output Bytes: 0.00 MB/sec
Input Frames: 0/sec, Output Frames: 0/sec
CONN<3>
Data connection: Local 10.1.1.6:3225, Remote 10.1.1.12:65483
TCP Parameters
Advertized window: Current: 24580 KB, Maximum: 24580 KB, Scale: 7
Peer receive window: Current: 488 KB, Maximum: 488 KB, Scale: 6
Congestion window: Current: 439 KB, Slow start threshold: 463 KB
Measured RTT : 500000 us Min RTT: 500000 us Max RTT: 0 us
Round trip time: Smoothed 24 ms, Variance: 12 Jitter: 150 us
TCP Connection Rate
Input Bytes: 0.00 MB/sec, Output Bytes: 0.00 MB/sec
Input Frames: 0/sec, Output Frames: 0/sec
CONN<4>
Control connection: Local 10.1.1.6:3225, Remote 10.1.1.12:65481
TCP Parameters
Advertized window: Current: 8123 KB, Maximum: 24580 KB, Scale: 7
Peer receive window: Current: 334 KB, Maximum: 334 KB, Scale: 6
Congestion window: Current: 50 KB, Slow start threshold: 373 KB
Measured RTT : 19 us Min RTT: 20 us Max RTT: 25 us
Round trip time: Smoothed 1 ms, Variance: 1 Jitter: 150 us
TCP Connection Rate
Input Bytes: 0.00 MB/sec, Output Bytes: 0.00 MB/sec
Input Frames: 0/sec, Output Frames: 0/sec
5 minutes input rate 288 bits/sec, 36 bytes/sec, 0 frames/sec
5 minutes output rate 224 bits/sec, 28 bytes/sec, 0 frames/sec
1130 frames input, 137324 bytes
1130 Class F frames input, 137324 bytes
0 Class 2/3 frames input, 0 bytes
0 Reass frames
0 Error frames timestamp error 0
1132 frames output, 119060 bytes
1132 Class F frames output, 119060 bytes
0 Class 2/3 frames output, 0 bytes
0 Error frames
```

### Displaying Detailed FCIP Interface Standard Counter Information

```
switch# show interface fcip 1 counters
fcip1
TCP Connection Information
5 Active TCP connections
27 Attempts for active connections, 2 close of connections
Path MTU 1500 bytes
Current retransmission timeout is 6400 ms
Current Send Buffer Size: 87080 KB, Requested Send Buffer Size: 62500 KB
CWM Burst Size: 50 KB
CONN<0>
Data connection: Local 10.1.1.12:65489, Remote 10.1.1.6:3225
TCP Parameters
Advertized window: Current: 24580 KB, Maximum: 24580 KB, Scale: 6
Peer receive window: Current: 8191 KB, Maximum: 8191 KB, Scale: 7
Congestion window: Current: 7372 KB, Slow start threshold: 8094 KB
Measured RTT : 500000 us Min RTT: 500000 us Max RTT: 0 us
Round trip time: Smoothed 0 ms, Variance: 100 Jitter: 150 us
TCP Connection Rate
Input Bytes: 0.00 MB/sec, Output Bytes: 0.00 MB/sec
Input Frames: 0/sec, Output Frames: 0/sec
CONN<1>
Data connection: Local 10.1.1.12:65487, Remote 10.1.1.6:3225
TCP Parameters
Advertized window: Current: 487 KB, Maximum: 24580 KB, Scale: 6
Peer receive window: Current: 8191 KB, Maximum: 8191 KB, Scale: 7
Congestion window: Current: 870 KB, Slow start threshold: 8094 KB
Measured RTT : 500000 us Min RTT: 7573 us Max RTT: 0 us
Round trip time: Smoothed 8 ms, Variance: 4 Jitter: 150 us
TCP Connection Rate
Input Bytes: 0.00 MB/sec, Output Bytes: 0.00 MB/sec
Input Frames: 0/sec, Output Frames: 0/sec
CONN<2>
Data connection: Local 10.1.1.12:65485, Remote 10.1.1.6:3225
TCP Parameters
Advertized window: Current: 477 KB, Maximum: 24580 KB, Scale: 6
Peer receive window: Current: 8191 KB, Maximum: 8191 KB, Scale: 7
Congestion window: Current: 853 KB, Slow start threshold: 8094 KB
Measured RTT : 500000 us Min RTT: 7419 us Max RTT: 0 us
Round trip time: Smoothed 8 ms, Variance: 4 Jitter: 150 us
TCP Connection Rate
Input Bytes: 0.00 MB/sec, Output Bytes: 0.00 MB/sec
Input Frames: 0/sec, Output Frames: 0/sec
CONN<3>
Data connection: Local 10.1.1.12:65483, Remote 10.1.1.6:3225
TCP Parameters
Advertized window: Current: 488 KB, Maximum: 24580 KB, Scale: 6
Peer receive window: Current: 8191 KB, Maximum: 8191 KB, Scale: 7
Congestion window: Current: 872 KB, Slow start threshold: 8094 KB
Measured RTT : 500000 us Min RTT: 7590 us Max RTT: 0 us
Round trip time: Smoothed 8 ms, Variance: 4 Jitter: 150 us
TCP Connection Rate
Input Bytes: 0.00 MB/sec, Output Bytes: 0.00 MB/sec
Input Frames: 0/sec, Output Frames: 0/sec
CONN<4>
Control connection: Local 10.1.1.12:65481, Remote 10.1.1.6:3225
TCP Parameters
Advertized window: Current: 333 KB, Maximum: 24580 KB, Scale: 6
Peer receive window: Current: 8122 KB, Maximum: 8122 KB, Scale: 7
Congestion window: Current: 50 KB, Slow start threshold: 8069 KB
Measured RTT : 500000 us Min RTT: 18 us Max RTT: 0 us
Round trip time: Smoothed 1 ms, Variance: 1 Jitter: 150 us
```

```
TCP Connection Rate
Input Bytes: 0.00 MB/sec, Output Bytes: 0.00 MB/sec
Input Frames: 0/sec, Output Frames: 0/sec
5 minutes input rate 224 bits/sec, 28 bytes/sec, 0 frames/sec
5 minutes output rate 288 bits/sec, 36 bytes/sec, 0 frames/sec
1686 frames input, 177260 bytes
1686 Class F frames input, 177260 bytes
0 Class 2/3 frames input, 0 bytes
0 Reass frames
0 Error frames timestamp error 0
1691 frames output, 206072 bytes
1691 Class F frames output, 206072 bytes
0 Class 2/3 frames output, 0 bytes
0 Error frames
IP compression statistics
83360 rxbytes
55942 rxbytes compressed, 0 rxbytes non-compressed
1.49 rx compression ratio
99132 txbytes
64015 txbytes compressed, 1152 txbytes non-compressed
1.52 tx compression ratio
IP compression flow control statistics
0 bytes queued for hw compression
0 queued for hardware compression
0 queued for hardware decompression
0 slowed tcp flow control
0 accelerated tcp flow control
0 side band flow control ON
2346 side band flow control OFF
IP compression hung statistics
0 times compression engine hung detected
0 jobs replayed for hardware compression
0 jobs replayed for hardware decompression
0 compression jobs not processed during compression engine reset
0 compression response job not processed during compression engine reset
0 decompression jobs not processed during decompression engine reset
0 decompression response job not processed during decompression engine reset
```

### Displaying the FCIP Interface Description

```
switch# show interface fcip 51 description
FCIP51
Sample FCIP interface
```

The transmitted bytes shown in the total *txbytes* counter is the amount of data before compression. After compression, the compressed *txbytes* bytes are transmitted with compression and the uncompressed *txbytes* bytes are transmitted without compression. A packet may be transmitted without compression, if it becomes bigger after compression.

### Displaying Brief FCIP Interface Counter Information (Cisco MDS 9250i Multiservice Fabric Switch)

```
switch# show interface fcip 1-12 counters brief
--------------------------------------------------------------------------------
Interface Input (rate is 5 min avg) Output (rate is 5 min avg)
------------------------------ ------------------------------
Rate Total Rate Total
MB/s Frames MB/s Frames
--------------------------------------------------------------------------------
fcip1 191 1155974124 225 1363537690
fcip2 173 1046686124 227 1372311228
```

```
fcip3 0 0 0 0
fcip4 0 0 0 0
fcip5 0 0 0 0
fcip6 0 0 0 0
fcip7 189 1143612956 221 1339130294
fcip8 194 1167499884 218 1317700800
fcip9 0 0 0 0
fcip10 0 0 0 0
fcip11 0 0 0 0
fcip12 0 0 0 0
```

#### Displaying Brief FCIP Interface Counter Information (24/10 port SAN Extension Module)

```
switch# show interface fcip 41 counters brief
--------------------------------------------------------------------------------
Interface Input (rate is 5 min avg) Output (rate is 5 min avg)
------------------------------ ------------------------------
Rate Total Rate Total
MB/s Frames MB/s Frames
--------------------------------------------------------------------------------
fcip41 191 1155974124 225 1363537690
```

# Advanced FCIP Features

You can significantly improve application performance by configuring one or more of the following options for the FCIP interface:

## FCIP Write Acceleration

The FCIP Write Acceleration feature enables you to significantly improve the application write performance when storage traffic is routed over wide area networks using FCIP. When FCIP Write Acceleration is enabled, WAN throughput is maximized by acknowledging the write frames from the sender at the closer FCIP switch, thereby eliminating the impact of WAN latency for write operations. The Write Acceleration feature is disabled by default and must be enabled on both sides of the FCIP link.

**Note** FCIP links using Write Acceleration (WA) must be ensured that all accelerated flows go through a single FCIP link (or port channel). This applies to both commands and responses in both directions. If that does not occur, then FCIP WA will fail. Consequently, FCIP WA cannot be used across FSPF equal cost paths because commands and responses could take different paths.

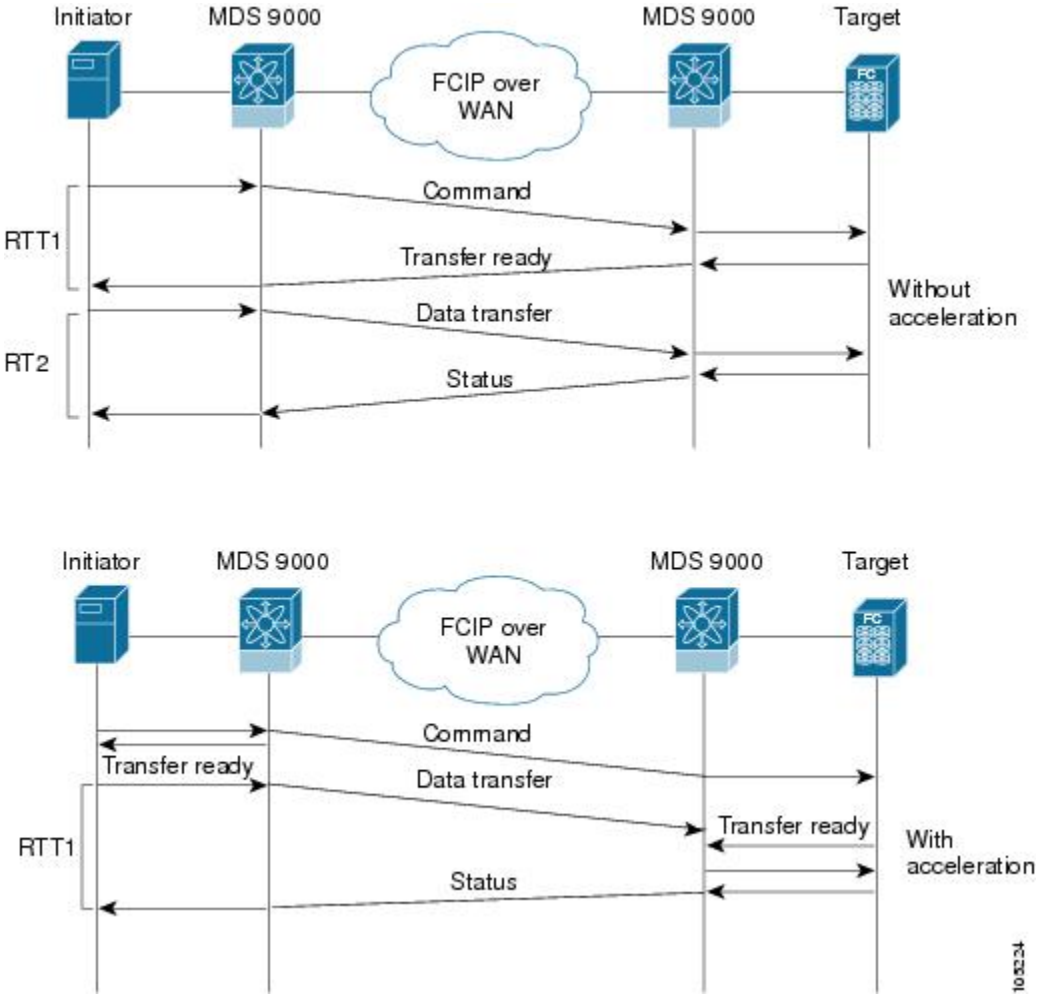
**Note** Ensure that all FCIP links in a port channel have the same attributes such as 2 or 5 connections, WA, TA, and so on. Otherwise, you will encounter undesirable results during an upgrade.

**Note** The FCIP Write Acceleration feature accelerates FC standard complaint SCSI WRITE commands only.

**Note** IBM Peer-to-Peer Remote Copy (PPRC) is not supported by FCIP Write Acceleration.

In the below figure, the SCSI WRITE command without Write Acceleration requires two round-trip transfers (RTT), while the SCSI WRITE command with Write Acceleration only requires one RTT. The maximum sized Transfer Ready is sent from the host side of the FCIP link back to the host before the SCSI WRITE command reaches the target. This enables the host to start sending the write data without waiting for the long latency over the FCIP link of the SCSI WRITE command and Transfer Ready. It also eliminates the delay caused by multiple Transfer Readys needed for the exchange going over the FCIP link.

***Figure 8: FCIP Link Write Acceleration***



**Tip** FCIP Write Acceleration (WA) can be enabled for multiple FCIP links if the links are part of a port channel configured with "channel mode active". These are port channels constructed with Port Channel Protocol (PCP). Write Acceleration cannot be used across FSPF equal cost paths in FCIP deployments. Native Fibre Channel Write Acceleration can be used with port channels. Also, FCIP Write Acceleration can be used in port channels configured with channel mode active or constructed with Port Channel Protocol (PCP). FCIP WA does not work if multiple non-port channel FCIP links exist with equal cost between the initiator and the target ports. This configuration might cause either SCSI discovery failure or failed WRITE or READ operations. When FCIP WA is used the FSPF routing should ensure that a single FCIP Port-Channel or ISL is always in the path between the initiator and the target ports.

Only one FCIP port channel is supported per VSAN on FCIPs configured on Cisco MDS 9700 Series switches with Write Acceleration configured.

**Tip** Do not enable time stamp control on an FCIP interface with Write Acceleration configured.

**Note** From Cisco MDS NX-OS Release 7.3(1)DY(1) and later, FCIP Write Acceleration can be enabled when FCIP port channels are configured between a Cisco MDS 9250i switch and a Cisco MDS 24/10 port SAN Extension Module in a Cisco MDS 9700 Director.

Ensure that the following prerequisites are met before enabling Write Acceleration:

- Use the **fcip-enhanced** command on the Cisco MDS 9220i and MDS 9250i switches while creating new port channels for FCIP ports. For more information on creating port channels, see the Configuring Port Channels chapter in the Cisco MDS 9000 Series Interfaces Configuration Guide. For more information on the **fcip-enhanced** command, see the Cisco MDS 9000 Series Command Reference.
- When you configure FCIP port channels between Cisco MDS 9250i and MDS 9700 switches and if write acceleration is enabled on the FCIP ports, then ensure that you enable write acceleration on the FCIP port-channels using the **fcip-enhanced** command on the Cisco MDS 9250i switch. For more information on creating port channels, see the Configuring Port Channels chapter in the Cisco MDS 9000 Series Interfaces Configuration Guide. For more information on the **fcip-enhanced** command, see the Cisco MDS 9000 Series Command Reference.
- Use the **show port-channel database** command to ensure that only FCIP interfaces are members of port channels.
- Enable **passive-mode** on FCIP interfaces created on a Cisco MDS 24/10 port SAN Extension Module in a Cisco MDS 9700 Director. For more information on enabling passive mode, see the Configuring Active Connections, on page 27 section.

**Tip** In Cisco MDS NX-OS Release 8.1(1) and later, Write Acceleration on FCIP port channels should not be configured between two MDS 9700 Series switches with Cisco MDS 24/10 port SAN Extension Module under the following conditions:

- If traffic flows between an FCoE port and an FC port through an FCIP port channel.
- If traffic flows through two successive FCIP port channels, one after another.

This section includes the following topics:

## Configuring FCIP Write Acceleration

To enable Write Acceleration, follow these steps:

**Procedure**

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | switch1# **configure terminal** | Enters configuration mode. |
| **Step 2** | switch1(config)# **interface fcip 51** | Creates an FCIP interface (51). |
| **Step 3** | switch1(config-if)# **write-accelerator** | Enables Write Acceleration. |

## Displaying Write Acceleration Activity Information

The following examples show how to display information about Write Acceleration activity.

### Displaying the Exchanges Processed by Write Acceleration at the Specified Host-End FCIP Link

```
switch# show fcip host-map 100

MAP TABLE (5 entries TOTAL entries 5)

OXID | RXID | HOST FCID| TARG FCID| VSAN | Index
------+------+----------+----------+------+---------
0xd490|0xffff|0x00690400|0x00620426|0x0005|0x0000321f
0xd4a8|0xffff|0x00690400|0x00620426|0x0005|0x00003220
0xd4c0|0xffff|0x00690400|0x00620426|0x0005|0x00003221
0xd4d8|0xffff|0x00690400|0x00620426|0x0005|0x00003222
0xd4f0|0xffff|0x00690400|0x00620426|0x0005|0x00003223
```

### Displaying Exchanges Processed by Write Acceleration at the Specified Target End FCIP Link

```
switch# show fcip target-map 100

MAP TABLE (3 entries TOTAL entries 3)

OXID | RXID | HOST FCID| TARG FCID| VSAN | Index
------+------+----------+----------+------+---------
0xc308|0xffff|0x00690400|0x00620426|0x0005|0x00003364
```

```
0xc320|0xffff|0x00690400|0x00620426|0x0005|0x00003365
0xc338|0xffff|0x00690400|0x00620426|0x0005|0x00003366
```

### Displaying Detailed FCIP Interface Write Acceleration Counter Information

```
switch# show interface fcip 4 counters
fcip4
TCP Connection Information
...
Write Accelerator statistics
6091 packets in 5994 packets out
0 frames dropped 0 CRC errors
0 rejected due to table full
0 ABTS sent 0 ABTS received
0 tunnel synchronization errors
37 writes recd 37 XFER_RDY sent (host)
0 XFER_RDY rcvd (target)
37 XFER_RDY rcvd (host)
0 XFER_RDY not proxied due to flow control (host)
0 bytes queued for sending
0 estimated bytes queued on the other side for sending
0 times TCP flow ctrl(target)
0 bytes current TCP flow ctrl(target)
```

# FCIP Tape Acceleration

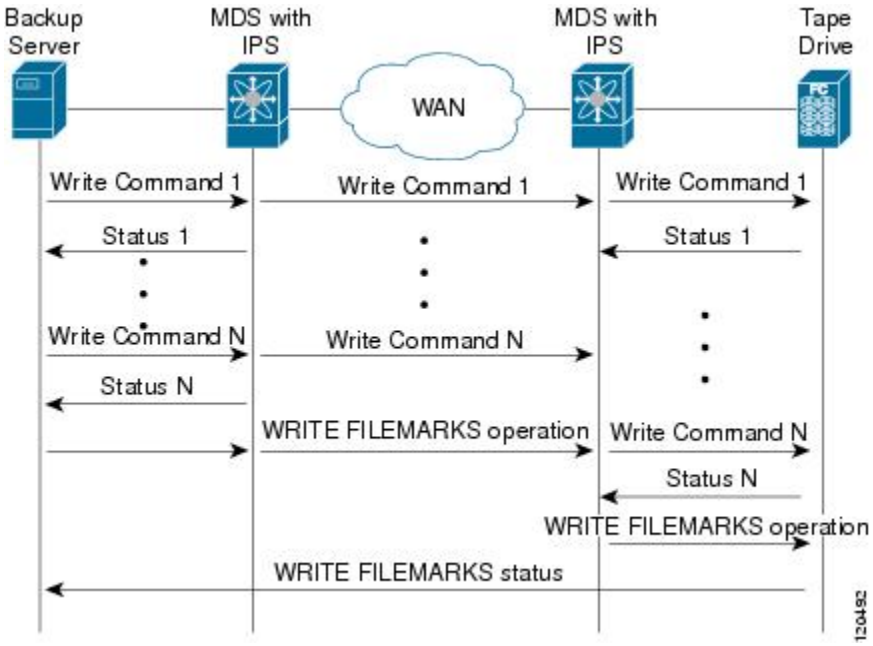
The FCIP Tape Acceleration feature enables you to significantly improve application write performance when storage traffic is routed over wide area networks using FCIP. When FCIP Tape Acceleration is enabled, WAN throughput is maximized by acknowledging the frames from the sender at the closer FCIP switch, thereby eliminating the impact of WAN latency for write operations. The FCIP TAPE Acceleration feature is disabled by default and must be enabled on both sides of the FCIP link.

Tapes are storage devices that store and retrieve user data sequentially. Cisco MDS NX-OS provides both tape write and read acceleration.

Applications that access tape drives normally have only one SCSI WRITE or READ operation outstanding to it. This single command process limits the benefit of the tape acceleration feature when using an FCIP link over a long-distance WAN link. It impacts backup and performance because each SCSI WRITE or READ operation does not complete until the host receives a good status response from the tape drive. The FCIP tape acceleration feature helps solve this problem. It improves tape backup, archive, and restore operations by allowing faster data streaming between the host and tape drive over the WAN link.

In an example of tape acceleration for write operations, the backup server in the below figure issues write operations to a drive in the tape library. Acting as a proxy for the remote tape drives, the local Cisco MDS switch proxies a transfer ready to signal the host to start sending data. After receiving all the data, the local Cisco MDS switch proxies the successful completion of the SCSI WRITE operation. This response allows the host to start the next SCSI WRITE operation. This proxy method results in more data being sent over the FCIP link in the same time period compared to the time taken to send data without proxying. The proxy method improves the performance on WAN links.

*Figure 9: FCIP Link Tape Acceleration for Write Operations*



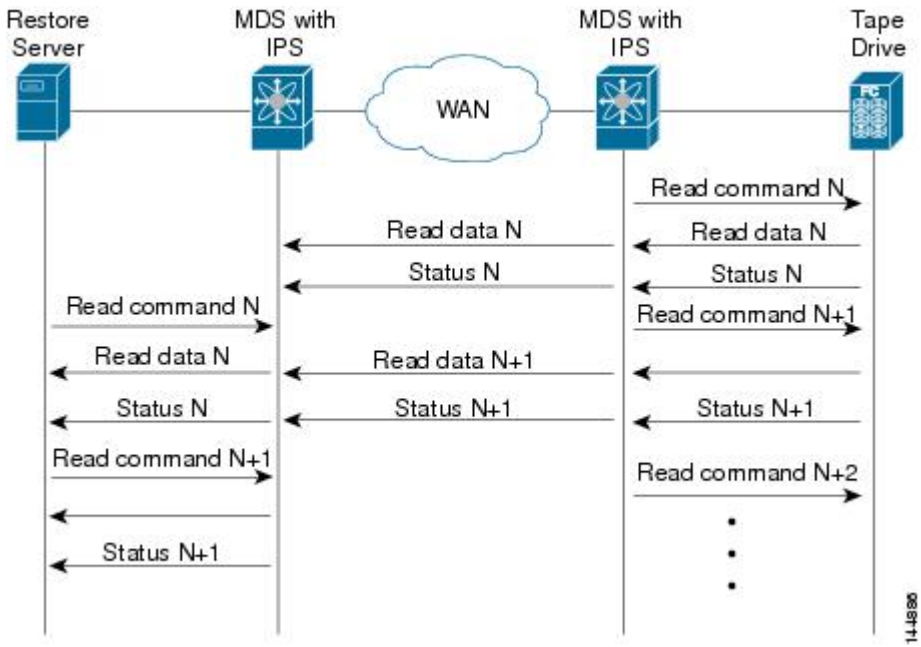

At the tape end of the FCIP link, another Cisco MDS switch buffers the command and data it has received. It then acts as a backup server to the tape drive by listening to a transfer ready from the tape drive before forwarding the data.

**Note** In some cases such as a quick link up/down event (FCIP link, Server/Tape Port link) in a tape library environment that exports Control LUN or a Medium Changer as LUN 0 and tape drives as other LUNs, tape acceleration may not detect the tape sessions and may not accelerate these sessions. You need to keep the FCIP link disabled for a couple of minutes before enabling the link. This does not apply to tape environments where the tape drives are either direct FC attached or exported as LUN 0.

The Cisco NX-OS provides reliable data delivery to the remote tape drives using TCP/IP over the WAN. It maintains write data integrity by allowing the WRITE FILEMARKS operation to complete end-to-end without proxying. The WRITE FILEMARKS operation signals the synchronization of the buffer data with the tape library data. While tape media errors are returned to backup servers for error handling, tape busy errors are retried automatically by the Cisco NX-OS software.

In an example of tape acceleration for read operations, the restore server in the below figure issues read operations to a drive in the tape library. During the restore process, the remote Cisco MDS switch at the tape end, in anticipation of more SCSI read operations from the host, sends out SCSI read operations on its own to the tape drive. The prefetched read data is cached at the local Cisco MDS switch. The local Cisco MDS switch on receiving SCSI read operations from the host, sends out the cached data. This method results in more data being sent over the FCIP link in the same time period compared to the time taken to send data without read acceleration for tapes. This improves the performance for tape reads on WAN links.

*Figure 10: FCIP Link Tape Acceleration for Read Operations*




The Cisco NX-OS provides reliable data delivery to the restore application using TCP/IP over the WAN. While tape media errors during the read operation are returned to the restore server for error handling, the Cisco NX-OS software recovers from any other errors.

**Note** The FCIP tape acceleration feature is disabled by default and must be enabled on both sides of the FCIP link. If it is only enabled on one side of the FCIP link, the tape acceleration feature is operationally off.

**Tip** FCIP tape acceleration does not work if the FCIP port is part of a port channel or if there are multiple paths between the initiator and the target port. Such a configuration might cause either SCSI discovery failure or broken write or read operations.

When tape acceleration is enabled in an FCIP interface, a FICON VSAN cannot be enabled in that interface. Likewise, if an FCIP interface is up in a FICON VSAN, tape acceleration cannot be enabled on that interface.

**Note** When you enable the tape acceleration feature for an FCIP link, the link is reinitialized and the write and read acceleration feature is also automatically enabled.

In tape acceleration for writes, after a certain amount of data has been buffered at the remote Cisco MDS switch, the write operations from the host are flow controlled by the local Cisco MDS switch by not proxying the Transfer Ready. On completion of a write operation when some data buffers are freed, the local Cisco MDS switch resumes the proxying. Likewise, in tape acceleration for reads, after a certain amount of data has been buffered at the local Cisco MDS switch, the read operations to the tape drive are flow controlled by

the remote Cisco MDS switch by not issuing any further reads. On completion of a read operation, when some data buffers are freed, the remote Cisco MDS switch resumes issuing reads.

The default flow control buffering uses the **automatic** option. This option takes the WAN latencies and the speed of the tape into account to provide optimum performance. You can also specify a flow control buffer size (the maximum buffer size is 12 MB).

We recommend that you use the default option for flow-control buffering.

**Note** FCIP tape acceleration will work with five TCP connections.

**Tip** Do not enable time-stamp control on an FCIP interface with tape acceleration configured.

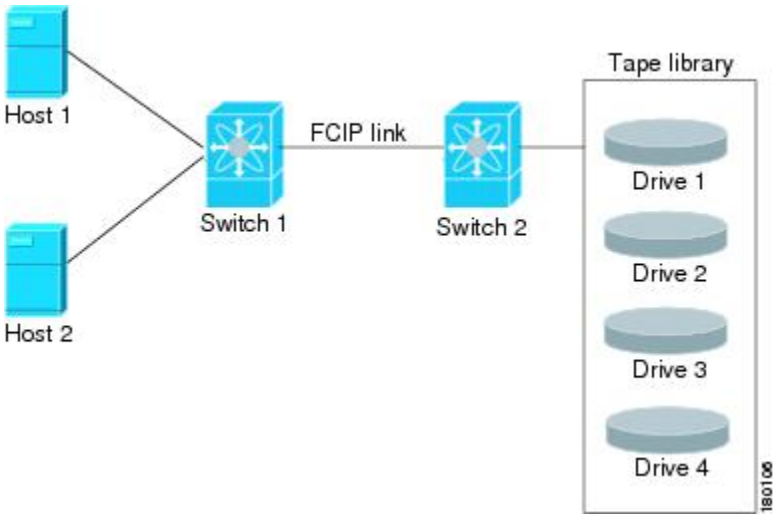
This section includes the following topics:

## Tape Library LUN Mapping for FCIP Tape Acceleration

If a tape library provides logical unit (LU) mapping and FCIP tape acceleration is enabled, you must assign a unique LU number (LUN) to each physical tape drive accessible through a target port.

The following figure shows tape drives connected to Switch 2 through a single target port. If the tape library provides LUN mapping, then all the four tape drives should be assign unique LUNs.

***Figure 11: FCIP LUN Mapping***




For the mappings described in Table 2: Correct LUN Mapping Example with Single Host Access, on page 44 and Table 3: Incorrect LUN Mapping Example with Single Hosts Access, on page 44, Host 1 has access to Drive 1 and Drive 2, and Host 2 has access to Drive 3 and Drive 4.

The following table describes correct tape library LUN mapping.

*Table 2: Correct LUN Mapping Example with Single Host Access*

| Host | LUN Mapping | Drive |
|---|---|---|
| Host 1 | LUN 1 | Drive 1 |
| | LUN 2 | Drive 2 |
| Host 2 | LUN 3 | Drive 3 |
| | LUN 4 | Drive 4 |

The following table describes incorrect tape library LUN mapping.

*Table 3: Incorrect LUN Mapping Example with Single Hosts Access*

| Host | LUN Mapping | Drive |
|---|---|---|
| Host 1 | LUN 1 | Drive 1 |
| | LUN 2 | Drive 2 |
| Host 2 | LUN 1 | Drive 3 |
| | LUN 2 | Drive 4 |

Another example setup is when a tape drive is shared by multiple hosts through a single tape port. For instance, Host 1 has access to Drive1 and Drive2, and Host 2 has access to Drive 2, Drive 3, and Drive 4. A correct LUN mapping configuration for such a setup is shown in the following table.

*Table 4: Correct LUN Mapping Example with Multiple Host Access*

| Host | LUN Mapping | Drive |
|---|---|---|
| Host 1 | LUN 1 | Drive 1 |
| | LUN 2 | Drive 2 |

| Host | LUN Mapping | Drive |
|---|---|---|
| Host 2 | LUN 2 | Drive 2 |
| | LUN 3 | Drive 3 |
| | LUN 4 | Drive 4 |

## Configuring FCIP Tape Acceleration



**Note** In an FCIP tape acceleration link, if the trunk mode is **on** for TA enabled links, then the trunk mode allowed VSAN should be configured such that each VSAN's traffic passes through only one link. If the traffic passes through multiple links, it may cause traffic failures.

To enable FCIP tape acceleration, follow these steps:

**Procedure**

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | switch1# **configure terminal** | Enters configuration mode. |
| **Step 2** | switch1(config)# **interface fcip 5** | Creates an FCIP interface (5). |
| **Step 3** | switch1(config-if)# **write-accelerator tape-accelerator** | Enables tape acceleration (and Write Acceleration—if not already enabled). |
| **Step 4** | switch1(config-if)# **write-accelerator tape-accelerator flow-control-buffer-size auto** | Enables tape acceleration with automatic flow control (default). |
| **Step 5** | switch1(config-if)# **write-accelerator tape-accelerator flow-control-buffer-size 2048** | Sets tape acceleration flow control buffer size to 2 MB. |
| **Step 6** | (Optional) switch1(config-if)# **no write-accelerator tape-accelerator** | Disables tape acceleration (default) and resets the FCIP link.<br>**Note**<br>The Write Acceleration feature remains enabled. |
| **Step 7** | (Optional) switch1(config-if)# **no write-accelerator tape-accelerator flow-control-buffer-size 2048** | Changes the flow control buffer size to the default value of automatic. The tape acceleration and Write Acceleration features remain enabled. This command does not reset the FCIP link. |
| **Step 8** | (Optional) switch1(config-if)# **no write-accelerator** | Disables both the Write Acceleration and tape acceleration features and resets the FCIP link. |

# Displaying Tape Acceleration Activity Information

The following examples show how to display information about tape acceleration activity.

### Displaying Information About Tapes for Which Exchanges are Tape Accelerated

```
switch# show fcip tape-session summary

--------------------------------------------------------------------------------
Tunnel Tunnel End host-fcid tape-fcid lun vsan TCP Connection
--------------------------------------------------------------------------------

16 host-end 0x7c0006 0x390006 0x0000 3000 0
16 host-end 0x7c0004 0x390004 0x0000 3000 0
16 host-end 0x7c0003 0x390003 0x0000 3000 0
16 host-end 0x7c0007 0x390007 0x0000 3000 0
16 host-end 0x7c0005 0x390005 0x0000 3000 0
16 host-end 0x7c0000 0x390000 0x0000 3000 0
16 host-end 0x7c0002 0x390002 0x0000 3000 0
```

### Displaying Information About Tapes for Which Exchanges are Tape Accelerated at the Host-End FCIP Link

```
switch# show fcip tape-session tunnel 1 host-end

HOST TAPE SESSIONS (1 entries TOTAL entries 1)

Host Tape Session #1
FCID 0xEF0001, VSAN 1, LUN 0x0002
Outstanding Exchanges 0, Outstanding Writes 0
Target End Write Buffering 0 Bytes, Auto Max Writes 3
Flags 0x0, FSM state Non TA Mode
Cached Reads 0
First index 0xfffffff7, Last index 0xfffffff7, RA index 0x0000f99a
Current index=0xfffffffe, Els Oxid 0xfff7
Hosts 1
FCID 0x770100
```

### Displaying Information About Tapes for Which Exchanges are Tape Accelerated at the Target-End FCIP Link

```
switch# show fcip tape-session tunnel 1 targ-end

TARGET TAPE SESSIONS (1 entries TOTAL entries 1)

Target Tape Session #1
FCID 0xEF0001, VSAN 1, LUN 0x0002
Outstanding Exchanges 0, Outstanding Writes 0
Host End Read Buffering 0 Bytes, Auto Max Read Blocks 3
Flags 0x800, Timer Flags 0x0
FSM State Default, Prev FSM State Bypass
Relative Block offset 0
First index 0xfffffff7, Last index 0xfffffff7, RA index 0x0000f99a
Current index=0xfffffffe, Els Oxid 0xfff7
Hosts 1
FCID 0x770100
```

### Displays Detailed FCIP Interface Tape Acceleration Counter Information, if Enabled

```
switch# show interface fcip 1 counters
fcip1
TCP Connection Information
....
Tape Accelerator statistics
1 Host Tape Sessions
0 Target Tape Sessions
Host End statistics
Received 31521 writes, 31521 good status, 0 bad status
Sent 31517 proxy status, 4 not proxied
Estimated Write buffer 0 writes 0 bytes
Received 31526 reads, 10 status
Sent 31516 cached reads
Read buffer 0 reads, 0 bytes
Host End error recovery statistics
Sent REC 0, received 0 ACCs, 0 Rejects
Sent ABTS 0, received 0 ACCs
Received 31 RECs, sent 2 ACCs, 0 Rejects
Received 0 SRRs, sent 0 ACCs, 0 Rejects
Received 0 TMF commands
Target End statistics
Received 0 writes, 0 good status, 0 bad status
Write Buffer 0 writes, 0 bytes
Received 0 reads, 0 good status, 0 bad status
Sent 0 reads, received 0 good status, 0 bad status
Sent 0 rewinds, received 0 good status, 0 bad status
Estimated Read buffer 0 reads, 0 bytes
Target End error recovery statistics
Sent REC 0, received 0 ACCs, 0 Rejects
Sent SRR 0, received 0 ACCs
Sent ABTS 0, received 0 ACCs
Received 0 TMF commands
```

# FCIP Compression

The FCIP compression option allows IP packets to be compressed on an FCIP link. This option is enabled per FCIP link. The FCIP maximum bandwidth configuration is calculated on the size of the FC frames. Since compression is done at the IP layer and FC is above the IP layer, enabling compression will not cause FCIP interfaces to transmit more than without it (that is, not increase FC throughput). Rather, it helps in reducing the amount of IP traffic sent over the IP network (that is, reduces the IP load). By default, FCIP compression is disabled. When enabled without specifying a compression mode, a compression algorithm appropriate for the FCIP service engine and link speed (as configured in the TCP parameters of the FCIP profile) is automatically used.



**Note** The main purpose of the FCIP compression feature is to increase throughput on low bandwidth IP connections which are hitting maximum bandwidth usage. Depending on the IP Services switch or module used, the performance of the built-in IP compression engine can be significantly lower than the maximum line rate of IPStorage ports. Therefore, configuring FCIP compression on high bandwidth links can be detrimental to the overall application level performance.

The Cisco MDS 9250i switch and Cisco 24/10 port SAN Extension Module supports Auto, Mode1 and Mode2 compression modes. All of these modes internally use the hardware compression engine in the module. Auto

mode is enabled by default. Mode2 uses a larger batch size for compression than Auto-mode, which results in higher compression throughput. However, Mode2 incurs a small latency due to the compression throughput. For those deployments where aggressive throughput is most important, Mode2 can be used. Mode1 gives the best compression ratio when compared to all other modes. For those deployments where compression ratio is most important, Mode1 can be used.

**Note** If both ends of an FCIP link are running Cisco MDS NX-OS Release 8.1(1) or later, and you enable compression at one end of the FCIP link, be sure to enable it at the other end of the link also.

**Note** When using FCIP compression, the rates specified in **tcp max-bandwidth-xxxx** and **min-available-bandwidth-xxxx** in the FCIP profile are in compressed bites.

This section includes the following topics:

## Configuring FCIP Compression

To enable FCIP compression, follow these steps:

**Procedure**

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | switch# **configure terminal** | Enters configuration mode. |
| **Step 2** | switch(config)# **interface fcip 51** | Creates an FCIP interface (51). |
| **Step 3** | switch(config-if)# **ip-compression mode2** | Enables high compression. |
| **Step 4** | (Optional) switch(config-if)# **ip-compression auto** | Defaults to using the **auto** mode. |
| **Step 5** | (Optional) switch(config-if)# **no ip-compression** | Disables (default) the FCIP compression feature. |

## Displaying FCIP Compression Information

The following examples show how to display FCIP compression information.

### Displaying Detailed FCIP Interface Compression Information, if Enabled

```
switch# show interface fcip 4 counters
fcip4
TCP Connection Information
...
IP compression statistics
208752 rxbytes, 208752 rxbytes compressed
5143584 txbytes
0 txbytes compressed, 5143584 txbytes non-compressed
1.00 tx compression ratio
```

### Displaying the Compression Engine Statistics for the 9250i

```
switch# (config-vsan-db)# show ips stats hw-comp interface IPStorage 1/1
HW Compression statistics for port IPStorage1/1
Compression stats
10444189094728 input bytes, 2822607905236 output compressed bytes
85952406 input pkts, 85952065 output compressed pkts
Decompression stats
8596899248 input compressed bytes, 27669956608 output bytes
45879853 input compressed pkts, 45879669 output pkts
Passthru stats
0 input bytes, 0 output bytes
0 input pkts, 0 output pkts
Miscellaneous stats
0 min input pktlen, 638570 max input pktlen
0 min output pktlen, 185641 max output pktlen
0 len mismatch, 0 incomplete processing
0 invalid result, 0 invalid session drop
0 comp expanded
Errors stats
0 decomp tx error, 0 post comp error
0 post decomp error, 0 comp packets expanded
```

### Displaying the Compression Engine Statistics for 24/10 port SAN Extension Module

```
switch# show ips stats hw-comp interface iPStorage 5/1

HW Compression statistics for port IPStorage5/1
Compression stats
53280732 input bytes, 44561835 output compressed bytes
544700 input pkts, 544700 output compressed pkts
Decompression stats
41760802 input compressed bytes, 49574684 output bytes
511886 input compressed pkts, 511886 output pkts
Passthru stats
0 input bytes, 0 output bytes
0 input pkts, 0 output pkts
Miscellaneous stats
0 min input pktlen, 3816 max input pktlen
0 min output pktlen, 1485 max output pktlen
0 len mismatch, 0 incomplete processing
0 invalid result, 0 invalid session drop
0 comp expanded
Errors stats
0 decomp tx error, 0 post comp error
0 post decomp error, 0 comp packets expanded
```

# Configuring FCIP Links for Maximum Performance

This section describes how to configure FCIP links for optimum performance between two Cisco MDS 9250i switches, or 24/10 port SAN Extension Modules. We recommend that the maximum and minimum bandwidth parameters in an FCIP profile be the same on both the sides.

**Note**

- FCIP links with a **tcp max-bandwidth-mbps** of 33 Mbps or less will result in FSPF cost of 30000. This makes the interface unusable. Starting from Cisco MDS NX-OS Releases 8.2(1), the FSPF cost for low bandwidth FCIP links is set to 28999. Because this value is less than the FSPF maximum cost of 30000, and it will allow the traffic to be routed across the interface. It also allows additional FC or FCoE hops (including the FCIP hop) in the end-to-end path. The total FSPF cost of these additional hops should not exceed 1000, because the path will not be usable. If the FSPF cost of 28999 is not applicable for a specific topology, it should be manually configured using the **fspf cost** interface configuration command. To check the FSPF cost of an interface, use the **show fspf interface** command.

  For more information on FSPF Cost, see the Cisco MDS Fabric Configuration Guide.

- FSPF equal cost multipath (ECMP) cannot be used to load balance the traffic across different types of ISLs. In other words, two or more equal cost ISLs of a different type between two switches (e.g. FCIP + FC, or FCIP + FCoE) is not supported.

- On Cisco MDS 24/10 Port SAN Extension Module, configuring multiple ECMP port channels with FCIP members in the same VSAN is not a valid configuration. If this is configured, then traffic will flow through only one of the port channels.

- Maximum FCIP performance is achieved with no packet loss between the FCIP TCP endpoints. FCIP will perform best when the TCP retransmission rate is below 0.05%. To determine if the TCP retransmission rate is above 0.05% use the **show logging onboard error-stats** command to display 60 second intervals of time, where the TCP retransmission rate is above 0.05% on a per FCIP interface and per TCP connection basis.

This section includes the following topics:

# Configuring FCIP Links for Maximum Performance on a Cisco MDS 9250i Switch

This section includes the following topics:

## Configuring FCIP performance in 1 Gbps mode

To achieve maximum FCIP performance in 1 Gbps mode, the following configuration is recommended:

1. Create an FCIP link on the IPStorage port.

**Note** If more than one FCIP link is bound to an IPStorage interface at 1 Gbps, the combined maximum bandwidth of all links bound to that interface must not exceed 1 Gbps.

2. Set the TCP maximum and minimum bandwidth as 1000 Mbps and 800 Mbps respectively.

**Note** If the TCP maximum bandwidth is set to any value more than 1000 Mbps, we recommend that you set the number of TCP connections to five.

3. Configure two TCP connections on each FCIP link.

4. Set the MTU size to 2500 for the IPStorage port.
5. Enable compression on each FCIP link.

To achieve maximum FCIP performance in 1 Gbps mode, follow these configuration steps:

### Procedure

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | switch# **configure terminal** | Enters global configuration mode. |
| **Step 2** | switch(config)# **fcip profile***profile-id* | Configures an FCIP profile and enters FCIP profile configuration mode. |
| **Step 3** | switch(config-profile)# **ip address**ip-address | Assigns an IP address to the FCIP profile. The assigned IP address can be an IPv4 or an IPv6 address. |
| **Step 4** | switch(config-profile)# **tcp max-bandwidth-mbps** *1000* **min-available-bandwidth-mbps** *800* **round-trip-time-ms** *milliseconds* | Sets the maximum and minimum available bandwidth of the FCIP link to 1000 Mbps and 800 Mbps respectively, and configures the round-trip time in milliseconds. |
| **Step 5** | switch(config-profile)# **exit** | Exits FCIP profile configuration mode and returns to global configuration mode. |
| **Step 6** | switch(config)# **interface fcip** *interface-number* | Enters FCIP interface configuration mode. |
| **Step 7** | switch(config-if)# **use-profile** *profile-id* | Binds the specified profile to the FCIP link. |
| **Step 8** | switch(config-if)# **peer-info ipaddr** *ip-address* | Configures the peer IP address (IPv4 or IPv6). |
| **Step 9** | switch(config-if)# **tcp-connections** *2* | Sets the number of TCP connections to 2. This value must be the same at the peer end. |
| **Step 10** | switch(config-if)# **ip-compression mode2** | Sets the compression algorithm to **mode2** for the interface. The other modes that can be set are **auto** and **mode1**. |
| **Step 11** | switch(config-if)# **no shutdown** | Enables the FCIP interface. |
| **Step 12** | switch (config-if)# **exit** | Exits FCIP interface configuration mode and returns to global configuration mode. |
| **Step 13** | switch(config)# **interface IPStorage** *slot-number/port-number* | Enters IPStorage interface configuration mode. |
| **Step 14** | switch(config-if)# ip address *ip-address subnet-mask* | Assigns an IP address to the interface. |
| **Step 15** | switch(config-if)# **switchport mtu 2500** | Sets the MTU size to 2500 for the interface. The valid range for MTU is from 576 to 9216. |
| **Step 16** | switch(config-if)# **no shutdown** | Enables the interface. |
| **Step 17** | switch(config-if)# **end** | Exits interface configuration mode and returns to privileged EXEC mode. |

# Configuring FCIP performance in 10 Gbps mode

To achieve maximum FCIP performance in 10 Gbps mode, the following configuration is recommended:

1. Create an FCIP link on the IPStorage port.

   If more than two FCIP links are bound to an IPStorage interface at 10 Gbps, the combined maximum bandwidth of all links bound to that interface must not exceed 10 Gbps.

2. Set the TCP maximum and minimum bandwidth to 5000 Mbps and 4000 Mbps respectively (default value).

3. Configure five TCP connections on each FCIP link.

4. Set the MTU size to 2500 on the IPStorage port.

To achieve maximum FCIP performance in 10 Gbps mode, follow these configuration steps:

**Procedure**

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | switch# **configure terminal** | Enters global configuration mode. |
| **Step 2** | switch(config)# **fcip profile** *profile-id* | Configures an FCIP profile and enters FCIP profile configuration mode. |
| **Step 3** | switch(config-profile)# **ip address** *ip-address* | Assigns an IP address to the FCIP profile. The assigned IP address can be an IPv4 or an IPv6 address. |
| **Step 4** | switch(config-profile)# **tcp max-bandwidth-mbps***5000***min-available-bandwidth-mbps***4000 round-trip-time-ms***milliseconds* | Sets the maximum and minimum available bandwidth of the FCIP link to 5000 Mbps and 4000 Mbps respectively, and configures the round-trip time in milliseconds. |
| **Step 5** | switch(config-profile)# **exit** | Exits FCIP profile configuration mode and returns to global configuration mode. |
| **Step 6** | switch(config)# **interface fcip interface-number** | Enters FCIP interface configuration mode. |
| **Step 7** | switch(config-if)# **use-profile***profile-id* | Binds the specified profile to the FCIP link. |
| **Step 8** | switch(config-if)# **peer-info ipaddr***ip-address* | Configures the peer IP address (IPv4 or IPv6). |
| **Step 9** | switch(config-if)# **tcp-connections***5* | Sets the number of TCP connections to 5. This value must be the same at the peer end. |
| **Step 10** | switch(config-if)# **no shutdown** | Enables the FCIP interface. |
| **Step 11** | switch(config-if)# **exit** | Exits FCIP interface configuration mode and returns to global configuration mode. |
| **Step 12** | switch(config)# **interface IPStorage***slot-number/port-number* | Enters IPStorage interface configuration mode. |
| **Step 13** | switch(config-if)# **ip address***ip-address subnet-mask* | Assigns an IP address to the interface. |

| | Command or Action | Purpose |
|---|---|---|
| **Step 14** | switch(config-if)# **switchport mtu 2500** | Sets the MTU size to 2500 for the interface. The valid range for MTU is from 576 to 9216. |
| **Step 15** | switch(config-if)# **no shutdown** | Enables the interface. |
| **Step 16** | switch(config-if)# **end** | Exits interface configuration mode and returns to privileged EXEC mode. |

# Configuring FCIP Links for Maximum Performance on Cisco MDS 24/10 port SAN Extension Module and M9220i switch

This section includes the following topics:

## Configuring FCIP performance in 1 Gbps mode

To achieve maximum FCIP performance in 1 Gbps mode, the following configuration is recommended:

1. Create an FCIP link on the IPStorage port.



**Note** If more than one FCIP link is bound to an IPStorage interface at 1 Gbps, the combined maximum bandwidth of all links bound to that interface must not exceed 1 Gbps.

2. Set the TCP maximum and minimum bandwidth as 1000 Mbps and 800 Mbps respectively.



**Note** If the TCP maximum bandwidth is set to a value that is more than 1000 Mbps, we recommend that you set the number of TCP connections to five.

3. Configure two TCP connections on each FCIP link.
4. Set the MTU size to 2500 for the IPStorage port.

To achieve maximum FCIP performance in 1 Gbps mode, follow these configuration steps:

**Procedure**

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | switch# **configure terminal** | Enters global configuration mode. |
| **Step 2** | switch(config)# **fcip profile** *profile-id* | Configures an FCIP profile and enters FCIP profile configuration mode. |
| **Step 3** | switch(config-profile)# **ip address** *ip-address* | Assigns an IP address to the FCIP profile. The assigned IP address can be an IPv4 or an IPv6 address. |

| | Command or Action | Purpose |
|---|---|---|
| **Step 4** | switch(config-profile)# **tcp max-bandwidth-mbps** *1000* **min-available-bandwidth-mbps** *800* **round-trip-time-ms** *milliseconds* | Sets the maximum and minimum available bandwidth of the FCIP link to 1000 Mbps and 800 Mbps respectively, and configures the round-trip time in milliseconds. |
| **Step 5** | switch(config-profile)# **exit** | Exits FCIP profile configuration mode and returns to global configuration mode. |
| **Step 6** | switch(config)# **interface fcip** *interface-number* | Enters FCIP interface configuration mode. |
| **Step 7** | switch(config-if)# **use-profile** *profile-id* | Binds the specified profile to the FCIP link. |
| **Step 8** | switch(config-if)# **peer-info ipaddr** *ip-address* | Configures the peer IP address (IPv4 or IPv6). |
| **Step 9** | switch(config-if)# **tcp-connections** *2* | Sets the number of TCP connections to 2. This value must be the same at the peer end. |
| **Step 10** | switch(config-if)# **ip-compression** *mode2* | Sets the compression algorithm to **mode2** for the interface. The other modes that can be set are **auto** and **mode1** . |
| **Step 11** | switch (config-if)# **no shutdown** | Enables the FCIP interface. |
| **Step 12** | switch (config-if)# **exit** | Exits FCIP interface configuration mode and returns to global configuration mode. |
| **Step 13** | switch(config)# **interface IPStorage** *slot-number/port-number* | Enters IPStorage interface configuration mode. |
| **Step 14** | switch(config-if)# **ip address** *ip-address subnet-mask* | Assigns an IP address to the interface. |
| **Step 15** | switch(config-if)# **switchport mtu** *2500* | Sets the MTU size to 2500 for the interface. The valid range for MTU is from 576 to 9216. |
| **Step 16** | switch(config-if)# **no shutdown** | Enables the interface. |
| **Step 17** | switch(config-if)# **end** | Exits interface configuration mode and returns to privileged EXEC mode. |

## Configuring FCIP performance in 10 Gbps mode

To achieve maximum FCIP performance in 10 Gbps mode, the following configuration is recommended:

In the releases prior to Cisco MDS NX-OS Release 9.3(1), the maximum throughput on a single FCIP tunnel is 7 Gbps. For maximum performance, create two fcip tunnels on 10 Gbps IPS ports.

1. Create an FCIP link on the IPStorage port.

   If more than two FCIP links are bound to an IPStorage interface at 10 Gbps, the combined maximum bandwidth of all links bound to that interface must not exceed 10 Gbps.

2. Set the TCP maximum and minimum bandwidth to 10000 Mbps and 8000 Mbps respectively (default value).

3. Configure five TCP connections on each FCIP link.

4. Set the MTU size to 2500 on the IPStorage port.

To achieve maximum FCIP performance in 10 Gbps mode, follow these configuration steps:

### Procedure

**Step 1** switch# **configure terminal**

Enters global configuration mode.

**Step 2** switch(config)# **fcip profile** *profile-id*

Configures an FCIP profile and enters FCIP profile configuration mode.

**Step 3** switch(config-profile)# **ip address** *ip-address*

Assigns an IP address to the FCIP profile. The assigned IP address can be an IPv4 or an IPv6 address.

**Step 4** switch(config-profile)# **tcp max-bandwidth-mbps** *10000* **min-available-bandwidth-mbps** *8000* **round-trip-time-ms** *milliseconds*

Sets the maximum and minimum available bandwidth of the FCIP link to 10000 Mbps and 8000 Mbps respectively, and configures the round-trip time in milliseconds.

**Step 5** switch(config-profile)# **exit**

Exits FCIP profile configuration mode and returns to global configuration mode.

**Step 6** switch(config)# **interface fcip** *interface-number*

Enters FCIP interface configuration mode.

**Step 7** switch(config-if)# **use-profile** *profile-id*

Binds the specified profile to the FCIP link.

**Step 8** switch(config-if)# **peer-info ipaddr** *ip-address*

Configures the peer IP address (IPv4 or IPv6).

**Step 9** switch(config-if)# **tcp-connections***5*

Sets the number of TCP connections to 5. This value must be the same at the peer end.

**Step 10** switch(config-if)# **no shutdown**

Enables the FCIP interface.

**Step 11** switch(config-if)# **exit**

Exits FCIP interface configuration mode and returns to global configuration mode.

**Step 12** switch(config)# **interface IPStorage** *slot-number/port-number*

Enters IPStorage interface configuration mode.

**Step 13** switch(config-if)# **ip address** *ip-address subnet-mask*

Assigns an IP address to the interface.

**Step 14** switch(config-if)# **switchport mtu 2500**

Sets the MTU size to 2500 for the interface. The valid range for MTU is from 576 to 9216.

**Step 15** switch(config-if)# **no shutdown**

Enables the interface.

**Step 16** switch(config-if)# **end**

Exits interface configuration mode and returns to privileged EXEC mode.

## Configuring FCIP performance in 25 Gbps mode

To achieve maximum FCIP performance in 25 Gbps mode, the following configuration is recommended:

The maximum bandwidth of FCIP tunnel is 10 Gbps. Therefore, configure two FCIP tunnels on each 25 Gbps IPS port.



**Note** For maximum performance, create two VLAN sub interfaces on each 25 Gbps IPStorage port. On each VLAN subinterface, create FCIP tunnel.

1. Create VLAN sub interfaces on the 25 Gbps IPStorage port.
2. Create an FCIP link on each VLAN sub interface.
3. Set the TCP maximum and minimum bandwidth to 10000 Mbps (10 Gbps) and 8000 Mbps (8Gbps) respectively (default value).
4. Configure five TCP connections on each FCIP link.
5. Set the MTU size to 2500 on the IPStorage port.

To achieve maximum FCIP performance in 25 Gbps mode, follow these configuration steps:

**Procedure**

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | switch# **configure terminal** | Enters global configuration mode. |
| **Step 2** | switch(config)# **interface IPStorage** *slot-number/port-number VLAN-ID* | Configure VLAN subinterface (VLAN ID) on 25Gbps IPStorage port. |
| **Step 3** | switch(config-if)# **ip address** *ip-address subnet-mask* | Assigns an IP address to the VLAN subinterface. |
| **Step 4** | switch(config-if)# **switchport mtu** *2500* | Sets the MTU size to 2500 for the VLAN subinterface. The valid range for MTU is from 576 to 9216. |
| **Step 5** | switch(config-if)# **no shutdown** | Enables the VLAN subinterface. |
| **Step 6** | switch(config-if)# **end** | Exits interface configuration mode and returns to privileged EXEC mode. |

| | Command or Action | Purpose |
|---|---|---|
| **Step 7** | switch(config)# **fcip profile** *profile-id* | Configures an FCIP profile and enters FCIP profile configuration mode. |
| **Step 8** | switch(config-profile)# **ip address** *ip-address* | Assign IP address of VLAN subinterface configured in Step 3 to the FCIP profile. |
| **Step 9** | switch(config-profile)# **tcp max-bandwidth-mbps** *10000* **min-available-bandwidth-mbps** *8000* **round-trip-time-ms** *milliseconds* | Sets the maximum and minimum available bandwidth of the FCIP link to 10000 Mbps and 8000 Mbps respectively, and configures the round-trip time in milliseconds. |
| **Step 10** | switch(config-profile)# **exit** | Exits FCIP profile configuration mode and returns to global configuration mode. |
| **Step 11** | switch(config)# **interface fcip** *interface-number* | Enters FCIP interface configuration mode. |
| **Step 12** | switch(config-if)# **use-profile** *profile-id* | Binds the specified profile to the FCIP link. |
| **Step 13** | switch(config-if)# **peer-info ipaddr** *ip-address* | Configures the peer IP address (IPv4 or IPv6). |
| **Step 14** | switch(config-if)# **tcp-connections** *5* | Sets the number of TCP connections to 5. This value must be the same at the peer end. |
| **Step 15** | switch(config-if)# **no shutdown** | Enables the FCIP interface. |
| **Step 16** | switch(config-if)# **exit** | Exits FCIP interface configuration mode and returns to global configuration mode. |

## Configuring FCIP performance in 40 Gbps mode

To achieve maximum FCIP performance in 40 Gbps mode, the following configuration is recommended:

- The maximum bandwidth of FCIP tunnel is 10 Gbps. Therefore, configure four FCIP tunnels on the 40 Gbps IPS port.
- For maximum performance, create four VLAN sub interfaces on the 40 Gbps IPStorage port. On each VLAN subinterface, create FCIP tunnel.
  - Releases prior to Cisco MDS NX-OS Rel 9.3(1), the maximum throughput on a single FCIP tunnel on Cisco MDS 9220i is 7 Gbps. On the 40 Gbps IPS port, the maximum throughput is 28.5 Gbps when 4 FCIP tunnels are created on separate VLAN subinterfaces.
  - From the Release Cisco MDS NX-OS Rel 9.3(1), the maximum throughput on a single FCIP tunnel on Cisco MDS 9220i is 10Gbps. On the 40 Gbps IPS port, the maximum throughput is 34.6 Gbps when 4 FCIP tunnels are created on separate VLAN subinterfaces.

To achieve maximum FCIP performance in 40 Gbps mode, follow these configuration steps to configure four FCIP tunnels:

1. Create VLAN sub interfaces on the 40 Gbps IPStorage port.
2. Create an FCIP link on each VLAN sub interface.

3. Set the TCP maximum and minimum bandwidth to 10000 Mbps (10 Gbps) and 8000 Mbps (8Gbps) respectively (default value).
4. Configure five TCP connections on each FCIP link.
5. Set the MTU size to 2500 on the IPStorage port.

To achieve maximum FCIP performance in 40 Gbps mode, follow these configuration steps:

### Procedure

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | switch# **configure terminal** | Enters global configuration mode. |
| **Step 2** | switch(config)# **interface IPStorage** *slot-number/port-number VLAN-ID* | Configure VLAN subinterface (VLAN ID) on 40 Gbps IPStorage port. |
| **Step 3** | switch(config-if)# **ip address** *ip-address subnet-mask* | Assigns an IP address to the VLAN subinterface. |
| **Step 4** | switch(config-if)# **switchport mtu 2500** | Sets the MTU size to 2500 for the VLAN subinterface. The valid range for MTU is from 576 to 9216. |
| **Step 5** | switch(config-if)# **no shutdown** | Enables the VLAN subinterface. |
| **Step 6** | switch(config-if)# **end** | Exits interface configuration mode and returns to privileged EXEC mode. |
| **Step 7** | switch(config)# **fcip profile** *profile-id* | Configures an FCIP profile and enters FCIP profile configuration mode. |
| **Step 8** | switch(config-profile)# **ip address** *ip-address* | Assign IP address of VLAN subinterface configured in Step 3 to the FCIP profile. |
| **Step 9** | switch(config-profile)# **tcp max-bandwidth-mbps** *10000* **min-available-bandwidth-mbps** *8000* **round-trip-time-ms** *milliseconds* | Sets the maximum and minimum available bandwidth of the FCIP link to 10000 Mbps and 8000 Mbps respectively, and configures the round-trip time in milliseconds. |
| **Step 10** | switch(config-profile)# **exit** | Exits FCIP profile configuration mode and returns to global configuration mode. |
| **Step 11** | switch(config)# **interface fcip***interface-number* | Enters FCIP interface configuration mode. |
| **Step 12** | switch(config-if)# **use-profile** *profile-id* | Binds the specified profile to the FCIP link. |
| **Step 13** | switch(config-if)# **peer-info ipaddr***ip-address* | Configures the peer IP address (IPv4 or IPv6). |
| **Step 14** | switch(config-if)# **tcp-connections***5* | Sets the number of TCP connections to 5. This value must be the same at the peer end. |
| **Step 15** | switch(config-if)# **no shutdown** | Enables the FCIP interface. |
| **Step 16** | switch(config-if)# **exit** | Exits FCIP interface configuration mode and returns to global configuration mode. |

# Verifying FCIP Configuration

To verify FCIP configurations, use the following commands:

| Command | Purpose |
|---|---|
| **show fcip host-map** *fcip-id* | Displays the information for a specified map. |
| **show fcip profile** *profile-id* | Displays the information for a profile. |
| **show fcip summary** | Displays summary information. |
| **show interface fcip** | Displays the status of an interface. |
| **show interface fcip** *fcip-id* **counters** | Displays the interface counter information. |
| **show interface fcip** *fcip-id* **counters brief** | Displays brief information of interface counters. |
| **show interface fcip** *fcip-id* **description** | Displays the interface description. |
| **show fcip target-map** | Displays information for a specified target map. |
| **show fcip tape-session summary** | Displays tape session information. |
| **show fcip tape-session tunnel 1 host-end** | Displays information for a specified FCIP link ID. |
| **show fcip tape-session tunnel 1 targ-end** | Displays information for the target end. |
| **show ips stats hw-comp all** | Displays hardware compression statistics. |
| **show port-channel database** | Displays port channel database information. |

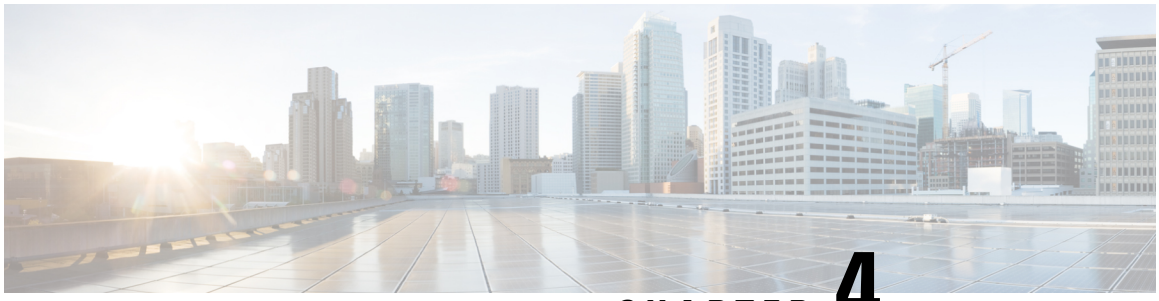
# Default Settings for FCIP Parameters

The following table lists the default settings for FCIP parameters.

***Table 5: Default FCIP Parameters***

| Parameters | Default |
|---|---|
| TCP default port for FCIP | 3225 |
| Base port for TCP connections to peer | 2 |
| Minimum retransmit time | 200 milliseconds |
| Keepalive timeout | 60 sec |
| Maximum retransmissions | 4 retransmissions |
| PMTU discovery | Enabled |
| PMTU enable reset timeout | 3600 seconds |
| SACK | Enabled |
| Maximum bandwidth (Cisco MDS 24/10 port SAN Extension Module) | 10 Gbps |
| Maximum bandwidth (Cisco MDS 9220i Fabric Switch) | 10 Gbps |

| | |
|---|---|
| Maximum bandwidth (Cisco MDS 9250i Multiservice Fabric Switch) | 5 Gbps |
| Minimum available bandwidth (Cisco MDS 24/10 port SAN Extension Module) | 8 Gbps |
| Minimum available bandwidth (Cisco MDS 9220i Fabric Switch) | 8 Gbps |
| Minimum available bandwidth (Cisco MDS 9250i Multiservice Fabric Switch) | 4 Gbps |
| Round trip time | 1 millisecond |
| Buffer size | 0 KB |
| Control TCP and data connection | No packets are transmitted |
| TCP congestion window monitoring | Enabled |
| Burst size | 50 KB |
| TCP connection mode | Active mode |
| FCIP timestamp | Disabled |
| Acceptable difference range to accept packets | +/– 2000 milliseconds |
| B port keepalive responses | Disabled |
| Write Acceleration | Disabled |
| Tape Acceleration | Disabled |



CHAPTER 4

# Configuring the SAN Extension Tuner

The SAN Extension Tuner (SET) feature is unique to the Cisco MDS 9000 Family of switches. This feature helps you optimize FCIP performance by generating either direct access (magnetic disk) or sequential access (magnetic tape) SCSI I/O commands and directing such traffic to a specific virtual target. You can specify the size of the test I/O transfers and how many concurrent or serial I/Os to generate while testing. The SET reports the resulting I/Os per second (IOPS) and I/O latency, which helps you determine the number of concurrent I/Os needed to maximize FCIP throughput.

This chapter includes the following sections:


- Overview of SAN Extension Tuner, on page 61
- SAN Extension Tuner Setup, on page 63
- Data Pattern, on page 63
- License Prerequisites, on page 64
- Configuring the SAN Extension Tuner, on page 64
- Verifying the SAN Extension Tuner Configuration, on page 70
- Default Settings for Tuning Parameters, on page 72


## Overview of SAN Extension Tuner

The SAN extension tuner (SET) feature is unique to the Cisco MDS 9000 Family of switches. This feature helps you optimize FCIP performance by generating either direct access (magnetic disk) or sequential access (magnetic tape) SCSI I/O commands and directing such traffic to a specific virtual target. Applications such as remote copy and data backup use FCIP over an IP network to connect across geographically distributed SANs. SET is implemented in IPS ports. When enabled, this feature can be used to generate SCSI I/O commands (read and write) to the virtual target based on your configured options.

**Note** SAN Extension Tuner is not supported on the Cisco Fabric Switch for HP c-Class BladeSystem, the Cisco Fabric Switch for IBM BladeCenter, and 16-Port Storage Services Node (SSN-16).

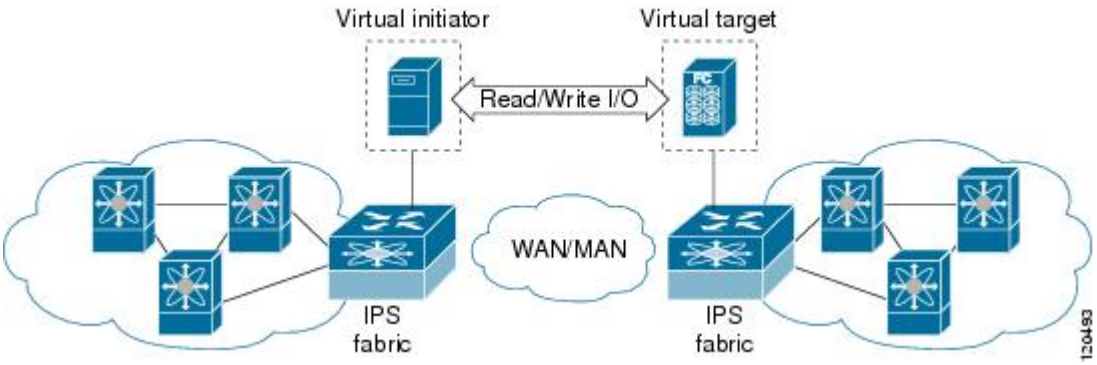
**Note** As of Cisco MDS SAN-OS Release 3.3(1a), SAN Extension Tuner is supported on the Multiservice Module (MSM) and the Multiservice Modular Switch.

Applications such as remote copy and data backup use FCIP over an IP network to connect across geographically distributed SANs. To achieve maximum throughput performance across the fabric, you can tune the following configuration parameters:

- The TCP parameters for the FCIP profile (see Window Management, on page 22 for more information).
- The number of concurrent SCSI I/Os generated by the application.
- The transfer size used by the application over an FCIP link.

SET is implemented in IPS ports. When enabled, this feature can be used to generate SCSI I/O commands (read and write) to the virtual target based on your configured options (see the following figure).

***Figure 12: SCSI Command Generation to the Virtual Target***




The SET feature assists with tuning by generating varying SCSI traffic workloads. It also measures throughput and response time per I/ O over an FCIP link.

Before tuning the SAN fabric, be aware of the following guidelines:

- Following these implementation details:
  - The tuned configuration is not persistent.
  - The virtual N ports created do not register FC4 features supported with the name server. This is to avoid the hosts in the SAN from discovering these N ports as regular initiators or targets.
  - Login requests from other initiators in the SAN are rejected.
  - The virtual N ports do not implement the entire SCSI suite; it only implements the SCSI read and write commands.
  - Tuner initiators can only communicate with tuner targets.
- Verify that the Gigabit Ethernet interface is up at the physical layer (GBIC and Cable connected—an IP address is not required).
- Enable iSCSI on the switch (no other iSCSI configuration is required).
- Enable the interface (no other iSCSI interface configuration is required)
- Create an iSCSI interface on the Gigabit Ethernet interface and enable the interface (no other iSCSI interface configuration is required)

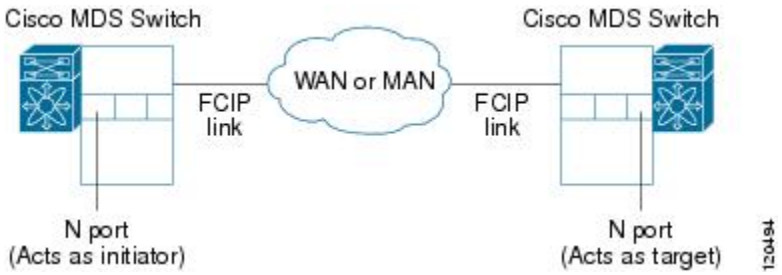
  See Creating iSCSI Interfaces, on page 77 for more information.

- Configure the virtual N ports in a separate VSAN or zone as required by your network.
- Be aware that a separate VSAN with only virtual N ports is not required, but is recommended as some legacy HBAs may fail if logins to targets are rejected.
- Do not use same Gigabit Ethernet interface to configure virtual N ports and FCIP links—use different Gigabit Ethernet interfaces. While this is not a requirement, it is recommended as the traffic generated by the virtual N ports may interfere with the performance of the FCIP link.

# SAN Extension Tuner Setup

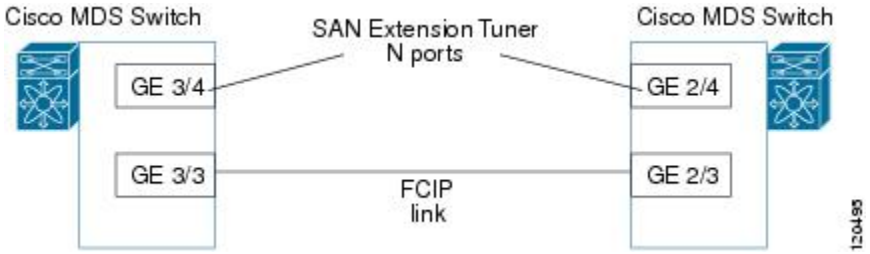
The below figure provides a sample physical setup in which the virtual N ports are created on ports that are not a part of the FCIP link for which the throughput and latency is measured.

***Figure 13: Port Tuning Configuration Physical Example***




The below figure provides a sample logical setup in which the virtual N ports are created on ports that are not a part of the FCIP link for which the throughput and latency is measured.

***Figure 14: Logical Example of N Port Tuning for an FCIP Link***




# Data Pattern

By default, an all-zero pattern is used as the pattern for data generated by the virtual N ports. You can optionally specify a file as the data pattern to be generated by selecting a data pattern file from one of three locations: the bootflash: directory, the volatile: directory, or the slot0: directory. This option is especially useful when testing compression over FCIP links. You can also use Canterbury corpus or artificial corpus files for benchmarking purposes.

# License Prerequisites

To use the SET, you need to obtain the SAN_EXTN_OVER_IP license (see the *Cisco Family NX-OS Licensing Guide*).

# Configuring the SAN Extension Tuner

This section includes the following topics:

## Tuning the FCIP Link

To tune the required FCIP link, follow these steps:

**Procedure**

**Step 1** Configure the nWWN for the virtual N ports on the switch.

**Step 2** Enable iSCSI on the interfaces on which you want to create the N ports.

**Step 3** Configure the virtual N ports on either side of the FCIP link.

**Step 4** Ensure that the virtual N ports are not visible to real initiators in the SAN. You can use zoning (see the Cisco MDS 9000 Family NX-OS Fabric Configuration Guide) to segregate the real initiators.

Ensure that the zoning configuration is set up to allow the virtual N ports to communicate with each other.

**Step 5** Start the SCSI read and write I/Os.

**Step 6** Add more N ports (as required) to other Gigabit Ethernet ports in the switch to obtain maximum throughput. One scenario that may require additional N ports is if you use FCIP PortChannels.

## Enabling the Tuner

The tuning feature is disabled by default in all switches in the Cisco 9000 Family. When you enable this feature, tuning is globally enabled for the entire switch.

To enable the tuning feature, follow these steps:

**Procedure**

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | switch# **configure terminal** | Enters configuration mode. |
| **Step 2** | switch(config)# **feature san-ext-tuner** | Enables tuning. |
| **Step 3** | switch(config)# **no feature san-ext-tuner** | Removes the currently applied tuning configuration and disables tuning (default). |

# Configuring nWWN

To configure the nWWNs for the tuner in this switch, follow these steps:

**Procedure**

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | switch# **san-ext-tuner** | Enters the SET configuration submode. |
| **Step 2** | switch(san-ext)# **nWWN 10:00:00:00:00:00:00:00** | Configures the nWWN for the SAN extension tuner. |

# Configuring the Virtual N Port

To configure the virtual N port for tuning, follow these steps:

**Procedure**

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | switch# **configure terminal** | Enters configuration mode. |
| **Step 2** | switch(config)# **feature iscsi** | Enables iSCSI globally. |
| **Step 3** | switch(config)# **iscsi enable module 1** | Enables iSCSI on module 1. |
| **Step 4** | switch(config)# **interface iscsi 1/1** | Creates an iSCSI interface and enters interface configuration submode. |
| **Step 5** | switch(config-if)# **no shutdown** | Enables the iSCSI interface. |
| **Step 6** | switch(config-if)# **end** | Returns to EXEC mode. |
| **Step 7** | switch# **san-ext-tuner** | Enters the SET configuration submode. |
| **Step 8** | switch(san-ext)# **nport pWWN 12:00:00:00:00:00:00:56 vsan 200 interface gigabitethernet 1/1** | Creates a virtual N port on the specified Gigabit Ethernet port and VSAN. This N port can act as an initiator or a target. |
| **Step 9** | (Optional) switch(san-ext)# **no nport pWWN 22:34:56:78:90:12:34:56 vsan 200 interface gigabitethernet 1/1** | Removes a virtual N port on the specified Gigabit Ethernet port and VSAN. |

# Generating SCSI Disk Read/Write IO

You can assign SCSI read and write commands on a one-time basis or on a continuous basis.

This section includes the following topics:

## Generating SCSI Disk Read/Write IO One Time

To generate SCSI read or write commands on a one-time basis, follow these steps:

**Procedure**

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | switch# **san-ext-tuner** | Enters the SET configuration submode. |
| **Step 2** | switch(san-ext)# **nport pWWN 12:00:00:00:00:00:00:56 vsan 200 interface gigabitethernet 1/1** | Creates a virtual N port on the specified Gigabit Ethernet port and VSAN. This N port can act as an initiator or a target. |
| **Step 3** | switch(san-ext-nport)# **read command-id 100 target 22:22:22:22:22:22:22:22 transfer-size 512000 outstanding-ios 2 num-transactions 5000000** | Specifies a transfer size of 512,000 bytes with two outstanding I/Os in the **read** command. The total number of I/Os is 5,000,000 bytes. |
| **Step 4** | switch(san-ext-nport)# **write command-id 101 target 22:22:22:22:22:22:22:22 transfer-size 512000 outstanding-ios 2 num-transactions 5000000** | Specifies a transfer size of 512,000 bytes with two outstanding I/Os in the **write** command received by the target. The total number of I/Os is 5,000,000 bytes. |
| **Step 5** | switch(san-ext-nport)# **stop command-id 100** | Stops the command with the specified ID. |
| **Step 6** | (Optional) switch(san-ext-nport)# **stop all** | Stops all outstanding commands. |
| **Step 7** | switch(san-ext-nport)# **clear counters** | Clears the counters associated with this N port. |
| **Step 8** | switch(san-ext-nport)# **end** | Exits the SAN extension tuner submode. |

## Generating SCSI Disk Read/Write IO Continuously

To generate SCSI read or write commands continuously, follow these steps:

**Procedure**

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | switch# **san-ext-tuner** | Enters the SET configuration submode. |
| **Step 2** | switch(san-ext)# **nport pWWN 12:00:00:00:00:00:00:56 vsan 200 interface gigabitethernet 1/1** | Creates a virtual N port on the specified Gigabit Ethernet port and VSAN. This N port can act as an initiator or a target. |
| **Step 3** | switch(san-ext-nport)# **read command-id 100 target 22:22:22:22:22:22:22:22 transfer-size 512000 outstanding-ios 2 continuous** | Configures SCSI commands to be read continuously.<br>**Tip**<br>Use the **stop command-id** command to stop the outstanding configuration. |
| **Step 4** | switch(san-ext-nport)# **write command-id 100 target 22:22:22:22:22:22:22:22 transfer-size 512000 outstanding-ios 2 continuous** | Configures SCSI commands to be written continuously. |

| | Command or Action | Purpose |
|---|---|---|
| **Step 5** | switch(san-ext-nport)# **stop command-id 100** | Stops the command with the specified ID. |
| **Step 6** | (Optional) switch(san-ext-nport)# **stop command-id all** | Stops all outstanding commands. |
| **Step 7** | (Optional) switch(san-ext-nport)# **stop all** | Stops all outstanding commands. |
| **Step 8** | switch(san-ext-nport)# **clear counters** | Clears the counters associated with this N port. |
| **Step 9** | switch(san-ext-nport)# **end** | Exits the SAN extension tuner submode. |

## Transferring Ready Size for SCSI

To specify a transfer ready size for a SCSI write command, follow these steps:

**Procedure**

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | switch# **san-ext-tuner** | Enters the SET configuration submode. |
| **Step 2** | switch(san-ext)# **nport pWWN 12:00:00:00:00:00:00:56 vsan 200 interface gigabitethernet 1/1** | Creates a virtual N port on the specified Gigabit Ethernet port and VSAN. This N port can act as an initiator or a target. |
| **Step 3** | switch(san-ext-nport)# **write command-id 100 target 22:22:22:22:22:22:22:22 transfer-size 512000 outstanding-ios 2 num-transactions 5000000** | Specifies a transfer size of 512,000 bytes with two outstanding I/Os in the **write** command received by the target. The total number of I/Os is 5,000,000 bytes. |
| **Step 4** | switch(san-ext-nport)# **transfer-ready-size 512000** | Specifies the maximum transfer ready size of 512,000 bytes as a target for SCSI write commands. For a SCSI **write** command with a larger size, the target performs multiple transfers based on the specified transfer size. |
| **Step 5** | switch(san-ext-nport)# **no transfer-ready-size 512000** | Removes the specified transfer ready size configuration for SCSI write commands. |
| **Step 6** | switch(san-ext-nport)# **stop command-id 100** | Stops the command with the specified ID. |
| **Step 7** | switch(san-ext-nport)# **end** | Exits the SAN extension tuner submode. |

# Generating SCSI Tape Read/Write IO

You can assign SCSI tape read and write commands on a one-time basis or on a continuous basis.

**Note**

- Ensure that the zoning configuration is set up to allow the virtual N-ports to communicate with each other.
- There is only one outstanding I/O at a time to the virtual N-port that emulates the tape behavior.

This section contains the following topics:

## Generating SCSI Tape Read/Write IO One Time

To generate SCSI tape read and or write commands on a one-time basis, follow these steps:

**Procedure**

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | switch# **san-ext-tuner** | Enters the SET configuration submode. |
| **Step 2** | switch(san-ext)# **nport pWWN 12:00:00:00:00:00:00:56 vsan 200 interface gigabitethernet 1/1** | Creates a virtual N port on the specified Gigabit Ethernet port and VSAN. This N port can act as an initiator or a target. |
| **Step 3** | switch(san-ext-nport)# **tape-read command-id 100 target 22:22:22:22:22:22:22:22 transfer-size 512000 num-transactions 5000000 filemark-frequency 32** | Specifies a transfer size of 512,000 bytes with space over the filemark every 32 SCSI read commands. The total number of I/Os is 5,000,000 bytes. |
| **Step 4** | switch(san-ext-nport)# **tape-write command-id 101 target 22:22:22:22:22:22:22:22 transfer-size 512000 num-transactions 5000000 filemark-frequency 32** | Specifies a transfer size of 512,000 bytes with filemarks written every 32 SCSI write commands. The total number of I/Os is 5,000,000 bytes. |
| **Step 5** | switch(san-ext-nport)# **stop command-id 100** | Stops the command with the specified ID. |
| **Step 6** | (Optional) switch(san-ext-nport)# **stop all** | Stops all outstanding commands. |
| **Step 7** | switch(san-ext-nport)# **clear counters** | Clears the counters associated with this N port. |
| **Step 8** | switch(san-ext-nport)# **end** | Exits the SAN extension tuner submode. |

## Generating SCSI Tape Read/Write IO Continuously

To generate SCSI tape read or write commands continuously, follow these steps:

**Procedure**

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | switch# **san-ext-tuner** | Enters the SET configuration submode. |
| **Step 2** | switch(san-ext)# **nport pWWN 12:00:00:00:00:00:00:56 vsan 200 interface gigabitethernet 1/1** | Creates a virtual N port on the specified Gigabit Ethernet port and VSAN. This N port can act as an initiator or a target. |
| **Step 3** | switch(san-ext-nport)# **tape-read command-id 100 target 22:22:22:22:22:22:22:22 transfer-size 512000 continuous filemark-frequency 32** | Configures SCSI tape read commands to be issued continuously.<br>**Tip**<br>Use the **stop command-id** command to stop the outstanding configuration. |

| | Command or Action | Purpose |
|---|---|---|
| **Step 4** | switch(san-ext-nport)# **tape-write command-id 100 target 22:22:22:22:22:22:22:22 transfer-size 512000 continuous filemark-frequency 32** | Configures SCSI tape write commands to be issued continuously. |
| **Step 5** | switch(san-ext-nport)# **stop command-id 100** | Stops the command with the specified ID. |
| **Step 6** | (Optional) switch(san-ext-nport)# **stop command-id all** | Stops all outstanding commands. |
| **Step 7** | switch(san-ext-nport)# **clear counters** | Clears the counters associated with this N port. |
| **Step 8** | switch(san-ext-nport)# **end** | Exits the SAN extension tuner submode. |

## Transferring Ready Size for SCSI

To specify a transfer ready size for a SCSI write command, follow these steps:

**Procedure**

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | switch# **san-ext-tuner** | Enters the SET configuration submode. |
| **Step 2** | switch(san-ext)# **nport pWWN 12:00:00:00:00:00:00:56 vsan 200 interface gigabitethernet 1/1** | Creates a virtual N port on the specified Gigabit Ethernet port and VSAN. This N port can act as an initiator or a target. |
| **Step 3** | switch(san-ext-nport)# **write command-id 100 target 22:22:22:22:22:22:22:22 transfer-size 512000 outstanding-ios 2 num-transactions 5000000** | Specifies a transfer size of 512,000 bytes with two outstanding I/Os in the **write** command received by the target. The total number of I/Os is 5,000,000 bytes. |
| **Step 4** | switch(san-ext-nport)# **transfer-ready-size 512000** | Specifies the maximum transfer ready size of 512,000 bytes as a target for SCSI write commands. For a SCSI **write** command with a larger size, the target performs multiple transfers based on the specified transfer size. |
| **Step 5** | switch(san-ext-nport)# **no transfer-ready-size 512000** | Removes the specified transfer ready size configuration for SCSI write commands. |
| **Step 6** | switch(san-ext-nport)# **stop command-id 100** | Stops the command with the specified ID. |
| **Step 7** | switch(san-ext-nport)# **end** | Exits the SAN extension tuner submode. |

# Configuring a Data Pattern

To optionally configure a data pattern for SCSI commands, follow these steps:

**Procedure**

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | switch# **san-ext-tuner** | Enters the SET configuration submode. |
| **Step 2** | switch(san-ext)# **nport pWWN 12:00:00:00:00:00:00:56 vsan 200 interface gigabitethernet 1/1** | Creates a virtual N port on the specified Gigabit Ethernet port and VSAN. This N port can act as an initiator or a target. |
| **Step 3** | switch(san-ext-nport)# **data-pattern-file bootflash://DataPatternFile** | Specifies the data pattern sent by the virtual N port when it is a target for read commands and an initiator for write commands.<br>**Tip**<br>This command should be configured on the target to change the data returned by read commands and on the initiator for write commands. This command is useful to define data sets which contain certain bit patterns or have certain compression ratios. The default data set of all zeros is very homogenous and very compressible. |
| **Step 4** | (Optional) switch(san-ext-nport)# **no data-pattern-file** | Removes the specified data pattern configuration for SCSI read and write commands. The default is to send an all zero data pattern. |
| **Step 5** | switch(san-ext-nport)# **write command-id 100 target 22:22:22:22:22:22:22:22 transfer-size 512000 outstanding-ios 2 num-transactions 5000000** | Specifies a transfer size of 512,000 bytes with two outstanding I/Os. The total number of I/Os is 5,000,000 bytes. |
| **Step 6** | switch(san-ext-nport)# **stop command-id 100** | Stops the command with the specified ID. |
| **Step 7** | switch(san-ext-nport)# **clear counters** | Clears the counters associated with this N port. |
| **Step 8** | switch(san-ext-nport)# **end** | Exits the SAN extension tuner submode. |

# Verifying the SAN Extension Tuner Configuration

The **show** commands display the current SAN extension tuner settings for the Cisco MDS switch.

### Displaying Entries in the FLOGI Database

```
switch# show flogi database
--------------------------------------------------------------------------------
INTERFACE VSAN FCID     PORT NAME               NODE NAME
--------------------------------------------------------------------------------
iscsi1/1   200 0x050000 12:00:00:00:00:00:00:56 10:00:00:00:00:00:00:00
```

### Displaying Details for a VSAN Entry in the FLOGI Database

```
switch# show fcns database vsan 200
VSAN 200
--------------------------------------------------------------------------
FCID TYPE PWWN (VENDOR) FC4-TYPE:FEATURE
--------------------------------------------------------------------------
0x020000 N 22:22:22:22:22:22:22:22 scsi-fcp
0x050000 N 12:00:00:00:00:00:00:56 scsi-fcp
```

### Displaying All Virtual N Ports Configured on the Specified Interface

```
switch# show san-ext-tuner interface gigabitethernet 3/4 nport pWWN 12:00:00:00:00:00:00:56
 vsan 200 counters
Statistics for nport
Node name 10:00:00:00:00:00:00:00 Port name 12:00:00:00:00:00:00:56
I/Os per second : 148
Read : 0%
Write : 100%
Ingress MB per second : 0.02 MBs/sec (Max -0.02 MBs/sec)
Egress MB per second : 73.97 MBs/sec (Max -75.47 MBs/sec))
Average Response time per I/O : Read - 0 us, Write - 13432 us
Maximum Response time per I/O : Read - 0 us, Write - 6953 us
Minimum Response time per I/O : Read - 0 us, Write - 19752 us
Errors : 0
```

### Displaying N Ports Configured on a Specified Gigabit Ethernet Interface

```
switch# show san-ext-tuner interface gigabitethernet 3/1
--------------------------------------------------------------------------------
Interface          NODE NAME               PORT                    NAME VSAN
--------------------------------------------------------------------------------
GigabitEthernet3/1 10:00:00:00:00:00:00:00 10:00:00:00:00:00:00:01 91
```

### Displaying the Transfer Ready Size Configured for a Specified N Port

```
switch# show san-ext-tuner interface gigabitethernet 3/1 nport pWWN 10:0:0:0:0:0:0:1 vsan
91
Node name : 10:00:00:00:00:00:00:00
Port name : 10:00:00:00:00:00:00:01
Transfer ready size : all
```

### Displaying All Virtual N Ports Configured in a Switch

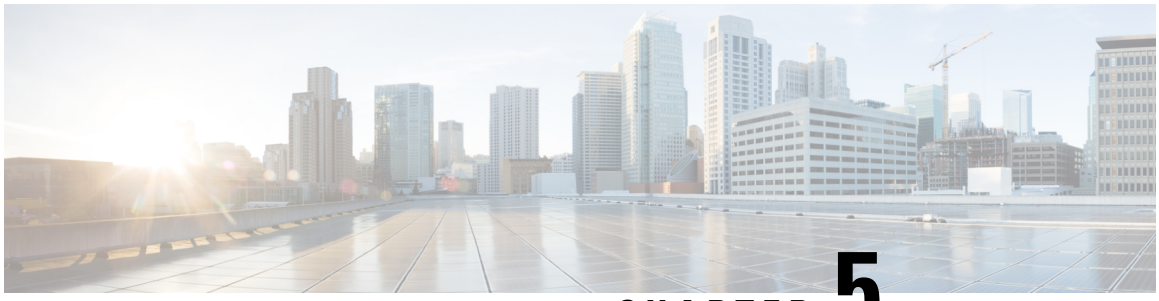
```
switch# show san-ext-tuner nports
--------------------------------------------------------------------------------
Interface          NODE NAME               PORT NAME               VSAN
--------------------------------------------------------------------------------
GigabitEthernet3/1 10:00:00:00:00:00:00:00 10:00:00:00:00:00:00:01 91
```

# Default Settings for Tuning Parameters

The following table lists the default settings for tuning parameters.

***Table 6: Default Tuning Parameters***

| Parameters | Default |
|---|---|
| Tuning | Disabled |
| Transfer ready size | Same as the transfer size in the SCSI **write** command |
| Outstanding I/Os | 1 |
| Number of transactions | 1 |
| Data generation format | All-zero format |
| File mark frequency | 0 |



CHAPTER 5

# Configuring iSCSI

Cisco MDS 9000 Family IP storage (IPS) services extend the reach of Fibre Channel SANs by using open-standard, IP-based technology. The switch allows IP hosts to access Fibre Channel storage using the Internet Small Computer Systems Interface (iSCSI) protocol.

**Note** The iSCSI feature is specific to the Fibre Channel module with IPS ports and is available in Cisco MDS 9200 Switches or Cisco MDS 9500 Directors. In Cisco MDS NX-OS Release 7.3(0)DY(1) and later, iSCSI is not supported on Cisco MDS 9700 Directors with 24/10 port SAN Extension modules.

The Cisco MDS 9216i switch and the 14/2 Multiprotocol Services (MPS-14/2) module also allow you to use Fibre Channel, FCIP, and iSCSI features. The MPS-14/2 module is available for use in any switch in the Cisco MDS 9200 Series or Cisco MDS 9500 Series.

This chapter includes the following sections:


- Overview of iSCSI, on page 73
- **Configuring iSCSI**, on page 76
- Configuring iSLB, on page 120
- Load Balancing Using VRRP, on page 132
- iSLB Configuration Distribution Using CFS, on page 138
- Distributing iSLB Configuration Using CFS, on page 139
- iSCSI High Availability, on page 143
- iSCSI Authentication Setup Guidelines and Scenarios, on page 149
- Overview of Internet Storage Name Service, on page 169
- Default Settings, on page 186


# Overview of iSCSI

Cisco MDS 9000 Family IP Storage (IPS) services extend the reach of Fibre Channel SANs by using open-standard, IP-based technology. The iSCSI feature consists of routing iSCSI requests and responses between iSCSI hosts in an IP network and Fibre Channel storage devices in the Fibre Channel SAN that are accessible from any Fibre Channel interface of the Cisco MDS 9000 Family switch. Using the iSCSI protocol, the iSCSI driver allows an iSCSI host to transport SCSI requests and responses over an IP network. To use the iSCSI feature, you must explicitly enable iSCSI on the required switches in the fabric.

**Note** The iSCSI feature is not supported on the Cisco Fabric Switch for HP c-Class Bladesystem and Cisco Fabric Switch for IBM BladeCenter.

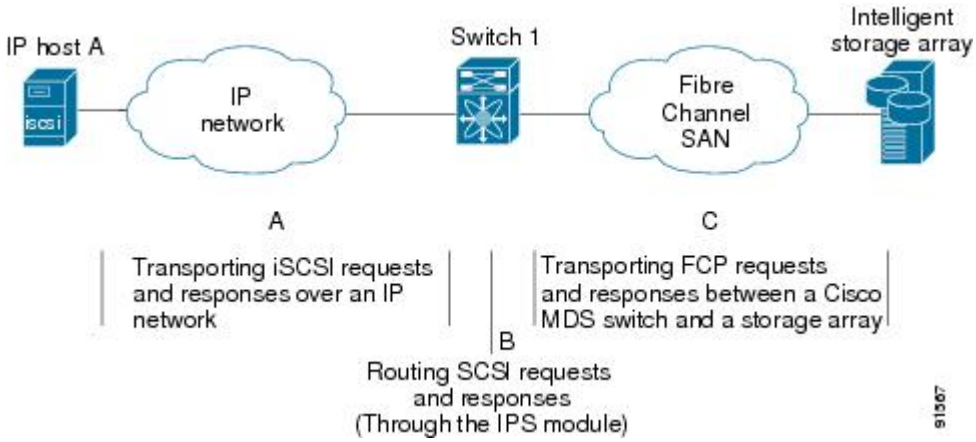
The iSCSI feature consists of routing iSCSI requests and responses between iSCSI hosts in an IP network and Fibre Channel storage devices in the Fibre Channel SAN that are accessible from any Fibre Channel interface of the Cisco MDS 9000 Family switch.

Each iSCSI host that requires access to storage through the Fibre Channel module with IPS ports or MPS-14/2 module needs to have a compatible iSCSI driver installed. Using the iSCSI protocol, the iSCSI driver allows an iSCSI host to transport SCSI requests and responses over an IP network. From the host operating system perspective, the iSCSI driver appears to be an SCSI transport driver similar to a Fibre Channel driver in the host.

The Fibre Channel module with IPS ports or MPS-14/2 module provides transparent SCSI routing. IP hosts using the iSCSI protocol can transparently access targets on the Fibre Channel network.

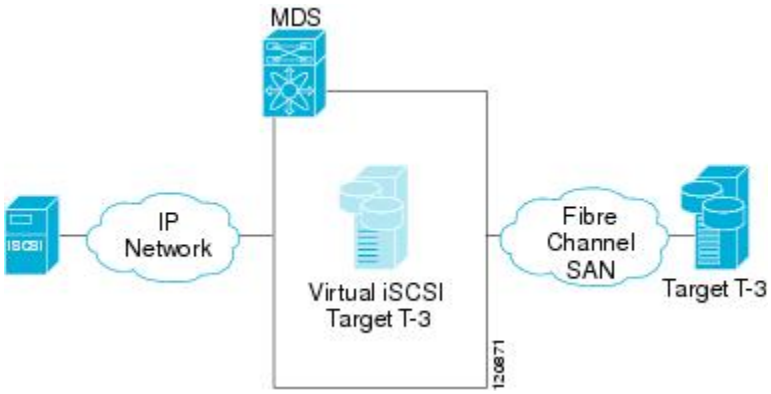
The following figure provides an example of a typical configuration of iSCSI hosts connected to an Fibre Channel module with IPS ports or MPS-14/2 module through the IP network access Fibre Channel storage on the Fibre Channel SAN.

***Figure 15: Transporting iSCSI Requests and Responses for Transparent iSCSI Routing***



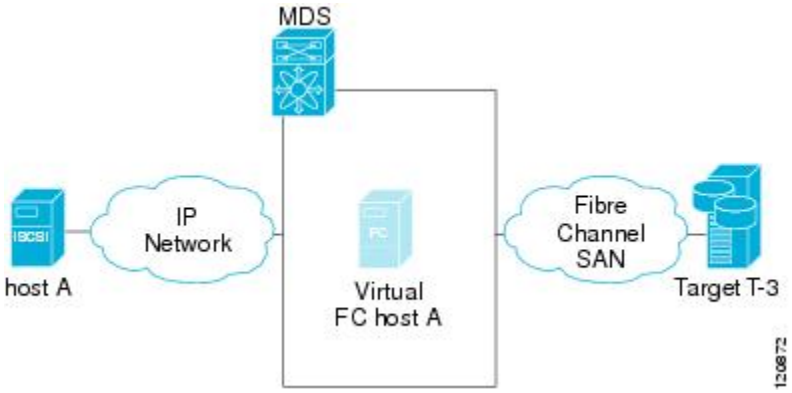

The Fibre Channel module with IPS ports or MPS-14/2 module create a separate iSCSI SAN view and Fibre Channel SAN view. For the iSCSI SAN view, the Fibre Channel module with IPS ports or MPS-14/2 module creates iSCSI virtual targets and then maps them to physical Fibre Channel targets available in the Fibre Channel SAN. They present the Fibre Channel targets to IP hosts as if the physical iSCSI targets were attached to the IP network.

*Figure 16: iSCSI SAN View—iSCSI Virtual Targets*



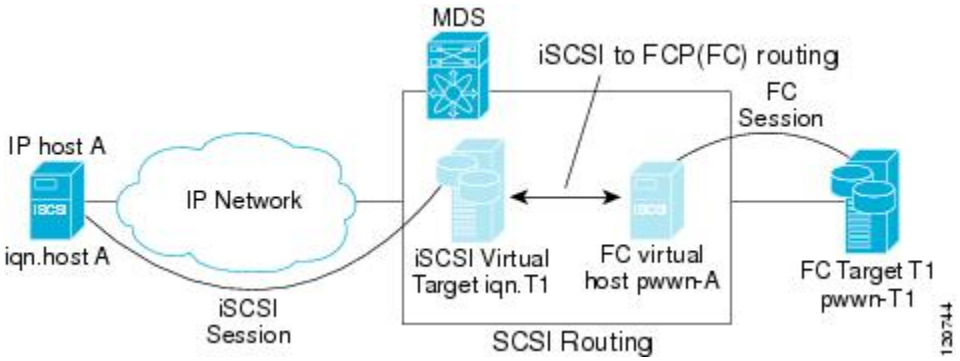

For the Fibre Channel SAN view, the Fibre Channel module with IPS ports or MPS-14/2 module presents iSCSI hosts as a virtual Fibre Channel host. The storage devices communicate with the virtual Fibre Channel host similar to communications performed with real Fibre Channel hosts .

*Figure 17: Fibre Channel SAN View—iSCSHI Host as an HBA*




The Fibre Channel module with IPS ports or MPS-14/2 modules transparently map the command between the iSCSI virtual target and the virtual Fibre Channel host.

*Figure 18: iSCSI to FCP (Fibre Channel) Routing*




Routing SCSI from the IP host to the Fibre Channel storage device consists of the following main actions:

- The iSCSI requests and responses are transported over an IP network between the hosts and the Fibre Channel module with IPS ports or MPS-14/2 module.
- The SCSI requests and responses are routed between the hosts on an IP network and the Fibre Channel storage device (converting iSCSI to FCP and vice versa). The Fibre Channel module with IPS ports or MPS-14/2 module performs this conversion and routing.
- The FCP requests or responses are transported between the Fibre Channel module with IPS ports or MPS-14/2 module and the Fibre Channel storage devices.



**Note** The Fibre Channel equivalent of iSCSI (FCP) carries SCSI commands over a Fibre Channel SAN.Refer to the IETF standards for IP storage at http://www.ietf.org for information on the iSCSI protocol.

# iSCSI Configuration Limits

iSCSI configuration has the following limits:

- The maximum number of iSCSI and iSLB initiators supported in a fabric is 2000.
- The maximum number of iSCSI and iSLB initiators supported is 200 per port.
- The maximum number of iSCSI and iSLB sessions supported by an IPS port in either transparent or proxy initiator mode is 500.
- The maximum number of iSCSI and iSLB session support by switch is 5000.
- The maximum number of iSCSI and iSLB targets supported in a fabric is 6000.

# Configuring iSCSI

Cisco MDS 9000 Family IP storage (IPS) services extend the reach of Fibre Channel SANs by using open-standard, IP-based technology. The switch allows IP hosts to access Fibre Channel storage using the Internet Small Computer Systems Interface (iSCSI) protocol.



**Note** The iSCSI feature is specific to the Fibre Channel module with IPS ports and is available in Cisco MDS 9200 Switches or Cisco MDS 9500 Directors. In Cisco MDS NX-OS Release 7.3(0)DY(1) and later, iSCSI is not supported on Cisco MDS 9700 Directors with 24/10 port SAN Extension modules.

The Cisco MDS 9216i switch and the 14/2 Multiprotocol Services (MPS-14/2) module also allow you to use Fibre Channel, FCIP, and iSCSI features. The MPS-14/2 module is available for use in any switch in the Cisco MDS 9200 Series or Cisco MDS 9500 Series.



**Note** For information on configuring Gigabit Ethernet interfaces, see Basic Gigabit Ethernet Configuration for IPv4 section.

This chapter includes the following sections:

# Enabling iSCSI

To use the iSCSI feature, you must explicitly enable iSCSI on the required switches in the fabric. Alternatively, you can enable or disable the iSCSI feature directly on the required modules using Fabric Manager or Device Manager. By default, this feature is disabled in all switches in the Cisco MDS 9000 Family.

To enable iSCSI on any participating switch, follow these steps:

### Before you begin

When you disable this feature, all related configurations are automatically discarded.

### Procedure

**Step 1** Enter the configuration commands, one per line. End with CNTL/Z

switch# **config terminalerminal**

**Step 2** Enable iSCSI on that switch.

switch(config)# **feature iscsi**

**Step 3** Disable (default) iSCSI on that switch.

switch(config)# **no feature iscsi**

**Step 4** Enable iSCSI modules on the switch.

switch(config)# **iscsi enable module <x>**

**Note**
New command added so that SME and iSCSI are available on the same switch

**Step 5** Disable the iSCSI module on the switch.

switch(config)# **no iscsi enable module <x>**

# Presenting iSCSI Hosts as Virtual Fibre Channel Hosts

The Fibre Channel module with IPS ports or MPS-14/2 module connects to the Fibre Channel storage devices on behalf of the iSCSI host to send commands and transfer data to and from the storage devices. These modules use a virtual Fibre Channel N port to access the Fibre Channel storage devices on behalf of the iSCSI host. iSCSI hosts are identified by either iSCSI qualified name (IQN) or IP address.

## Creating iSCSI Interfaces

Each physical Gigabit Ethernet interface on an Fibre Channel module with IPS ports, MPS-14/2 module or 1/10Gbps IPStorage port on a Cisco MDS 9250i Multiservice Fabric Switch can be used to translate and route

iSCSI requests to Fibre Channel targets and responses in the opposite direction. To enable this capability, the corresponding iSCSI interface must be in an enabled state.

To enable iSCSI interfaces, follow these steps:

### Before you begin

Use the **tcp maximum-bandwidth-kbps** and **tcp maximum-bandwidth-mbps** commands to configure the iSCSI speed and the **switchport speed** command to set the Physical IPStorage ports to 1Gbps or 10Gbps speed. The Cisco MDS switches do not limit the configuration of the iSCSI tcp maximum-bandwidth-kbps and maximum-bandwidth-mbps based on the speed of the underlying physical Gigabit Ethernet or IPStorage ports. Consequently, it is possible to configure iSCSI **tcp maximum-bandwidth-kbps** and **tcp maximum-bandwidth-mbps** commands to the equivalent of 10Gbps on a physical IPStorage port that is running at a 1Gbps speed. When configuring the tcp maximum bandwidth, ensure that it does not exceed the maximum speed of the physical IPStorage port.

### Procedure

**Step 1** Enable the required Gigabit Ethernet interface.

```
switch# config terminal
switch(config)# interface gigabitethernet 2/1
switch(config-if)# no shutdown
switch(config-if)# exit
switch(config)#
```

**Step 2** Create the required iSCSI interface and enable the interface.

```
switch(config)# interface iscsi 2/1
switch(config-if)# no shutdown
```

## Presenting Fibre Channel Targets as iSCSI Targets

The Fibre Channel module with IPS ports or MPS-14/2 module presents physical Fibre Channel targets as iSCSI virtual targets, allowing them to be accessed by iSCSI hosts. The module presents these targets in one of the two ways:

- Dynamic mapping—Automatically maps all the Fibre Channel target devices/ports as iSCSI devices. Use this mapping to create automatic iSCSI target names

  .
- Static mapping—Manually creates iSCSI target devices and maps them to the whole Fibre Channel target port or a subset of Fibre Channel LUNs. With this mapping, you must specify unique iSCSI target names.

Static mapping should be used when iSCSI hosts should be restricted to subsets of LUs in the Fibre Channel targets and/or iSCSI access control is needed (see the iSCSI Access Control section). Also, static mapping allows the configuration of transparent failover if the LUs of the Fibre Channel targets are reachable by redundant Fibre Channel ports (see the Transparent Target Failover section).

**Note** The Fibre Channel module with IPS ports or MPS-14/2 module does not import Fibre Channel targets to iSCSI by default. Either dynamic or static mapping must be configured before the Fibre Channel module with IPS ports or MPS-14/2 module makes Fibre Channel targets available to iSCSI initiators.

### *Dynamic Mapping*

When you configure dynamic mapping the Fibre Channel module with IPS ports or MPS-14/2 module imports all Fibre Channel targets to the iSCSI domain and maps each physical Fibre Channel target port as one iSCSI target. That is, all LUs accessible through the physical storage target port are available as iSCSI LUs with the same LU number (LUN) as in the physical Fibre Channel target port.

The iSCSI target node name is created automatically using the iSCSI qualified name (IQN) format. The iSCSI qualified name is restricted to a maximum name length of 223 alphanumeric characters and a minimum length of 16 characters.

The Fibre Channel module with IPS ports or MPS-14/2 module creates an IQN formatted iSCSI target node name using the following conventions because the name must be unique in the SAN:

- IPS Gigabit Ethernet ports that are not part of a Virtual Router Redundancy Protocol (VRRP) group or port channel use this format:

  `iqn.1987-05.com.cisco:05.<mgmt-ip-address>.<slot#>-<port#>-<sub-intf#>.<Target-pWWN>`
- IPS ports that are part of a VRRP group use this format:

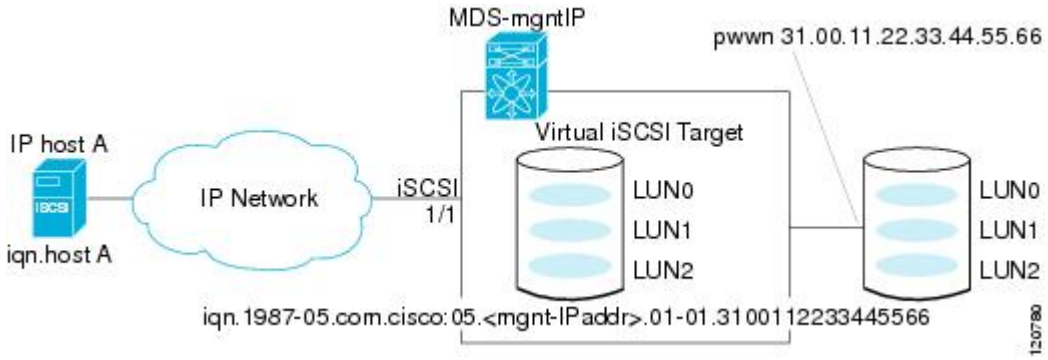
  `iqn.1987-05.com.cisco:05.vrrp-<vrrp-ID#>-<vrrp-IP-addr>.<Target-pWWN>`
- Ports that are part of a port channel use this format:`iqn.1987-02.com.cisco:02.<mgmt-ip-address>.pc-<port-ch-sub-intf#>.<Target-pWWN>`

**Note** If you have configured a switch name, then the switch name is used instead of the management IP address. If you have not configured a switch name, the management IP address is used.

With this convention, each IPS port in a Cisco MDS 9000 Family switch creates a unique iSCSI target node name for the same Fibre Channel target port in the SAN.

For example, if an iSCSI target was created for a Fibre Channel target port with pWWN 31:00:11:22:33:44:55:66 and that pWWN contains LUN 0, LUN 1, and LUN 2, those LUNs would become available to an IP host through the iSCSI target node name iqn.1987-05.com.cisco:05. *MDS_switch_management_IP_address* .

*Figure 19: Dynamic Target Mapping*




**Note** Each iSCSI initiator may not have access to all targets depending on the configured access control mechanisms (see the iSCSI Access Control section).

### *Enabling Dynamic Mapping of Fibre Channel Targets into iSCSI*

To enable dynamic mapping of Fibre Channel targets into iSCSI, follow these steps:

**Procedure**

**Step 1** Enter configuration mode.

```
switch# config terminal
switch(config)#
```

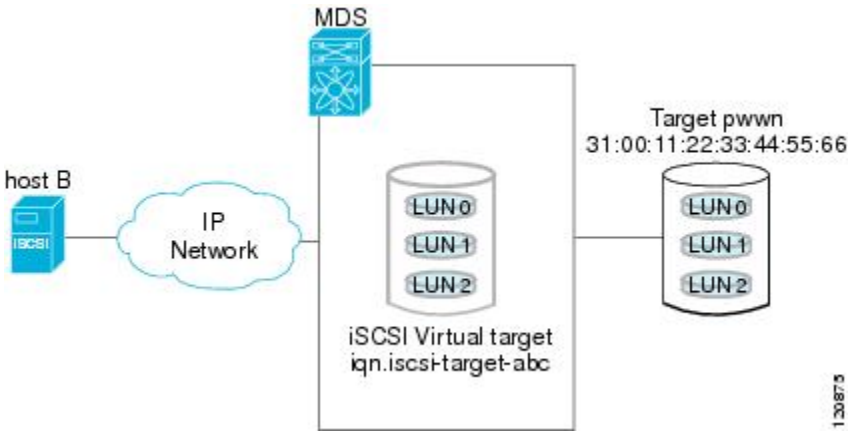
**Step 2** Fibre Channel module with IPS ports and MPS-14/2 modules dynamically import all Fibre Channel targets in the Fibre Channel SAN into the IP network.

```
switch(config)# iscsi import target fc
```

### *Static Mapping*

You can manually (statically) create an iSCSI target by assigning a user-defined unique iSCSI node name to it. The iSCSI qualified name is restricted to a minimum length of 16 characters and a maximum of 223 characters. A statically mapped iSCSI target can either map the whole Fibre Channel target port (all LUNs in the target port mapped to the iSCSI target), or it can contain one or more LUs from a Fibre Channel target port.

*Figure 20: Statically Mapped iSCSI Targets*




**Note** An iSCSI target cannot contain more than one Fibre Channel target port. If you have already mapped the whole Fibre Channel target port, you cannot use the LUN mapping option.

See the iSCSI-Based Access Control section for more information on controlling access to statically mapped targets.

### Advertising Static iSCSI Targets

You can limit the Gigabit Ethernet interfaces through which static iSCSI targets are advertised. By default iSCSI targets are advertised on all Gigabit Ethernet interfaces, subinterfaces, port channel interfaces, and port channel subinterfaces.

## Configuing an Interface to Advertise the iSCSI Virtual Target

To configure a specific interface that should advertise the iSCSI virtual target, follow these steps:

### Procedure

**Step 1** Advertise the virtual target only on the specified interface. By default, it is advertised on all interfaces in all Fibre Channel module with IPS ports or MPS-14/2 modules.

```
switch(config-iscsi-tgt)# advertise interface GigabitEthernet 2/5
```

**Note**
To advertise the virtual target on multiple interfaces, issue the command for each interface.

**Step 2** Remove this interface from the list of interfaces from which this target is advertised

```
switch(config-iscsi-tgt)# no advertise interface GigabitEthernet 2/5
```

## iSCSI Virtual Target Configuration Examples

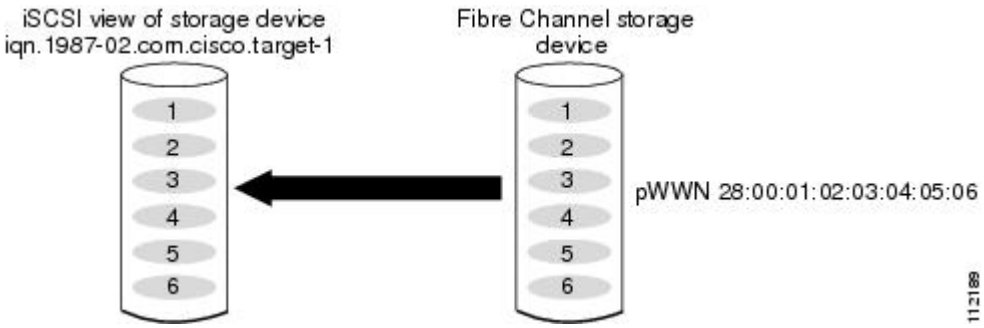
This section provides three examples of iSCSI virtual target configurations.

### Example 1

This example assigns the whole Fibre Channel target as an iSCSI virtual target. All LUNs that are part of the Fibre Channel target are available as part of the iSCSI target.

```
iscsi virtual-target name iqn.1987-02.com.cisco.target-1
pWWN 28:00:01:02:03:04:05:06
```

*Figure 21: Assigning iSCSI Node Names*

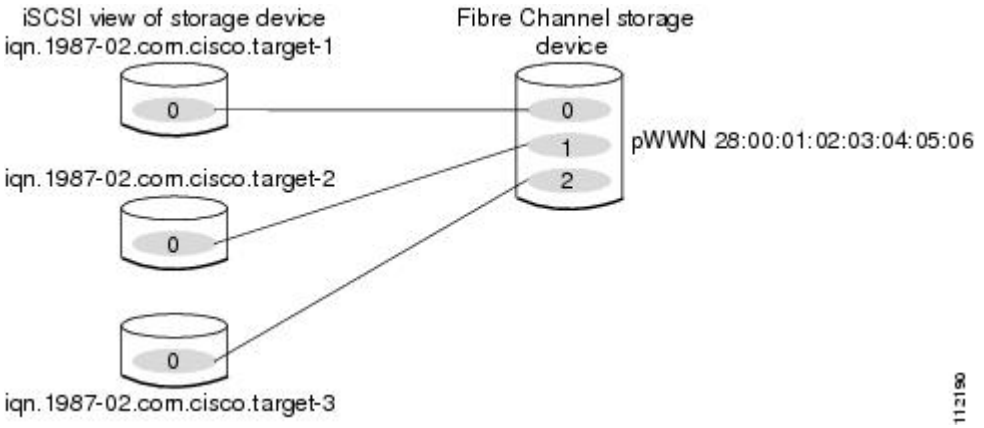



### Example 2

This example maps a subset of LUNs of a Fibre Channel target to three iSCSI virtual targets. Each iSCSI target only has one LUN.

```
iscsi virtual-target name iqn.1987-02.com.cisco.target-1
pWWN 28:00:01:02:03:04:05:06 fc-lun 0 iscsi-lun 0
iscsi virtual-target name iqn.1987-02.com.cisco.target-2
pWWN 28:00:01:02:03:04:05:06 fc-lun 1 iscsi-lun 0
iscsi virtual-target name iqn.1987-02.com.cisco.target-3
pWWN 28:00:01:02:03:04:05:06 fc-lun 2 iscsi-lun 0
```

*Figure 22: Mapping LUNs to an iSCSI Node Name*

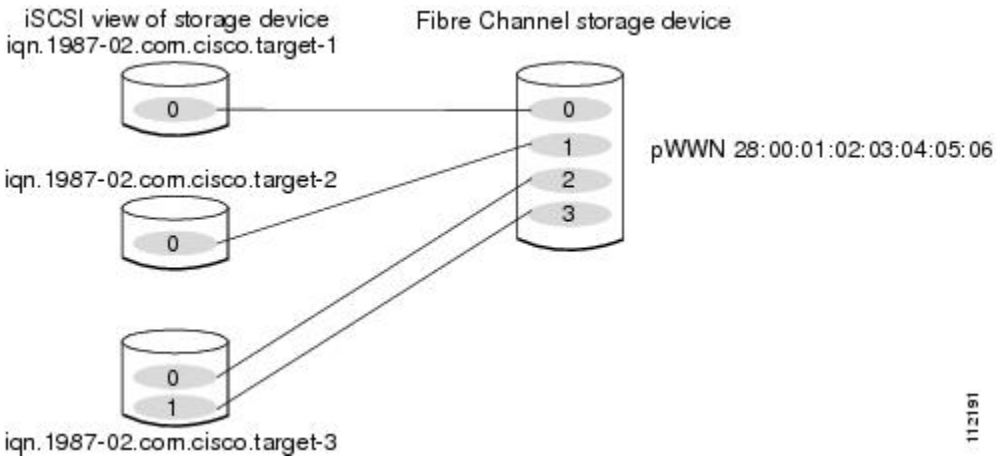


### Example 3

This example maps three subsets of Fibre Channel LUN targets to three iSCSI virtual targets. Two iSCSI targets have one LUN and the third iSCSI target has two LUNs.

```
iscsi virtual-target name iqn.1987-02.com.cisco.target-1
pWWN 28:00:01:02:03:04:05:06 fc-lun 0 iscsi-lun 0
iscsi virtual-target name iqn.1987-02.com.cisco.target-2
pWWN 28:00:01:02:03:04:05:06 fc-lun 1 iscsi-lun 0
iscsi virtual-target name iqn.1987-02.com.cisco.target-3
pWWN 28:00:01:02:03:04:05:06 fc-lun 2 iscsi-lun 0
pWWN 28:00:01:02:03:04:05:06 fc-lun 3 iscsi-lun 1
```

***Figure 23: Mapping LUNs to Multiple iSCSI Node Names***




## Presenting Fibre Channel Targets as iSCSI Targets

The Fibre Channel module with IPS ports or MPS-14/2 module presents physical Fibre Channel targets as iSCSI virtual targets, allowing them to be accessed by iSCSI hosts. The module presents these targets in one of the two ways:

- Dynamic mapping—Automatically maps all the Fibre Channel target devices/ports as iSCSI devices. Use this mapping to create automatic iSCSI target names

  .
- Static mapping—Manually creates iSCSI target devices and maps them to the whole Fibre Channel target port or a subset of Fibre Channel LUNs. With this mapping, you must specify unique iSCSI target names.

Static mapping should be used when iSCSI hosts should be restricted to subsets of LUs in the Fibre Channel targets and/or iSCSI access control is needed (see the iSCSI Access Control section). Also, static mapping allows the configuration of transparent failover if the LUs of the Fibre Channel targets are reachable by redundant Fibre Channel ports (see the Transparent Target Failover section).

**Note** The Fibre Channel module with IPS ports or MPS-14/2 module does not import Fibre Channel targets to iSCSI by default. Either dynamic or static mapping must be configured before the Fibre Channel module with IPS ports or MPS-14/2 module makes Fibre Channel targets available to iSCSI initiators.

# Dynamic Mapping

When you configure dynamic mapping the Fibre Channel module with IPS ports or MPS-14/2 module imports all Fibre Channel targets to the iSCSI domain and maps each physical Fibre Channel target port as one iSCSI target. That is, all LUs accessible through the physical storage target port are available as iSCSI LUs with the same LU number (LUN) as in the physical Fibre Channel target port.

The iSCSI target node name is created automatically using the iSCSI qualified name (IQN) format. The iSCSI qualified name is restricted to a maximum name length of 223 alphanumeric characters and a minimum length of 16 characters.

The Fibre Channel module with IPS ports or MPS-14/2 module creates an IQN formatted iSCSI target node name using the following conventions because the name must be unique in the SAN:

- IPS Gigabit Ethernet ports that are not part of a Virtual Router Redundancy Protocol (VRRP) group or port channel use this format:

  `iqn.1987-05.com.cisco:05.<mgmt-ip-address>.<slot#>-<port#>-<sub-intf#>.<Target-pWWN>`
- IPS ports that are part of a VRRP group use this format:

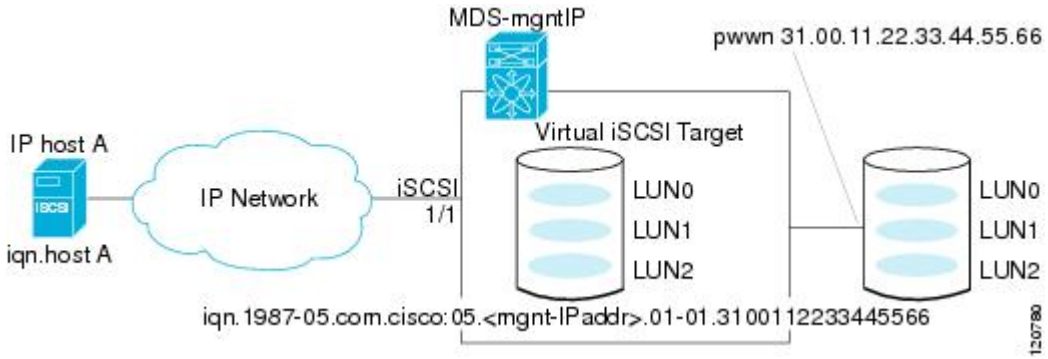
  `iqn.1987-05.com.cisco:05.vrrp-<vrrp-ID#>-<vrrp-IP-addr>.<Target-pWWN>`
- Ports that are part of a port channel use this format:`iqn.1987-02.com.cisco:02.<mgmt-ip-address>.pc-<port-ch-sub-intf#>.<Target-pWWN>`

**Note** If you have configured a switch name, then the switch name is used instead of the management IP address. If you have not configured a switch name, the management IP address is used.

With this convention, each IPS port in a Cisco MDS 9000 Family switch creates a unique iSCSI target node name for the same Fibre Channel target port in the SAN.

For example, if an iSCSI target was created for a Fibre Channel target port with pWWN 31:00:11:22:33:44:55:66 and that pWWN contains LUN 0, LUN 1, and LUN 2, those LUNs would become available to an IP host through the iSCSI target node name iqn.1987-05.com.cisco:05. *MDS_switch_management_IP_address* .

***Figure 24: Dynamic Target Mapping***

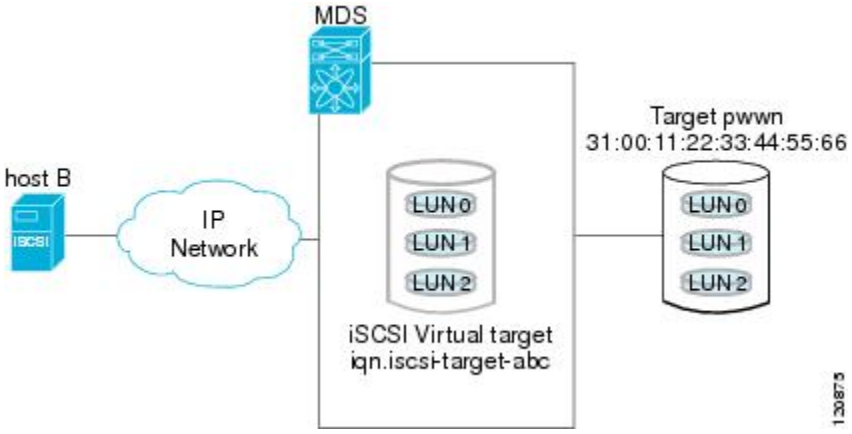


**Note** Each iSCSI initiator may not have access to all targets depending on the configured access control mechanisms (see the iSCSI Access Control section).

# Static Mapping

You can manually (statically) create an iSCSI target by assigning a user-defined unique iSCSI node name to it. The iSCSI qualified name is restricted to a minimum length of 16 characters and a maximum of 223 characters. A statically mapped iSCSI target can either map the whole Fibre Channel target port (all LUNs in the target port mapped to the iSCSI target), or it can contain one or more LUs from a Fibre Channel target port.

***Figure 25: Statically Mapped iSCSI Targets***




**Note** An iSCSI target cannot contain more than one Fibre Channel target port. If you have already mapped the whole Fibre Channel target port, you cannot use the LUN mapping option.

See the iSCSI-Based Access Control section for more information on controlling access to statically mapped targets.

### Advertising Static iSCSI Targets

You can limit the Gigabit Ethernet interfaces through which static iSCSI targets are advertised. By default iSCSI targets are advertised on all Gigabit Ethernet interfaces, subinterfaces, port channel interfaces, and port channel subinterfaces.

# Configuing an Interface to Advertise the iSCSI Virtual Target

To configure a specific interface that should advertise the iSCSI virtual target, follow these steps:

**Procedure**

**Step 1** Advertise the virtual target only on the specified interface. By default, it is advertised on all interfaces in all Fibre Channel module with IPS ports or MPS-14/2 modules.

```
switch(config-iscsi-tgt)# advertise interface GigabitEthernet 2/5
```

**Note**
To advertise the virtual target on multiple interfaces, issue the command for each interface.

**Step 2** Remove this interface from the list of interfaces from which this target is advertised

```
switch(config-iscsi-tgt)# no advertise interface GigabitEthernet 2/5
```

# Presenting iSCSI Hosts as Virtual Fibre Channel Hosts

The Fibre Channel module with IPS ports or MPS-14/2 module connects to the Fibre Channel storage devices on behalf of the iSCSI host to send commands and transfer data to and from the storage devices. These modules use a virtual Fibre Channel N port to access the Fibre Channel storage devices on behalf of the iSCSI host. iSCSI hosts are identified by either iSCSI qualified name (IQN) or IP address.

## Specifying the Initiator Identification Mode

To specify the initiator identification mode, follow these steps:

**Procedure**

**Step 1** Select the iSCSI interface on the switch that identifies all the initiators.

```
switch# config terminal
switch(config)# interface iscsi 4/1
switch(config-if)#
```

**Step 2** Identify the iSCSI initiator based on the IP address.

```
switch(config-if)# switchport initiator id ip-address
```

**Step 3** Identify the iSCSI initiator based on the initiator node name. This is the default behavior.

```
switch(config-if)# switchport initiator id name
```

# iSCSI Virtual Target Configuration Examples

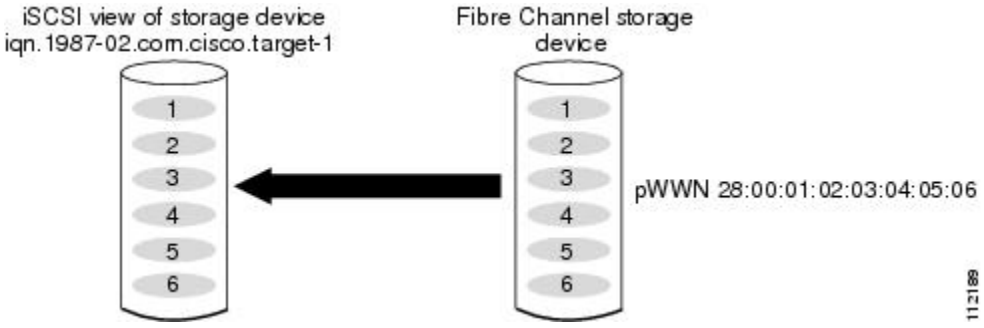
This section provides three examples of iSCSI virtual target configurations

### Assigning the whole Fibre Channel target as an iSCSI virtual target

This example assigns the whole Fibre Channel target as an iSCSI virtual target. All LUNs that are part of the Fibre Channel target are available as part of the iSCSI target.

```
iscsi virtual-target name iqn.1987-02.com.cisco.target-1
pWWN 28:00:01:02:03:04:05:06
```

*Figure 26: Assigning iSCSI Node Names*

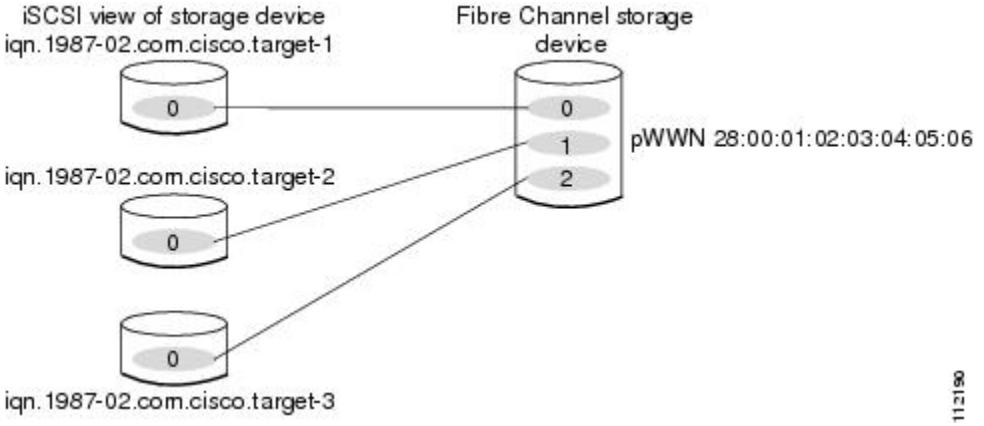



### Mapping a subset of LUNs of a Fibre Channel target to three iSCSI virtual targets

This example maps a subset of LUNs of a Fibre Channel target to three iSCSI virtual targets. Each iSCSI target only has one LUN.

```
iscsi virtual-target name iqn.1987-02.com.cisco.target-1
pWWN 28:00:01:02:03:04:05:06 fc-lun 0 iscsi-lun 0
iscsi virtual-target name iqn.1987-02.com.cisco.target-2
pWWN 28:00:01:02:03:04:05:06 fc-lun 1 iscsi-lun 0
iscsi virtual-target name iqn.1987-02.com.cisco.target-3
pWWN 28:00:01:02:03:04:05:06 fc-lun 2 iscsi-lun 0
```

*Figure 27: Mapping LUNs to an iSCSI Node Name*



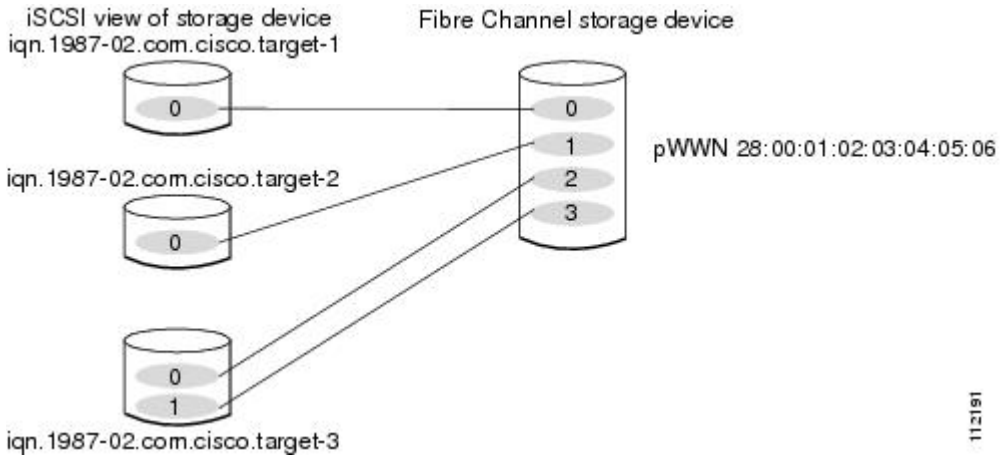

### Mapping three subsets of Fibre Channel LUN targets to three iSCSI virtual targets

This example maps three subsets of Fibre Channel LUN targets to three iSCSI virtual targets. Two iSCSI targets have one LUN and the third iSCSI target has two LUNs.

```
iscsi virtual-target name iqn.1987-02.com.cisco.target-1
pWWN 28:00:01:02:03:04:05:06 fc-lun 0 iscsi-lun 0
iscsi virtual-target name iqn.1987-02.com.cisco.target-2
```

```
pWWN 28:00:01:02:03:04:05:06 fc-lun 1 iscsi-lun 0
iscsi virtual-target name iqn.1987-02.com.cisco.target-3
pWWN 28:00:01:02:03:04:05:06 fc-lun 2 iscsi-lun 0
pWWN 28:00:01:02:03:04:05:06 fc-lun 3 iscsi-lun 1
```

***Figure 28: Mapping LUNs to Multiple iSCSI Node Names***




# Initiator Presentation Modes

Two modes are available to present iSCSI hosts in the Fibre Channel fabric: transparent initiator mode and proxy initiator mode.

- In transparent initiator mode, each iSCSI host is presented as one virtual Fibre Channel host. The benefit of transparent mode is it allows a finer level of Fibre Channel access control configuration (similar to managing a "real" Fibre Channel host). Because of the one-to-one mapping from iSCSI to Fibre Channel, each host can have different zoning or LUN access control on the Fibre Channel storage device.
- In proxy initiator mode, there is only one virtual Fibre Channel host per one IPS port and all iSCSI hosts use that to access Fibre Channel targets. In a scenario where the Fibre Channel storage device requires explicit LUN access control for every host, the static configuration for each iSCSI initiator can be overwhelming. In this case, using the proxy initiator mode simplifies the configuration.

**Note** Enabling proxy initiator mode of an iSCSI interface that is part of an iSLB VRRP group impacts load balancing on the interface. See the Changing iSCSI Interface Parameters and the Impact on Load Balancing section

The Cisco MDS switches support the following iSCSI session limits:

- The maximum number of iSCSI sessions on a switch is 5000.
- The maximum number of iSCSI sessions per IPS port in transparent initiator mode is 500.
- The maximum number of iSCSI sessions per IPS port in proxy initiator mode is 500.
- The maximum number of concurrent sessions an IPS port can create is five (but the total number of sessions that can be supported is 500).

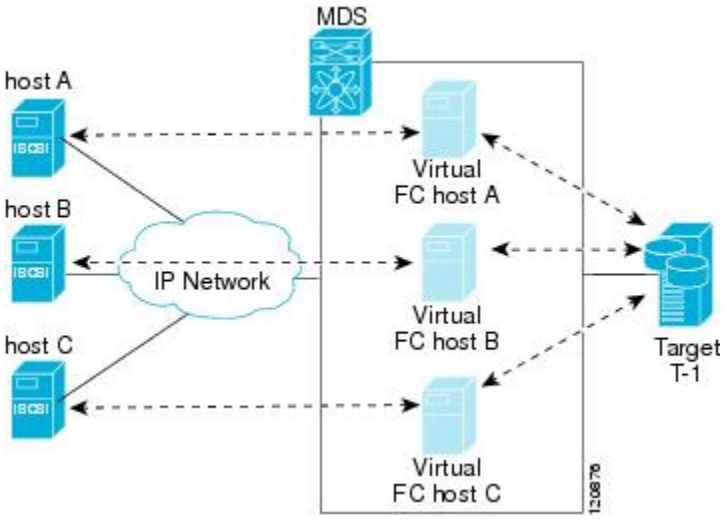
**Note** If more than five iSCSI sessions try to come up simultaneously on a port, the initiator receives a temporary error and later retries to create a session.

### Transparent Initiator Mode

Each iSCSI host is presented as one virtual Fibre Channel host (that is, one Fibre Channel N port). The benefit of transparent mode is it allows a finer-level of Fibre Channel access control configuration. Because of the one-to-one mapping from iSCSI to Fibre Channel, each host can have different zoning or LUN access control on the Fibre Channel storage device.

When an iSCSI host connects to the Fibre Channel module with IPS ports or MPS-14/2 module, a virtual host N port (HBA port) is created for the host (see the below figure). Every Fibre Channel N port requires a unique Node WWN and Port WWN.

***Figure 29: Virtual Host HBA Port***




After the virtual N port is created with the WWNs, a fabric login (FLOGI) is done through the virtual iSCSI interface of the IPS port. After the FLOGI is completed, the virtual N port is online in the Fibre Channel SAN and virtual N port is registered in the Fibre Channel name server. The Fibre Channel module with IPS ports or MPS-14/2 module registers the following entries in the Fibre Channel name server:

- IP address of the iSCSI host in the IP-address field on the name server
- IQN of the iSCSI host in the symbolic-node-name field of the name server
- SCSI_FCP in the FC-4 type field of the name server
- Initiator flag in the FC-4 feature of the name server
- Vendor-specific iSCSI GW flag in the FC-4 type field to identify the N-port device as an iSCSI gateway device in the name server.

When all the iSCSI sessions from the iSCSI host are terminated, the Fibre Channel module with IPS ports or MPS-14/2 modules perform an explicit Fabric logout (FLOGO) to remove the virtual N-port device from the Fibre Channel SAN (this indirectly de-registers the device from the Fibre Channel name server).

For every iSCSI session from the host to the iSCSI virtual target there is a corresponding Fibre Channel session to the real Fibre Channel target. There are three iSCSI hosts, and all three of them connect to the same Fibre Channel target. There is one Fibre Channel session from each of the three virtual Fibre Channel hosts to the target.

### iSCSI Initiator Idle Timeout

iSCSI initiator idle timeout specifies the time for which the virtual Fibre Channel N port is kept idle after the initiator logs out from its last iSCSI session. The default value for this timer is 300 seconds. This is useful to avoid N ports logging in to and logging off of the Fibre Channel SAN as transient failure occurs in the IP network. This helps reduce unnecessary RSCNs being generated in the Fibre Channel SAN.

## Configuring Initiator Idle Timeout

To configure the initiator idle timeout, follow these steps:

### Procedure

**Step 1** Enter the configuration mode.

```
switch# config terminal
switch(config)#
```

**Step 2** Configure the iSCSI initiators to have an idle timeout value of 10 seconds.

```
switch(config)# iscsi initiator idle-timeout 10
```

# WWN Assignment for iSCSI Initiators

An iSCSI host is mapped to an N port's WWNs by one of the following mechanisms:

- Dynamic mapping (default)
- Static mapping

### Dynamic Mapping

With dynamic mapping, an iSCSI host is mapped to a dynamically generated port WWN (pWWN) and node WWN (nWWN). Each time the iSCSI host connects it might be mapped to a different WWN. Use this option if no access control is required on the Fibre Channel target device (because the target device access control is usually configured using the host WWN).

The WWNs are allocated from the MDS switch's WWN pool. The WWN mapping to the iSCSI host is maintained as long as the iSCSI host has at least one iSCSI session to the IPS port. When all iSCSI sessions from the host are terminated and the Fibre Channel module with IPS ports or MPS-14/2 module performs an FLOGO for the virtual N port of the host, the WWNs are released back to the switch's Fibre Channel WWN pool. These addresses are then available for assignment to other iSCSI hosts requiring access to the Fibre Channel Fabric.

The following are three dynamic initiator modes are supported:

- iSCSI—Dynamic initiators are treated as iSCSI initiators and can access dynamic virtual targets and configured iSCSI virtual targets.
- iSLB—Dynamic initiators are treated as iSLB initiators.
- Deny—Dynamic initiators are not allowed to log in to the MDS switch.

iSCSI dynamic mapping is the default mode of operation. This configuration is distributed using CFS

### Static Mapping

With static mapping, an iSCSI host is mapped to a specific pWWN and nWWN. This mapping is maintained in persistent storage and each time the iSCSI host connects, the same WWN mapping is used. This mode is required if you use access control on the target device.

You can implement static mapping in one of two ways:

- User assignment: You can specify your own unique WWN by providing them during the configuration process.
- System assignment: You can request that the switch provide a WWN from the switch's Fibre Channel WWN pool and keep the mapping in its configuration.

We recommend using the **system-assign** option. If you manually assign a WWN, you must ensure its uniqueness (see the Cisco Fabric Manager Fabric Configuration Guide and Cisco MDS 9000 Family NX-OS Fabric Configuration Guide for more information). You should not use any previously assigned WWNs.

### Making the Dynamic iSCSI Initiator WWN Mapping Static

After a dynamic iSCSI initiator has already logged in, you may decide to permanently keep the automatically assigned nWWN/pWWN mapping so this initiator uses the same mapping the next time it logs in.

You can convert a dynamic iSCSI initiator to static iSCSI initiator and make its WWNs persistent.



**Note** You cannot convert a dynamic iSCSI initiator to a static iSLB initiator or a dynamic iSLB initiator to a static iSCSI initiator.



**Note** Making the dynamic pWWNs static after the initiator is created is supported only through the CLI, not through Device Manager or Fabric Manager. In Fabric Manager or Device Manager, you must delete and then recreate this initiator to have the pWWNs static.

### Checking for WWN Conflicts

WWNs assigned to static iSCSI initiators by the system can be inadvertently returned to the system when an upgrade fails or you downgrade the system software (manually booting up an older Cisco MDS SAN-OS release without using the **install all** command). In these instances, the system can later assign those WWNs to other iSCSI initiators (dynamic or static) and cause conflicts.

You can address this problem by checking for and removing any configured WWNs that belong to the system whenever such scenarios occur.

## Configuring Dynamic Mapping Using the Name Option for an iSCSI Initiator

To configure dynamic mapping (using the **name** option) for an iSCSI initiator, follow these steps:

### Procedure

**Step 1** Enter configuration mode.

```
switch# config terminal
switch(config)#
```

**Step 2** Specify iSLB dynamic initiator mode.

```
switch(config)# iscsi dynamic initiator islb
```

**Step 3** Prohibit dynamic initiators from logging on to the MDS switch

```
switch(config)# iscsi dynamic initiator deny
```

**Step 4** Revert to iSCSI mode (default).

```
switch(config)# no iscsi dynamic initiator islb
```

## Configuring Static Mapping Using the Name Option for an iSCSI Initiator

To configure static mapping (using the **name** option) for an iSCSI initiator, follow these steps:

### Procedure

**Step 1** Enter configuration mode.

```
switch# config terminal
switch(config)#
```

**Step 2** Configure an iSCSI initiator using the iSCSI name of the initiator node. The maximum name length is restricted to 223 alphanumeric characters. The minimum length is 16.

```
switch(config)# iscsi initiator name iqn.1987-02.com.cisco.initiator
switch(config-iscsi-init)#
```

**Step 3** Delete the configured iSCSI initiator.

```
switch(config)# no iscsi initiator name iqn.1987-02.com.cisco.initiator
```

## Configuring Static Mapping Using the IP-Address Option for an iSCSI Initiator

To configure static mapping (using the **ip-address** option) for an iSCSI initiator, follow these steps:

### Procedure

**Step 1** Enter configuration mode.

```
switch# config terminal
switch(config)#
```

**Step 2** Configure an iSCSI initiator using the IPv4 address of the initiator node.

```
switch(config)# iscsi initiator ip-address 10.50.0.0
switch(config-iscsi-init)#
```

**Step 3** Configure an iSCSI initiator using the IPv6 unicast address of the initiator node.

```
switch(config)# iscsi initiator ip-address 2001:0DB8:800:200C::417A
switch(config-iscsi-init)#
```

**Step 4** (Optional) Deletes the configured iSCSI initiator.

```
switch(config)# no iscsi initiator ip-address 2001:0DB8:800:200C::417A
```

## Assigning the WWN for an iSCSI Initiator

To assign the WWN for an iSCSI initiator, follow these steps:

### Before you begin

If the system-assign option is used to configure WWNs for an iSCSI initiator, when the configuration is saved to an ASCII file the system-assigned WWNs are also saved. Subsequently if you perform a write erase, you must manually delete the WWN configuration from the ASCII file. Failing to do so can cause duplicate WWN assignments if the ASCII configuration file is reapplied on the switch.

### Procedure

**Step 1** Use the switch's WWN pool to allocate the nWWN for this iSCSI initiator and keeps it persistent.

```
switch(config-iscsi-init)# static nWWN system-assign
```

**Step 2** Assign the user provided WWN as the nWWN for the iSCSI initiator. You can only specify one nWWN for each iSCSI node.

```
switch(config-iscsi-init)# static nWWN 20:00:00:05:30:00:59:11
```

**Step 3** Use the switch's WWN pool to allocate two pWWNs for this iSCSI initiator and keeps them persistent. The range is from 1 to 64.

```
switch(config-iscsi-init)# static pWWN system-assign 2
```

**Step 4** Assign the user provided WWN as the pWWN for the iSCSI initiator.

```
switch(config-iscsi-init)# static pWWN 21:00:00:20:37:73:3b:20
```

## Saving the Automatically Assigned nWWN/pWWN Mapping

To permanently keep the automatically assigned nWWN/pWWN mapping, follow these steps:

### Procedure

**Step 1** Enter configuration mode.

```
switch# config terminal
switch(config)#
```

**Step 2** Save the nWWN and pWWNs that have automatically been assigned to the iSCSI initiator whose name is specified.

```
switch(config)# iscsi save-initiator name iqn.1987-02.com.cisco.initiator
```

**Step 3** Save the nWWN and pWWNs that have automatically been assigned to the iSCSI initiator whose IPv4 address is specified.

```
switch(config)# iscsi save-initiator ip-address 10.10.100.11
```

**Step 4** Save the nWWN and pWWNs that have automatically been assigned to the iSCSI initiator whose IPv6 unicast address is specified.

```
switch(config)# iscsi save-initiator ip-address 2001:0DB8:800:200C::417A
```

**Step 5** Save the nWWN and pWWNs that have automatically been assigned to all the initiators.

```
switch(config)# iscsi save-initiator
```

**Step 6** Return to EXEC mode.

```
switch(config)# exit

switch#
```

**Step 7** Save the nWWN/pWWN mapping configuration across system reboots.

```
switch# copy running-config startup-config
```

## Checking and Removing WWN Conflicts

To check for and remove WWN conflicts, follow these steps:

### Procedure

**Step 1** Enter configuration mode.

```
switch# config terminal
switch(config)#
```

**Step 2** Check for WWN conflicts.

```
switch(config)# iscsi duplicate-wwn-check

List of Potential WWN Conflicts:
--------------------------------
Node : iqn.test-local-nwwn:1-local-pwwn:1
nWWN : 22:03:00:0d:ec:02:cb:02
pWWN : 22:04:00:0d:ec:02:cb:02
```

**Step 3** Enter iSCSI initiator configuration mode for the initiator named iqn.test-local-nwwn:1-local-pwwn:1.

```
switch(config)# iscsi initiator name iqn.test-local-nwwn:1-local-pwwn:1
```

**Step 4** Remove a conflicting nWWN.

```
switch(config-iscsi-init)# no static nWWN 22:03:00:0d:ec:02:cb:02
```

**Step 5** Remove a conflicting pWWN.

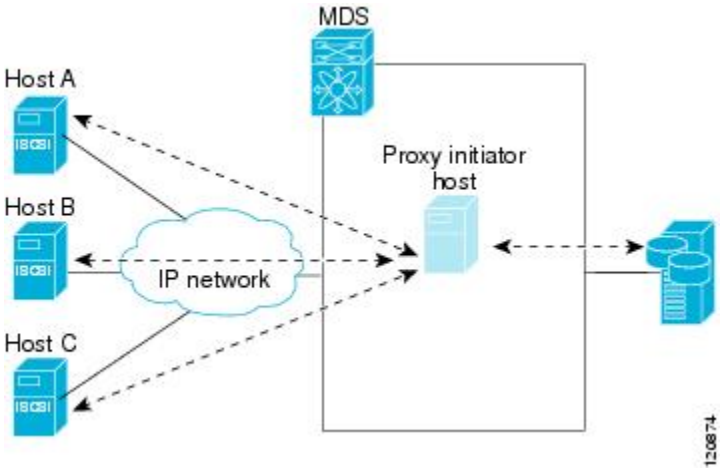
```
switch(config-iscsi-init)# no static pWWN 22:04:00:0d:ec:02:cb:02
```

# Proxy Initiator Mode

In the event that the Fibre Channel storage device requires explicit LUN access control for every host use the transparent initiator mode (presenting one iSCSI host as one Fibre Channel host). Every iSCSI host has to be configured statically. This can mean several configuration tasks for each iSCSI host. If you do not need explicit LUN access control, using the proxy initiator mode simplifies the configuration.

In this mode, only one virtual host N port (HBA port) is created per IPS port. All the iSCSI hosts connecting to that IPS port will be multiplexed using the same virtual host N port. This mode simplifies the task of statically binding WWNs. LUN mapping and assignment on the Fibre Channel storage array must be configured to allow access from the proxy virtual N port's pWWN for all LUNs used by each iSCSI initiator that connects through this IPS port. The LUN is then assigned to each iSCSI initiator by configuring iSCSI virtual targets (see the Static Mapping section) with LUN mapping and iSCSI access control.

***Figure 30: Multiplexing IPS Ports***




Proxy initiator mode can be configured on a per IPS port basis, in which case only iSCSI initiators terminating on that IPS port will be in this mode.

When an IPS port is configured in proxy-initiator mode, fabric login (FLOGI) is done through the virtual iSCSI interface of the IPS port. After the FLOGI is completed, the proxy-initiator virtual N port is online in the Fibre Channel fabric and virtual N port is registered in the Fibre Channel name server. The Fibre Channel module with IPS ports or MPS-14/2 module registers the following entries in the Fibre Channel name server:

- iSCSI interface name iSCSI slot /port is registered in the symbolic-node-name field of the name server
- SCSI_FCP in the FC-4 type field of the name server
- Initiator flag in the FC-4 feature of the name server
- Vendor specific flag (iscsi-gw) in the FC-4 type field to identify the N-port device as an iSCSI gateway device in the name server

Similar to transparent initiator mode, the user can provide a pWWN and nWWN or request a system assigned WWN for the proxy initiator N port.

**Note** Enabling the proxy initiator mode of an iSCSI interface that is part of an iSLB VRRP group impacts load balancing on the interface. See the Changing iSCSI Interface Parameters and the Impact on Load Balancing section.

**Note** When an interface is in proxy initiator mode, you can only configure Fibre Channel access control (zoning) based on the iSCSI interface's proxy N port attributes—the WWN pairs or the FC ID. You cannot configure zoning using iSCSI attributes such as IP address or IQN of the iSCSI initiator. To enforce initiator-based access control, use iSCSI based access control

## Configuring Proxy Initiator

To configure the proxy initiator, follow these steps:

### Procedure

**Step 1** Enter configuration mode.

```
switch# config terminal
switch(config)#
```

**Step 2** Select the iSCSI interface on the switch that initiators will connect to.

```
switch(config)# interface iscsi 4/1
switch(config-if)#
```

**Step 3** Configure the proxy initiator mode with system-assignment nWWN and pWWN.

```
switch(config-if)# switchport proxy-initiator
```

**Step 4** (Optional) Disable the proxy initiator mode.

```
switch(config-if)# no switchport proxy-initiator
```

**Step 5** (Optional) Configure the proxy initiator mode using the specified WWNs.

```
switch(config-if)# switchport proxy-initiator nWWN 11:11:11:11:11:11:11:11 pwwn 22:22:22:22:22:22:22:22
```

**Step 6** (Optional) Disable the proxy initiator mode.

```
switch(config-if)# no switchport proxy-initiator nWWN 11:11:11:11:11:11:11:11 pwwn
22:22:22:22:22:22:22:22
```

# VSAN Membership for iSCSI

VSAN membership can be configured for an iSCSI interface, called the portVSAN. All the iSCSI devices that connect to this interface automatically become members of this VSAN, if it is not explicitly configured in a VSAN. The default port VSAN of an iSCSI interface is VSAN 1. Similar to Fibre Channel devices, iSCSI devices have two mechanisms by which VSAN membership can be defined.

- iSCSI host: VSAN membership to iSCSI host. (This method takes precedent over the iSCSI interface).
- iSCSI interface: VSAN membership to iSCSI interface. (All iSCSI hosts connecting to this iSCSI interface inherit the interface VSAN membership if the host is not configured in any VSAN by the iSCSI host method).

Individual iSCSI hosts can be configured to be in a specific VSAN. The specified VSAN overrides the iSCSI interface VSAN membership.

## Assigning VSAN Membership for iSCSI Hosts

To assign VSAN membership for iSCSI hosts, follow these steps:

### Before you begin

When an initiator is configured in any other VSAN (other than VSAN 1), for example VSAN 2, the initiator is automatically removed from VSAN 1. If you also want it to be present in VSAN 1, you must explicitly configure the initiator in VSAN 1.

### Procedure

**Step 1** Enter configuration mode.

```
switch# config terminal
switch(config)#
```

**Step 2** Configure an iSCSI initiator.

```
switch(config)# iscsi initiator name iqn.1987-02.com.cisco.initiator
switch(config-iscsi-init)#
```

**Step 3** Assign the iSCSI initiator node to a specified VSAN.

```
switch(config-iscsi-init)# vsan 3
```

**Note**
You can assign this host to one or more VSANs.

**Step 4** Remove the iSCSI node from the specified VSAN.

```
switch(config-iscsi-init)# no vsan 5
```

## Configuring Default Port VSAN for iSCSI Interfaces

VSAN membership can be configured for an iSCSI interface, called the *port VSAN* . All the iSCSI devices that connect to this interface automatically become members of this VSAN, if it is not explicitly configured in a VSAN. In other words, the port VSAN of an iSCSI interface is the default VSAN for all dynamic iSCSI initiators. The default port VSAN of an iSCSI interface is VSAN 1.

To change the default port VSAN for an iSCSI interface, follow these steps:

### Before you begin

Changing the VSAN membership of an iSCSI interface that is part of an iSLB VRRP group impacts load balancing on the interface. See the "Changing iSCSI Interface Parameters and the Impact on Load Balancing" section.

### Procedure

**Step 1** Enter configuration mode.

```
switch# config terminal
switch(config)#
```

**Step 2** Configure VSAN membership for iSCSI interfaces.

```
switch(config)# iscsi interface vsan-membership
```

**Step 3** Configure the database for a VSAN. Application specific VSAN parameters cannot be configured from this prompt.

```
switch(config)# vsan database
switch(config-vsan-db)#
```

**Step 4** Assign the membership of the iscsi 2/1 interface to the specified VSAN (VSAN 2)

```
switch(config - vsan-db)# vsan 2 interface iscsi 2/1
```

**Step 5** Revert to using the default VSAN as the port VSAN of the iSCSI interface.

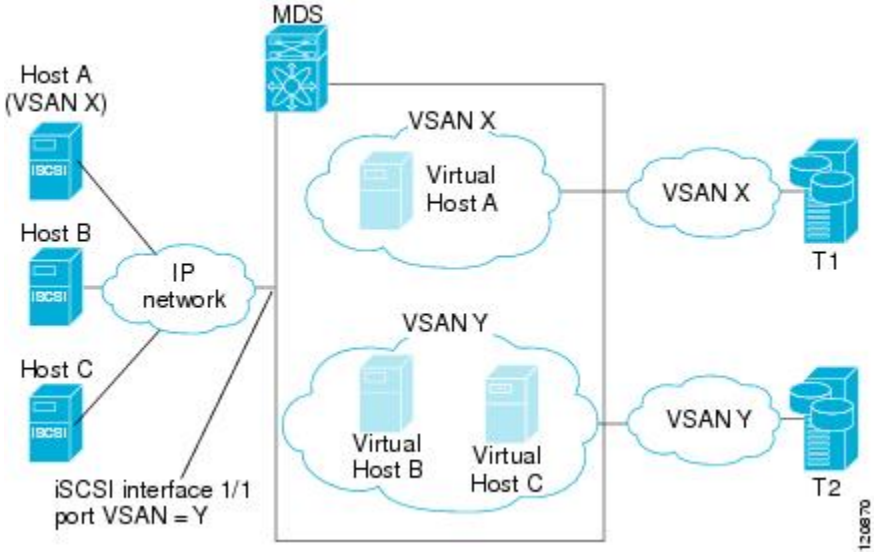
```
switch(config - vsan-db)# no vsan 2 interface iscsi 2/1
```

## VSAN Membership for iSCSI Devices

Example of VSAN membership for iSCSI devices are as follows:

- SCSI interface 1/1 is a member of VSAN Y.
- iSCSI initiator host A has explicit VSAN membership to VSAN X.
- Three iSCSI initiators (host A, host B, and host C) C connect to iSCSI interface 1/1.

*Figure 31: VSAN Membership for iSCSI Interfaces*




Host A's virtual Fibre Channel N port will be added to VSAN X because of explicit membership for the initiator. The virtual host-B and host-C N ports do not have any explicit membership configuration so they will inherit the iSCSI interface VSAN membership and be part of VSAN Y.

### Advanced VSAN Membership for iSCSI Hosts

An iSCSI host can be a member of multiple VSANs. In this case, multiple virtual Fibre Channel hosts are created, one in each VSAN in which the iSCSI host is a member. This configuration is useful when certain resources such as Fibre Channel tape devices need to be shared among different VSANs.

## iSCSI Access Control

Two methods of access control are available for iSCSI devices. Depending on the initiator mode used to present the iSCSI hosts in the Fibre Channel fabric, either or both of the access control methods can be used.

- Fiber Channel zoning-based access control: Fibre Channel zoning has been extended to support iSCSI devices, and this extension has the advantage of having a uniform, flexible access control mechanism across the whole SAN. In the case of iSCSI, multiple iSCSI devices may be connected behind an iSCSI interface. Interface-based zoning may not be useful because all iSCSI devices behind the interface will automatically be within the same zone.
- iSCSI ACL-based access control: iSCSI-based access control is applicable only if static iSCSI virtual targets are created. For a static iSCSI target, you can configure a list of iSCSI initiators that are allowed to access the targets. By default, static iSCSI virtual targets are not accessible to any iSCSI host.

Depending on the initiator mode used to present the iSCSI hosts in the Fibre Channel fabric, either or both the access control mechanisms can be used.

The following topics are included in this section:

- Fibre Channel Zoning-Based Access Control
- iSCSI-Based Access Control
- Enforcing Access Control

## Fibre Channel Zoning-Based Access Control

Cisco SAN-OS Release 3.x and NX-OS Release 4.1(1b) VSAN and zoning concepts have been extended to cover both Fibre Channel devices and iSCSI devices. Zoning is the standard access control mechanism for Fibre Channel devices, which is applied within the context of a VSAN. Fibre Channel zoning has been extended to support iSCSI devices, and this extension has the advantage of having a uniform, flexible access control mechanism across the whole SAN.

Common mechanisms for identifying members of a Fibre Channel zone are the following:

- Fibre Channel device pWWN.
- Interface and switch WWN. Device connecting via that interface is within the zone.

See the Cisco Fabric Manager Fabric Configuration Guide and Cisco MDS 9000 Family NX-OS Fabric Configuration Guide for details on Fibre Channel zoning.

In the case of iSCSI, multiple iSCSI devices may be connected behind an iSCSI interface. Interface-based zoning may not be useful because all the iSCSI devices behind the interface will automatically be within the same zone.

In transparent initiator mode (where one Fibre Channel virtual N port is created for each iSCSI host as described in the Transparent Initiator Mode), if an iSCSI host has static WWN mapping then the standard Fibre Channel device pWWN-based zoning membership mechanism can be used.

Zoning membership mechanism has been enhanced to add iSCSI devices to zones based on the following:

- IPv4 address/subnet mask
- IPv6 address/prefix length
- iSCSI qualified name (IQN)
- Symbolic-node-name (IQN)

For iSCSI hosts that do not have a static WWN mapping, the feature allows the IP address or iSCSI node name to be specified as zone members. Note that iSCSI hosts that have static WWN mapping can also use these features. IP address based zone membership allows multiple devices to be specified in one command by providing the subnet mask.

**Note** In proxy initiator mode, all iSCSI devices connecting to an IPS port gain access to the Fibre Channel fabric through a single virtual Fibre Channel N port. Zoning based on the iSCSI node name or IP address will not have any effect. If zoning based on pWWN is used, then all iSCSI devices connecting to that IPS port will be put in the same zone. To implement individual initiator access control in proxy initiator mode, configure an iSCSI ACL on the virtual target (see the iSCSI-Based Access Control section).

### *Adding iSCSI Initiator to the Zone Database*

To add an iSCSI initiator to the zone database, follow these steps:

#### Procedure

**Step 1** Enter configuration mode.

```
switch# config terminal
switch(config)#
```

**Step 2** Create a zone name for the iSCSI devices in the Fibre Channel module with IPS ports or MPS-14/2 module to be included.

```
switch(config)# zone name iSCSIzone vsan 1
switch(config-zone)
```

**Step 3** Assign an iSCSI node name-based membership into a zone.

```
switch(config-zone)# member symbolic-nodename iqn.1987-02.com.cisco.initiator1
```

**Step 4** (Optional) Delete the specified device from a zone.

```
switch(config-zone)# no member symbolic-nodename iqn.1987-02.com.cisco.init1
```

**Step 5** Assign an iSCSI IPv4 address-based membership into a zone.

```
switch(config-zone)# member ip-address 10.50.1.1
```

**Step 6** (Optional) Delete the specified device from a zone.

```
switch(config-zone)# no member ip-address 10.50.1.1
```

**Step 7** Assign an iSCSI IPv6 address-based membership into a zone.

```
switch(config-zone)# member ipv6-address 2001:0DB8:800:200C::417A
```

**Step 8** Delete the specified device from a zone.

```
switch(config-zone)# no member ipv6-address 2001:0DB8:800:200C::417A
```

**Step 9** Assign an iSCSI port WWN-based membership into a zone.

```
switch(config-zone)# member pwwn 20:00:00:05:30:00:59:11
```

**Step 10** Delete the device identified by the port WWN from a zone.

```
switch(config-zone)# no member pwwn 20:00:00:05:30:00:59:11
```

## iSCSI-Based Access Control

iSCSI-based access control is applicable only if static iSCSI virtual targets are created (see the Static Mapping section). For a static iSCSI target, you can configure a list of iSCSI initiators that are allowed to access the targets.

By default, static iSCSI virtual targets are not accessible to any iSCSI host. You must explicitly configure accessibility to allow an iSCSI virtual target to be accessed by all hosts. The initiator access list can contain one or more initiators. The iSCSI initiator can be identified by one of the following mechanisms:

- iSCSI node name
- IPv4 address and subnet
- IPv6 address

To configure access control in iSCSI follow these steps:

### Before you begin

For a transparent mode iSCSI initiator, if both Fibre Channel zoning and iSCSI ACLs are used, then for every static iSCSI target that is accessible to the iSCSI host, the initiator's virtual N port should be in the same Fibre Channel zone as the Fibre Channel target.

### Procedure

**Step 1** Enter configuration mode.

```
switch# config terminal
switch(config)#
```

**Step 2** Create the iSCSI target name iqn.1987-02.com.cisco.initiator.

```
switch(config)# iscsi virtual-target name iqn.1987-02.com.cisco.initiator
switch(config-iscsi-tgt)#
```

**Step 3** Map a virtual target node to a Fibre Channel target.

```
switch(config-iscsi-tgt)# pWWN 26:00:01:02:03:04:05:06
switch(config-iscsi-tgt)#
```

**Step 4** Allow the specified iSCSI initiator node to access this virtual target. You can issue this command multiple times to allow multiple initiators.

```
switch(config-iscsi-tgt)# initiator iqn.1987-02.com.cisco.initiator1 permit
```

**Step 5** (Optional) Prevent the specified initiator node from accessing virtual targets.

```
switch(config-iscsi-tgt)# no initiator iqn.1987-02.com.cisco.initiator1 permit
```

**Step 6** Prevent the specified IPv4 address from accessing virtual targets.

```
switch(config-iscsi-tgt)# no initiator ip address 10.50.1.1 permit
```

**Step 7** Allow all initiators in this IPv4 subnetwork (10.50.1/24) to access this virtual target.

```
switch(config-iscsi-tgt)# initiator ip address 10.50.1.0 255.255.255.0 permit
```

**Step 8** Prevent all initiators in this IPv4 subnetwork from accessing virtual targets.

```
switch(config-iscsi-tgt)# no initiator ip address 10.50.1.0 255.255.255.0 permit
```

**Step 9** Allow all initiators in this IPv6 subnetwork (2001:0DB8:800:200C::/64) to access this virtual target.

```
switch(config-iscsi-tgt)# initiator ip address 2001:0DB8:800:200C::/64 permit
```

**Step 10** Prevent all initiators in this IPv6 subnetwork from accessing virtual targets.

```
switch(config-iscsi-tgt)# no initiator ip address 2001:0DB8:800:200C::/64 permit
```

**Step 11** Allow all initiator nodes to access this virtual target.

```
switch(config-iscsi-tgt)# all-initiator-permit
```

**Step 12** Prevent any initiator from accessing virtual targets (default).

```
switch(config-iscsi-tgt)# no all-initiator-permit
```

## Enforcing Access Control

Fibre Channel module with IPS ports and MPS-14/2 modules use both iSCSI and Fibre Channel zoning-based access control lists to enforce access control. Access control is enforced both during the iSCSI discovery phase and the iSCSI session creation phase. Access control enforcement is not required during the I/O phase because the Fibre Channel module with IPS ports or MPS-14/2 module is responsible for the routing of iSCSI traffic to Fibre Channel.

- iSCSI discovery phase—When an iSCSI host creates an iSCSI discovery session and queries for all iSCSI targets, the Fibre Channel module with IPS ports or MPS-14/2 module returns only the list of iSCSI targets this iSCSI host is allowed to access based on the access control policies discussed in the previous section. The Fibre Channel module with IPS ports or MPS-14/2 module does this by querying the Fibre Channel name server for all the devices in the same zone as the initiator in all VSANs. It then filters out the devices that are initiators by looking at the FC4-feature field of the FCNS entry. (If a device does not register as either initiator or target in the FC4-feature field, the Fibre Channel module with IPS ports or MPS-14/2 module will advertise it). It then responds to the iSCSI host with the list of targets. Each will have either a static iSCSI target name that you configure or a dynamic iSCSI target name that the Fibre Channel module with IPS ports or MPS-14/2 module creates for it (see the Dynamic Mapping section).
- iSCSI session creation—When an IP host initiates an iSCSI session, the Fibre Channel module with IPS ports or MPS-14/2 module verifies if the specified iSCSI target (in the session login request) is allowed by both the access control mechanisms described in the iSCSI-Based Access Control section.

If the iSCSI target is a static mapped target, the Fibre Channel module with IPS ports or MPS-14/2 module verifies if the iSCSI host is allowed within the access list of the iSCSI target. If the IP host does not have access, its login is rejected. If the iSCSI host is allowed, it validates if the virtual Fibre Channel N port used by the iSCSI host and the Fibre Channel target mapped to the static iSCSI virtual target are in the same Fibre Channel zone.

If the iSCSI target is an autogenerated iSCSI target, then the Fibre Channel module with IPS ports or MPS-14/2 module extracts the WWN of the Fibre Channel target from the iSCSI target name and verifies if the initiator and the Fibre Channel target is in the same Fibre Channel zone or not. If they are, then access is allowed.

The Fibre Channel module with IPS ports or MPS-14/2 module uses the Fibre Channel virtual N port of the iSCSI host and does a zone-enforced name server query for the Fibre Channel target WWN. If the FC ID is returned by the name server, then the iSCSI session is accepted. Otherwise, the login request is rejected.

### iSCSI Session Authentication

The Fibre Channel module with IPS ports or MPS-14/2 module supports the iSCSI authentication mechanism to authenticate the iSCSI hosts that request access to the storage devices. By default, the Fibre Channel module with IPS ports or MPS-14/2 modules allow CHAP or None authentication of iSCSI initiators. If authentication is always used, you must configure the switch to allow only CHAP authentication.

For CHAP user name or secret validation, you can use any method supported and allowed by the Cisco MDS AAA infrastructure. AAA authentication supports a RADIUS, TACACS+, or local authentication device. See the Cisco Fabric Manager Security Configuration Guide.

The **aaa authentication iscsi** command enables AAA authentication for the iSCSI host and specifies the method to use. See Cisco MDS 9000 Family NX-OS Security Configuration Guide

# Configuring AAA Authentication for iSCSI User

To configure AAA authentication for an iSCSI user, follow these steps:

### Procedure

**Step 1** Enter configuration mode.

```
switch# config terminal
switch(config)#
```

**Step 2** Use RADIUS servers that are added in the group called RadServerGrp for the iSCSI CHAP authentication.

```
switch(config)# aaa authentication iscsi default group RadServerGrp
```

**Step 3** Use TACACS+ servers that are added in the group called TacServerGrp for the iSCSI CHAP authentication.

```
switch(config)# aaa authentication iscsi default group TacServerGrp
```

**Step 4** Use the local password database for iSCSI CHAP authentication.

```
switch(config)# aaa authentication iscsi default local
```

# Configuring Authentication Mechanism for iSCSI

To configure the authentication mechanism for iSCSI, follow these steps:.

### Procedure

**Step 1** Enter configuration mode.

```
switch# config terminal
switch(config)#
```

**Step 2** Configure CHAP as the default authentication mechanism globally for the Cisco MDS switch. CHAP authentication is required for all iSCSI sessions.

```
switch(config)# iscsi authentication chap
```

# Configuring Authentication Mechanism for iSCSI sessions on an Interface

To configure the authentication mechanism for iSCSI sessions to a particular interface, follow these steps:

### Procedure

**Step 1** Enter configuration mode.

```
switch# config terminal
switch(config)#
```

**Step 2** Select the Gigabit Ethernet interface.

```
switch(config)# interface GigabitEthernet 2/1.100
switch(config-if)#
```

**Step 3** Specify that no authentication is required for iSCSI sessions to the selected interface.

```
switch(config-if)# iscsi authentication none
```

# Configuring Local Authentication

To configure iSCSI users for local authentication, follow these steps:

### Procedure

**Step 1** Enter configuration mode

```
switch# config terminal
switch(config)#
```

**Step 2** Configure a user name (iscsiuser) and password (ffsffsfsffs345353554535) in the local database for iSCSI login authentication.

```
switch(config)# username iscsiuser password ffsffsfsffs345353554535 iscsi
```

# Restricting iSCSI Initiator Authentication

By default, the iSCSI initiator can use any user name in the RADIUS server or in the local database in authenticating itself to the Fibre Channel module with IPS ports or MPS-14/2 module (the CHAP user name is independent of the iSCSI initiator name). The Fibre Channel module with IPS ports or MPS-14/2 module allows the initiator to log in as long as it provides a correct response to the CHAP challenge sent by the switch. This can be a problem if one CHAP user name and password has been compromised.

To restrict an initiator to use a specific user name for CHAP authentication, follow these steps:

### Procedure

**Step 1** Enter configuration mode.

```
switch# config terminal
switch(config)#
```

**Step 2** Enter the configuration submode for the initiator iqn.1987-02.com.cisco.init.

```
switch(config)# iscsi initiator name iqn.1987-02.com.cisco.init
switch(config-iscsi-init)#
```

**Step 3** Restrict the initiator iqn.1987-02.com.cisco.init to only authenticate using user1 as its CHAP user name.

```
switch(config-iscsi-init)# username user1
```

**Note**
Be sure to define user1 as an iSCSI user in the local AAA database or the RADIUS server.

# Configuring Mutual CHAP Authentication

The Fibre Channel module with IPS ports or MPS-14/2 module supports a mechanism by which the iSCSI initiator can authenticate the Cisco MDS switch's iSCSI target during the iSCSI login phase. This authentication is available in addition to the Fibre Channel module with IPS ports or MPS-14/2 module authentication of the iSCSI initiator.

In addition to the Fibre Channel module with IPS ports or MPS-14/2 module authentication of the iSCSI initiator, the Fibre Channel module with IPS ports or MPS-14/2 module also supports a mechanism for the iSCSI initiator to authenticate the Cisco MDS switch's iSCSI target during the iSCSI login phase. This authentication requires the user to configure a user name and password for the switch to present to the iSCSI initiator. The provided password is used to calculate a CHAP response to a CHAP challenge sent to the IPS port by the initiator.

To configure a global iSCSI target user name and password to be used by the switch to authenticate itself to an initiator, follow these steps:

### Procedure

**Step 1** Enter configuration mode.

```
switch# config terminal
switch(config)#
```

**Step 2** Configure the switch user account (testuser) along with a password (abc123) specified in clear text (default) for all initiators. The password is limited to 128 characters.

```
switch(config)# iscsi authentication username testuser password abc123
```

**Step 3** Configures the switch user account (user1) along with the encrypted password specified by 7 (!@*asdsfsdfjh!@df) for all initiators.

```
switch(config)# iscsi authentication username user1 password 7!@*asdsfsdfjh!@df
```

**Step 4** Configure the switch user account (user1) along with a password (abcd12AAA) specified in clear text (indicated by 0—default) for all initiators. The password is limited to 128 characters.

```
switch(config)# iscsi authentication username user1 password 0 abcd12AAA
```

**Step 5** Remove the global configuration for all initiators.

```
switch(config)# no iscsi authentication username testuser
```

# Configuring Per-Initiator iSCSI Target User Name and Password for Authentication

To configure a per-initiator iSCSI target's user name and password used by the switch to authenticate itself to an initiator, follow these steps:

### Before you begin

Use the **show running-config** and the **show iscsi global** commands to display the global configuration. Use the **show running-config** and the **show iscsi initiator configured** commands to display the initiator specific configuration.(See the Displaying iSCSI Information section for command output examples).

### Procedure

**Step 1** Enter configuration mode.

```
switch# config terminal
switch(config)#
```

**Step 2** Configure an iSCSI initiator using the iSCSI name of the initiator node.

```
switch(config)# iscsi initiator name iqn.1987-02.com.cisco.initiator
switch(config-iscsi-init)#
```

**Step 3** Configure the switch user account (testuser) along with a password (abcd12AAA) specified in clear text (default). The password is limited to 128 characters.

```
switch(config-iscsi-init)# mutual-chap username testuser password abcd12AAA
```

**Step 4** Configure the switch user account (user1) along with the encrypted password specified by 7 (!@*asdsfsdfjh!@df).

```
switch(config-iscsi-init)# mutual-chap username user1 password 7!@*asdsfsdfjh!@df
```

**Step 5** Remove the switch authentication configuration.

```
switch(config-iscsi-init)# no mutual-chap username testuser
```

# iSCSI Immediate Data and Unsolicited Data Features

Cisco MDS switches support the iSCSI immediate data and unsolicited data features if requested by the initiator during the login negotiation phase. Immediate data is iSCSI write data contained in the data segment of an iSCSI command protocol data unit (PDU), such as combining the write command and write data together in one PDU. Unsolicited data is iSCSI write data that an initiator sends to the iSCSI target, such as an MDS switch, in an iSCSI data-out PDU without having to receive an explicit ready to transfer (R2T) PDU from the target.

These two features help reduce I/O time for small write commands because it removes one round-trip between the initiator and the target for the R2T PDU. As an iSCSI target, the MDS switch allows up to 64 KB of unsolicited data per command. This is controlled by the FirstBurstLength parameter during iSCSI login negotiation phase.

If an iSCSI initiator supports immediate data and unsolicited data features, these features are automatically enabled on the MDS switch with no configuration required.

### iSCSI Interface Advanced Features

Advanced configuration options are available for iSCSI interfaces on a per-IPS port basis. These configurations are similar to the advanced FCIP configurations and are already explained in that section (see Advanced FCIP Profile Configuration for more information).

Use the following example to access these commands from the iSCSI interface:

```
switch# config terminal
switch(config)# interface iscsi 4/1
switch(config-if)#
```

# Advanced Features for iSCSI Interfaces for Cisco MDS Switches

Cisco MDS switches support the following advanced features for iSCSI interfaces:

- iSCSI Listener Port
- TCP Tuning Parameters
- iSCSI Routing Modes
- Setting QoS Values

### iSCSI Listener Port

You can configure the TCP port number for the iSCSI interface that listens for new TCP connections. The default port number is 3260. Once you change the TCP port number, the iSCSI port only accepts TCP connections on the newly configured port.

### TCP Tuning Parameters

You can configure the following TCP parameters:

- Minimum retransmit timeout (See the Minimum Retransmit Timeout for more information).
- Keepalive timeout (See the Keepalive Timeout section for more information).

- Maximum retransmissions (See the Maximum Retransmissions section for more information).
- Path MTU (See the Path MTUs section for more information).
- SACK (SACK is enabled by default for iSCSI TCP configurations).
- Window management (The iSCSI defaults are max-bandwidth is 1 Gbps, min-available-bandwidth is 70 Mbps, and round-trip-time is 1 msec). (See the Window Management section for more information).
- Buffer size (The iSCSI default send buffer size is 4096 KB) (See the Displaying FCIP Profile Information section for more information).
- Window congestion monitoring (enabled by default and the default burst size is 50 KB) (See the Monitoring Congestion section for more information).
- Maximum delay jitter (enabled by default and the default time is 500 microseconds).

### iSCSI Routing Modes

Cisco MDS 9000 Family switches support multiple iSCSI routing modes. Each mode negotiates different operational parameters, has different advantages and disadvantages, and is suitable for different usages.

- Pass-thru mode: In pass-thru mode, the port on the Fibre Channel module with IPS ports or MPS 14/2 module converts and forwards read data frames from the Fibre Channel target to the iSCSI host frame-by-frame without buffering. This means that one data-in frame received is immediately sent out as one iSCSI data-in PDU.

  In the opposite direction, the port on the Fibre Channel module with IPS ports or MPS 14/2 module limits the maximum size of iSCSI write data-out PDU that the iSCSI host can send to the maximum data size that the Fibre Channel target specifies that it can receive. The result is one iSCSI data-out PDU received sent out as one Fibre Channel data frame to the Fibre Channel target.

  The absence of buffering in both directions leads to an advantage of lower forwarding latency. However, a small maximum data segment length usually results in lower data transfer performance from the host because of a higher processing overhead by the host system. Another benefit of this mode is iSCSI data digest can be enabled. This helps protect the integrity of iSCSI data carried in the PDU over what TCP checksum offers.

- Store-and-forward mode (default): In store-and-forward mode, the port on the Fibre Channel module with IPS ports or MPS 14/2 module assembles all the Fibre Channel data frames of an exchange to build one large iSCSI data-in PDU before forwarding it to the iSCSI client.

  In the opposite direction, the port on the Fibre Channel module with IPS ports or MPS 14/2 module does not impose a small data segment size on the host so the iSCSI host can send an iSCSI data-out PDU of any size (up to 256 KB). The port then waits until the whole iSCSI data-out PDU is received before it converts, or splits, the PDU, and forwards Fibre Channel frames to the Fibre Channel target.

  The advantage of this mode is higher data transfer performance from the host. The disadvantages are higher transfer latency and that the iSCSI data digest (CRC) cannot be used.



**Note** The store-and-forward mode is the default forwarding mode.

- Cut-through mode: Cut-through mode improves the read operation performance over store-and-forward mode. The port on the Fibre Channel module with IPS ports or MPS 14/2 module achieves this by

forwarding each Fibre Channel data-in frame to the iSCSI host as it is received without waiting for the whole exchange complete. There is no difference for write data-out operations from store-and-forward mode.

*Figure 32: iSCSI Routing Modes*

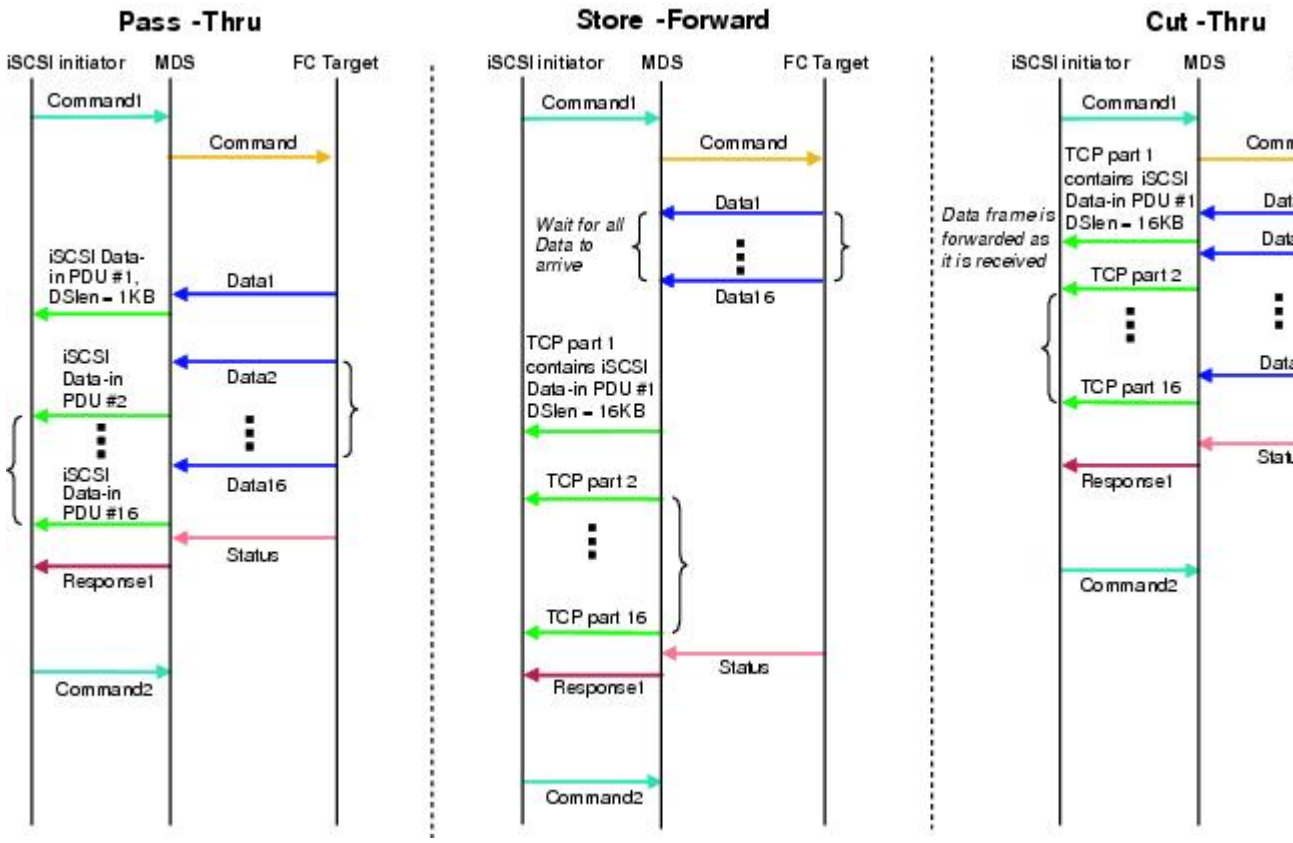



*Table 7: Comparison of iSCSI Routing Modes*

| Mode | Advantages | Disadvantages |
|---|---|---|
| Pass-thru | Low-latency<br>Data digest can be used | Lower data transfer performance. |
| Store-and-forward | Higher data transfer performance | Data digest cannot be used. |
| Cut-thru | Improved read performance over store-and-forward | If the Fibre Channel target sent read data for different commands interchangeably, data of the first command is forwarded in cut-thru mode but the data of subsequent commands is buffered and the behavior is the same as store-and-forward mode.<br>Data digest cannot be used. |

**Note** Changing the forwarding mode of an iSCSI interface that is part of an iSLB VRRP group impacts load balancing on the interface. See the Changing iSCSI Interface Parameters and the Impact on Load Balancing section.

## Setting QoS Values

To set the QoS values, follow these steps:

### Procedure

**Step 1** Configure the differentiated services code point (DSCP) value of 3 to be applied to all outgoing IP packets in this iSCSI interface. The valid range for the iSCSI DSCP value is from 0 to 63.

```
switch(config-if)# qos 3
```

**Step 2** Reverts the switch to its factory default (marks all packets with DSCP value 0).

```
switch(config-if)# no qos 5
```

# Displaying iSCSI Information

Use the **show iscsi** command to obtain detailed information about iSCSI configurations.

This section includes the following topics:

- Displaying iSCSI Interfaces
- Displaying iSCSI Statistics
- Displaying Proxy Initiator Information
- Displaying Global iSCSI Information
- Displaying iSCSI Sessions
- Displaying iSCSI Initiators
- Displaying iSCSI Virtual Targets
- Displaying iSCSI User Information

### Displaying iSCSI Interfaces

Use the **show iscsi interface** command to view the summary, counter, description, and status of the iSCSI interface. Use the output to verify the administrative mode, the interface status, TCP parameters currently used, and brief statistics.

Example of Displaying the iSCSI Interface Information:

```
switch# show interface iscsi 4/1
iscsi4/1 is up
Hardware is GigabitEthernet
Port WWN is 20:cf:00:0c:85:90:3e:80
Admin port mode is ISCSI
Port mode is ISCSI
Speed is 1 Gbps
iSCSI initiator is identified by name
Number of iSCSI session: 0 (discovery session: 0)
Number of TCP connection: 0
Configured TCP parameters
Local Port is 3260
PMTU discover is enabled, reset timeout is 3600 sec
Keepalive-timeout is 60 sec
Minimum-retransmit-time is 300 ms
Max-retransmissions 4
Sack is enabled
QOS code point is 0
Maximum allowed bandwidth is 1000000 kbps
Minimum available bandwidth is 70000 kbps
Estimated round trip time is 1000 usec
Send buffer size is 4096 KB
Congestion window monitoring is enabled, burst size is 50 KB
Configured maximum jitter is 500 us
Forwarding mode: store-and-forward
TMF Queueing Mode : disabled
Proxy Initiator Mode : disabled
5 minutes input rate 0 bits/sec, 0 bytes/sec, 0 frames/sec
5 minutes output rate 0 bits/sec, 0 bytes/sec, 0 frames/sec
iSCSI statistics
Input 0 packets, 0 bytes
Command 0 pdus, Data-out 0 pdus, 0 bytes
Output 0 packets, 0 bytes
Response 0 pdus (with sense 0), R2T 0 pdus
Data-in 0 pdus, 0 bytes
```

### Displaying iSCSI Statistic

Use the **show iscsi stats** command to view brief or detailed iSCSI statistics per iSCSI interface. For more information, see the following examples.

The following example displays iSCSI throughput on an IPS port in both inbound and outbound directions. It also displays the number of different types of iSCSI PDU received and transmitted by this IPS port

Example of Displaying Brief iSCSI Statistics for an iSCSI Interface:

```
switch# show iscsi stats iscsi 2/1
iscsi2/1
5 minutes input rate 704 bits/sec, 88 bytes/sec, 1 frames/sec
5 minutes output rate 704 bits/sec, 88 bytes/sec, 1 frames/sec
iSCSI statistics
974756 packets input, 142671620 bytes
Command 2352 pdus, Data-out 44198 pdus, 92364800 bytes, 0 fragments, unsolicited 0 bytes
output 1022920 packets, 143446248 bytes
Response 2352 pdus (with sense 266), R2T 1804 pdus
Data-in 90453 pdus, 92458248 bytes
```

The following example displays detailed iSCSI statistics for an IPS port. Along with the traffic rate and the number of each iSCSI PDU type, it shows the number of FCP frames received and forwarded, the number of iSCSI login attempts, successes, and failures. It also shows the number of different types of iSCSI PDUs sent and received that are noncritical or occur less frequently, such as NOP in and out (NOP-In and NOP-Out),

text request and response (Text-REQ and Text-RESP), and task management request and response (TMF-REQ and TMF-RESP).

Various types of errors and PDU or frame drop occurrences are also counted and displayed. For example, Bad header digest shows the number of iSCSI PDUs received that have a header digest that fails CRC verification. The iSCSI Drop section shows the number of PDUs that were dropped because of reasons such as target down, LUN mapping fail, Data CRC error, or unexpected Immediate or Unsolicited data. These statistics are helpful for debugging purposes when the feature is not working as expected.

The last section, Buffer Stats, gives statistics on the internal IPS packet buffer operation. This section is for debugging purposes only.

Example of displaying detailed iSCSI statistics for the iSCSI interface:

```
switch# show iscsi stats iscsi 2/1 detail
iscsi2/1
5 minutes input rate 704 bits/sec, 88 bytes/sec, 1 frames/sec
5 minutes output rate 704 bits/sec, 88 bytes/sec, 1 frames/sec
iSCSI statistics
974454 packets input, 142656516 bytes
Command 2352 pdus, Data-out 44198 pdus, 92364800 bytes, 0 fragments, unsolicited 0 bytes
output 1022618 packets, 143431144 bytes
Response 2352 pdus (with sense 266), R2T 1804 pdus
Data-in 90453 pdus, 92458248 bytes
iSCSI Forward:
Command:2352 PDUs (Rcvd:2352)
Data-Out (Write):16236 PDUs (Rcvd 44198), 0 fragments, 92364800 bytes, unsolicited 0 bytes
FCP Forward:
Xfer_rdy:1804 (Rcvd:1804)
Data-In:90453 (Rcvd:90463), 92458248 bytes
Response:2352 (Rcvd:2362), with sense 266
TMF Resp:0

iSCSI Stats:
Login:attempt:13039, succeed:110, fail:12918, authen fail:0
Rcvd:NOP-Out:914582, Sent:NOP-In:914582
NOP-In:0, Sent:NOP-Out:0
TMF-REQ:0, Sent:TMF-RESP:0
Text-REQ:18, Sent:Text-RESP:27
SNACK:0
Unrecognized Opcode:0, Bad header digest:0
Command in window but not next:0, exceed wait queue limit:0
Received PDU in wrong phase:0
SCSI Busy responses:0
Immediate data failure::Separation:0
Unsolicited data failure::Separation:0, Segment:0
Add header:0
Sequence ID allocation failure:0
FCP Stats:
Total:Sent:47654
Received:96625 (Error:0, Unknown:0)
Sent:PLOGI:10, Rcvd:PLOGI_ACC:10, PLOGI_RJT:0
PRLI:10, Rcvd:PRLI_ACC:10, PRLI_RJT:0, Error:0, From initiator:0
LOGO:4, Rcvd:LOGO_ACC:0, LOGO_RJT:0
PRLO:4, Rcvd:PRLO_ACC:0, PRLO_RJT:0
ABTS:0, Rcvd:ABTS_ACC:0
TMF REQ:0
Self orig command:10, Rcvd:data:10, resp:10
Rcvd:PLOGI:156, Sent:PLOGI_ACC:0, PLOGI_RJT:156
LOGO:0, Sent:LOGO_ACC:0, LOGO_RJT:0
PRLI:8, Sent:PRLI_ACC:8, PRLI_RJT:0
PRLO:0, Sent:PRLO_ACC:0, PRLO_RJT:0
ADISC:0, Sent:ADISC_ACC:0, ADISC_RJT:0
```

```
ABTS:0

iSCSI Drop:
Command:Target down 0, Task in progress 0, LUN map fail 0
CmdSeqNo not in window 0, No Exchange ID 0, Reject 0
No task:0
Data-Out:0, Data CRC Error:0
TMF-Req:0, No task:0
Unsolicited data:0, Immediate command PDU:0
FCP Drop:
Xfer_rdy:0, Data-In:0, Response:0

Buffer Stats:
Buffer less than header size:0, Partial:45231, Split:322
Pullup give new buf:0, Out of contiguous buf:0, Unaligned m_data:0
```

### Displaying Proxy Initiator Information

If the proxy initiator feature is enabled in the iSCSI interface, use the **show interface iscsi** command to display configured proxy initiator information.

Example of displaying proxy initiator information for the iSCSI interface with system-sssigned WWNs:

```
switch# show interface iscsi 4/1
iscsi4/1 is up
Hardware is GigabitEthernet
Port WWN is 20:c1:00:05:30:00:a7:9e
Admin port mode is ISCSI
Port mode is ISCSI
Speed is 1 Gbps
iSCSI initiator is identified by name
Number of iSCSI session: 0, Number of TCP connection: 0
Configured TCP parameters
Local Port is 3260
PMTU discover is enabled, reset timeout is 3600 sec
Keepalive-timeout is 60 sec
Minimum-retransmit-time is 300 ms
Max-retransmissions 4
Sack is disabled
QOS code point is 0
Forwarding mode: pass-thru
TMF Queueing Mode : disabled
Proxy Initiator Mode : enabled<-----------------------------Proxy initiator is enabled
nWWN is 28:00:00:05:30:00:a7:a1 (system-assigned)<----System-assigned nWWN
pWWN is 28:01:00:05:30:00:a7:a1 (system-assigned)<---- System-assigned pWWN
5 minutes input rate 0 bits/sec, 0 bytes/sec, 0 frames/sec
5 minutes output rate 0 bits/sec, 0 bytes/sec, 0 frames/sec
iSCSI statistics
Input 7 packets, 2912 bytes
Command 0 pdus, Data-out 0 pdus, 0 bytes
Output 7 packets, 336 bytes
Response 0 pdus (with sense 0), R2T 0 pdus
Data-in 0 pdus, 0 bytes
```

Example of displaying proxy initiator information for the iSCSI interface with user-assigned WWNs:

```
switch# show interface iscsi 4/2
iscsi4/2 is up
Hardware is GigabitEthernet
Port WWN is 20:c1:00:05:30:00:a7:9e
Admin port mode is ISCSI
Port mode is ISCSI
Speed is 1 Gbps
iSCSI initiator is identified by name
```

```
Number of iSCSI session: 0, Number of TCP connection: 0
Configured TCP parameters
Local Port is 3260
PMTU discover is enabled, reset timeout is 3600 sec
Keepalive-timeout is 60 sec
Minimum-retransmit-time is 300 ms
Max-retransmissions 4
Sack is disabled
QOS code point is 0
Forwarding mode: pass-thru
TMF Queueing Mode : disabled
Proxy Initiator Mode : enabled
nWWN is 11:11:11:11:11:11:11:11 (manually-configured)<----User-assigned nWWN
pWWN is 22:22:22:22:22:22:22:22 (manually-configured)<----User-assigned pWWN
5 minutes input rate 0 bits/sec, 0 bytes/sec, 0 frames/sec
5 minutes output rate 0 bits/sec, 0 bytes/sec, 0 frames/sec
iSCSI statistics
Input 7 packets, 2912 bytes
Command 0 pdus, Data-out 0 pdus, 0 bytes
Output 7 packets, 336 bytes
Response 0 pdus (with sense 0), R2T 0 pdus
Data-in 0 pdus, 0 bytes
```

### Displaying Global iSCSI Information

Use the **show iscsi global** command to view the overall configuration and the iSCSI status.

Example of displaying the current global iSCSI configuration and state:

```
switch# show iscsi global
iSCSI Global information
Authentication: CHAP, NONE
Import FC Target: Enabled
Initiator idle timeout: 300 seconds
Number of target node: 0
Number of portals: 11
Number of session: 0
Failed session: 0, Last failed initiator name:
```

### Displaying iSCSI Sessions

Use the **show iscsi session** command to view details about the current iSCSI sessions in the switch. Without parameters, this command displays all sessions. The output can be filtered by specifying an initiator, a target, or both. The following example shows one iSCSI initiator configured based on the IQN (iqn.1987-05.com.cisco:02.3021b0f2fda0.avanti12-w2k) and another based on its IPv4 address (10.10.100.199).

Example of Displaying Brief Information of All iSCSI Sessions:

```
switch# show iscsi session
Initiator iqn.1987-05.com.cisco:02.3021b0f2fda0.avanti12-w2k
Initiator ip addr (s): 10.10.100.116
Session #1
Discovery session, ISID 00023d000043, Status active
Session #2
Target VT1
VSAN 1, ISID 00023d000046, Status active, no reservation
Session #3
Target VT2
VSAN 1, ISID 00023d000048, Status active, no reservation
Initiator 10.10.100.199
Initiator name iqn.1987-05.com.cisco.01.7e3183ae458a94b1cd6bc168cba09d2e
Session #1
Target VT2
```

```
VSAN 1, ISID 246700000000, Status active, no reservation
Session #2
Target VT1
VSAN 1, ISID 246b00000000, Status active, no reservation
Session #3
Target iqn.1987-05.com.cisco:05.switch.04-01.2100002037a6be32
VSAN 1, ISID 246e00000000, Status active, no reservation
```

The following example display the iSCSI initiator configured based on its IPv4 address (10.10.100.199).

Eample of displaying brief information about the specified iSCSI session:

```
switch# show iscsi session initiator 10.10.100.199 target VT1
Initiator 10.10.100.199
Initiator name iqn.1987-05.com.cisco.01.7e3183ae458a94b1cd6bc168cba09d2e
Session #1
Target VT1
VSAN 1, ISID 246b00000000, Status active, no reservation
```

Example of displaying detailed information about the specified iSCSI session:

```
switch# show iscsi session initiator 10.10.100.199 target VT1 detail
Initiator 10.10.100.199 (oasis-qa)
Initiator name iqn.1987-05.com.cisco.01.7e3183ae458a94b1cd6bc168cba09d2e
Session #1 (index 3)
Target VT1
VSAN 1, ISID 246b00000000, TSIH 384, Status active, no reservation
Type Normal, ExpCmdSN 39, MaxCmdSN 54, Barrier 0
MaxBurstSize 0, MaxConn 0, DataPDUInOrder No
DataSeqInOrder No, InitialR2T Yes, ImmediateData No
Registered LUN 0, Mapped LUN 0
Stats:
PDU: Command: 38, Response: 38
Bytes: TX: 8712, RX: 0
Number of connection: 1
Connection #1
Local IP address: 10.10.100.200, Peer IP address: 10.10.100.199
CID 0, State: LOGGED_IN
StatSN 62, ExpStatSN 0
MaxRecvDSLength 1024, our_MaxRecvDSLength 1392
CSG 3, NSG 3, min_pdu_size 48 (w/ data 48)
AuthMethod none, HeaderDigest None (len 0), DataDigest None (len 0)
Version Min: 2, Max: 2
FC target: Up, Reorder PDU: No, Marker send: No (int 0)
Received MaxRecvDSLen key: No
```

### Displaying iSCSI Initiators

Use the **show iscsi initiator** command to display information about all initiators connected to an iSCSI interface in the switch. The information can be filtered to display only the desired iSCSI initiator by specifying the initiator name. Detailed output of the iSCSI initiator can be obtained by specifying the **detail** option. The **iscsi-session** (and optionally **detail** ) parameter displays only iSCSI session information. The **fcp-session** (and optionally **detail** ) parameter displays only FCP session information. The output includes static and dynamic initiators.

Example of displaying information about connected iSCSI initiators:

```
switch# show iscsi initiator
iSCSI Node name is iqn.1987-05.com.cisco:02.3021b0f2fda0.avanti12-w2k
Initiator ip addr (s): 10.10.100.116
iSCSI alias name: AVANTI12-W2K
Node WWN is 22:01:00:05:30:00:10:e1 (configured)
Member of vsans: 1, 2, 10
Number of Virtual n_ports: 1
```

```
Virtual Port WWN is 22:04:00:05:30:00:10:e1 (configured)
Interface iSCSI 4/1, Portal group tag: 0x180
VSAN ID 1, FCID 0x6c0202
VSAN ID 2, FCID 0x6e0000
VSAN ID 10, FCID 0x790000

iSCSI Node name is 10.10.100.199
iSCSI Initiator name: iqn.1987-05.com.cisco.01.7e3183ae458a94b1cd6bc168cba09d2e
iSCSI alias name: oasis-qa
Node WWN is 22:03:00:05:30:00:10:e1 (configured)
Member of vsans: 1, 5
Number of Virtual n_ports: 1
Virtual Port WWN is 22:00:00:05:30:00:10:e1 (configured)
Interface iSCSI 4/1, Portal group tag: 0x180
VSAN ID 5, FCID 0x640000
VSAN ID 1, FCID 0x6c0203
```

Example of displaying detailed information about the iSCSI initiator:

```
switch# show iscsi initiator iqn.1987-05.com.cisco:02.3021b0f2fda0.avanti12-w2k detail
iSCSI Node name is iqn.1987-05.com.cisco:02.3021b0f2fda0.avanti12-w2k
Initiator ip addr (s): 10.10.100.116
iSCSI alias name: AVANTI12-W2K
Node WWN is 22:01:00:05:30:00:10:e1 (configured)
Member of vsans: 1, 2, 10
Number of Virtual n_ports: 1

Virtual Port WWN is 22:04:00:05:30:00:10:e1 (configured)
Interface iSCSI 4/1, Portal group tag is 0x180
VSAN ID 1, FCID 0x6c0202
1 FC sessions, 1 iSCSI sessions
iSCSI session details <-------------------iSCSI session details
Target: VT1
Statistics:
PDU: Command: 0, Response: 0
Bytes: TX: 0, RX: 0
Number of connection: 1
TCP parameters
Local 10.10.100.200:3260, Remote 10.10.100.116:4190
Path MTU: 1500 bytes
Retransmission timeout: 310 ms
Round trip time: Smoothed 160 ms, Variance: 38
Advertized window: Current: 61 KB, Maximum: 62 KB, Scale: 0
Peer receive window: Current: 63 KB, Maximum: 63 KB, Scale: 0
Congestion window: Current: 1 KB

FCP Session details <-------------------FCP session details
Target FCID: 0x6c01e8 (S_ID of this session: 0x6c0202)
pWWN: 21:00:00:20:37:62:c0:0c, nWWN: 20:00:00:20:37:62:c0:0c
Session state: CLEANUP
1 iSCSI sessions share this FC session
Target: VT1
Negotiated parameters
RcvDataFieldSize 1392 our_RcvDataFieldSize 1392
MaxBurstSize 0, EMPD: FALSE
Random Relative Offset: FALSE, Sequence-in-order: Yes
Statistics:
PDU: Command: 0, Response: 0
```

Use the **show fcns database** command (and optionally **detail**) to display the Fibre Channel name server entry for the Fibre Channel N port created for iSCSI initiators in the SAN.

Example of displaying the FCNS database contents:

```
switch# show fcns database
VSAN 1:
------------------------------------------------------------------------
FCID TYPE PWWN (VENDOR) FC4-TYPE:FEATURE
------------------------------------------------------------------------
0x020101 N 22:04:00:05:30:00:35:e1 (Cisco) scsi-fcp:init isc..w <--iSCSI
0x020102 N 22:02:00:05:30:00:35:e1 (Cisco) scsi-fcp:init isc..w initiator
0x0205d4 NL 21:00:00:04:cf:da:fe:c6 (Seagate) scsi-fcp:target
0x0205d5 NL 21:00:00:04:cf:e6:e4:4b (Seagate) scsi-fcp:target
...
Total number of entries = 10

VSAN 2:
------------------------------------------------------------------------
FCID TYPE PWWN (VENDOR) FC4-TYPE:FEATURE
------------------------------------------------------------------------
0xef0001 N 22:02:00:05:30:00:35:e1 (Cisco) scsi-fcp:init isc..w
Total number of entries = 1

VSAN 3:
------------------------------------------------------------------------
FCID TYPE PWWN (VENDOR) FC4-TYPE:FEATURE
------------------------------------------------------------------------
0xed0001 N 22:02:00:05:30:00:35:e1 (Cisco) scsi-fcp:init isc..w
Total number of entries = 1
```

Example of displaying the FCNS database in detail:

```
switch# show fcns database detail
------------------------
VSAN:1 FCID:0x020101
------------------------
port-wwn (vendor) :22:04:00:05:30:00:35:e1 (Cisco)
node-wwn :22:03:00:05:30:00:35:e1
class :2,3
node-ip-addr :10.2.2.12 <--- iSCSI initiator's IPv4 address
ipa :ff ff ff ff ff ff ff ff
fc4-types:fc4_features:scsi-fcp:init iscsi-gw
symbolic-port-name :
symbolic-node-name :iqn.1991-05.com.microsoft:oasis2-dell <--- iSCSI initiator's IQN
port-type :N
port-ip-addr :0.0.0.0
fabric-port-wwn :22:01:00:05:30:00:35:de
hard-addr :0x000000
------------------------
VSAN:1 FCID:0x020102
------------------------
port-wwn (vendor) :22:02:00:05:30:00:35:e1 (Cisco)
node-wwn :22:01:00:05:30:00:35:e1
class :2,3
node-ip-addr :10.2.2.11
ipa :ff ff ff ff ff ff ff ff
fc4-types:fc4_features:scsi-fcp:init iscsi-gw
symbolic-port-name :
symbolic-node-name :iqn.1987-05.com.cisco.01.14ac33ba567f986f174723b5f9f2377
port-type :N
port-ip-addr :0.0.0.0
fabric-port-wwn :22:01:00:05:30:00:35:de
hard-addr :0x000000
...
Total number of entries = 10
==========================================================================
------------------------
VSAN:2 FCID:0xef0001
```

```
------------------------
port-wwn (vendor) :22:02:00:05:30:00:35:e1 (Cisco)
node-wwn :22:01:00:05:30:00:35:e1
class :2,3
node-ip-addr :10.2.2.11
ipa :ff ff ff ff ff ff ff ff
fc4-types:fc4_features:scsi-fcp:init iscsi-gw
symbolic-port-name :
symbolic-node-name :iqn.1987-05.com.cisco.01.14ac33ba567f986f174723b5f9f2377
port-type :N
port-ip-addr :0.0.0.0
fabric-port-wwn :22:01:00:05:30:00:35:de
hard-addr :0x000000
Total number of entries = 1
...
```

Use the **show iscsi initiator configured** command to display information about all the configured iSCSI initiators. Specifying the name shows information about the desired initiator.

Example of displaying information about configured initiators:

```
switch# show iscsi initiator configured
iSCSI Node name is iqn.1987-05.com.cisco:02.3021b0f2fda0.avanti12-w2k
Member of vsans: 1, 2, 10
Node WWN is 22:01:00:05:30:00:10:e1
No. of PWWN: 5
Port WWN is 22:04:00:05:30:00:10:e1
Port WWN is 22:05:00:05:30:00:10:e1
Port WWN is 22:06:00:05:30:00:10:e1
Port WWN is 22:07:00:05:30:00:10:e1
Port WWN is 22:08:00:05:30:00:10:e1

iSCSI Node name is 10.10.100.199
Member of vsans: 1, 5
Node WWN is 22:03:00:05:30:00:10:e1
No. of PWWN: 4
Port WWN is 22:00:00:05:30:00:10:e1
Port WWN is 22:09:00:05:30:00:10:e1
Port WWN is 22:0a:00:05:30:00:10:e1
Port WWN is 22:0b:00:05:30:00:10:e1

User Name for Mutual CHAP: testuser
```

### Displaying iSCSI Virtual Targets

Use the **show iscsi virtual-target** command to display information about the Fibre Channel targets exported as iSCSI virtual targets to the iSCSI initiators. The output includes static as well as dynamic targets.

Example of displaying exported targets:

```
switch# show iscsi virtual-target
target: VT1
* Port WWN 21:00:00:20:37:62:c0:0c
Configured node
all initiator permit is enabled
target: VT2
Port WWN 21:00:00:04:cf:4c:52:c1
Configured node
all initiator permit is disabled
target: iqn.1987-05.com.cisco:05.switch.04-01.2100002037a6be32
Port WWN 21:00:00:20:37:a6:be:32 , VSAN 1
Auto-created node
```

### Displaying iSCSI User Information

The **show user-account iscsi** command displays all configured iSCSI user names.

Example of displaying iSCSI user names:

```
switch# show user-account iscsi
username:iscsiuser
secret: dsfffsffsffasffsdffg
username:user2
secret:cshadhdhsadadjajdjas
```

# Configuring iSLB

The iSCSI server load balancing (iSLB) feature provides a means to easily configure large scale iSCSI deployments containing hundreds or even thousands of initiators. iSLB provides the following features:

- The iSLB initiator configuration is simplified with support for initiator targets and auto-zones.
- Cisco Fabric Services (CFS) eliminates the need for manual configuration by distributing the iSLB initiator configuration among all MDS switches in the fabric.
- Dynamic load balancing of iSLB initiators is available using iSCSI login redirect and VRRP.

When not using iSLB, configuring iSCSI requires the following:

- You need to perform multiple configuration steps on the MDS switch, including the following:
  - Initiator configuration using static pWWN and VSAN.
  - Zoning configuration for initiators and targets.
  - Optional create virtual target and give access to the initiator.
  - Configuration of target LUN mapping and masking on the storage system for the initiator based on the static pWWN created for the initiator on the MDS switch.
- You need to duplicate the configuration manually on multiple MDS switches.
- There is no load balancing for IPS ports. For example:
  - The Virtual Router Redundancy Protocol (VRRP) only supports active and backup, not load balancing.
  - You must use multiple VRRP groups and configure hosts in different groups.

iSLB provides the following features:

- The iSLB initiator configuration is simplified with support for initiator targets and auto-zones.
- Cisco Fabric Services (CFS) eliminates the need for manual configuration by distributing the iSLB initiator configuration among all MDS switches in the fabric.

**Note** Only statically mapped iSLB initiator configuration is distributed throughout the fabric using CFS. Dynamically and statically mapped iSCSI initiator configurations are not distributed.

- Dynamic load balancing of iSLB initiators is available using iSCSI login redirect and VRRP.

**Note** Before configuring iSLB, you must enable iSCSI (see the Enabling iSCSI section).

For iSLB, all switches in the fabric must be running Cisco MDS SAN-OS Release 2.1(1a) or later.

This section covers the following topics:

# iSLB Configuration Limits

iSLB configuration has the following limits:

- The maximum number of iSLB and iSCSI initiators supported in a fabric is 2000.
- The maximum number of iSLB and iSCSI sessions supported by an IPS port in either transparent or proxy initiator mode is 500.
- The maximum number of iSLB initiators supported in a fabric is 2000.
- The maximum number of iSLB initiators and iSCSI sessions supported by a switch is 5000.
- The maximum number of iSLB sessions per IPS port in either transparent or proxy initiator mode is 500.
- The maximum number of iSLB and iSCSI targets supported in a fabric is 6000.
- The maximum number of switches in a fabric that can have iSLB with CFS distribution enabled is four.
- No more than 200 new iSLB initiators can be added to the pending configuration. Before adding more initiators, you must commit the configuration.
- You cannot disable iSCSI if you have more than 200 iSLB initiators in the running configuration. Reduce the number of iSLB initiators to fewer than 200 before disabling iSCSI.
- iSLB can be used without CFS distribution but if iSLB auto-zone feature is used, traffic is disrupted when any zoneset is activated.
- If IVR and iSLB features are enabled in the same fabric, you should have at least one switch in the fabric where both these features are enabled. Any zoning-related configuration and activation (for normal zones, IVR zones, or iSLB zones) must be performed on this switch. Otherwise, there may be traffic disruption in the fabric.

# iSLB Configuration Prerequisites

Perform the following prerequisite actions prior to configuring iSLB:

- Enable iSCSI (see the Enabling iSCSI section for more information).

- Configure the Gigabit Ethernet interfaces (see the Basic Gigabit Ethernet Configuration for IPv4 section).
- Configure the VRRP groups (see the Configuring Load Balancing Using VRRP section).
- Configure and activate a zone set (see the Cisco Fabric Manager Fabric Configuration Guide and Cisco MDS 9000 Family NX-OS Fabric Configuration Guide for more information).
- Enable CFS distribution for iSLB (see the Enabling iSLB Configuration Distribution section).

# iSLB Initiators

iSLB initiators provide the following features in addition to those supported by iSCSI initiators:

- An iSLB initiator also supports iSLB virtual targets.
- Initiator targets–These targets are configured for a particular initiator.
- Load balancing using iSCSI login redirect and VRRP–If iSCSI login redirect is enabled, the IPS Manager redirects incoming sessions to the best interface based on the calculated load for each interface.
- Configuration distribution to other switches using CFS.

iSLB initiators provide the following features in addition to those supported by iSCSI initiators:

- An iSLB initiator also supports iSLB virtual targets. These targets are very similar to iSCSI virtual targets with the exception that they do not include the advertise interface option and as a result are distributable using CFS.
- Initiator targets——These targets are configured for a particular initiator.
- Load balancing using iSCSI login redirect and VRRP—If load balancing is enabled, the IPS Manager redirects incoming sessions to the best interface based on the calculated load for each interface.
- Configuration distribution to other switches using CFS.

# Configuring iSLB Initiators

This section includes the following topics:

## Configuring iSLB Initiator Names

To enter iSLB initiator configuration submode using the **name** option for an iSLB initiator, follow these steps:

### Before you begin

You must specify the iSLB initiator name or IP address before configuring it.

Specifying the iSLB initiator name or IP address is the same as for an iSCSI initiator. See the Static Mapping section.

### Procedure

**Step 1** Configure an iSLB initiator using the iSCSI name of the initiator node (iqn.1987-02.com.cisco.initiator) and enters iSLB initiator configuration submode. The maximum name length is 223 alphanumeric characters. The minimum length is 16.

```
switch# config terminal
switch(config)#
switch(config)# islb initiator name iqn.1987-02.com.cisco.initiator
switch(config-islb-init)#
```

**Step 2** Delete the configured iSLB initiator.

```
switch(config)# no lslb initiator name iqn.1987-02.com.cisco.initiator
```

## Configuring iSLB Initiator IP Address

To enter iSLB initiator configuration submode using the **ip-address** option for an iSLB initiator, follow these steps:

### Before you begin

You must specify the iSLB initiator name or IP address before configuring it.

Specifying the iSLB initiator name or IP address is the same as for an iSCSI initiator. See the Static Mapping section.

### Procedure

**Step 1** Configure an iSLB initiator using the IPv4 address of the initiator node and enters iSLB initiator configuration submode.

```
switch# config terminal
switch(config)# islb initiator ip-address 10.1.1.3
```

**Step 2** Delete the configured iSLB initiator.

```
switch(config)# no islb initiator ip-address 10.1.1.3
```

**Step 3** Configure an iSLB initiator using the IPv6 unicast address of the initiator node and enters iSLB initiator configuration submode.

```
switch(config)# islb initiator ip-address 2001:0DB8:800:200C::417A
switch(config-islb-init)#
```

**Step 4** Delete the configured iSLB initiator.

```
switch(config)# no islb initiator ip-address 2001:0DB8:800:200C::417A
```

# Assigning WWNs to iSLB Initiators

An iSLB host is mapped to an N port's WWNs by one of the following mechanisms:

- Dynamic mapping (default)
- Static mapping

**Note** Assigning WWNs for iSLB initiators is the same as for iSCSI initiators. For information on dynamic and static mapping, see the WWN Assignment for iSCSI Initiators section.

We recommend using the **SystemAssign system-assign** option. If you manually assign a WWN, you must ensure its uniqueness (see the Cisco Fabric Manager Fabric Configuration Guide and Cisco MDS 9000 Family NX-OS Fabric Configuration Guide for more information). You should not use any previously assigned WWNs.

### Making the Dynamic iSLB Initiator WWN Mapping Static

After a dynamic iSLB initiator has logged in, you may decide to permanently keep the automatically assigned nWWN/pWWN mapping to allow this initiator to use the same mapping the next time it logs in (see the Dynamic Mapping section).

You can convert a dynamic iSLB initiator to a static iSLB initiator and make its WWNs persistent

**Note** You cannot convert a dynamic iSCSI initiator to a static iSLB initiator or a dynamic iSLB initiator to a static iSCSI initiator. For more information, see, Dynamic Mapping).

**Note** Making the dynamic mapping for iSLB initiators static is the same as for iSCSI..

**Note** Only statically mapped iSLB initiator configuration is distributed throughout the fabric using CFS. Dynamically and statically configured iSCSI initiator configurations are not distributed.

### iSLB Target Access Mapping

In iSLB, all fabric distributed configurations including iSCSI virtual target access are part of the iSCSI initiator configuration. Access is granted using the pWWN or the device alias. You can specify the one or more of the following optional parameters:

- Secondary pWWN
- Secondary device alias
- LUN mapping
- IQN

In addition, you can disable auto zoning.

If you configure an IQN for an initiator target then that name is used to identify the target. Otherwise, an unique IQN is generated for the initiator target.

### Configuring Permanent nWWN/pWWN Mapping

To permanently keep the automatically assigned nWWN/pWWN mapping, follow these steps:

#### Procedure

**Step 1** Save the nWWNs and pWWNs that have automatically been assigned to the iSLB initiator whose name is specified.

```
switch# config terminal
switch(config)# islb save-initiator name iqn.1987-02.com.cisco.initiator
```

**Step 2** Save the nWWNs and pWWNs that have automatically been assigned to the iSLB initiator whose IPv4 address is specified.

```
switch(config)# islb save-initiator 10.10.100.11
```

**Step 3** Save the nWWNs and pWWNs that have automatically been assigned to the iSCSI initiator whose IPv6 unicast address is specified.

```
switch(config)# iscsi save-initiator ip-address 2001:0DB8:800:200C::417A
```

**Step 4** Save the nWWNs and pWWNs that have automatically been assigned to all the iSLB initiators

```
switch(config)# islb save-initiator
switch(config)# exit
```

**Step 5** Save the nWWN/pWWN mapping configuration across system reboots.

```
switch# copy running-config startup-config
```

### Configuring iSLB Initiator Access to iSCSI Virtual Target

To configure iSLB initiator access to an iSCSI virtual target, follow these steps:

#### Procedure

**Step 1** Configure the initiator using a name and enters iSLB initiator configuration submode.

```
switch# config terminal
switch(config)# islb initiator ip-address 10.1.1.3
switch(config-iscsi-islb-init)#
```

**Step 2** Grant iSLB initiator access to the target using a pWWN with auto zoning enabled (default).

```
switch(config-iscsi-islb-init)# target pwwn 26:00:01:02:03:04:05:06
```

**Step 3** Grant iSLB initiator access to the target using a pWWN with auto zoning disabled.

```
switch(config-iscsi-islb-init)# target pwwn 26:00:01:02:03:04:05:06 no-zone
```

**Step 4** Grant iSLB initiator access to the target using a device alias with auto zoning enabled (default).

```
switch(config-iscsi-islb-init)# target device-alias SampleAlias
```

**Step 5** Grant iSLB initiator access to the target using a devices alias and optional LUN mapping.

```
switch(config-iscsi-islb-init)# target device-alias SampleAlias fc-lun 0x1234 iscsi-lun 0x2345
```

**Step 6** Grant iSLB initiator access to the target using a devices alias and an optional IQN.

```
switch(config-iscsi-islb-init)# target device-alias SampleAlias iqn-name iqn.1987-01.com.cisco.initiator
```

**Step 7** Grant iSLB initiator access to the target using a devices alias and an optional secondary device alias.

```
switch(config-iscsi-islb-init)# target device-alias SampleAlias sec-device-alias SecondaryAlias
```

**Step 8** Grant iSLB initiator access to the target using a devices alias and an optional secondary pWWN.

```
switch(config-iscsi-islb-init)# target device-alias SampleAlias sec-pwwn 26:01:02:03:04:05:06:07
```

**Step 9** Remove the target access.

```
switch(config-iscsi-init)# no target pwwn 26:00:01:02:03:04:05:06
```

### Verifying the iSLB Target Configuration

To verify the iSLB target configuration, use the **show islb initiator configured** command.

```
switch# show islb initiator configured
iSCSI Node name is 10.1.1.3
Number of Initiator Targets: 1
Initiator Target: iqn.1987-05.com.cisco:05.ips-hac4
Port WWN 50:06:04:82:ca:e1:26:8d
Zoning Enabled
No. of LU mapping: 3
iSCSI LUN: 0x0001, FC LUN: 0x0001
iSCSI LUN: 0x0002, FC LUN: 0x0002
iSCSI LUN: 0x0003, FC LUN: 0x0003
```

# Assigning VSAN Membership for iSLB Initiators

Individual iSLB hosts can be configured to be in a specific VSAN (similar to the DPVM feature for Fibre Channel. The specified VSAN overrides the iSCSI interface VSAN membership.

For more information, see the Cisco MDS 9000 Family NX-OS Fabric Manager Fabric Configuration Guide.

To assign VSAN membership for iSLB initiators, follow these steps:

### Before you begin

Specifying the iSLB initiator VSAN is the same as for an iSCSI initiator. See VSAN Membership for iSCSI.

When an iSLB initiator is configured in any other VSAN (other than VSAN 1, the default VSAN), for example VSAN 2, the initiator is automatically removed from VSAN 1. If you also want it to be present in VSAN 1, you must explicitly configure the initiator in VSAN 1.

### Procedure

**Step 1** Configure an iSLB initiator using its IPv4 address and enters iSLB initiator configuration submode

```
switch# config terminal
switch(config)# islb initiator ip-address 10.1.1.3
switch(config-islb-init)#
```

**Step 2** Assigns the iSLB initiator node to a specified VSAN. You can assign this host to one or more VSANs.

```
switch(config-islb-init)# vsan 3
```

**Step 3** Remove the iSLB initiator from the specified VSAN.

```
switch(config-islb-init)# no vsan 3
```

# Configuring Metrics for Load Balancing

You can assign a load metric to each initiator for weighted load balancing. The load calculated is based on the number of initiators on a given iSCSI interface. This feature accommodates initiators with different bandwidth requirements. For example, you could assign a higher load metric to a database server than to a web server. Weighted load balancing also accommodates initiators with different link speeds.

Also, you can configure initiator targets using the device alias or the pWWN. If you configure an IQN for an initiator target, then that name is used to identify the initiator target. Otherwise, a unique IQN is generated for the initiator target

For more information on load balancing, see the Load Balancing Using VRRP section.

To configure a weight for load balancing, follow these steps:

### Procedure

**Step 1** Configure an iSLB initiator using the name of the initiator node and enters iSLB initiator configuration mode.

```
switch# config terminal
switch(config)# islb initiator name iqn.1987-02.com.cisco.initiator
switch(config-iscsi-init)#
```

**Step 2** Assign 100 as the weight metric for this iSLB initiator.

```
switch(config-iscsi-init)# metric 100
```

**Step 3** Revert to the default value (1000).

```
switch(config-iscsi-init)# no metric 100
```

# Verifying iSLB Initiator Configuration

To verify the iSLB initiator configuration, use the **show islb initiator configured** command.

```
switch# show islb initiator configured
iSCSI Node name is 10.1.1.2
Member of vsans: 10
Node WWN is 23:02:00:0c:85:90:3e:82
Load Balance Metric: 100
Number of Initiator Targets: 1
```

```
Initiator Target: test-targt
Port WWN 01:01:01:01:02:02:02:02
Primary PWWN VSAN 1
Zoning support is enabled
Trespass support is disabled
Revert to primary support is disabled
```

# Configuring iSLB Initiator Targets

You can configure initiator targets using the device alias or the pWWN. You can also optionally specify one or more of the following optional parameters:

- Secondary pWWN
- Secondary device alias
- LUN mapping
- IQN
- VSAN identifier

In addition, you can also disable auto-zoning. If you configure an IQN for an initiator target, then that name is used to identify the initiator target. Otherwise, a unique IQN is generated for the initiator target.

To configure iSLB initiator targets, follow these steps:

### Before you begin

The VSAN identifier is optional if the target is online. If the target is not online, the VSAN identifier is required.

### Procedure

**Step 1** Configure an iSLB initiator using its IPv4 address and enters iSLB initiator configuration submode.

```
switch# config terminal
switch(config)# islb initiator ip-address 10.1.1.3
switch(config-islb-init)#
```

**Step 2** Configure the iSLB initiator target using a pWWN with auto-zoning enabled (default).

```
switch(config-iscsi-islb-init)# target pwwn 26:00:01:02:03:04:05:06
```

**Step 3** Configure the iSLB initiator target using a pWWN with auto-zoning disabled.

```
switch(config-iscsi-islb-init)# target pwwn 26:00:01:02:03:04:05:06 no-zone
```

**Step 4** Configure the iSLB initiator target using a device alias with auto-zoning enabled (default).

```
switch(config-iscsi-islb-init)# target device-alias SampleAlias
```

**Step 5** Configure the iSLB initiator target using a device alias and optional LUN mapping. The CLI interprets the LUN identifier value as a hexadecimal value whether or not the **0x** prefix is included.

```
switch(config-iscsi-islb-init)# target device-alias SampleAlias fc-lun 0x1234 iscsi-lun 0x2345
```

**Step 6** Configure the iSLB initiator target using a device alias and an optional IQN.

```
switch(config-iscsi-islb-init)# target device-alias SampleAlias iqn-name
iqn.1987-01.com.cisco.initiator
```

**Step 7** Configure the iSLB initiator target using a device alias and an optional secondary device alias.

```
switch(config-iscsi-islb-init)# target device-alias SampleAlias sec-device-alias SecondaryAlias
```

**Step 8** Configure the iSLB initiator target using a device alias and an optional secondary pWWN.

```
switch(config-iscsi-islb-init)# target device-alias SampleAlias sec-pwwn 26:01:02:03:04:05:06:07
```

**Step 9** Configure the iSLB initiator target using a device alias and the VSAN identifier.

```
switch(config-iscsi-islb-init)# target device-alias SampleAlias vsan 10
```

**Step 10** Remove the iSLB initiator target.

```
switch(config-iscsi-init)# no target pwwn 26:00:01:02:03:04:05:06
```

# Configuring and Activating Zones for iSLB Initiators and Initiator Targets

You can configure a zone name where the iSLB initiators and initiator targets are added. If you do not specify a zone name, the IPS manager creates one dynamically. iSLB zone sets have the following considerations:

- Auto-zoning of the initiator with the initiator targets is enabled by default.
- A zone set must be active in a VSAN for auto-zones to be created in that VSAN.
- iSLB zone set activation might fail if another zone set activation is in process or if the zoning database is locked. Retry the iSLB zone set activation if a failure occurs. To avoid this problem, only perform only one zoning related operation (normal zones, IVR zones, or iSLB zones) at a time.
- Auto-zones are created when the zone set is activated and there has been at least one change in the zoneset. The activation has no effect if only the auto-zones have changed.

To configure the iSLB initiator optional auto-zone name and activate the zone set, follow these steps:

### Before you begin

If IVR and iSLB are enabled in the same fabric, at least one switch in the fabric must have both features enabled. Any zoning related configuration or activation operation (for normal zones, IVR zones, or iSLB zones) must be performed on this switch. Otherwise, traffic might be disrupted in the fabric.

### Procedure

**Step 1** Configure an iSLB initiator using its IPv4 address and enters iSLB initiator configuration submode.

```
switch# config terminal
switch(config)# islb initiator ip-address 10.1.1.3
switch(config-islb-init)#
```

**Step 2** (Optional) Specify the zone name where the initiators and the initiator targets are added.

```
switch(config-islb-init)# zonename IslbZone
```

**Step 3** (default) Remove the initiators and initiator targets from the zone and adds them to a dynamically created zone

```
switch(config-islb-init)# no zonename IslbZone
```

**Step 4** Return to configuration mode.

```
switch(config-islb-init)# exit
```

**Step 5** Activate zoning for the iSLB initiators and initiator targets with zoning enabled and creates auto-zones if no zone names are configured.

This step is not required if CFS is enabled. CFS automatically activates the zone when the configuration changes are committed.

```
switch(config)# islb zoneset activate
```

# Verifying iSLB Zoning Configuration

The following example shows the **show zoneset active** command output when the dynamically generated zone name is used:

```
switch# show zoneset active
zoneset name zoneset-1 vsan 1
zone name ips_zone_5d9603bcff68008a6fc5862a6670ca09 vsan 1
* fcid 0x010009 [ip-address 10.1.1.3]
pwwn 22:00:00:04:cf:75:28:4d
pwwn 22:00:00:04:cf:75:ed:53
pwwn 22:00:00:04:cf:75:21:d5
pwwn 22:00:00:04:cf:75:ee:59
.
.
.
```

The following example shows the **show zoneset active** command output when the configured zone name IslbZone is used:

```
switch# show zoneset active
zoneset name zoneset-1 vsan 1
zone name ips_zone_IslbZone vsan 1
ip-address 10.1.1.3
pwwn 22:00:00:04:cf:75:28:4d
pwwn 22:00:00:04:cf:75:ed:53
pwwn 22:00:00:04:cf:75:21:d5
pwwn 22:00:00:04:cf:75:ee:59
.
.
.
```

# Configuring iSLB Session Authentication

The Fibre Channel module with IPS ports and MPS-14/2 module support the iSLB authentication mechanism to authenticate iSLB hosts that request access to storage. By default, the Fibre Channel module with IPS ports and MPS-14/2 module allow CHAP or None authentication of iSCSI initiators. If authentication is always used, you must configure the switch to allow only CHAP authentication.

For CHAP user name or secret validation you can use any method supported and allowed by the Cisco MDS AAA infrastructure (see the Cisco Fabric Manager Security Configuration GuideCisco MDS 9000 Family

NX-OS Security Configuration Guide for more information). AAA authentication supports RADIUS, TACACS+, or a local authentication device.



**Note** Specifying the iSLB session authentication is the same as for iSCSI. See the iSCSI Session Authentication section.

# Restricting iSLB Initiator Authentication

By default, the iSLB initiator can use any user name in the RADIUS or local AAA database in authenticating itself to the Fibre Channel module with IPS ports or MPS-14/2 module (the CHAP user name is independent of the iSLB initiator name). The Fibre Channel module with IPS ports or MPS-14/2 module allows the initiator to log in as long as it provides a correct response to the CHAP challenge sent by the switch. This can be a problem if one CHAP user name and password have been compromised.

To restrict an initiator to use a specific user name for CHAP authentication, follow these steps:

### Procedure

**Step 1** Configure an iSLB initiator using the IQN of the initiator node and enters iSLB initiator configuration mode.

```
switch# config terminal
switch(config)# islb initiator name iqn.1987-02.com.cisco.init
switch(config-islb-init)#
```

**Step 2** Restricts the initiator `iqn.1987-02.com.cisco.init` to only authenticate using `user1` as its CHAP user name.

**Note**
Be sure to define user1 as an iSCSI user in the local AAA database or the RADIUS server.

```
switch(config-islb-init)#
switch(config-islb-init)# username user1
```

# Mutual CHAP Authentication

In addition to the Fibre Channel module with IPS ports and MPS-14/2 module authentication of the iSLB initiator, the Fibre Channel module with IPS ports and MPS-14/2 module also support a mechanism for the iSLB initiator to authenticate the Cisco MDS switch's initiator target during the iSCSI login phase. This authentication requires the user to configure a user name and password for the switch to present to the iSLB initiator. The provided password is used to calculate a CHAP response to a CHAP challenge sent to the IPS port by the initiator.

To configure a per-initiator user name and password used by the switch to authenticate itself to an initiator, follow these steps:

### Procedure

**Step 1** Configure an iSLB initiator using the name of the initiator node and enters iSLB initiator configuration mode.

```
switch# config terminal
switch(config)# islb initiator name iqn.1987-02.com.cisco.initiator
switch(config-islb-init)#
```

**Step 2** Configure the switch user account (testuser) along with a password (dcba12LKJ) specified in clear text (default). The password is limited to 128 characters.

```
switch(config-islb-init)# mutual-chap username testuser password dcba12LKJ
```

**Step 3** Configure the switch user account (testuser) along with the encrypted password specified by 7 (!@*asdsfsdfjh!@df).

```
switch(config-islb-init)# mutual-chap username testuser password 7!@*asdsfsdfjh!@df
```

**Step 4** Remove the switch authentication configuration.

```
switch(config-iscsi-init)# no mutual-chap username testuser
```

# Verifying iSLB Authentication Configuration

Use the **show running-config** and the **show iscsi global** commands to display the global configuration. Use the **show running-config** and the **show islb initiator configured** commands to display the initiator specific configuration.

To verify the iSLB user name and mutual CHAP configuration, use the **show islb initiator configured** command:

```
switch# show islb initiator configured
iSCSI Node name is 10.1.1.3
Member of vsans: 3
User Name for login authentication: user1
User Name for Mutual CHAP: testuser
Load Balance Metric: 1000 Number of Initiator Targets: 1
Number of Initiator Targets: 1
Initiator Target: iqn.1987-05.com.cisco:05.ips-hac4
Port WWN 50:06:04:82:ca:e1:26:8d
Zoning Enabled
No. of LU mapping: 3
iSCSI LUN: 0x0001, FC LUN: 0x0001
iSCSI LUN: 0x0002, FC LUN: 0x0002
iSCSI LUN: 0x0003, FC LUN: 0x0003
```

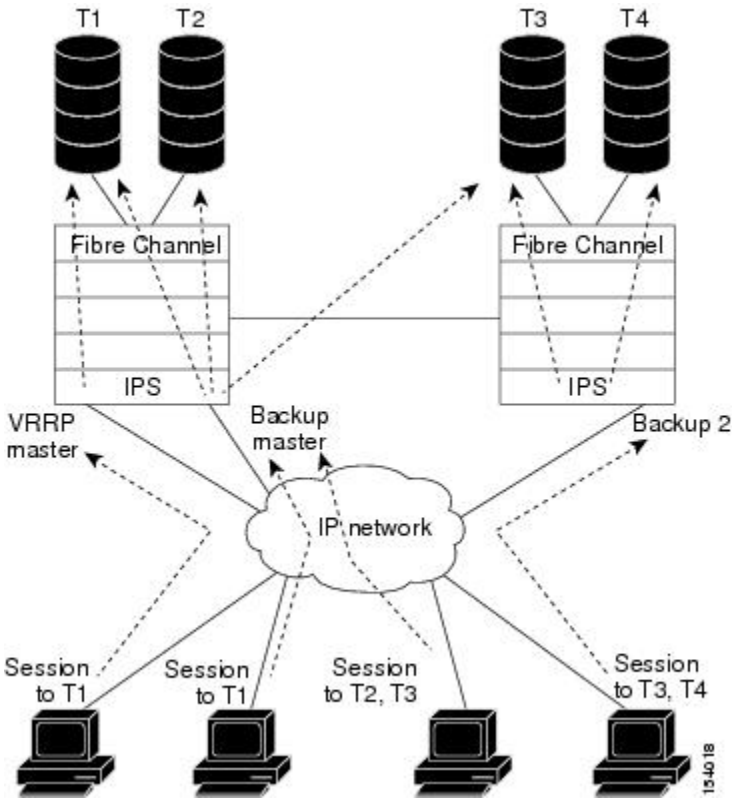
# Load Balancing Using VRRP

You can configure Virtual Router Redundancy Protocol (VRRP) load balancing for iSLB. The host is configured with a VRRP address as the portal address. When the VRRP master port receives the first iSCSI session from an initiator, it assigns a backup port to serve that particular host. The information is synchronized to all switches through CFS if recovery is needed when a master port fails. The initiator gets a temporary redirect iSCSI login response. The host then logs in to the backup port at its physical IP address. All iSCSI interfaces in a VRRP

group that has load balancing enabled must have the same interface VSAN, authentication, proxy initiator mode, and forwarding mode.

You can configure Virtual Router Redundancy Protocol (VRRP) load balancing for iSLB. The following figure shows an example of load balancing using iSLB.

***Figure 33: iSLB Initiator Load Balancing Example***




The host is configured with a VRRP address as the portal address. When the VRRP master port receives the first iSCSI session from an initiator, it assigns a backup port to serve that particular host. This information is synchronized to all switches through CFS if recovery is needed when a master port fails. The initiator gets a temporary redirect iSCSI login response. The host then logs in to the backup port at its physical IP address. If the backup port goes down, the host will revert to the master port. The master port knows through CFS that the backup port has gone down and redirects the host to another backup port.

**Note** If an Ethernet port channel is configured between the Fibre Channel module with IPS ports and an Ethernet switch, the load balancing policy on the Ethernet switch must be based on source/destination IP address only, not port numbers, for load balancing with VRRP to operate correctly.

**Note** An initiator can also be redirected to the physical IP address of the master interface.

iSLB VRRP load balancing is based on the number of iSLB initiators and not number of sessions. Any iSLB initiator that has more targets configured than the other iSLB initiators (resulting in more sessions) should be

configured with a higher load metric. For example, you can increase the load metric of the iSLB initiator with more targets to 3000 from the default value of 1000.

A Gigabit Ethernet interface configured for iSLB can only be in one VRRP group because redirected sessions do not carry information about the VRRP IP address or group. This restriction allows the slave backup port to uniquely identify the VRRP group to which it belongs.

### Changing iSCSI Interface Parameters and the Impact on Load Balancing

All iSCSI interfaces in a VRRP group that has load balancing enabled must have the same interface VSAN, authentication, proxy initiator mode, and forwarding mode. When you need to change any of these parameters for the iSCSI interfaces in a VRRP group, you must do so one interface at a time. During the transition time when the parameter is changed on some interfaces in the VRRP group and not the others, the master port does not redirect new initiators and instead handles them locally.

Changing the VSAN, proxy initiator, authentication, and forwarding mode for iSCSI interfaces in a VRRP group can cause sessions to go down multiple times.

# VRRP Load Balancing Algorithm for Selecting Gigabit Ethernet Interfaces

When the VRRP master receives an iSCSI session request from an initiator, it first checks for an existing mapping to one of the interfaces in that VRRP group. If such a mapping exists, the VRRP master redirects the initiator to that interface. If no such mapping exists, the VRRP master selects the least loaded interface and updates the selected interface's load with the initiator's iSLB metric (weight).



**Note** The VRRP master interface is treated specially and it needs to take a lower load compared to the other interfaces. This is to account for the redirection work performed by the master interface for every session. A new initiator is assigned to the master interface only if the following is true for every other interface: VRRP backup interface load > [2 * VRRP master interface load + 1]

The following examples are based on the following configurations:

- GigabitEthernet2/1.441 is the VRRP master interface for Switch1.
- GigabitEthernet2/2.441 is the VRRP backup interface for Switch1.
- GigabitEthernet1/1.441 is the VRRP backup interface for Switch2.
- GigabitEthernet1/2.441 is the VRRP backup interface for Switch2.

### Load distribution with the Default Metric

The follow example output shows the initial load distribution for three initiators with the default load metric value:

```
switch# show islb vrrp summary
.
.
.
--------------------------------------------------------------------------------
VR Id VRRP IP Switch WWN Ifindex Load
--------------------------------------------------------------------------------
M 1 10.10.122.115 20:00:00:0b:5f:3c:01:80 GigabitEthernet2/1.441 0
1 10.10.122.115 20:00:00:0b:5f:3c:01:80 GigabitEthernet2/2.441 1000
```

```
1 10.10.122.115 20:00:00:0c:ce:5c:5b:c0 GigabitEthernet1/1.441 1000
1 10.10.122.115 20:00:00:0c:ce:5c:5b:c0 GigabitEthernet1/2.441 1000
-- Initiator To Interface Assignment --
--------------------------------------------------------------------------------
Initiator VR Id VRRP IP Switch WWN Ifindex
--------------------------------------------------------------------------------
iqn.cisco.test-linux.init0 1 10.10.122.115 20:00:00:0b:5f:3c:01:80 GigabitEthernet2/2.441
iqn.cisco.test-linux.init1 1 10.10.122.115 20:00:00:0c:ce:5c:5b:c0 GigabitEthernet1/1.441
iqn.cisco.test-linux.init 1 10.10.122.115 20:00:00:0c:ce:5c:5b:c0 GigabitEthernet1/2.441
```

The following example output shows load distribution for four initiators. The interface load metric value for the master interface changed from 0 to 1000.

```
switch# show islb vrrp summary
.
.
.
--------------------------------------------------------------------------------
VVR Id VRRP IP Switch WWN Ifindex Load
--------------------------------------------------------------------------------
M 1 10.10.122.115 20:00:00:0b:5f:3c:01:80 GigabitEthernet2/1.441 1000
1 10.10.122.115 20:00:00:0b:5f:3c:01:80 GigabitEthernet2/2.441 1000
1 10.10.122.115 20:00:00:0c:ce:5c:5b:c0 GigabitEthernet1/1.441 1000
1 10.10.122.115 20:00:00:0c:ce:5c:5b:c0 GigabitEthernet1/2.441 1000
-- Initiator To Interface Assignment --
--------------------------------------------------------------------------------
Initiator VR Id VRRP IP Switch WWN Ifindex
--------------------------------------------------------------------------------
iqn.cisco.test-linux.init0 1 10.10.122.115 20:00:00:0b:5f:3c:01:80 GigabitEthernet2/2.441
iqn.cisco.test-linux.init1 1 10.10.122.115 20:00:00:0c:ce:5c:5b:c0 GigabitEthernet1/1.441
iqn.cisco.test-linux.init2 1 10.10.122.115 20:00:00:0c:ce:5c:5b:c0 GigabitEthernet1/2.441
iqn.cisco.test-linux.init3 1 10.10.122.115 20:00:00:0b:5f:3c:01:80 GigabitEthernet2/1.441
```

The following example output shows load distribution for nine initiators. The interface load metric values for the backup interfaces have changed.

```
switch# show islb vrrp summary
.
.
.
--------------------------------------------------------------------------------
VVR Id VRRP IP Switch WWN Ifindex Load
--------------------------------------------------------------------------------
M 1 10.10.122.115 20:00:00:0b:5f:3c:01:80 GigabitEthernet2/1.441 1000
1 10.10.122.115 20:00:00:0b:5f:3c:01:80 GigabitEthernet2/2.441 3000
1 10.10.122.115 20:00:00:0c:ce:5c:5b:c0 GigabitEthernet1/1.441 3000
1 10.10.122.115 20:00:00:0c:ce:5c:5b:c0 GigabitEthernet1/2.441 2000
-- Initiator To Interface Assignment --
--------------------------------------------------------------------------------
Initiator VR Id VRRP IP Switch WWN Ifindex
--------------------------------------------------------------------------------
iqn.cisco.test-linux.init0 1 10.10.122.115 20:00:00:0b:5f:3c:01:80 GigabitEthernet2/2.441
iqn.cisco.test-linux.init1 1 10.10.122.115 20:00:00:0c:ce:5c:5b:c0 GigabitEthernet1/1.441
iqn.cisco.test-linux.init2 1 10.10.122.115 20:00:00:0c:ce:5c:5b:c0 GigabitEthernet1/2.441
iqn.cisco.test-linux.init3 1 10.10.122.115 20:00:00:0b:5f:3c:01:80 GigabitEthernet2/1.441
iqn.cisco.test-linux.init4 1 10.10.122.115 20:00:00:0b:5f:3c:01:80 GigabitEthernet2/2.441
iqn.cisco.test-linux.init5 1 10.10.122.115 20:00:00:0c:ce:5c:5b:c0 GigabitEthernet1/1.441
iqn.cisco.test-linux.init6 1 10.10.122.115 20:00:00:0c:ce:5c:5b:c0 GigabitEthernet1/2.441
iqn.cisco.test-linux.init7 1 10.10.122.115 20:00:00:0b:5f:3c:01:80 GigabitEthernet2/2.441
iqn.cisco.test-linux.init8 1 10.10.122.115 20:00:00:0c:ce:5c:5b:c0 GigabitEthernet1/1.441
```

### Load Distribution with the Metric Set to 3000 on One Initiator

The following example output shows the initial load distribution for three initiators with one initiator having load metric of 3000 and the remaining initiator with the default metric value:

```
switch# show islb vrrp summary
.
.
.
--------------------------------------------------------------------------------
VVR Id VRRP IP Switch WWN Ifindex Load
--------------------------------------------------------------------------------
M 1 10.10.122.115 20:00:00:0b:5f:3c:01:80 GigabitEthernet2/1.441 0
1 10.10.122.115 20:00:00:0b:5f:3c:01:80 GigabitEthernet2/2.441 1000
1 10.10.122.115 20:00:00:0c:ce:5c:5b:c0 GigabitEthernet1/1.441 3000
1 10.10.122.115 20:00:00:0c:ce:5c:5b:c0 GigabitEthernet1/2.441 1000
-- Initiator To Interface Assignment --
--------------------------------------------------------------------------------
Initiator VR Id VRRP IP Switch WWN Ifindex
--------------------------------------------------------------------------------
iqn.cisco.test-linux.init0 1 10.10.122.115 20:00:00:0c:ce:5c:5b:c0 GigabitEthernet1/1.441
iqn.cisco.test-linux.init1 1 10.10.122.115 20:00:00:0b:5f:3c:01:80 GigabitEthernet2/2.441
iqn.cisco.test-linux.init2 1 10.10.122.115 20:00:00:0c:ce:5c:5b:c0 GigabitEthernet1/2.441
```

The following example output shows load distribution for four initiators. The interface load metric value for the master interface changed from 0 to 1000:

```
switch# show islb vrrp summary
.
.
.
--------------------------------------------------------------------------------
VVR Id VRRP IP Switch WWN Ifindex Load
--------------------------------------------------------------------------------
M 1 10.10.122.115 20:00:00:0b:5f:3c:01:80 GigabitEthernet2/1.441 1000
1 10.10.122.115 20:00:00:0b:5f:3c:01:80 GigabitEthernet2/2.441 3000
1 10.10.122.115 20:00:00:0c:ce:5c:5b:c0 GigabitEthernet1/1.441 1000
1 10.10.122.115 20:00:00:0c:ce:5c:5b:c0 GigabitEthernet1/2.441 1000
-- Initiator To Interface Assignment --
--------------------------------------------------------------------------------
Initiator VR Id VRRP IP Switch WWN Ifindex
--------------------------------------------------------------------------------
iqn.cisco.test-linux.init0 1 10.10.122.115 20:00:00:0b:5f:3c:01:80 GigabitEthernet2/2.441
iqn.cisco.test-linux.init1 1 10.10.122.115 20:00:00:0c:ce:5c:5b:c0 GigabitEthernet1/2.441
iqn.cisco.test-linux.init2 1 10.10.122.115 20:00:00:0c:ce:5c:5b:c0 GigabitEthernet1/1.441
iqn.cisco.test-linux.init3 1 10.10.122.115 20:00:00:0b:5f:3c:01:80 GigabitEthernet2/1.441
```

The following example output shows load distribution for nine initiators. The interface load metric values for the backup interfaces have changed:

```
switch# show islb vrrp summary
.
.
.
--------------------------------------------------------------------------------
VVR Id VRRP IP Switch WWN Ifindex Load
--------------------------------------------------------------------------------
M 1 10.10.122.115 20:00:00:0b:5f:3c:01:80 GigabitEthernet2/1.441 2000
1 10.10.122.115 20:00:00:0b:5f:3c:01:80 GigabitEthernet2/2.441 3000
1 10.10.122.115 20:00:00:0c:ce:5c:5b:c0 GigabitEthernet1/1.441 3000
1 10.10.122.115 20:00:00:0c:ce:5c:5b:c0 GigabitEthernet1/2.441 3000
-- Initiator To Interface Assignment --
--------------------------------------------------------------------------------
Initiator VR Id VRRP IP Switch WWN Ifindex
--------------------------------------------------------------------------------
```

```
iqn.cisco.test-linux.init0 1 10.10.122.115 20:00:00:0b:5f:3c:01:80 GigabitEthernet2/2.441
iqn.cisco.test-linux.init1 1 10.10.122.115 20:00:00:0c:ce:5c:5b:c0 GigabitEthernet1/1.441
iqn.cisco.test-linux.init2 1 10.10.122.115 20:00:00:0c:ce:5c:5b:c0 GigabitEthernet1/2.441
iqn.cisco.test-linux.init3 1 10.10.122.115 20:00:00:0b:5f:3c:01:80 GigabitEthernet2/1.441
iqn.cisco.test-linux.init4 1 10.10.122.115 20:00:00:0c:ce:5c:5b:c0 GigabitEthernet1/1.441
iqn.cisco.test-linux.init5 1 10.10.122.115 20:00:00:0c:ce:5c:5b:c0 GigabitEthernet1/2.441
iqn.cisco.test-linux.init6 1 10.10.122.115 20:00:00:0c:ce:5c:5b:c0 GigabitEthernet1/1.441
iqn.cisco.test-linux.init7 1 10.10.122.115 20:00:00:0c:ce:5c:5b:c0 GigabitEthernet1/2.441
iqn.cisco.test-linux.init8 1 10.10.122.115 20:00:00:0b:5f:3c:01:80 GigabitEthernet2/1.441
```

# Configuring Load Balancing Using VRRP

You must first configure VRRP on the Gigabit Ethernet interfaces on the switch that connect to the IP network before configuring VRRP for iSLB. For information on how to configure VRRP on a Gigabit Ethernet interface, see the Virtual Router Redundancy Protocol section.

## Enabling VRRP for Load Balancing

To enable or disable VRRP for iSLB, follow these steps:

### Procedure

**Step 1** Enables iSLB VRRP for IPv4 VR group 10.

```
switch# config terminal
switch(config)# islb vrrp 10 load-balance
```

**Step 2** Disables iSLB VRRP for IPv4 VR group 10.

```
switch(config)# no islb vrrp 10 load-balance
```

**Step 3** Enables iSLB VRRP for IPv6 VR group 20.

```
switch(config)# islb vrrp ipv6 20 load-balance
```

**Step 4** Disables iSLB VRRP for IPv6 VR group 20.

```
switch(config)# no islb vrrp ipv6 20 load-balance
```

## Verifying iSLB VRRP Load Balancing Configuration

To verify the iSLB VRRP load balancing configuration for IPv4, use the **show vrrp vr** command:

```
switch# show vrrp vr 1
Interface VR IpVersion Pri Time Pre State VR IP addr
---------------------------------------------------------------------------
GigE1/5 1 IPv4 100 1 s master 10.10.10.1
GigE1/6 1 IPv4 100 1 s master 10.10.10.1
```

To verify the iSLB VRRP load balancing configuration for IPv6, use the **show vrrp ipv6 vr** command:

```
switch# show vrrp ipv6 vr 1
Interface VR IpVersion Pri Time Pre State VR IP addr
---------------------------------------------------------------------
GigE6/2 1 IPv6 100 100cs master 5000:1::100
PortCh 4 1 IPv6 100 100cs master 5000:1::100
```

## Displaying iSLB VRRP Information

Use the **show islb vrrp summary vr** command to display VRRP load-balancing information:

```
switch# show islb vrrp summary vr 30

-- Groups For Load Balance --
-------------------------------------------------------------------------------------------
VR Id VRRP Address Type Configured Status
-------------------------------------------------------------------------------------------
30 IPv4 Enabled

-- Interfaces For Load Balance --
-------------------------------------------------------------------------------------------
VR Id VRRP IP Switch WWN Ifindex Load
-------------------------------------------------------------------------------------------
30 192.168.30.40 20:00:00:0d:ec:02:cb:00 GigabitEthernet3/1 2000
30 192.168.30.40 20:00:00:0d:ec:02:cb:00 GigabitEthernet3/2 2000
30 192.168.30.40 20:00:00:0d:ec:0c:6b:c0 GigabitEthernet4/1 2000
M 30 192.168.30.40 20:00:00:0d:ec:0c:6b:c0 GigabitEthernet4/2 1000
```

# iSLB Configuration Distribution Using CFS

You can distribute the configuration for iSLB initiators and initiator targets on an MDS switch. This feature lets you synchronize the iSLB configuration across the fabric from the console of a single MDS switch. The iSCSI initiator idle timeout, global authentication, and iSCSI dynamic initiator mode parameters are also distributed. CFS distribution is disabled by default.

Configuration for iSLB initiators and initiator targets on an MDS switch can be distributed using the Cisco Fabric Services (CFS). This feature allows you to synchronize the iSLB configuration across the fabric from the console of a single MDS switch. The iSCSI initiator idle timeout, iSCSI dynamic initiator mode, and global authentication parameters are also distributed. CFS distribution is disabled by default (see the Cisco Fabric Manager System Management Configuration Guide and Cisco MDS 9000 Family NX-OS System Management Configuration Guide for more information).

After enabling the distribution, the first configuration starts an implicit session. All server configuration changes entered thereafter are stored in a temporary database and applied to all switches in the fabric (including the originating one) when you explicitly commit the database.

When CFS is enabled for iSLB, the first iSLB configuration operation starts a CFS session and locks the iSLB configuration in the fabric. The configuration changes are applied to the pending configuration database. When you make the changes to the fabric, the pending configuration is distributed to all the switches in the fabric. Each switch then validates the configuration. This check ensures the following:

- The VSANs assigned to the iSLB initiators are configured on all the switches.
- The static WWNs configured for the iSLB initiators are unique and available on all the switches.
- The iSLB initiator node names do not conflict with the iSCSI initiators on all the switches.

After the check completes successfully, all the switches commit the pending configuration to the running configuration. If any check fails, the entire commit fails.

**Note** iSLB is only fully supported when CFS is enabled. Using iSLB auto-zoning without enabling CFS mode may cause traffic disruption when any zone set is activated

CFS does not distribute non-iSLB initiator configurations or import Fibre Channel target settings.

Non-iSLB virtual targets will continue to support advertised interfaces option.

**Note** The pending changes are only available in the volatile directory and are discarded if the switch is restarted.

# Distributing iSLB Configuration Using CFS

This section contains the following:

## Enabling iSLB Configuration Distribution

To enable CFS distribution of the iSLB configuration, follow these steps:

**Procedure**

**Step 1** Enters configuration mode.

```
switch# config terminal
```

**Step 2** Enables iSLB configuration distribution.

```
switch(config)# islb distribute
```

**Step 3** Disables (default) iSLB configuration distribution.

```
switch(config)# no islb distribute
```

## Locking the Fabric

The first action that modifies the existing configuration creates the pending configuration and locks the feature in the fabric. Once you lock the fabric, the following conditions apply:

- No other user can make any configuration changes to this feature.
- A pending configuration is created by copying the active configuration. Modifications from this point on are made to the pending configuration and remain there until you commit the changes to the active configuration (and other switches in the fabric) or discard them.

**Note** iSCSI configuration changes are not allowed when an iSLB CFS session is active.

# Committing Changes to the Fabric

To apply the pending iSLB configuration changes to the active configuration and to other MDS switches in the fabric, you must commit the changes. The pending configuration changes are distributed and, on a successful commit, the configuration changes are applied to the active configuration in the MDS switches throughout the fabric, the automatic zones are activated, and the fabric lock is released.

To commit iSLB configuration changes to other MDS switches in the fabric, activate iSLB automatic zones, and release the fabric lock, follow these steps:

### Procedure

**Step 1** Enters configuration mode.

```
switch# config terminal
```

**Step 2** Commits the iSLB configuration distribution, activates iSLB automatic zones, and releases the fabric lock

```
switch(config)# islb commit
```

# Discarding Pending Changes

At any time, you can discard the pending changes to the iSLB configuration and release the fabric lock. This action has no affect on the active configuration on any switch in the fabric.

To discard the pending iSLB configuration changes and release the fabric lock, follow these steps:

### Procedure

**Step 1** Enters configuration mode.

```
switch# config terminal
```

**Step 2** Commits the iSLB configuration distribution.

```
switch(config)# islb abort
```

# Clearing a Fabric Lock

If you have performed an iSLB configuration task and have not released the lock by either committing or discarding the changes, an administrator can release the lock from any switch in the fabric. If the administrator performs this task, your pending changes are discarded and the fabric lock is released.

### Before you begin

The pending changes are only available in the volatile directory and are discarded if the switch is restarted.

### Procedure

To release a fabric lock, issue the **clear islb session** command in EXEC mode using a login ID that has administrative privileges:

```
switch# clear islb session
```

# CFS Merge Process

When two fabrics merge, CFS attempts to merge the iSLB configuration from both the fabrics. A designated switch (called the *dominant switch* ) in one fabric sends its iSLB configuration to a designated switch (called the *subordinate switch* ) in the other fabric. The subordinate switch compares its running configuration to the received configuration for any conflicts. If no conflicts are detected, it merges the two configurations and sends it to all the switches in both the fabrics. Each switch then validates the configuration. This check ensures the following:

- VSANs assigned to the iSLB initiators are configured on all the switches.
- The static WWNs configured for the iSLB initiators are unique and available on all the switches.
- The iSLB initiator node names have no conflicts with iSCSI initiators on all the switches.

If this check completes successfully, the subordinate switch directs all the switches to commit the merged configuration to running configuration. If any check fails, the merge fails.

The **show islb merge status** command displays the exact reason for the failure. The first successful commit request after a merge failure takes the fabric out of the merge failure stat

# Displaying Pending iSLB Configuration Changes

You can display the pending configuration changes using the **show islb pending** command:

```
switch# show islb pending
iscsi initiator idle-timeout 10
islb initiator ip-address 10.1.1.1
static pWWN 23:01:00:0c:85:90:3e:82
static pWWN 23:06:00:0c:85:90:3e:82
username test1
islb initiator ip-address 10.1.1.2
static nWWN 23:02:00:0c:85:90:3e:82
```

You can display the differences between the pending configuration and the current configuration using the **show islb pending-diff** command:

```
switch# show islb pending-diff
+iscsi initiator idle-timeout 10
islb initiator ip-address 10.1.1.1
+ static pWWN 23:06:00:0c:85:90:3e:82
+islb initiator ip-address 10.1.1.2
+ static nWWN 23:02:00:0c:85:90:3e:82
```

# Displaying iSLB CFS Status

### Displaying iSLB CFS status using the show islb session status command:

You can display the iSLB CFS status using the **show islb session status** command:

```
switch# show islb status
iSLB Distribute is enabled
iSLB CFS Session exists
```

### Displaying iSLB CFS Distribution Session Status

You can display the status of the iSLB CFS distribution session using the **show islb cfs-session status** command:

```
switch# show islb cfs-session status
last action : fabric distribute enable
last action result : success
last action failure cause : success
```

### Displaying iSLB CFS Merge Status

You can display the iSLB CFS merge status using the **show islb merge status** command:

```
switch# show islb merge status
Merge Status: Success
```

# iSLB CFS Merge Status Conflicts

Merge conflicts may occur. User intervention is required for the following merge conflicts:

- The iSCSI global authentication or iSCSI initiator idle timeout parameters are not configured the same in the two fabrics.
- The same iSLB initiator is configured differently in the two fabrics.
- An iSLB initiator in one fabric has the same name as an iSCSI initiator in the other fabric.
- Duplicate pWWN/nWWN configuration is detected in the two fabric. For example, a pWWN/nWWN configured for an iSLB initiator on one fabric is configured for an iSCSI initiator or a different iSLB initiator in the other fabric.
- A VSAN configured for an iSLB initiator in one fabric does not exist in the other fabric.

Check the syslog for details on merge conflicts. User intervention is not required when the same iSLB initiator has a different set of non-conflicting initiator targets. The merged configuration is the union of all the initiator targets.

# iSCSI High Availability

The following high availability features are available for iSCSI configurations:

## Transparent Target Failover

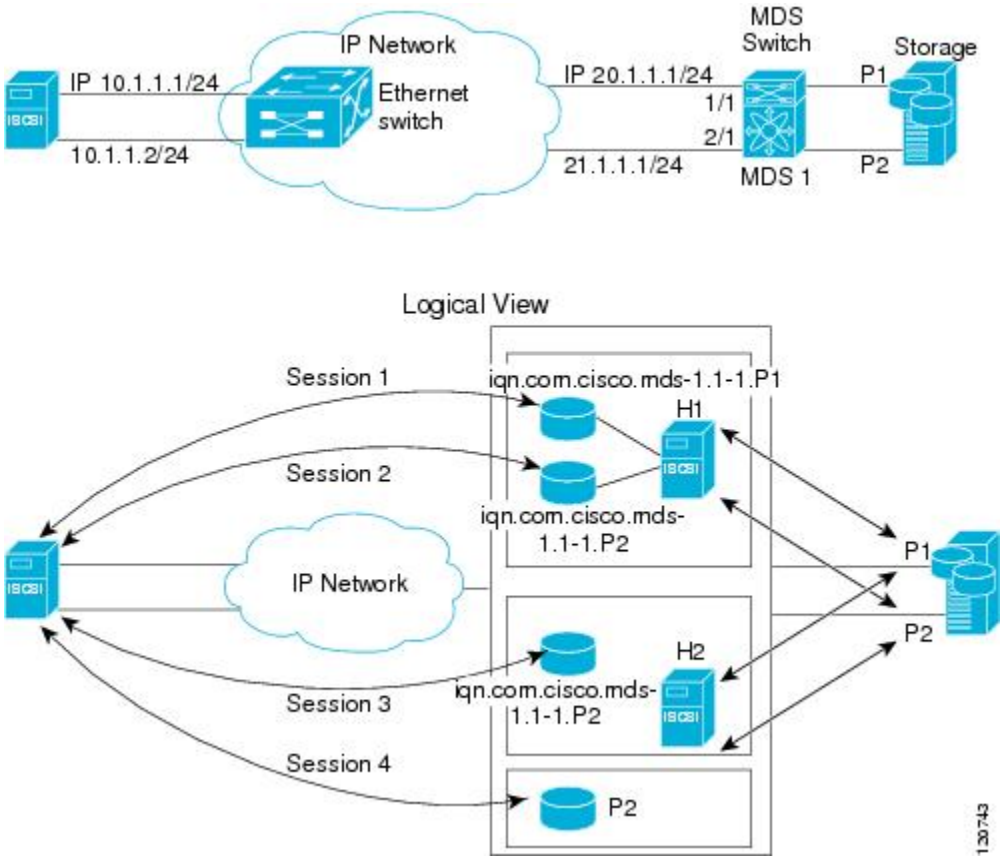
The following high-availability features are available for iSCSI configurations:

- iSCSI high availability with host running multi-path software—In this topology, you have recovery from failure of any of the components. The host multi-path software takes care of load balancing or failover across the different paths to access the storage.
- iSCSI high availability with host not having multi-path software—Without multi-path software, the host does not have knowledge of the multiple paths to the same storage.

### iSCSI High Availability with Host Running Multipath Software

The following figure shows the physical and logical topology for an iSCSI HA solution for hosts running multi-path software. In this scenario, the host has four iSCSI sessions. There are two iSCSI sessions from each host NIC to the two IPS ports.

*Figure 34: Host Running Multipath Software*



Each IPS ports is exporting the same two Fibre Channel target ports of the storage but as different iSCSI target names if you use dynamic iSCSI targets). So the two IPS ports are exporting a total of four iSCSI target devices. These four iSCSI targets map the same two ports of the Fibre Channel target.

The iSCSI host uses NIC-1 to connect to IPS port 1 and NIC-2 to connect to IPS port 2. Each IPS port exports two iSCSI targets, so the iSCSI host creates four iSCSI sessions.

If the iSCSI host NIC-1 fails, then sessions 1 and 2 fail but we still have sessions 3 and 4.

If the IPS port 1 fails, the iSCSI host cannot connect to the IPS port, and sessions 1 and 2 fail. But sessions 3 and 4 are still available.

If the storage port 1 fails, then the IPS ports will terminate sessions 1 and 3 (put iSCSI virtual target iqn.com.cisco.mds-5.1-2.p1 and iqn-com.cisco.mds-5.1-1.p1 in offline state). But sessions 2 and 4 are still available.

In this topology, you have recovery from failure of any of the components. The host multi-path software takes care of load-balancing or failover across the different paths to access the storage.

### iSCSI HA with Host Not Having Any Multipath Software

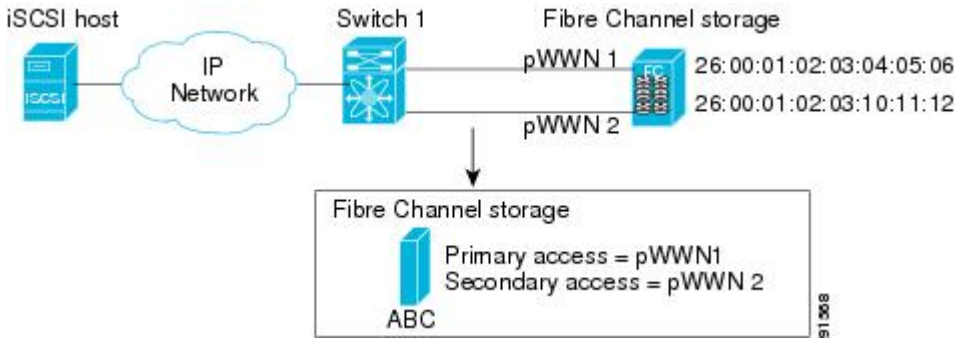
The above topology will not work if the host does not have multi-path software because the host has multiple sessions to the same storage. Without multi-path software the host does not have knowledge of the multiple paths to the same storage.

IP storage has two additional features that provide an HA solution in this scenario.

- IPS ports support the VRRP feature (see Configuring VRRP for IPStorage Interfaces section) to provide failover for IPS ports.
- IPS has transparent Fibre Channel target failover for iSCSI static virtual targets.

Statically imported iSCSI targets have an additional option to provide a secondary pWWN for the Fibre Channel target. This can be used when the physical Fibre Channel target is configured to have an LU visible across redundant ports. When the active port fails, the secondary port becomes active and the iSCSI session switches to use the new active port.

***Figure 35: Static Target Importing Through Two Fibre Channel Ports***




In the above figure, you can create an iSCSI virtual target that is mapped to both pWWN1 and pWWN2 to provide redundant access to the Fibre Channel targets.

The failover to a secondary port is done transparently by the IPS port without impacting the iSCSI session from the host. All outstanding I/Os are terminated with a check condition status when the primary port fails. New I/Os received during the failover are not completed and receive a busy status.

If you use LUN mapping, you can define a different secondary Fibre Channel LUN if the LU number is different.

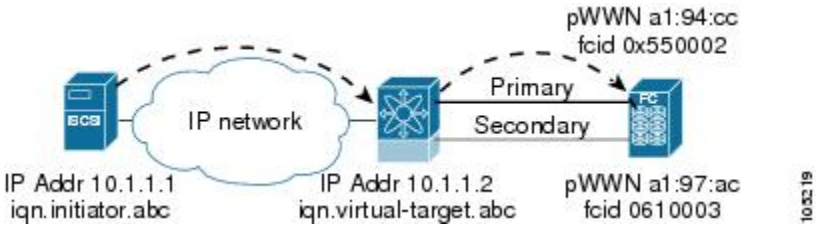
Enable the optional **revert-primary-port** option to direct the IPS port to switch back to the primary port when the primary port is up again. If this option is disabled (default) and the primary port is up again after a switchover, the old sessions will remain with the secondary port and do not switch back to the primary port. However, any new session will use the primary port. This is the only situation when both the primary and secondary ports are used at the same time.

### LUN Trespass for Storage Port Failover

In addition to the high availability of statically imported iSCSI targets, the trespass feature is available to enable the move of LUs, on an active port failure, from the active to the passive port of a statically imported iSCSI target.

In physical Fibre Channel targets, which are configured to have LUs visible over two Fibre Channel N ports, when the active port fails, the passive port takes over. Some physical Fibre Channel targets require that the trespass feature be used to move the LUs from the active port to the passive port. A statically imported iSCSI target's secondary pWWN option and an additional option of enabling the trespass feature is available for a physical Fibre Channel target with redundant ports. When the active port fails, the passive port becomes active, and if the trespass feature is enabled, the Cisco MDS switch sends a request to the target to move the LUs on the new active port. The iSCSI session switches to use the new active port and the moved LUs are accessed over the new active port.

*Figure 36: Virtual Target with an Active Primary Port*




# Createing a Static iSCSI Virtual Target

To create a static iSCSI virtual target, follow these steps:

### Procedure

**Step 1** Create the iSCSI target name iqn.1987-02.com.cisco.initiator.

```
switch# config terminal
switch(config)# iscsi virtual-target name iqn.1987-02.com.cisco.initiator
```

**Step 2** Configure the primary port for this virtual target.

```
switch(config-iscsi-tgt)# pwwn 26:00:01:02:03:04:05:06
```

**Step 3** Configure the primary and secondary ports for this virtual target.

```
switch(config-iscsi-tgt)# pwwn 26:00:01:02:03:04:05:06 secondary-pwwn 26:00:01:02:03:10:11:12
```

**Step 4** Configure the primary port for this virtual target with LUN mapping and different LUN on the secondary Fibre Channel port.

```
switch(config-iscsi-tgt)# pwwn 26:00:01:02:03:04:05:06 fc-lun 0x1 iscsi-lun 0x0 sec-lun 0x3
```

**Step 5** Remove the primary port, secondary port, and LUN mapping configuration for this virtual target.

```
switch(config-iscsi-tgt)# no pwwn 26:00:01:02:03:04:05:06
```

**Step 6** Configure the session failover redundancy for this virtual-target to switch all sessions back to primary port when the primary port comes back up.

```
switch(config-iscsi-tgt)# revert-primary-port
```

**Step 7** Direct the switch to continue using the secondary port for existing sessions and to use the primary port for new sessions (default).

```
switch(config-iscsi-tgt)# no revert-primary-port
```

## Enabling Trespass Feature for a Static iSCSI Virtual Target

To enable the trespass feature for a static iSCSI virtual target, follow these steps:

### Procedure

**Step 1** Create the iSCSI target name iqn.1987-02.com.cisco.initiator.

```
switch# config terminal
switch(config)# iscsi virtual-target name iqn.1987-02.com.cisco.initiator
switch(config-iscsi-tgt)#
```

**Step 2** Map a virtual target node to a Fibre Channel target and configures a secondary pWWN.

```
switch(config-iscsi-tgt)# pwwn 50:00:00:a1:94:cc secondary-pwwn 50:00:00:a1:97:ac
```

**Step 3** Enable the trespass feature.

```
switch(config-iscsi-tgt)# trespass
```

**Step 4** Disable the trespass feature (default).

```
switch(config-iscsi-tgt)# no trespass
```

### Verifying trespass feature for a static iSCSI virtual target

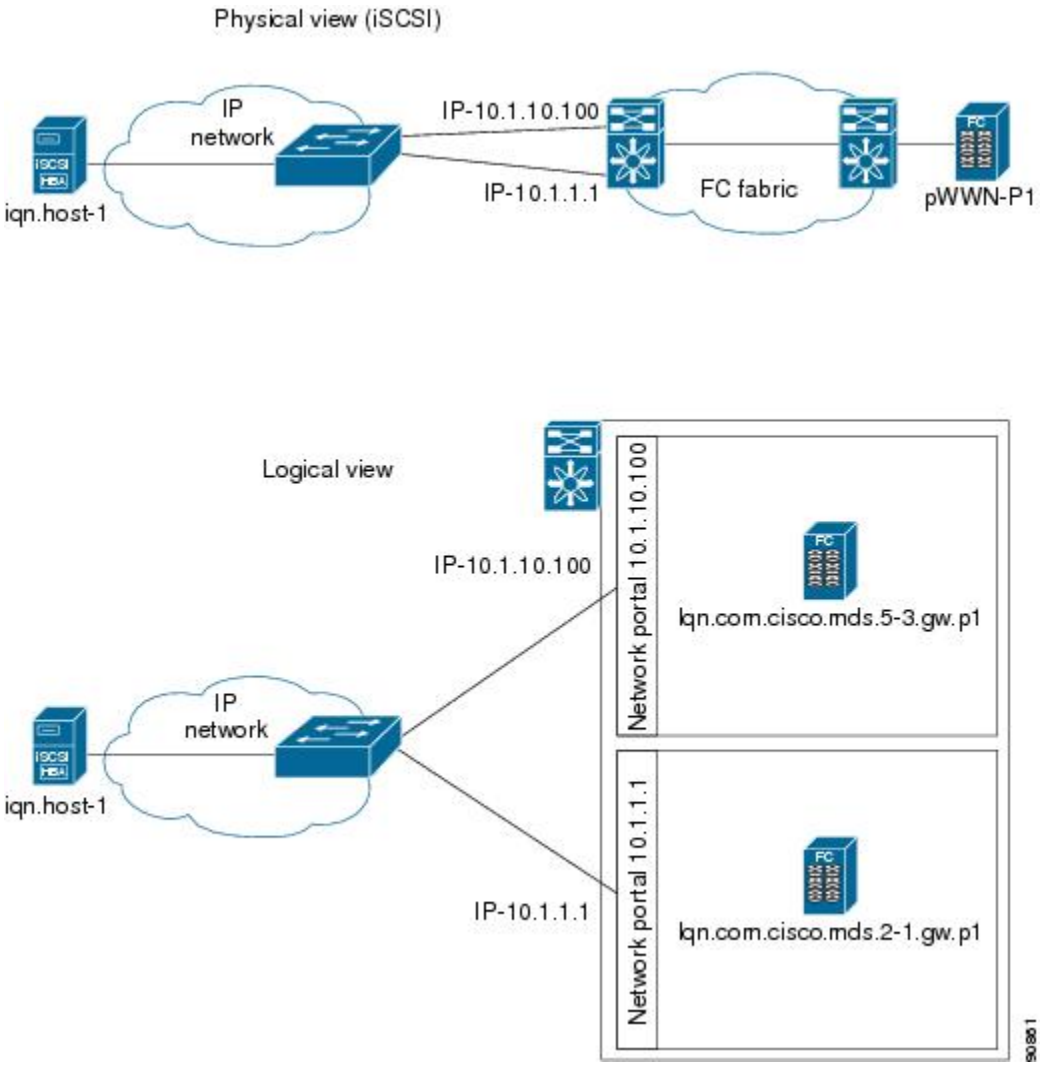
Use the **show iscsi virtual-target** command to verify trespass feature for a static iSCSI virtual target:

```
switch# show iscsi virtual-target iqn.1987-02.com.cisco.initiator
target: 1987-02.com.cisco.initiator
Port WWN 10:20:10:00:56:00:70:50
Configured node
all initiator permit is disabled
trespass support is enabled
```

# Multiple IPS Ports Connected to the Same IP Network

In the following figure, each iSCSI host discovers two iSCSI targets for every physical Fibre Channel target (with different names). The multi-pathing software on the host provides load-balancing over both paths. If one Gigabit Ethernet interface fails, the host multi-pathing software is not affected because it can use the second path.

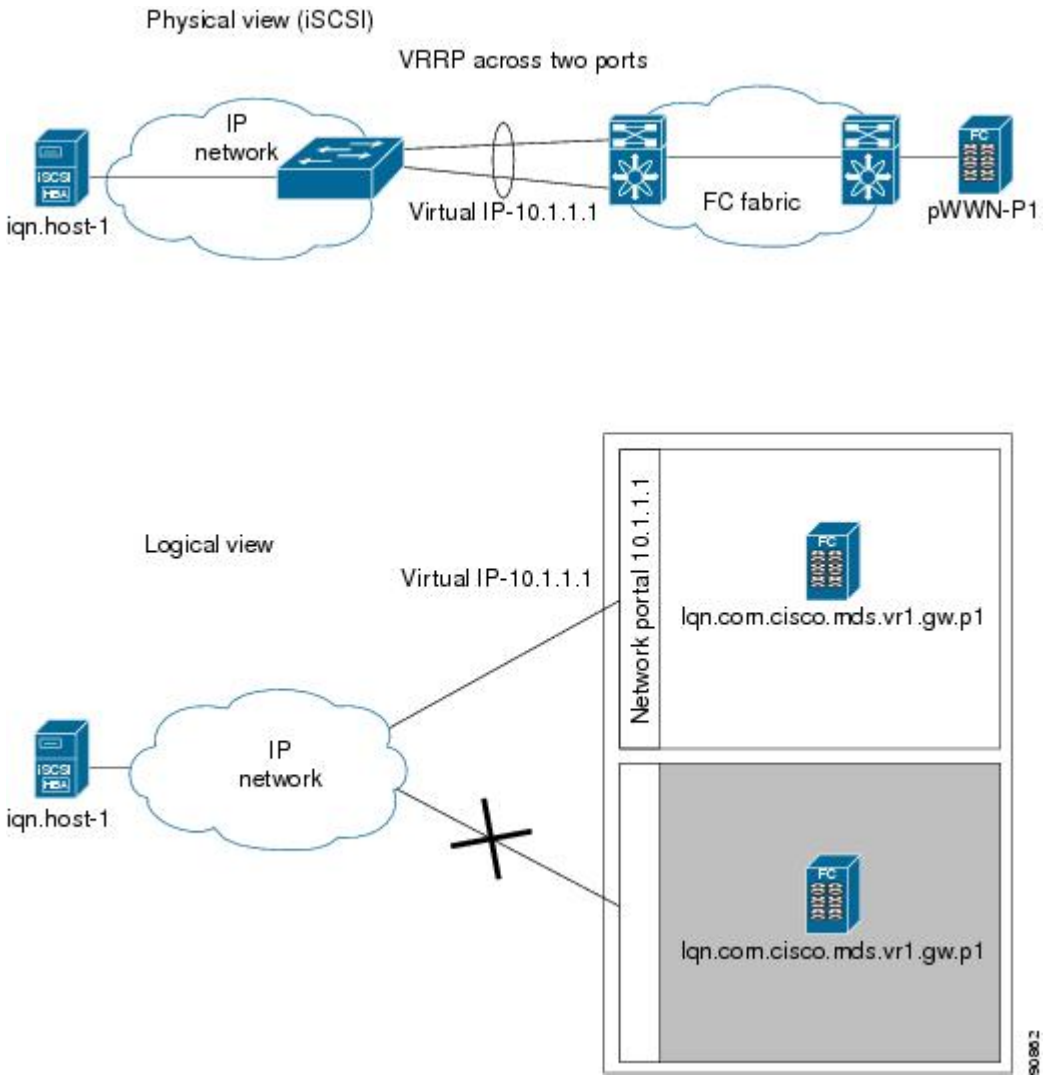
***Figure 37: Multiple Gigabit Ethernet Interfaces in the Same IP Network***




# VRRP-Based High Availability

In the following figure, each iSCSI host discovers one iSCSI target for every physical Fibre Channel target. When the Gigabit Ethernet interface of the VRRP master fails, the iSCSI session is terminated. The host then reconnects to the target and the session comes up because the second Gigabit Ethernet interface has taken over the virtual IP address as the new master.

*Figure 38: VRRP-Based iSCSI High Availability*



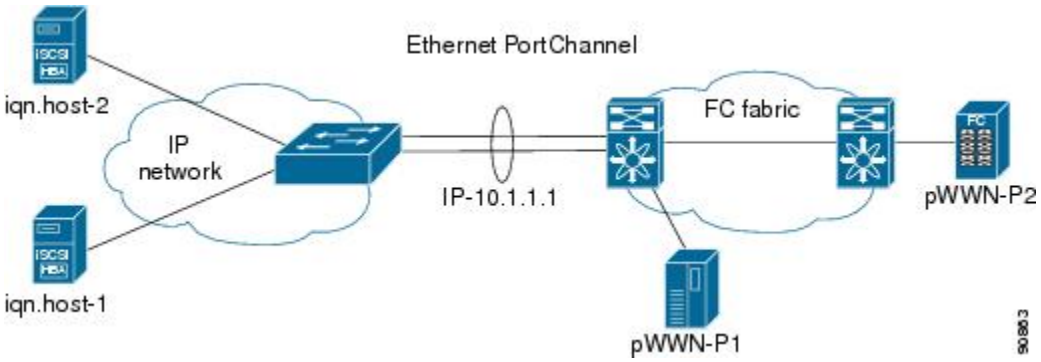

# Ethernet Port Channel-Based High Availability

In the following figure, each iSCSI host discovers one iSCSI target for every physical Fibre Channel target. The iSCSI session from the iSCSI host to the iSCSI virtual target (on the IPS port) uses one of the two physical interfaces (because an iSCSI session uses one TCP connection). When the Gigabit Ethernet interface fails, the Fibre Channel module with IPS ports and the Ethernet switch transparently forwards all the frames on to the second Gigabit Ethernet interface.

**Note** All iSCSI data traffic for one iSCSI link is carried on one TCP connection. Consequently, the aggregated bandwidth is 1 Gbps for that iSCSI link.

**Note** If an Ethernet port channel is configured between the Fibre Channel module with IPS ports and an Ethernet switch, the load balancing policy on the Ethernet switch must be based on source/destination IP address only, not port numbers, for load balancing with VRRP to operate correctly.

*Figure 39: Ethernet Port Channel-Based iSCSI High Availability*




# iSCSI Authentication Setup Guidelines and Scenarios

This section provides guidelines on iSCSI authentication possibilities, setup requirements, and sample scenarios.

**Note** This section does not specify the steps to enter or exit EXEC mode, configuration mode, or any submode. Be sure to verify the prompt before entering any command.

**Note** Changing the authentication of an iSCSI interface that is part of an iSLB VRRP group impacts load balancing on the interface. See the Changing iSCSI Interface Parameters and the Impact on Load Balancing section.

It includes the following authentication setup guidelines:

## Configuring No Authentication

You can set the iSCSI authentication to **none** to configure a network with no authentication. Use the following ommand to configure no authentication:

```
switch(config)# iscsi authentication none
```

## Configuring CHAP with Local Password Database

To configure authentication using the CHAP option with the local password database, follow these steps:

### Procedure

**Step 1** Set the AAA authentication to use the local password database for the iSCSI protocol:

```
switch(config)# aaa authentication iscsi default local
```

**Step 2** Set the iSCSI authentication method to require CHAP for all iSCSI clients:

```
switch(config)# iscsi authentication chap
```

**Step 3** Configure the user names and passwords for iSCSI users:

```
switch(config)# username iscsi-user password abcd iscsi
```

**Note**
If you do not specify the **iscsi** option, the user name is assumed to be a Cisco MDS switch user instead of an iSCSI user.

**Step 4** Verify the global iSCSI authentication setup:

```
switch# show iscsi global
iSCSI Global information Authentication: CHAP <----Verify
Import FC Target: Disabled
.
.
.
```

# Configuring CHAP with External RADIUS Server

To configure authentication using the CHAP option with an external RADIUS server, follow these steps:

### Before you begin

### Procedure

**Step 1** Configure the password for the Cisco MDS switch as RADIUS client to the RADIUS server:

```
switch(config)# radius-server key mds-1
```

**Step 2** Configure the RADIUS server IP address by performing one of the following:

a) Configure an IPv4 address:

```
switch(config)# radius-server host 10.1.1.10
```

b) Configure an IPv6 address:

```
switch(config)# radius-server host 2001:0DB8:800:200C::417A
```

**Step 3** Configure the RADIUS server group IP address by performing one of the following:

a) Configure an IPv4 address:

```
switch(config)# aaa group server radius iscsi-radius-group
switch(config-radius)# server 10.1.1.1
```

b) Configure an IPv6 address:

```
switch(config)# aaa group server radius iscsi-radius-group
switch(config-radius)# server 001:0DB8:800:200C::4180
switch(config)# aaa authentication iscsi default group iscsi-radius-group
```

**Step 4** Set up the iSCSI authentication method to require CHAP for all iSCSI clients:

```
switch(config)# iscsi authentication chap
```

**Step 5** Verify that the global iSCSI authentication setup is for CHAP:

```
switch# show iscsi global
iSCSI Global information
Authentication: CHAP <---------------- Verify CHAP
.
.
.
```

**Step 6** Verify that the AAA authentication information is for iSCSI

```
switch# show aaa authentication
default: local
console: local
iscsi: group iscsi-radius-group <--------- Group name
dhchap: local

switch# show radius-server groups
total number of groups:2

following RADIUS server groups are configured:
group radius:
server: all configured radius servers
group iscsi-radius-group:
server: 10.1.1.1 on auth-port 1812, acct-port 1813

switch# show radius-server
Global RADIUS shared secret:mds-1 <-------- Verify secret
.
.
.
.

following RADIUS servers are configured:
10.1.1.1: <----------- Verify the server IPv4 address
available for authentication on port:1812
available for accounting on port:1813
```

# iSCSI Transparent Mode Initiator

This configuration assumes the following configuration:

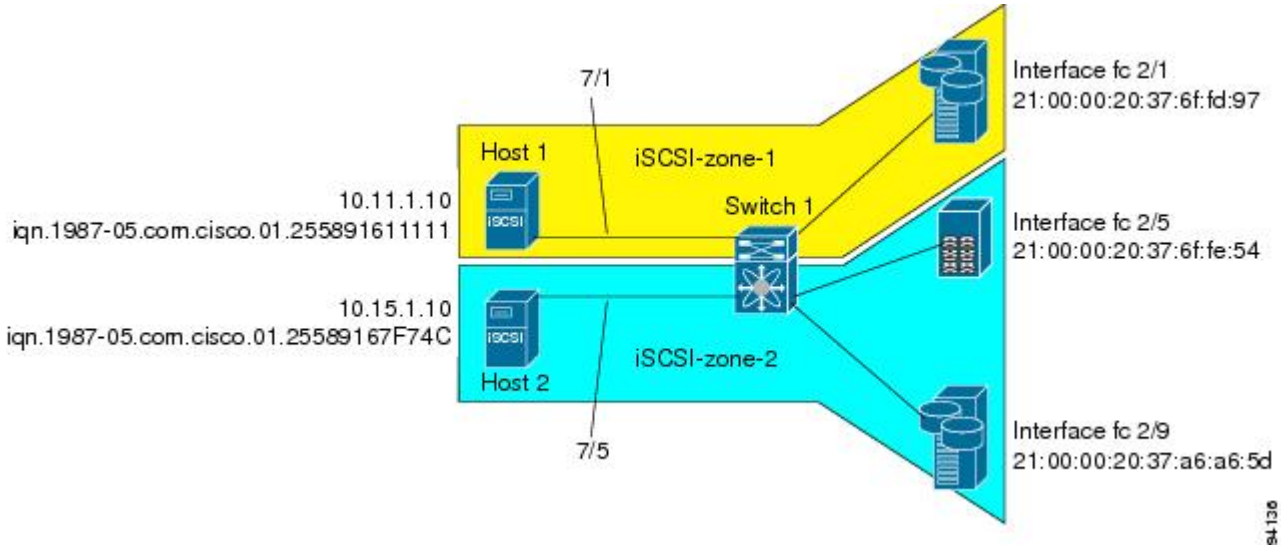
- No LUN mapping or LUN masking or any other access control for hosts on the target device
- No iSCSI login authentication (that is, login authentication set to none)
- The topology is as follows:
  - iSCSI interface 7/1 is configured to identify initiators by IP address.

- iSCSI interface 7/5 is configured to identify initiators by node name.
- The iSCSI initiator host 1 with IPv4 address 10.11.1.10 and name iqn.1987-05.com.cisco:01.255891611111 connects to IPS port 7/1 is identified using IPv4 address (host 1 = 10.11.1.10).
- The iSCSI initiator host 2 with IPv4 address 10.15.1.10 and node name iqn.1987-05.com.cisco:01.25589167f74c connects to IPS port 7/5.

## Configuring iSCSI Scenario 1

the following figure represents scenario1. To configure scenario 1, follow these steps:

***Figure 40: iSCSI Scenario 1***




### Procedure

**Step 1** Configure null authentication for all iSCSI hosts in Cisco MDS switches:

```
switch(config)# iscsi authentication none
```

**Step 2** Configure iSCSI to dynamically import all Fibre Channel targets into the iSCSI SAN using auto-generated iSCSI target names:

```
switch(config)# iscsi import target fc
```

**Step 3** Configure the Gigabit Ethernet interface in slot 7 port 1 with an IPv4 address and enable the interface:

```
switch(config)# interface gigabitethernet 7/1
switch(config-if)# ip address 10.11.1.1 255.255.255.0
switch(config-if)# no shutdown
```

**Note**
Host 2 is connected to this port.

**Step 4** Configure the iSCSI interface in slot 7 port 1 to identify all dynamic iSCSI initiators by their IP address, and enable the interface:

```
switch(config)# interface iscsi 7/1
switch(config-if)# switchport initiator id ip-address
switch(config-if)# no shut
```

**Step 5** Configure the Gigabit Ethernet interface in slot 7 port 5 with the IPv4 address and enable the interface:

```
switch(config)# interface gigabitethernet 7/5
switch(config-if)# ip address 10.15.1.1 255.255.255.0
switch(config-if)# no shutdown
```

**Step 6** Configure the iSCSI interface in slot 7 port 5 to identify all dynamic iSCSI initiators by node name and enable the interface:

```
switch(config)# interface iscsi 7/5
switch(config-if)# switchport initiator id name
switch(config-if)# no shutdown
```

**Note**
Host 1 is connected to this port.

**Step 7** Verify the available Fibre Channel targets:

```
switch# show fcns database
VSAN 1:
------------------------------------------------------------------------
FCID TYPE PWWN (VENDOR) FC4-TYPE:FEATURE
------------------------------------------------------------------------
0x6d0001 NL 21:00:00:20:37:6f:fd:97 (Seagate) scsi-fcp:target
0x6d0101 NL 21:00:00:20:37:6f:fe:54 (Seagate) scsi-fcp:target
0x6d0201 NL 21:00:00:20:37:a6:a6:5d (Seagate) scsi-fcp:target
Total number of entries = 3
```

**Step 8** Create a zone named *iscsi-zone-1* with host 1 and one Fibre Channel target in it:

```
switch(config)# zone name iscsi-zone-1 vsan 1
switch(config-zone)# member pwwn 21:00:00:20:37:6f:fd:97
switch(config-zone)# member ip-address 10.11.1.10
```

**Note**
Use the IP address of the host in zone membership configuration because the iSCSI interface is configured to identify all hosts based on IP address.

**Step 9** Create a zone named *iscsi-zone-2* with host 2 and two Fibre Channel targets in it:

```
switch(config)# zone name iscsi-zone-2 vsan 1
switch(config-zone)# member pwwn 21:00:00:20:37:6f:fe:54
switch(config-zone)# member pwwn 21:00:00:20:37:a6:a6:5d
switch(config-zone)# member symbolic-nodename iqn.1987-05.com.cisco:01.25589167f74c
```

**Note**
Use the symbolic node name of the iSCSI host in zone membership configuration because the iSCSI interface is configured to identify all hosts based on node name.

**Step 10** Create a zone set and add the two zones as members:

```
switch(config)# zoneset name zoneset-iscsi vsan 1
switch(config-zoneset)# member iscsi-zone-1
switch(config-zoneset)# member iscsi-zone-2
```

**Step 11** Activate the zone set:

```
switch(config)# zoneset activate name zoneset-iscsi vsan 1
```

**Step 12** Display the active zone set:

```
switch# show zoneset active
zoneset name zoneset-iscsi vsan 1
zone name iscsi-zone-1 vsan 1
* fcid 0x6d0001 [pwwn 21:00:00:20:37:6f:fd:97] <-----------Target
symbolic-nodename 10.11.1.10 <--------------------------------- iSCSI host (host 1, not online)

zone name iscsi-zone-2 vsan 1
* fcid 0x6d0101 [pwwn 21:00:00:20:37:6f:fe:54] <------------Target
* fcid 0x6d0201 [pwwn 21:00:00:20:37:a6:a6:5d] <------------Target
symbolic-nodename iqn.1987-05.com.cisco:01.25589167f74c <-iSCSI host (host 2, not online)
```

**Step 13** Bring up the iSCSI hosts (host 1 and host 2).

**Step 14** Show all the iSCSI sessions (use the **detail** option for detailed information):

```
switch# show iscsi session
Initiator iqn.1987-05.com.cisco:01.25589167f74c <-----Host 2
Initiator ip addr (s): 10.15.1.11
Session #1
Target iqn.1987-05.com.cisco:05.172.22.92.166.07-05.21000020376ffe54
VSAN 1, ISID 00023d000001, Status active, no reservation

Session #2
Target iqn.1987-05.com.cisco:05.172.22.92.166.07-05.2100002037a6a65d
VSAN 1, ISID 00023d000001, Status active, no reservation

Initiator 10.11.1.10 <----------------------------------Host 1
Initiator name iqn.1987-05.com.cisco:01.e41695d16b1a
Session #1
Target iqn.1987-05.com.cisco:05.172.22.92.166.07-01.21000020376ffd97
VSAN 1, ISID 00023d000001, Status active, no reservation
```

**Note**
The last part of the auto-created target name is the Fibre Channel target's pWWN.

**Step 15** Verify the details of the two iSCSI initiators:

Host 2: Initiator ID based on node name because the initiator is entering iSCSI interface 7/5:

```
switch# show iscsi initiator

iSCSI Node name is iqn.1987-05.com.cisco:01.25589167f74c
Initiator ip addr (s): 10.15.1.11
iSCSI alias name: oasis11.cisco.com
Node WWN is 20:02:00:0b:fd:44:68:c2 (dynamic)
Member of vsans: 1
Number of Virtual n_ports: 1
Virtual Port WWN is 20:03:00:0b:fd:44:68:c2 (dynamic)
Interface iSCSI 7/5 , Portal group tag: 0x304
VSAN ID 1, FCID 0x6d0300
```

Host 1: Initiator ID based on IPv4 address because the initiator is entering iSCSI interface 7/1:

```
iSCSI Node name is 10.11.1.10
iSCSI Initiator name: iqn.1987 - 05.com.cisco:01.e41695d16b1a
iSCSI alias name: oasis10.cisco.com
Node WWN is 20:04:00:0b:fd:44:68:c2 (dynamic)
Member of vsans: 1
Number of Virtual n_ports: 1
```

```
Virtual Port WWN is 20:05:00:0b:fd:44:68:c2 (dynamic)
Interface iSCSI 7/1 , Portal group tag: 0x300
VSAN ID 1, FCID 0x6d0301
```

**Step 16** View the active zone set. The iSCSI initiators' FC IDs are resolved:

FC ID resolved forhost 1:

```
switch# show zoneset active
zoneset name zoneset-iscsi vsan 1
zone name iscsi-zone-1 vsan 1
* fcid 0x6d0001 [pwwn 21:00:00:20:37:6f:fd:97]
* fcid 0x6d0301 [symbolic-nodename 10.11.1.10]
```

FC ID for host 2:

```
zone name iscsi-zone-2 vsan 1
* fcid 0x6d0101 [pwwn 21:00:00:20:37:6f:fe:54]
* fcid 0x6d0201 [pwwn 21:00:00:20:37:a6:a6:5d]
* fcid 0x6d0300 [symbolic-nodename iqn.1987-05.com.cisco:01.25589167f74c] <-------------------------
```

**Step 17** The Fibre Channel name server shows the virtual N ports created for the iSCSI hosts:

```
switch# show fcns database
VSAN 1:
--------------------------------------------------------------------------
FCID TYPE PWWN (VENDOR) FC4-TYPE:FEATURE
--------------------------------------------------------------------------
0x6d0001 NL 21:00:00:20:37:6f:fd:97 (Seagate) scsi-fcp:target
0x6d0101 NL 21:00:00:20:37:6f:fe:54 (Seagate) scsi-fcp:target
0x6d0201 NL 21:00:00:20:37:a6:a6:5d (Seagate) scsi-fcp:target
0x6d0300 N 20:03:00:0b:fd:44:68:c2 (Cisco) scsi-fcp:init isc..w
0x6d0301 N 20:05:00:0b:fd:44:68:c2 (Cisco) scsi-fcp:init isc..w
```

**Step 18** Verify the detailed output of the iSCSI initiator nodes in the Fibre Channel name server:

```
switch# show fcns database fcid 0x6d0300 detail vsan 1
------------------------
VSAN:1 FCID:0x6d0300
------------------------
port-wwn (vendor) :20:03:00:0b:fd:44:68:c2 (Cisco)
node-wwn :20:02:00:0b:fd:44:68:c2
class :2,3
node-ip-addr :10.15.1.11
ipa :ff ff ff ff ff ff ff ff
fc4-types:fc4_features:scsi-fcp:init iscsi-gw
symbolic-port-name :
symbolic-node-name :iqn.1987-05.com.cisco:01.25589167f74c <--------------------
port-type :N
port-ip-addr :0.0.0.0
fabric-port-wwn :21:91:00:0b:fd:44:68:c0
hard-addr :0x000000
Total number of entries = 1

switch# show fcns database fcid 0x6d0301 detail vsan 1
------------------------
VSAN:1 FCID:0x6d0301
------------------------
port-wwn (vendor) :20:05:00:0b:fd:44:68:c2 (Cisco)
node-wwn :20:04:00:0b:fd:44:68:c2
class :2,3
node-ip-addr :10.11.1.10
ipa :ff ff ff ff ff ff ff ff
fc4-types:fc4_features:scsi-fcp:init iscsi-gw
symbolic-port-name :
```

```
symbolic-node-name :10.11.1.10
port-type :N
port-ip-addr :0.0.0.0
fabric-port-wwn :21:81:00:0b:fd:44:68:c0
hard-addr :0x000000
```

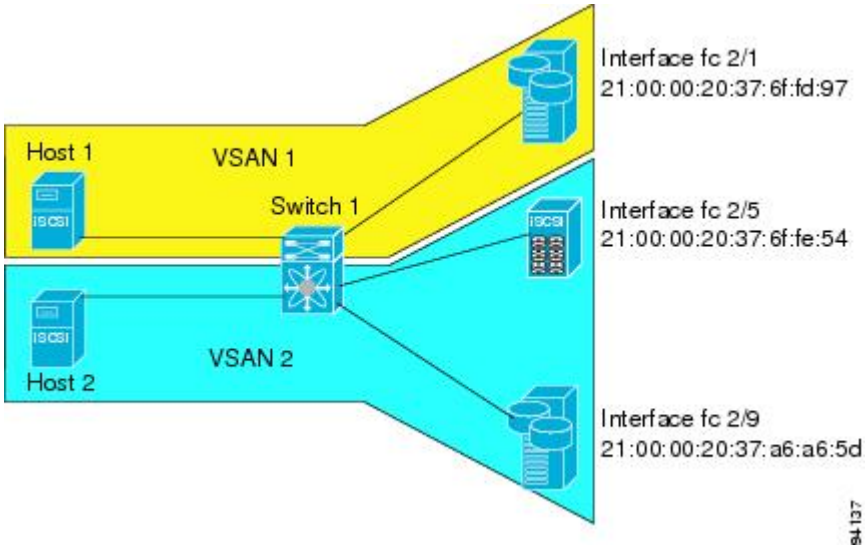
## Configuring iSCSI Scenario 2

Sample scenario 2 assumes the following configuration:

- Access control is based on Fibre Channel zoning.
- There is target-based LUN mapping or LUN masking.
- There is no iSCSI authentication (none).
- The iSCSI initiator is assigned to different VSANs.

The following figure represents scenario2. To configure scenario 2, follow these steps:

***Figure 41: iSCSI Scenario 2***




### Procedure

**Step 1** Configure null authentication for all iSCSI hosts:

```
switch(config)# iscsi authentication none
```

**Step 2** Configure iSCSI to dynamically import all Fibre Channel targets into the iSCSI SAN using auto-generated iSCSI target names:

```
switch(config)# iscsi import target fc
```

**Step 3** Configure the Gigabit Ethernet interface in slot 7 port 1 with an IPv4 address and enable the interface:

```
switch(config)# interface gigabitethernet 7/1
switch(config-if)# ip address 10.11.1.1 255.255.255.0
switch(config-if)# no shutdown
```

**Step 4** Configure the iSCSI interface in slot 7 port 1 to identify all dynamic iSCSI initiators by their IP address and enable the interface:

```
switch(config)# interface iscsi 7/1
switch(config-if)# switchport initiator id ip-address
switch(config-if)# no shutdown
```

**Step 5** Configure the Gigabit Ethernet interface in slot 7 port 5 with the IPv4 address and enable the interface:

```
switch(config)# interface gigabitethernet 7/5
switch(config-if)# ip address 10.15.1.1 255.255.255.0
switch(config-if)# no shutdown
```

**Step 6** Configure the iSCSI interface in slot 7 port 5 to identify all dynamic iSCSI initiators by IP address and enable the interface:

```
switch(config)# interface iscsi 7/5
switch(config-if)# switchport initiator id ip-address
switch(config-if)# no shutdown
```

**Step 7** Add static configuration for each iSCSI initiator:

```
switch(config)# iscsi initiator name iqn.1987-05.com.cisco:01.e41695d16b1a <-----Host 2
switch(config-iscsi-init)# static pWWN system-assign 1
switch(config-iscsi-init)# static nWWN system-assign
switch(config)# iscsi initiator ip address 10.15.1.11 <-----------------------------------Host 1
switch(config-iscsi-init)# static pwwn system-assigned 1
switch(config-iscsi-init)# vsan 2
```

**Note**
Host 1 is configured in VSAN 2.

**Step 8** View the configured WWNs:

```
switch# show iscsi initiator configured
iSCSI Node name is iqn.1987-05.com.cisco:01.e41695d16b1a
Member of vsans: 1
Node WWN is 20:03:00:0b:fd:44:68:c2
No. of PWWN: 1
Port WWN is 20:02:00:0b:fd:44:68:c2

iSCSI Node name is 10.15.1.11
Member of vsans: 2
No. of PWWN: 1
Port WWN is 20:06:00:0b:fd:44:68:c2
```

**Note**
The WWNs are assigned by the system. The initiators are members of different VSANs.

**Step 9** Create a zone with host 1:

```
switch(config)# zone name iscsi-zone-1 vsan 1
```

**Step 10** Add three members to the zone named *iscsi-zone-1* :

a) The following command is based on the symbolic node name.

```
switch(config-zone)# member symbolic-nodename iqn.1987-05.com.cisco:01.e41695d16b1a
```

b) The following command is based on the persistent pWWN assigned to the initiator. You can obtain the pWWN from the **show iscsi initiator** output.

```
switch(config-zone)# member pwwn 20:02:00:0b:fd:44:68:c2
```

**Step 11** Create a zone with host 2 and two Fibre Channel targets:

```
switch(config)# zone name iscsi-zone-2 vsan 2
```

**Note**
If the host is in VSAN 2, the Fibre Channel targets and zone must also be in VSAN 2.

**Step 12** Activate the zone set in VSAN 2:

```
switch(config)# zoneset activate name iscsi-zoneset-v2 vsan 2
Zoneset activation initiated. check zone status
switch# show zoneset active vsan 2
zoneset name iscsi-zoneset-v2 vsan 2
zone name iscsi-zone-2 vsan 2
* fcid 0x750001 [pwwn 21:00:00:20:37:6f:fe:54]
* fcid 0x750101 [pwwn 21:00:00:20:37:a6:a6:5d]
pwwn 20:06:00:0b:fd:44:68:c2
```

**Step 13** Start the iSCSI clients on both hosts and verify that sessions come up.

**Step 14** Display the iSCSI sessions to verify the Fibre Channel target and the configured WWNs:

```
switch# show iscsi session
Initiator iqn.1987-05.com.cisco:01.e41695d16b1a
Initiator ip addr (s): 10.11.1.10
Session #1
Discovery session, ISID 00023d000001, Status active
Session #2
Target iqn.1987-05.com.cisco:05.172.22.92.166.07-01.21000020376ffd97 <----
VSAN 1, ISID 00023d000001, Status active, no reservation
```

**Step 15** Display the iSCSI initiator to verify the configured nWWN and pWWN:

```
switch# show iscsi initiator
iSCSI Node name is iqn.1987-05.com.cisco:01.e41695d16b1a
Initiator ip addr (s): 10.11.1.10
iSCSI alias name: oasis10.cisco.com
Node WWN is 20:03:00:0b:fd:44:68:c2 (configured) <----------
Member of vsans: 1
Number of Virtual n_ports: 1
Virtual Port WWN is 20:02:00:0b:fd:44:68:c2 (configured) <----
Interface iSCSI 7/1, Portal group tag: 0x300
VSAN ID 1, FCID 0x680102
```

**Step 16** Check the Fibre Channel name server:

```
switch# show fcns database vsan 1
VSAN 1:
--------------------------------------------------------------------
FCID TYPE PWWN (VENDOR) FC4-TYPE:FEATURE
--------------------------------------------------------------------
0x680001 NL 21:00:00:20:37:6f:fd:97 (Seagate) scsi-fcp:target
0x680102 N 20:02:00:0b:fd:44:68:c2 (Cisco) scsi-fcp:init iscw
```

**Step 17** Verify the details of the iSCSI initiator's FC ID in the name server:

```
switch(config)# show fcns database fcid 0x680102 detail vsan 1
-----------------------
VSAN:1 FCID:0x680102
```

```
------------------------
port-wwn (vendor) :20:02:00:0b:fd:44:68:c2 (Cisco)
node-wwn :20:03:00:0b:fd:44:68:c2
class :2,3
node-ip-addr :10.11.1.10
ipa :ff ff ff ff ff ff ff ff
fc4-types:fc4_features:scsi-fcp:init iscsi-gw
symbolic-port-name :
symbolic-node-name :iqn.1987-05.com.cisco:01.e41695d16b1a
port-type :N
port-ip-addr :0.0.0.0
fabric-port-wwn :21:81:00:0b:fd:44:68:c0
iSCSI alias name: oasis10.cisco.com
```

**Step 18** Check the Fibre Channel name server:

```
switch# show fcns database vsan 1

VSAN 1:
--------------------------------------------------------------------
FCID TYPE PWWN (VENDOR) FC4-TYPE:FEATURE
--------------------------------------------------------------------
0x680001 NL 21:00:00:20:37:6f:fd:97 (Seagate) scsi-fcp:target
0x680102 N 20:02:00:0b:fd:44:68:c2 (Cisco) scsi-fcp:init isc..w
```

**Step 19** Verify the details of the iSCSI initiator's FC ID in the name server:

```
switch(config)# show fcns database fcid 0x680102 detail vsan 1
------------------------
VSAN:1 FCID:0x680102
------------------------
port-wwn (vendor) :20:02:00:0b:fd:44:68:c2 (Cisco)
node-wwn :20:03:00:0b:fd:44:68:c2
class :2,3
node-ip-addr :10.11.1.10
ipa :ff ff ff ff ff ff ff ff
fc4-types:fc4_features:scsi-fcp:init iscsi-gw
symbolic-port-name :
symbolic-node-name :iqn.1987-05.com.cisco:01.e41695d16b1a
port-type :N
port-ip-addr :0.0.0.0
fabric-port-wwn :21:81:00:0b:fd:44:68:c0
hard-addr :0x000000
```

**Step 20** Verify that zoning has resolved the FC ID for the iSCSI client:

```
switch# show zoneset active vsan 1
zoneset name iscsi-zoneset-v1 vsan 1
zone name iscsi-zone-1 vsan 1
* fcid 0x680001 [pwwn 21:00:00:20:37:6f:fd:97]
* fcid 0x680102 [pwwn 20:02:00:0b:fd:44:68:c2]
```

**Step 21** Verify that the second initiator is connected to the two Fibre Channel targets in VSAN 2:

The following command displays session to first target and session to second target:

```
switch# show iscsi session initiator 10.15.1.11
Initiator 10.15.1.11
Initiator name iqn.1987-05.com.cisco:01.25589167f74c
Session #1
Target iqn.1987-05.com.cisco:05.172.22.92.166.07-05.21000020376ffe54 <--
VSAN 2, ISID 00023d000001, Status active, no reservation
Session #2
Target iqn.1987-05.com.cisco:05.172.22.92.166.07-05.2100002037a6a65d <--
VSAN 2, ISID 00023d000001, Status active, no reservation
```

The following command displays dynamic WWN as static WWN not assigned and static pWWN for the initiator:

```
switch# show iscsi initiator
iSCSI Node name is 10.15.1.11
iSCSI Initiator name: iqn.1987-05.com.cisco:01.25589167f74c
iSCSI alias name: oasis11.cisco.com
Node WWN is 20:04:00:0b:fd:44:68:c2 (dynamic)
Member of vsans: 2
Number of Virtual n_ports: 1
Virtual Port WWN is 20:06:00:0b:fd:44:68:c2 (configured)
Interface iSCSI 7/5, Portal group tag: 0x304
VSAN ID 2, FCID 0x750200
```

The following output displays iSCSI initiator entry in name server

```
switch# show fcns database vsan 2
VSAN 2:
FCID TYPE PWWN (VENDOR) FC4-TYPE:FEATURE
0x750001 NL 21:00:00:20:37:6f:fe:54 (Seagate) scsi-fcp:target
0x750101 NL 21:00:00:20:37:a6:a6:5d (Seagate) scsi-fcp:target
0x750200 N 20:06:00:0b:fd:44:68:c2 (Cisco) scsi-fcp:init isc..w
Total number of entries = 3
```

```
switch# show fcns database fcid 0x750200 detail vsan 2
------------------------
VSAN:2 FCID:0x750200
------------------------
port-wwn (vendor) :20:06:00:0b:fd:44:68:c2 (Cisco)
node-wwn :20:04:00:0b:fd:44:68:c2
class :2,3
node-ip-addr :10.15.1.11
ipa :ff ff ff ff ff ff ff ff
fc4-types:fc4_features:scsi-fcp:init iscsi-gw
symbolic-port-name :
symbolic-node-name :10.15.1.11
port-type :N
port-ip-addr :0.0.0.0
fabric-port-wwn :21:91:00:0b:fd:44:68:c0
hard-addr :0x000000
Total number of entries = 1
```

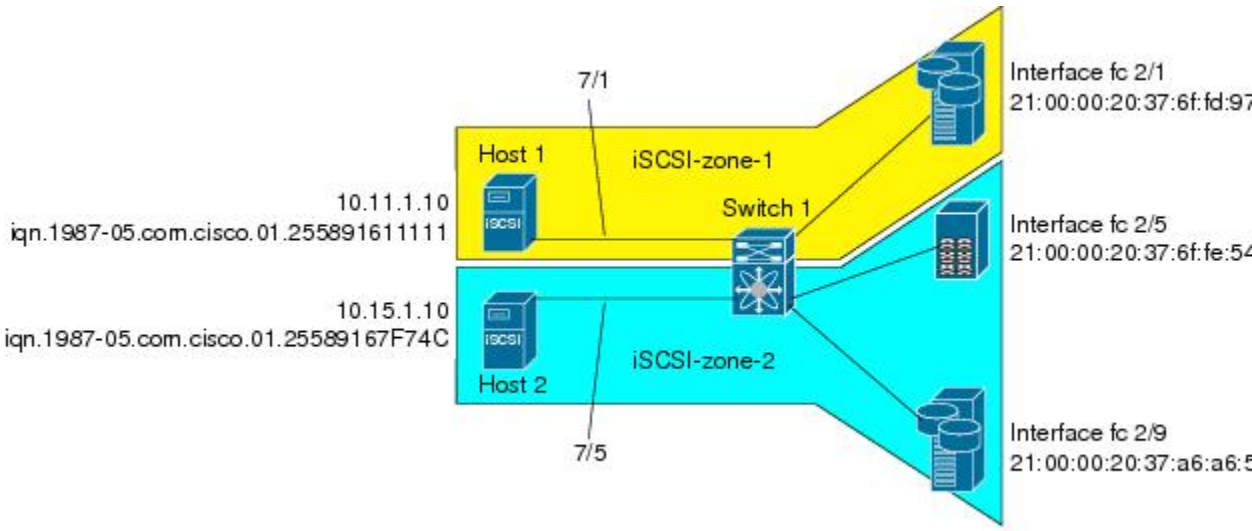
The following command displays FC ID resolved for iSCSI initiator:

```
switch# show zoneset active vsan 2
zoneset name iscsi-zoneset-v2 vsan 2
zone name iscsi-zone-2 vsan 2
* fcid 0x750001 [pwwn 21:00:00:20:37:6f:fe:54]
* fcid 0x750101 [pwwn 21:00:00:20:37:a6:a6:5d]
```

# Configuring iSCSI Scenario 1

the following figure represents scenario1. To configure scenario 1, follow these steps:

*Figure 42: iSCSI Scenario 1*




## Procedure

**Step 1** Configure null authentication for all iSCSI hosts in Cisco MDS switches:

```
switch(config)# iscsi authentication none
```

**Step 2** Configure iSCSI to dynamically import all Fibre Channel targets into the iSCSI SAN using auto-generated iSCSI target names:

```
switch(config)# iscsi import target fc
```

**Step 3** Configure the Gigabit Ethernet interface in slot 7 port 1 with an IPv4 address and enable the interface:

```
switch(config)# interface gigabitethernet 7/1
switch(config-if)# ip address 10.11.1.1 255.255.255.0
switch(config-if)# no shutdown
```

**Note**
Host 2 is connected to this port.

**Step 4** Configure the iSCSI interface in slot 7 port 1 to identify all dynamic iSCSI initiators by their IP address, and enable the interface:

```
switch(config)# interface iscsi 7/1
switch(config-if)# switchport initiator id ip-address
switch(config-if)# no shut
```

**Step 5** Configure the Gigabit Ethernet interface in slot 7 port 5 with the IPv4 address and enable the interface:

```
switch(config)# interface gigabitethernet 7/5
switch(config-if)# ip address 10.15.1.1 255.255.255.0
switch(config-if)# no shutdown
```

**Step 6** Configure the iSCSI interface in slot 7 port 5 to identify all dynamic iSCSI initiators by node name and enable the interface:

```
switch(config)# interface iscsi 7/5
switch(config-if)# switchport initiator id name
switch(config-if)# no shutdown
```

**Note**
Host 1 is connected to this port.

**Step 7** Verify the available Fibre Channel targets:

```
switch# show fcns database
VSAN 1:
--------------------------------------------------------------------------
FCID TYPE PWWN (VENDOR) FC4-TYPE:FEATURE
--------------------------------------------------------------------------
0x6d0001 NL 21:00:00:20:37:6f:fd:97 (Seagate) scsi-fcp:target
0x6d0101 NL 21:00:00:20:37:6f:fe:54 (Seagate) scsi-fcp:target
0x6d0201 NL 21:00:00:20:37:a6:a6:5d (Seagate) scsi-fcp:target
Total number of entries = 3
```

**Step 8** Create a zone named *iscsi-zone-1* with host 1 and one Fibre Channel target in it:

```
switch(config)# zone name iscsi-zone-1 vsan 1
switch(config-zone)# member pwwn 21:00:00:20:37:6f:fd:97
switch(config-zone)# member ip-address 10.11.1.10
```

**Note**
Use the IP address of the host in zone membership configuration because the iSCSI interface is configured to identify all hosts based on IP address.

**Step 9** Create a zone named *iscsi-zone-2* with host 2 and two Fibre Channel targets in it:

```
switch(config)# zone name iscsi-zone-2 vsan 1
switch(config-zone)# member pwwn 21:00:00:20:37:6f:fe:54
switch(config-zone)# member pwwn 21:00:00:20:37:a6:a6:5d
switch(config-zone)# member symbolic-nodename iqn.1987-05.com.cisco:01.25589167f74c
```

**Note**
Use the symbolic node name of the iSCSI host in zone membership configuration because the iSCSI interface is configured to identify all hosts based on node name.

**Step 10** Create a zone set and add the two zones as members:

```
switch(config)# zoneset name zoneset-iscsi vsan 1
switch(config-zoneset)# member iscsi-zone-1
switch(config-zoneset)# member iscsi-zone-2
```

**Step 11** Activate the zone set:

```
switch(config)# zoneset activate name zoneset-iscsi vsan 1
```

**Step 12** Display the active zone set:

```
switch# show zoneset active
zoneset name zoneset-iscsi vsan 1
zone name iscsi-zone-1 vsan 1
* fcid 0x6d0001 [pwwn 21:00:00:20:37:6f:fd:97] <-----------Target
symbolic-nodename 10.11.1.10 <--------------------------------- iSCSI host (host 1, not online)

zone name iscsi-zone-2 vsan 1
* fcid 0x6d0101 [pwwn 21:00:00:20:37:6f:fe:54] <------------Target
* fcid 0x6d0201 [pwwn 21:00:00:20:37:a6:a6:5d] <------------Target
symbolic-nodename iqn.1987-05.com.cisco:01.25589167f74c <-iSCSI host (host 2, not online)
```

**Step 13** Bring up the iSCSI hosts (host 1 and host 2).

**Step 14** Show all the iSCSI sessions (use the **detail** option for detailed information):

```
switch# show iscsi session
Initiator iqn.1987-05.com.cisco:01.25589167f74c <-----Host 2
Initiator ip addr (s): 10.15.1.11
Session #1
Target iqn.1987-05.com.cisco:05.172.22.92.166.07-05.21000020376ffe54
VSAN 1, ISID 00023d000001, Status active, no reservation

Session #2
Target iqn.1987-05.com.cisco:05.172.22.92.166.07-05.2100002037a6a65d
VSAN 1, ISID 00023d000001, Status active, no reservation

Initiator 10.11.1.10 <----------------------------------Host 1
Initiator name iqn.1987-05.com.cisco:01.e41695d16b1a
Session #1
Target iqn.1987-05.com.cisco:05.172.22.92.166.07-01.21000020376ffd97
VSAN 1, ISID 00023d000001, Status active, no reservation
```

**Note**
The last part of the auto-created target name is the Fibre Channel target's pWWN.

**Step 15** Verify the details of the two iSCSI initiators:

Host 2: Initiator ID based on node name because the initiator is entering iSCSI interface 7/5:

```
switch# show iscsi initiator

iSCSI Node name is iqn.1987-05.com.cisco:01.25589167f74c
Initiator ip addr (s): 10.15.1.11
iSCSI alias name: oasis11.cisco.com
Node WWN is 20:02:00:0b:fd:44:68:c2 (dynamic)
Member of vsans: 1
Number of Virtual n_ports: 1
Virtual Port WWN is 20:03:00:0b:fd:44:68:c2 (dynamic)
Interface iSCSI 7/5 , Portal group tag: 0x304
VSAN ID 1, FCID 0x6d0300
```

Host 1: Initiator ID based on IPv4 address because the initiator is entering iSCSI interface 7/1:

```
iSCSI Node name is 10.11.1.10
iSCSI Initiator name: iqn.1987 - 05.com.cisco:01.e41695d16b1a
iSCSI alias name: oasis10.cisco.com
Node WWN is 20:04:00:0b:fd:44:68:c2 (dynamic)
Member of vsans: 1
Number of Virtual n_ports: 1
Virtual Port WWN is 20:05:00:0b:fd:44:68:c2 (dynamic)
Interface iSCSI 7/1 , Portal group tag: 0x300
VSAN ID 1, FCID 0x6d0301
```

**Step 16** View the active zone set. The iSCSI initiators' FC IDs are resolved:

FC ID resolved forhost 1:

```
switch# show zoneset active
zoneset name zoneset-iscsi vsan 1
zone name iscsi-zone-1 vsan 1
* fcid 0x6d0001 [pwwn 21:00:00:20:37:6f:fd:97]
* fcid 0x6d0301 [symbolic-nodename 10.11.1.10]
```

FC ID for host 2:

```
zone name iscsi-zone-2 vsan 1
* fcid 0x6d0101 [pwwn 21:00:00:20:37:6f:fe:54]
```

```
* fcid 0x6d0201 [pwwn 21:00:00:20:37:a6:a6:5d]
* fcid 0x6d0300 [symbolic-nodename iqn.1987-05.com.cisco:01.25589167f74c] <-------------------------
```

**Step 17** The Fibre Channel name server shows the virtual N ports created for the iSCSI hosts:

```
switch# show fcns database
VSAN 1:
--------------------------------------------------------------------------
FCID TYPE PWWN (VENDOR) FC4-TYPE:FEATURE
--------------------------------------------------------------------------
0x6d0001 NL 21:00:00:20:37:6f:fd:97 (Seagate) scsi-fcp:target
0x6d0101 NL 21:00:00:20:37:6f:fe:54 (Seagate) scsi-fcp:target
0x6d0201 NL 21:00:00:20:37:a6:a6:5d (Seagate) scsi-fcp:target
0x6d0300 N 20:03:00:0b:fd:44:68:c2 (Cisco) scsi-fcp:init isc..w
0x6d0301 N 20:05:00:0b:fd:44:68:c2 (Cisco) scsi-fcp:init isc..w
```

**Step 18** Verify the detailed output of the iSCSI initiator nodes in the Fibre Channel name server:

```
switch# show fcns database fcid 0x6d0300 detail vsan 1
------------------------
VSAN:1 FCID:0x6d0300
------------------------
port-wwn (vendor) :20:03:00:0b:fd:44:68:c2 (Cisco)
node-wwn :20:02:00:0b:fd:44:68:c2
class :2,3
node-ip-addr :10.15.1.11
ipa :ff ff ff ff ff ff ff ff
fc4-types:fc4_features:scsi-fcp:init iscsi-gw
symbolic-port-name :
symbolic-node-name :iqn.1987-05.com.cisco:01.25589167f74c <--------------------
port-type :N
port-ip-addr :0.0.0.0
fabric-port-wwn :21:91:00:0b:fd:44:68:c0
hard-addr :0x000000
Total number of entries = 1
```

```
switch# show fcns database fcid 0x6d0301 detail vsan 1
------------------------
VSAN:1 FCID:0x6d0301
------------------------
port-wwn (vendor) :20:05:00:0b:fd:44:68:c2 (Cisco)
node-wwn :20:04:00:0b:fd:44:68:c2
class :2,3
node-ip-addr :10.11.1.10
ipa :ff ff ff ff ff ff ff ff
fc4-types:fc4_features:scsi-fcp:init iscsi-gw
symbolic-port-name :
symbolic-node-name :10.11.1.10
port-type :N
port-ip-addr :0.0.0.0
fabric-port-wwn :21:81:00:0b:fd:44:68:c0
hard-addr :0x000000
```

# Configuring iSCSI Scenario 2

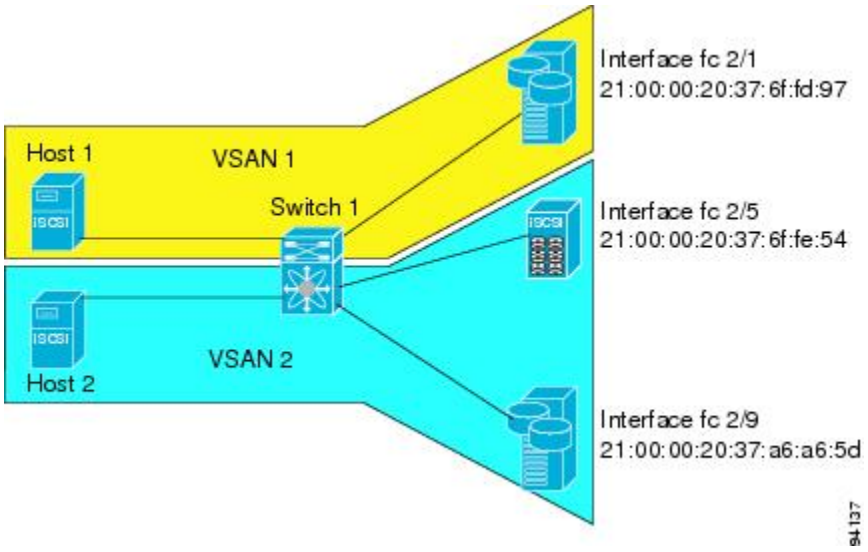
Sample scenario 2 assumes the following configuration:

- Access control is based on Fibre Channel zoning.
- There is target-based LUN mapping or LUN masking.

- There is no iSCSI authentication (none).
- The iSCSI initiator is assigned to different VSANs.

The following figure represents scenario2. To configure scenario 2, follow these steps:

***Figure 43: iSCSI Scenario 2***




## Procedure

**Step 1** Configure null authentication for all iSCSI hosts:

```
switch(config)# iscsi authentication none
```

**Step 2** Configure iSCSI to dynamically import all Fibre Channel targets into the iSCSI SAN using auto-generated iSCSI target names:

```
switch(config)# iscsi import target fc
```

**Step 3** Configure the Gigabit Ethernet interface in slot 7 port 1 with an IPv4 address and enable the interface:

```
switch(config)# interface gigabitethernet 7/1
switch(config-if)# ip address 10.11.1.1 255.255.255.0
switch(config-if)# no shutdown
```

**Step 4** Configure the iSCSI interface in slot 7 port 1 to identify all dynamic iSCSI initiators by their IP address and enable the interface:

```
switch(config)# interface iscsi 7/1
switch(config-if)# switchport initiator id ip-address
switch(config-if)# no shutdown
```

**Step 5** Configure the Gigabit Ethernet interface in slot 7 port 5 with the IPv4 address and enable the interface:

```
switch(config)# interface gigabitethernet 7/5
switch(config-if)# ip address 10.15.1.1 255.255.255.0
switch(config-if)# no shutdown
```

**Step 6** Configure the iSCSI interface in slot 7 port 5 to identify all dynamic iSCSI initiators by IP address and enable the interface:

```
switch(config)# interface iscsi 7/5
switch(config-if)# switchport initiator id ip-address
switch(config-if)# no shutdown
```

**Step 7** Add static configuration for each iSCSI initiator:

```
switch(config)# iscsi initiator name iqn.1987-05.com.cisco:01.e41695d16b1a <-----Host 2
switch(config-iscsi-init)# static pWWN system-assign 1
switch(config-iscsi-init)# static nWWN system-assign
switch(config)# iscsi initiator ip address 10.15.1.11 <----------------------------------Host 1
switch(config-iscsi-init)# static pwwn system-assigned 1
switch(config-iscsi-init)# vsan 2
```

**Note**
Host 1 is configured in VSAN 2.

**Step 8** View the configured WWNs:

```
switch# show iscsi initiator configured
iSCSI Node name is iqn.1987-05.com.cisco:01.e41695d16b1a
Member of vsans: 1
Node WWN is 20:03:00:0b:fd:44:68:c2
No. of PWWN: 1
Port WWN is 20:02:00:0b:fd:44:68:c2

iSCSI Node name is 10.15.1.11
Member of vsans: 2
No. of PWWN: 1
Port WWN is 20:06:00:0b:fd:44:68:c2
```

**Note**
The WWNs are assigned by the system. The initiators are members of different VSANs.

**Step 9** Create a zone with host 1:

```
switch(config)# zone name iscsi-zone-1 vsan 1
```

**Step 10** Add three members to the zone named *iscsi-zone-1* :

a) The following command is based on the symbolic node name.

```
switch(config-zone)# member symbolic-nodename iqn.1987-05.com.cisco:01.e41695d16b1a
```

b) The following command is based on the persistent pWWN assigned to the initiator. You can obtain the pWWN from the **show iscsi initiator** output.

```
switch(config-zone)# member pwwn 20:02:00:0b:fd:44:68:c2
```

**Step 11** Create a zone with host 2 and two Fibre Channel targets:

```
switch(config)# zone name iscsi-zone-2 vsan 2
```

**Note**
If the host is in VSAN 2, the Fibre Channel targets and zone must also be in VSAN 2.

**Step 12** Activate the zone set in VSAN 2:

```
switch(config)# zoneset activate name iscsi-zoneset-v2 vsan 2
Zoneset activation initiated. check zone status
switch# show zoneset active vsan 2
```

```
zoneset name iscsi-zoneset-v2 vsan 2
zone name iscsi-zone-2 vsan 2
* fcid 0x750001 [pwwn 21:00:00:20:37:6f:fe:54]
* fcid 0x750101 [pwwn 21:00:00:20:37:a6:a6:5d]
pwwn 20:06:00:0b:fd:44:68:c2
```

**Step 13** Start the iSCSI clients on both hosts and verify that sessions come up.

**Step 14** Display the iSCSI sessions to verify the Fibre Channel target and the configured WWNs:

```
switch# show iscsi session
Initiator iqn.1987-05.com.cisco:01.e41695d16b1a
Initiator ip addr (s): 10.11.1.10
Session #1
Discovery session, ISID 00023d000001, Status active
Session #2
Target iqn.1987-05.com.cisco:05.172.22.92.166.07-01.21000020376ffd97 <----
VSAN 1, ISID 00023d000001, Status active, no reservation
```

**Step 15** Display the iSCSI initiator to verify the configured nWWN and pWWN:

```
switch# show iscsi initiator
iSCSI Node name is iqn.1987-05.com.cisco:01.e41695d16b1a
Initiator ip addr (s): 10.11.1.10
iSCSI alias name: oasis10.cisco.com
Node WWN is 20:03:00:0b:fd:44:68:c2 (configured) <----------
Member of vsans: 1
Number of Virtual n_ports: 1
Virtual Port WWN is 20:02:00:0b:fd:44:68:c2 (configured) <----
Interface iSCSI 7/1, Portal group tag: 0x300
VSAN ID 1, FCID 0x680102
```

**Step 16** Check the Fibre Channel name server:

```
switch# show fcns database vsan 1
VSAN 1:
--------------------------------------------------------------------
FCID TYPE PWWN (VENDOR) FC4-TYPE:FEATURE
--------------------------------------------------------------------
0x680001 NL 21:00:00:20:37:6f:fd:97 (Seagate) scsi-fcp:target
0x680102 N 20:02:00:0b:fd:44:68:c2 (Cisco) scsi-fcp:init iscw
```

**Step 17** Verify the details of the iSCSI initiator's FC ID in the name server:

```
switch(config)# show fcns database fcid 0x680102 detail vsan 1
------------------------
VSAN:1 FCID:0x680102
------------------------
port-wwn (vendor) :20:02:00:0b:fd:44:68:c2 (Cisco)
node-wwn :20:03:00:0b:fd:44:68:c2
class :2,3
node-ip-addr :10.11.1.10
ipa :ff ff ff ff ff ff ff ff
fc4-types:fc4_features:scsi-fcp:init iscsi-gw
symbolic-port-name :
symbolic-node-name :iqn.1987-05.com.cisco:01.e41695d16b1a
port-type :N
port-ip-addr :0.0.0.0
fabric-port-wwn :21:81:00:0b:fd:44:68:c0
iSCSI alias name: oasis10.cisco.com
```

**Step 18** Check the Fibre Channel name server:

```
switch# show fcns database vsan 1
```

```
VSAN 1:
--------------------------------------------------------------------
FCID TYPE PWWN (VENDOR) FC4-TYPE:FEATURE
--------------------------------------------------------------------
0x680001 NL 21:00:00:20:37:6f:fd:97 (Seagate) scsi-fcp:target
0x680102 N 20:02:00:0b:fd:44:68:c2 (Cisco) scsi-fcp:init isc..w
```

**Step 19** Verify the details of the iSCSI initiator's FC ID in the name server:

```
switch(config)# show fcns database fcid 0x680102 detail vsan 1
------------------------
VSAN:1 FCID:0x680102
------------------------
port-wwn (vendor) :20:02:00:0b:fd:44:68:c2 (Cisco)
node-wwn :20:03:00:0b:fd:44:68:c2
class :2,3
node-ip-addr :10.11.1.10
ipa :ff ff ff ff ff ff ff ff
fc4-types:fc4_features:scsi-fcp:init iscsi-gw
symbolic-port-name :
symbolic-node-name :iqn.1987-05.com.cisco:01.e41695d16b1a
port-type :N
port-ip-addr :0.0.0.0
fabric-port-wwn :21:81:00:0b:fd:44:68:c0
hard-addr :0x000000
```

**Step 20** Verify that zoning has resolved the FC ID for the iSCSI client:

```
switch# show zoneset active vsan 1
zoneset name iscsi-zoneset-v1 vsan 1
zone name iscsi-zone-1 vsan 1
* fcid 0x680001 [pwwn 21:00:00:20:37:6f:fd:97]
* fcid 0x680102 [pwwn 20:02:00:0b:fd:44:68:c2]
```

**Step 21** Verify that the second initiator is connected to the two Fibre Channel targets in VSAN 2:

The following command displays session to first target and session to second target:

```
switch# show iscsi session initiator 10.15.1.11
Initiator 10.15.1.11
Initiator name iqn.1987-05.com.cisco:01.25589167f74c
Session #1
Target iqn.1987-05.com.cisco:05.172.22.92.166.07-05.21000020376ffe54 <--
VSAN 2, ISID 00023d000001, Status active, no reservation
Session #2
Target iqn.1987-05.com.cisco:05.172.22.92.166.07-05.2100002037a6a65d <--
VSAN 2, ISID 00023d000001, Status active, no reservation
```

The following command displays dynamic WWN as static WWN not assigned and static pWWN for the initiator:

```
switch# show iscsi initiator
iSCSI Node name is 10.15.1.11
iSCSI Initiator name: iqn.1987-05.com.cisco:01.25589167f74c
iSCSI alias name: oasis11.cisco.com
Node WWN is 20:04:00:0b:fd:44:68:c2 (dynamic)
Member of vsans: 2
Number of Virtual n_ports: 1
Virtual Port WWN is 20:06:00:0b:fd:44:68:c2 (configured)
Interface iSCSI 7/5, Portal group tag: 0x304
VSAN ID 2, FCID 0x750200
```

The following output displays iSCSI initiator entry in name server

```
switch# show fcns database vsan 2
VSAN 2:
```

```
FCID TYPE PWWN (VENDOR) FC4-TYPE:FEATURE
0x750001 NL 21:00:00:20:37:6f:fe:54 (Seagate) scsi-fcp:target
0x750101 NL 21:00:00:20:37:a6:a6:5d (Seagate) scsi-fcp:target
0x750200 N 20:06:00:0b:fd:44:68:c2 (Cisco) scsi-fcp:init isc..w
Total number of entries = 3
```

```
switch# show fcns database fcid 0x750200 detail vsan 2
-----------------------
VSAN:2 FCID:0x750200
-----------------------
port-wwn (vendor) :20:06:00:0b:fd:44:68:c2 (Cisco)
node-wwn :20:04:00:0b:fd:44:68:c2
class :2,3
node-ip-addr :10.15.1.11
ipa :ff ff ff ff ff ff ff ff
fc4-types:fc4_features:scsi-fcp:init iscsi-gw
symbolic-port-name :
symbolic-node-name :10.15.1.11
port-type :N
port-ip-addr :0.0.0.0
fabric-port-wwn :21:91:00:0b:fd:44:68:c0
hard-addr :0x000000
Total number of entries = 1
```

The following command displays FC ID resolved for iSCSI initiator:

```
switch# show zoneset active vsan 2
zoneset name iscsi-zoneset-v2 vsan 2
zone name iscsi-zone-2 vsan 2
* fcid 0x750001 [pwwn 21:00:00:20:37:6f:fe:54]
* fcid 0x750101 [pwwn 21:00:00:20:37:a6:a6:5d]
```

# Overview of Internet Storage Name Service

Internet Storage Name Service (iSNS) allows your existing TCP/IP network to function more effectively as a SAN by automating the discovery, management, and configuration of iSCSI devices. To facilitate these functions, the iSNS server and client function as follows:

- The iSNS client registers iSCSI portals and all iSCSI devices accessible through them with an iSNS server.
- The iSNS server provides the following services for the iSNS client:
  - Device registration
  - State change notification
  - Remote domain discovery services

All iSCSI devices (both initiator and target) acting as iSNS clients, can register with an iSNS server. iSCSI initiators can then query the iSNS server for a list of targets. The iSNS server will respond with a list of targets that the querying client can access based on configured access control parameters.

A Cisco MDS 9000 Family switch can act as an iSNS client and register all available iSCSI targets with an external iSNS server. All switches in the Cisco MDS 9000 Family with Fibre Channel module with IPS ports or MPS-14/2 modules installed support iSNS server functionality. This allows external iSNS clients, such as an iSCSI initiator, to register with the switch and discover all available iSCSI targets in the SAN.

This section includes the following topics:

# Overview of iSNS Client Functionality

Internet Storage Name Service (iSNS) allows your existing TCP/IP network to function more effectively as a SAN by automating the discovery, management, and configuration of iSCSI devices. The iSNS client registers iSCSI portals and all iSCSI devices accessible through them with an iSNS server. All iSCSI devices (both initiator and target) acting as iSNS clients can register with an iSNS server. When the iSNS client is unable to register or deregister objects with the iSNS server (for example, the client is unable to make a TCP connection to the iSNS server), it retries every minute to reregister all iSNS objects for the affected interfaces with the iSNS server.

The iSNS client functionality on each IPS interface (Gigabit Ethernet interface or subinterface or port channel) registers information with an iSNS server.

Once a profile is tagged to an interface, the switch opens a TCP connection to the iSNS server IP address (using the well-known iSNS port number 3205) in the profile and registers network entity and portal objects; a unique entity is associated with each IPS interface. The switch then searches the Fibre Channel name server (FCNS) database and switch configuration to find storage nodes to register with the iSNS server.

Statically mapped virtual targets are registered if the associated Fibre Channel pWWN is present in the FCNS database and no access control configuration prevents it. A dynamically mapped target is registered if dynamic target importing is enabled. See the Presenting Fibre Channel Targets as iSCSI Targets section for more details on how iSCSI imports Fibre Channel targets.

A storage node is deregistered from the iSNS server when it becomes unavailable when a configuration changes (such as access control change or dynamic import disabling) or the Fibre Channel storage port goes offline. It is registered again when the node comes back online.

When the iSNS client is unable to register or deregister objects with the iSNS server (for example, the client is unable to make a TCP connection to the iSNS server), it retries every minute to reregister all iSNS objects for the affected interfaces with the iSNS server. The iSNS client uses a registration interval value of 15 minutes. If the client fails to refresh the registration during this interval, the server will deregister the entries.

Untagging a profile also causes the network entity and portal to be deregistered from that interface.

**Note** The iSNS client is not supported on a VRRP interface.

# Creating an iSNS Client Profile

To create an iSNS profile, follow these steps:

### Procedure

**Step 1** Create a profile called MyIsns:

```
switch# config terminal
switch(config)# isns profile name MyIsns
switch(config-isns-profile)#
```

**Step 2** Specify an iSNS server IPv4 address for this profile.

```
switch(config-isns-profile)# server 10.10.100.211
```

**Step 3** Remove a configured iSNS server from this profile.

```
switch(config-isns-profile)# no server 10.10.100.211
```

**Step 4** Specify an iSNS server IPv6 address for this profile.

```
switch(config-isns-profile)# server 2003::11
```

**Step 5** Remove a configured iSNS server from this profile.

```
switch(config-isns-profile)# no server 10.20.100.211
```

# Tagging a Profile to an Interface

To tag a profile to an interface, follow these steps:

### Procedure

**Step 1** Enter configuration mode:

```
switch# config terminal
switch(config)#
```

**Step 2** Configure the specified Gigabit Ethernet interface:

```
switch(config)# interface gigabitethernet 4/1
switch(config-if)#
```

**Step 3** Tag a profile to an interface:

```
switch(config-if)# isns MyIsns
```

**Step 4** Untags a profile from an interface:

```
switch(config-if)# no isns OldIsns
```

# Verifying iSNS Client Configuration

#### Displaying configured iSNS profiles:

Use the **show isns profile** command to view configured iSNS profiles. Profile ABC has two portals registered with the iSNS server. Each portal corresponds to a particular interface. Profile XYZ has a specified iSNS server, but does not have any tagged interfaces configured.

Displaying Information for Configured iSNS Profiles:

```
switch# show isns profile
iSNS profile name ABC
tagged interface GigabitEthernet2/3
```

```
tagged interface GigabitEthernet2/2
iSNS Server 10.10.100.204

iSNS profile name XYZ
iSNS Server 10.10.100.211
```

Displaying a Specified iSNS Profile:

```
switch# show isns profile ABC
iSNS profile name ABC
tagged interface GigabitEthernet2/3
tagged interface GigabitEthernet2/2
iSNS Server 10.10.100.204
```

### Displaying all configured profiles with the iSNS PDU statistics for each tagged interface:

Use the **show isns profile counters** command to view all configured profiles with the iSNS PDU statistics for each tagged interface:

Displaying Configured Profiles with iSNS Statistics:

```
switch# show isns profile counters
iSNS profile name ABC
tagged interface port-channel 1
iSNS statistics
Input 54 pdus (registration/deregistration pdus only)
Reg pdus 37, Dereg pdus 17
Output 54 pdus (registration/deregistration pdus only)
Reg pdus 37, Dereg pdus 17
iSNS Server 10.10.100.204

iSNS profile name XYZ
tagged interface port-channel 2
iSNS statistics
Input 30 pdus (registration/deregistration pdus only)
Reg pdus 29, Dereg pdus 1
Output 30 pdus (registration/deregistration pdus only)
Reg pdus 29, Dereg pdus 1
iSNS Server 10.1.4.218
```

Displaying iSNS Statistics for a Specified Profile:

```
switch# show isns profile ABC counters
iSNS profile name ABC
tagged interface port-channel 1
iSNS statistics
Input 54 pdus (registration/deregistration pdus only)
Reg pdus 37, Dereg pdus 17
Output 54 pdus (registration/deregistration pdus only)
Reg pdus 37, Dereg pdus 17
iSNS Server 10.10.100.204
```

### Displaying all objects registered on the iSNS server and specified in the given profile

Use the **show isns** command to view all objects registered on the iSNS server and specified in the given profile.

Displaying iSNS Queries:

```
switch# show isns query ABC gigabitethernet 2/3
iSNS server: 10.10.100.204
Init: iqn.1991-05.com.w2k
Alias: <MS SW iSCSI Initiator>
Tgt : iqn.1987-05.com.cisco:05.172.22.94.22.02-03
```

```
Tgt : iqn.1987-05.com.cisco:05.172.22.94.22.02-03.210000203762fa34
nWWN: 200000203762fa34
```

### Displaying the iSNS profile to which an interface is tagged

Use the **show interface** command to view the iSNS profile to which an interface is tagged

Displaying Tagged iSNS Interfaces:

```
switch# show interface gigabitethernet 2/3
GigabitEthernet2/3 is up
Hardware is GigabitEthernet, address is 0005.3000.ae94
Internet address is 10.10.100.201/24
MTU 1500 bytes
Port mode is IPS
Speed is 1 Gbps
Beacon is turned off
Auto-Negotiation is turned on
iSNS profile ABC
^^^^^^^^^^^^^^^^
5 minutes input rate 112 bits/sec, 14 bytes/sec, 0 frames/sec
5 minutes output rate 0 bits/sec, 0 bytes/sec, 0 frames/sec
1935 packets input, 132567 bytes
4 multicast frames, 0 compressed
0 input errors, 0 frame, 0 overrun 0 fifo
1 packets output, 42 bytes, 0 underruns
0 output errors, 0 collisions, 0 fifo
0 carrier errors
```

# iSNS Server Functionality

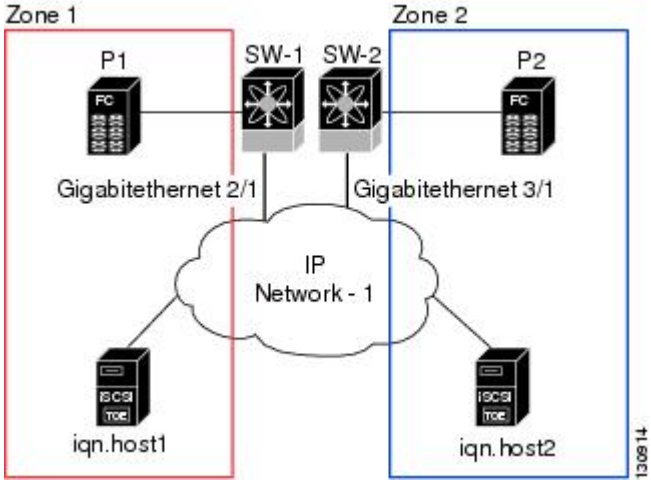
When enabled, the iSNS server on the Cisco 9000 Family MDS switch tracks all registered iSCSI devices. As a result, iSNS clients can locate other iSNS clients by querying the iSNS server. The iSNS server also provides the following functionalities:

- Allows iSNS clients to register, deregister, and query other iSNS clients registered with the iSNS server.
- Provides centralized management for enforcing access control to provide or deny access to targets from specific initiators.
- Provides a notification mechanism for registered iSNS clients to receive change notifications on the status change of other iSNS clients.
- Provides a single access control configuration for both Fibre Channel and iSCSI devices.
- Discovers iSCSI targets that do not have direct IP connectivity to the iSCSI initiators.

### Sample Scenario

The iSNS server provides uniform access control across Fibre Channel and iSCSI devices by utilizing both Fibre Channel zoning information and iSCSI access control information and configuration. An iSCSI initiator acting as an iSNS client only discovers devices it is allowed to access based on both sets of access control information. The following figure provides an example of this scenario.

*Figure 44: Using iSNS Servers in the Cisco MDS Environment*




The above figure displays iqn.host1 and iqn.host2 are iSCSI initiators. P1 and P2 are Fibre Channel targets. The two initiators are in different zones: Zone 1 consists of iqn.host1 and target P1, and Zone 2 consists of iqn.host2 and target P2. iSNS server functionality is enabled on both switches, SW-1 and SW-2. The registration process proceeds as follows:

1. Initiator iqn.host1 registers with SW-1, port Gigabitethernet2/1.
2. Initiator iqn.host2 registers with SW-2, port Gigabitethernet3/1.
3. Initiator iqn.host1 issues an iSNS query to SW-1 to determine all accessible targets.
4. The iSNS server in turn queries the Fibre Channel name server (FCNS) to obtain a list of devices that are accessible (that is, in the same zone) by the query originator. This query yields only P1.
5. The iSNS server then queries its own database to convert the Fibre Channel devices to the corresponding iSCSI targets. This is based on the iSCSI configuration, such as virtual-target and its access control setting or whether the dynamic Fibre Channel target import feature is enabled or disabled.
6. The iSNS server sends a response back to the query initiator. This response contains a list all iSCSI portals known to the iSNS server. This means iqn.host1 can choose to log in to target P1 through either SW-1 (at Gigabitethernet 2/1) or SW-2 (at Gigabitethernet 3/1).
7. If the initiator chooses to log in to SW-1 and later that port becomes inaccessible (for example, Gigabitethernet 2/1 goes down), the initiator has the choice to move to connect to target P1 through port Gigabitethernet 3/1 on SW-2 instead.
8. If the target either goes down or is removed from the zone, the iSNS server sends out an iSNS State Change Notification (SCN) message to the initiator so that the initiator can remove the session.

# Configuring an iSNS Server

This section describe how to configure an iSNS server on a Cisco MDS 9000 Family switch.

This section includes the following topics:

## Enabling an iSNS Server

To enable the iSNS server, follow these steps:

### Before you begin

Ensure iSCSI is enabled. For more information, see Enabling iSCSI section. When you disable iSCSI, iSNS is automatically disabled. When the iSNS server is enabled on a switch, every IPS port whose corresponding iSCSI interface is up is capable of servicing iSNS registration and query requests from external iSNS clients.

### Procedure

**Step 1** Enable the iSNS server:

```
switch# config terminal
switch(config)# isns-server enable
```

**Step 2** Disable (default) the iSNS server.

```
switch(config)# no isns-server enable
```

## iSNS Configuration Distribution

To enable iSNS configuration distribution using, follow these steps:

### Before you begin

You can use the CFS infrastructure to distribute the iSCSI initiator configuration to iSNS servers across the fabric. This allows the iSNS server running on any switch to provide a querying iSNS client a list of iSCSI devices available anywhere on the fabric. For information on CFS, see the Cisco Fabric Manager System Management Configuration Guide and Cisco MDS 9000 Family NX-OS System Management Configuration Guide.

### Procedure

**Step 1** Use the CFS infrastructure to distribute the iSCSI virtual target configuration to all switches in the fabric.

```
switch# config terminal
switch(config)# isns distribute
```

**Step 2** Stop (default) the distribution of iSCSI virtual target configuration to all switches in the fabric:

```
switch(config)# no isns distribute
```

## Configuring the ESI Retry Count

The iSNS client registers information with its configured iSNS server using an iSNS profile. At registration, the client can indicate an entity status inquiry (ESI) interval of 60 seconds or more. If the client registers with

an ESI interval set to zero (0), then the server does not monitor the client using ESI. In such cases, the client's registrations remain valid until explicitly deregistered or the iSNS server feature is disabled.

The ESI retry count is the number of times the iSNS server queries iSNS clients for their entity status. The default ESI retry count is 3. The client sends the server a response to indicate that it is still alive. If the client fails to respond after the configured number of retries, the client is deregistered from the server.

To configure the ESI retry count for an iSNS server, follow these steps:

**Procedure**

**Step 1** Configure the ESI to retry contacting the client up to 6 times. The range is 1 to 10:

```
switch# config terminal
switch(config)# isns esi retries 6
```

**Step 2** Revert to the default value of 3 retries:

```
switch(config)# no isns esi retries 6
```

## Configuring a Registration Period

The iSNS client specifies the registration period with the iSNS Server. The iSNS Server keeps the registration active until the end of this period. If there are no commands from the iSNS client during this period, then the iSNS server removes the client registration from its database.

If the iSNS client does not specify a registration period, the iSNS server assumes a default value of 0, which keeps the registration active indefinitely. You can also manually configure the registration period on the MDS iSNS Server.

To configure the registration period on an iSNS Server, follow these steps:

**Procedure**

**Step 1** Configure the registration to be active for 300 seconds. The permissible registration period is between 0 to 65536 seconds:

```
switch# config terminal
switch(config)# isns registration period 300
```

**Step 2** Revert to the client registered timeout value, or the default value of 0:

```
switch(config)# no isns registration period
```

# iSNS Client Registration and Deregistration

You can use the **show isns database** command to display all registered iSNS clients and their associated configuration.

An iSNS client cannot query the iSNS server until it has registered. iSNS client deregistration can occur either explicitly or when the iSNS server detects that it can no longer reach the client (through ESI monitoring).

iSNS client registration and deregistration result in status change notifications (SCNs) being generated to all interested iSNS clients.

### Target Discovery

iSCSI initiators discover targets by issuing queries to the iSNS server. The server supports DevGetNext requests to search the list of targets and DevAttrQuery to determine target and portal details, such as the IP address or port number to which to connect.

On receiving a query request from the iSCSI client, the iSNS server queries the Fibre Channel Name Server (FCNS) to obtain a list of Fibre Channel targets that are accessible by the querying initiator. The result of this query depends on zoning configuration currently active and current configuration(s) of the initiator. The iSNS server will subsequently use the iSCSI target configuration(s) (virtual target and dynamic import configuration) to translate the Fibre Channel target to an equivalent iSCSI target. At this stage it also applies any access control configured for the virtual target. A response message with the target details is then sent back to the query initiator.

The iSNS server sends a consolidated response containing all possible targets and portals to the querying initiator. For example, if a Fibre Channel target is exported as different iSCSI targets on different IPS interfaces, the iSNS server will respond with a list of all possible iSCSI targets and portals.

In order to keep the list of targets updated, the iSNS server sends state change notifications (SCN) to the client whenever an iSCSI target becomes reachable or unreachable. The client is then expected to rediscover its list of accessible targets by initiating another iSNS query. Reachability of iSCSI targets changes when any one of the following occurs:

- Target goes up or down.
- Dynamic import of FC target configuration changes.
- Zone set changes.
- Default zone access control changes.
- IPS interface state changes.
- Initiator configuration change makes the target accessible or inaccessible.

## Verifying the iSNS Server Configuration

### Viewing the ESI interval and the summary information about the iSNS database

Use the **show isns config** command to view the ESI interval and the summary information about the iSNS database contents.

Displaying the iSNS Server Configuration of ESI Interval and Database Contents:

```
switch# show isns config
Server Name: switch1(Cisco Systems) Up since: Fri Jul 30 04:08:16 2004
Index: 1 Version: 1 TCP Port: 3205
fabric distribute (remote sync): ON
ESI
Non Response Threshold: 5 Interval(seconds): 60
Database contents
```

```
Number of Entities: 2
Number of Portals: 3
Number of ISCSI devices: 4
Number of Portal Groups: 0
```

### Displaying detailed information about the contents of the iSNS database:

Use the **show isns database** command to view detailed information about the contents of the iSNS database. This command displays the full iSNS database giving all the entities, nodes, and portals registered in the database. This command without options only displays explicitly registered objects. The asterisk next to the VSAN ID indicates that the iSCSI node is in the default zone for that VSAN.

### Displaying Explicitly Registered Objects:

```
switch# show isns database
Entity Id: dp-204
Index: 2 Last accessed: Fri Jul 30 04:08:46 2004
iSCSI Node Name: iqn.1991-05.comdp-2041
Entity Index: 2
Node Type: Initiator(2) Node Index: 0x1
SCN Bitmap: OBJ_UPDATED|OBJ ADDED|OBJ REMOVED|TARGET&SELF
Node Alias: <MS SW iSCSI Initiator>
VSANS: 1(*), 5(*)
Portal IP Address: 192.168.100.2 TCP Port: 4179
Entity Index: 2 Portal Index: 1
ESI Interval: 0 ESI Port: 4180 SCN Port: 4180
```

### Displaying the Full Database with Both Registered and Configured Nodes and Portals:

```
switch# show isns database full
Entity Id: isns.entity.mds9000
Index: 1 Last accessed: Fri Jul 30 04:08:16 2004

iSCSI Node Name: iqn.com.cisco.disk1
Entity Index: 1
Node Type: Target(1) Node Index: 0x80000001
WWN(s):
22:00:00:20:37:39:dc:45
VSANS:
iSCSI Node Name: iqn.isns-first-virtual-target
Entity Index: 1
Node Type: Target(1) Node Index: 0x80000002

VSANS:
iSCSI Node Name: iqn.com.cisco.disk2
Entity Index: 1
Node Type: Target(1) Node Index: 0x80000003
WWN(s):
22:00:00:20:37:39:dc:45

VSANS:
Portal IP Address: 192.168.100.5 TCP Port: 3205
Entity Index: 1 Portal Index: 3

Portal IP Address: 192.168.100.6 TCP Port: 3205
Entity Index: 1 Portal Index: 5

Entity Id: dp-204
Index: 2 Last accessed: Fri Jul 30 04:08:46 2004

iSCSI Node Name: iqn.1991-05.com.microsoft:dp-2041
Entity Index: 2
Node Type: Initiator(2) Node Index: 0x1
SCN Bitmap: OBJ_UPDATED|OBJ ADDED|OBJ REMOVED|TARGET&SELF
```

```
Node Alias: <MS SW iSCSI Initiator>

VSANS: 1(*), 5(*)
Portal IP Address: 192.168.100.2 TCP Port: 4179
Entity Index: 2 Portal Index: 1
ESI Interval: 0 ESI Port: 4180 SCN Port: 4180
```



**Note** The *local option* is only available for virtual targets.

### Displaying the Virtual Target Information in a Local Switch:

```
switch# show isns database virtual-targets local
Entity Id: isns.entity.mds9000
Index: 1 Last accessed: Fri Jul 30 04:08:16 2004

iSCSI Node Name: iqn.com.cisco.disk1
Entity Index: 1
Node Type: Target(1) Node Index: 0x80000001
WWN(s):
22:00:00:20:37:39:dc:45

VSANS:
iSCSI Node Name: iqn.isns-first-virtual-target
Entity Index: 1
Node Type: Target(1) Node Index: 0x80000002

VSANS:
iSCSI Node Name: iqn.com.cisco.disk2
Entity Index: 1
Node Type: Target(1) Node Index: 0x80000003
WWN(s):
22:00:00:20:37:39:dc:45

VSANS:
Portal IP Address: 192.168.100.5 TCP Port: 3205
Entity Index: 1 Portal Index: 3

Portal IP Address: 192.168.100.6 TCP Port: 3205
Entity Index: 1 Portal Index: 5
```

### Displaying Virtual Target for a Specified Switch:

```
switch# show isns database virtual-targets switch 20:00:00:0d:ec:01:04:40
Entity Id: isns.entity.mds9000
Index: 1 Last accessed: Fri Jul 30 04:08:16 2004

iSCSI Node Name: iqn.com.cisco.disk1
Entity Index: 1
Node Type: Target(1) Node Index: 0x80000001
WWN(s):
22:00:00:20:37:39:dc:45

VSANS:
iSCSI Node Name: iqn.isns-first-virtual-target
Entity Index: 1
Node Type: Target(1) Node Index: 0x80000002

VSANS:
iSCSI Node Name: iqn.com.cisco.disk2
Entity Index: 1
Node Type: Target(1) Node Index: 0x80000003
WWN(s):
```

```
22:00:00:20:37:39:dc:45

VSANS:
Portal IP Address: 192.168.100.5 TCP Port: 3205
Entity Index: 1 Portal Index: 3

Portal IP Address: 192.168.100.6 TCP Port: 3205
Entity Index: 1 Portal Index: 5
```

### Displaying attributes of nodes registered with the iSNS server:

Use the **show isns node** command to display attributes of nodes registered with the iSNS server. If you do not specify any options, the server displays the name and node type attribute in a compact format; one per line.

**Displaying Explicitly Registered Objects**:

```
switch# show isns node all
--------------------------------------------------------------------------------
iSCSI Node Name Type
--------------------------------------------------------------------------------
iqn.1987-05.com.cisco:05.switch1.02-03.22000020375a6c8 Target
...
iqn.com.cisco.disk1 Target
iqn.com.cisco.ipdisk Target
iqn.isns-first-virtual-target Target
iqn.1991-05.cw22 Target
iqn.1991-05.cw53 Target
```

**Displaying the Specified Node**:

```
switch# show isns node name iqn.com.cisco.disk1
iSCSI Node Name: iqn.com.cisco.disk1
Entity Index: 1
Node Type: Target(1) Node Index: 0x80000001
WWN(s):
22:00:00:20:37:39:dc:45
VSANS: 1
```

**Displaying the Attribute Details for All Nodes**:

```
switch# show isns node all detail
iSCSI Node Name: iqn.1987-05.com.cisco:05.switch1.02-03.22000020375a6c8f
Entity Index: 1
Node Type: Target(1) Node Index: 0x3000003
Configured Switch WWN: 20:00:00:0d:ec:01:04:40
WWN(s):
22:00:00:20:37:5a:6c:8f
VSANS: 1
...
iSCSI Node Name: iqn.com.cisco.disk1
Entity Index: 1
Node Type: Target(1) Node Index: 0x80000001
Configured Switch WWN: 20:00:00:0d:ec:01:04:40
WWN(s):
22:00:00:20:37:39:dc:45
VSANS: 1

iSCSI Node Name: iqn.com.cisco.ipdisk
Entity Index: 1
Node Type: Target(1) Node Index: 0x80000002
Configured Switch WWN: 20:00:00:0d:ec:01:04:40
WWN(s):
22:00:00:20:37:5a:70:1a
```

```
VSANS: 1

iSCSI Node Name: iqn.isns-first-virtual-target
Entity Index: 1
Node Type: Target(1) Node Index: 0x80000003
Configured Switch WWN: 20:00:00:0d:ec:01:04:40

iSCSI Node Name: iqn.parna.121212
Entity Index: 1
Node Type: Target(1) Node Index: 0x80000004
Configured Switch WWN: 20:00:00:0d:ec:01:04:40

iSCSI Node Name: iqn.parna.121213
Entity Index: 1
Node Type: Target(1) Node Index: 0x80000005
Configured Switch WWN: 20:00:00:0d:ec:01:04:40
```

### Displaying the attributes of a portal along with its accessible nodes:

Use the **show isns portal** command to display the attributes of a portal along with its accessible nodes. You can specify portals by using the switch WWN-interface combination or the IP address-port number combination.

**Displaying the Attribute Information for All Portals**:

```
switch# show isns portal all
--------------------------------------------------------------------------------
IPAddress TCP Port Index SCN Port ESI port
--------------------------------------------------------------------------------
192.168.100.5 3205 3 - -
192.168.100.6 3205 5 - -
```

**Displaying Detailed Attribute Information for All Portals**:

```
switch# show isns portal all detail
Portal IP Address: 192.168.100.5 TCP Port: 3205
Entity Index: 1 Portal Index: 3

Portal IP Address: 192.168.100.6 TCP Port: 3205
Entity Index: 1 Portal Index: 5
```

**Displaying Virtual Portals**:

```
switch# show isns portal virtual
--------------------------------------------------------------------------------
IPAddress TCP Port Index SCN Port ESI port
--------------------------------------------------------------------------------
192.168.100.5 3205 3 - -
192.168.100.6 3205 5 - -
```

**Displaying Virtual Portals for a Specified Switch**:

```
switch# show isns portal virtual switch 20:00:00:0d:ec:01:04:40
--------------------------------------------------------------------------------
IPAddress TCP Port Index SCN Port ESI port
--------------------------------------------------------------------------------
192.168.100.5 3205 3 - -
192.168.100.6 3205 5 - -
```

**Displaying Detailed Information for the Virtual Portals in a Specified Switch**:

```
switch# show isns portal virtual switch 20:00:00:0d:ec:01:04:40 detail
Portal IP Address: 192.168.100.5 TCP Port: 3205
Entity Index: 1 Portal Index: 3
Switch WWN: 20:00:00:0d:ec:01:04:40
Interface: GigabitEthernet2/3
```

```
Portal IP Address: 192.168.100.6 TCP Port: 3205
Entity Index: 1 Portal Index: 5
Switch WWN: 20:00:00:0d:ec:01:04:40
Interface: GigabitEthernet2/5
```

### Displaying the attributes of an entity along with the list of portals and nodes in that entity:

Use the **show isns entity** command to display the attributes of an entity along with the list of portals and nodes in that entity. If you do not specify any option, this command displays the entity ID and number of nodes or portals associated with the entity in a compact format; one per line

**Displaying All Registered Entries**:

```
switch1# show isns entity
--------------------------------------------------------------------------------
Entity ID Last Accessed
--------------------------------------------------------------------------------
dp-204 Tue Sep 7 23:15:42 2004
```

**Displaying All Entities in the Database**:

```
switch# show isns entity all

--------------------------------------------------------------------------------
Entity ID Last Accessed
--------------------------------------------------------------------------------
isns.entity.mds9000 Tue Sep 7 21:33:23 2004

dp-204 Tue Sep 7 23:15:42 2004
```

**Displaying the Entity with a Specified ID**:

```
switch1# show isns entity id dp-204
Entity Id: dp-204
Index: 2 Last accessed: Tue Sep 7 23:15:42 2004
```

**Displaying Detailed Information for All Entities in the Database**:

```
switch1# show isns entity all detail
Entity Id: isns.entity.mds9000
Index: 1 Last accessed: Tue Sep 7 21:33:23 2004

Entity Id: dp-204
Index: 2 Last accessed: Tue Sep 7 23:16:34 2004
```

**Displaying Virtual Entities**:

```
switch# show isns entity virtual
Entity Id: isns.entity.mds9000
Index: 1 Last accessed: Thu Aug 5 00:58:50 2004

Entity Id: dp-204
Index: 2 Last accessed: Thu Aug 5 01:00:23 2004
```

### Displaying information about import targets

Use the **show iscsi global config** command to display information about import targets

**Displaying the Import Target Settings for a Specified Switch**:

```
switch# show isns iscsi global config switch 20:00:00:05:ec:01:04:00
iSCSI Global configuration:
Switch: 20:00:00:05:ec:01:04:00 iSCSI Auto Import: Enabled
```

**Displaying the Import Target Settings for All Switches**:

```
switch# show isns iscsi global config all
iSCSI Global configuration:
Switch: 20:00:44:0d:ec:01:02:40 iSCSI Auto Import: Enabled
```

### Displaying CFS peers switch information about the iSNS application

Use the **show cfs peers** command to display CFS peers switch information about the iSNS application:

**Displaying the CFS Peer Switch Information for the iSNS Application:**

```
switch# show cfs peers name isns

Scope : Physical
--------------------------------------------------
Switch WWN IP Address
--------------------------------------------------
20:00:00:00:ec:01:00:40 10.10.100.11 [Local]

Total number of entries = 1
```

# iSNS Cloud Discovery

You can configure iSNS cloud discovery to automate the process of discovering iSNS servers in the IP network.

This section includes the following topics:

- Cloud Discovery
- Configuring iSNS Cloud Discovery
- Verifying Cloud Discovery Status
- Verifying Cloud Discovery Membership
- Displaying Cloud Discovery Statistics

## Cloud Discovery

When an iSNS server receives a query request, it responds with a list of available targets and the portals through which the initiator can reach the target. The IP network configuration outside the MDS switch may result in only a subset of Gigabit Ethernet interfaces being reachable from the initiator. To ensure that the set of portals returned to the initiator is reachable, the iSNS server needs to know the set of Gigabit Ethernet interfaces that are reachable from a given initiator.



**Note** iSNS Cloud Discovery is not supported on the Cisco Fabric Switch for IBM BladeCenter and Cisco Fabric Switch for HP c-Class BladeSystem.

The iSNS cloud discovery feature provides information to the iSNS server on the various interfaces reachable from an initiator by partitioning the interfaces on a switch into disjointed IP clouds. This discovery is achieved by sending messages to all other known IPS ports that are currently up and, depending on the response (or the lack of it), determines if the remote IPS port is in the same IP network or in a different IP network.

Cloud discovery is initiated when the following events occur:

- Manual requests from the CLI initiate cloud discovery from the CLI. This action causes the destruction of existing memberships and makes new ones.

- Auto-discovery of the interface results in an interface being assigned to its correct cloud. All other cloud members are not affected. The membership of each cloud is built incrementally and is initiated by the following events:
  - A Gigabit Ethernet interface comes up. This can be a local or remote Gigabit Ethernet interface.
  - The IP address of a Gigabit Ethernet interface changes.
  - The VRRP configuration on a port changes.

The iSNS server distributes cloud and membership information across all the switches using CFS. Therefore, the cloud membership view is the same on all the switches in the fabric.



**Note** For CFS distribution to operate correctly for iSNS cloud discovery, all switches in the fabric must be running Cisco SAN-OS Release 3.0(1) or NX-OS 4.1(1b) and later.

## Configuring iSNS Cloud Discovery

This section describes how to configure iSNS cloud discovery and includes the following topics:

- Enabling iSNS Cloud Discovery
- Initiating On-Demand iSNS Cloud Discovery
- Configuring Automatic iSNS Cloud Discovery
- Verifying Automatic iSNS Cloud Discovery Configuration
- Configuring iSNS Cloud Discovery
- Configuring iSNS Cloud Discovery Message Types

### Enabling iSNS Cloud Discovery

To enable iSNS cloud discovery, follow these steps:

**Procedure**

**Step 1** Enable iSNS cloud discovery:

```
switch# config terminal
switch(config)# cloud-discovery enable
```

**Step 2** Disable (default) iSNS cloud discovery:

```
switch(config)# no cloud-discovery enable
```

### Initiating On-Demand iSNS Cloud Discovery

To initiate on-demand iSNS cloud discovery, use the **cloud discover** command in EXEC mode.

The following example shows how to initiate on-demand cloud discovery for the entire fabric:

```
switch# cloud discover
```

### Configuring Automatic iSNS Cloud Discovery

To configure automatic iSNS cloud discovery, follow these steps:

#### Procedure

**Step 1** Enable (default) automatic iSNS cloud discovery:

```
switch# config terminal
switch(config)# cloud discovery auto
```

**Step 2** Disable automatic iSNS cloud discovery:

```
switch(config)# no cloud discovery auto
```

**Step 3** Verify the automatic iSNS cloud discovery configuration:

```
switch# show cloud discovery config
Auto discovery: Enabled
```

### Configuring iSNS Cloud Discovery Distribution

To configure iSNS cloud discovery distribution using CFS, follow these steps:

#### Procedure

**Step 1** Enable (default) iSNS cloud discovery fabric distribution.

```
switch# config terminal
switch(config)# cloud discovery fabric distribute
```

**Step 2** Disable iSNS cloud discovery fabric distribution:

```
switch(config)# no cloud discovery fabric distribute
```

### Configuring iSNS Cloud Discovery Message Types

You can configure iSNS cloud discovery the type of message to use. By default, iSNS cloud discovery uses ICMP.

To configure iSNS cloud discovery message types, follow these steps:

#### Procedure

**Step 1** Enter configuration mode:

```
switch# config terminal
switch(config)#
```

**Step 2** Enable (default) iSNS cloud discovery using ICMP messages:

```
switch(config)# cloud discovery message icmp
```

**Note**
Only ICMP messages are supported.

### Verifying Cloud Discovery Status

Use the **show cloud discovery status** command to verify the status of the cloud discovery operation:

```
switch# show cloud discovery status
Discovery status: Succeeded
```

### Verifying Cloud Discovery Membership

Use the **show cloud membership all** command to verify the cloud membership for the switch:

```
switch# show cloud membership all
Cloud 2
GigabitEthernet1/5[20:00:00:0d:ec:02:c6:c0] IP Addr 10.10.10.5
GigabitEthernet1/6[20:00:00:0d:ec:02:c6:c0] IP Addr 10.10.10.6
#members=2
```

Use the **show cloud membership unresolved** command to verify the unresolved membership on the switch:

```
switch# show cloud membership unresolved
Undiscovered Cloud
No members
```

Use the **show cloud discovery statistics** command to display the statistics for the cloud discovery operations:

```
switch# show cloud discovery statistics
Global statistics
Number of Auto Discovery = 1
Number of Manual Discovery = 0
Number of cloud discovery (ping) messages sent = 1
Number of cloud discovery (ping) success = 1
```

# Default Settings

The following table lists the default settings for iSCSI parameters.

***Table 8: Default iSCSI Parameters***

| Parameters | Default |
|---|---|
| Number of TCP connections | One per iSCSI session |
| minimum-retransmit-time | 300 msec |
| keepalive-timeout | 60 seconds |

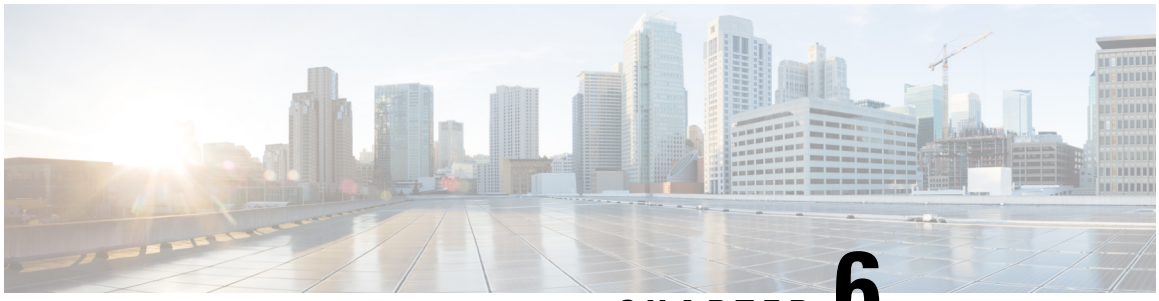
| Parameters | Default |
|---|---|
| max-retransmissions | 4 retransmissions |
| PMTU discovery | Enabled |
| pmtu-enable reset-timeout | 3600 sec |
| SACK | Enabled |
| max-bandwidth | 1 Gbps |
| min-available-bandwidth | 70 Mbps |
| round-trip-time | 1 msec |
| Buffer size | 4096 KB |
| Control TCP and data connection | No packets are transmitted |
| TCP congestion window monitoring | Enabled |
| Burst size | 50 KB |
| Jitter | 500 microseconds |
| TCP connection mode | Active mode is enabled |
| Fibre Channel targets to iSCSI | Not imported |
| Advertising iSCSI target | Advertised on all Gigabit Ethernet interfaces, subinterfaces, port channel interfaces, and port channel subinterfaces |
| iSCSI hosts mapping to virtual Fibre Channel hosts | Dynamic mapping |
| Dynamic iSCSI initiators | Members of the VSAN 1 |
| Identifying initiators | iSCSI node names |
| Advertising static virtual targets | No initiators are allowed to access a virtual target (unless explicitly configured) |
| iSCSI login authentication | CHAP or none authentication mechanism |
| revert-primary-port | Disabled |
| Header and data digest | Enabled automatically when iSCSI initiators send requests. This feature cannot be configured and is not available in store-and-forward mode. |
| iSNS registration interval | 60 sec (not configurable) |
| iSNS registration interval retries | 3 |
| Fabric distribution | Disabled |

The following table lists the default settings for iSLB parameters.

***Table 9: Default iSLB***

| | |
|---|---|
| Fabric distribution | Disabled |

| | |
|---|---|
| Load balancing metric | 1000 |



CHAPTER 6

# Configuring IP Services

Cisco MDS 9000 Family switches can route IP traffic between Ethernet and Fibre Channel interfaces. The IP static routing feature is used to route traffic between VSANs. To do so, each VSAN must be in a different IP subnetwork. Each Cisco MDS 9000 Family switch provides the following services for network management systems (NMSs):

- IP forwarding on the out-of-band Ethernet interface (mgmt0) on the front panel of the supervisor modules.
- IP forwarding on in-band Fibre Channel interface using the IP over Fibre Channel (IPFC) function—IPFC specifies how IP frames can be transported over Fibre Channel using encapsulation techniques. IP frames are encapsulated into Fibre Channel frames so NMS information can cross the Fibre Channel network without using an overlay Ethernet network.
- IP routing (default routing and static routing)—If your configuration does not need an external router, you can configure a default route using static routing.

Switches are compliant with RFC 2338 standards for Virtual Router Redundancy Protocol (VRRP) features. VRRP is a restartable application that provides a redundant, alternate path to the gateway switch.

**Note** From Cisco MDS NX-OS Release 8.3(1) and later, the VRRP feature is not supported on Cisco MDS 9000 Series Switches.

For information about configuring IPv6, see Configuring IPv4 for Gigabit Ethernet Interfaces , on page 229

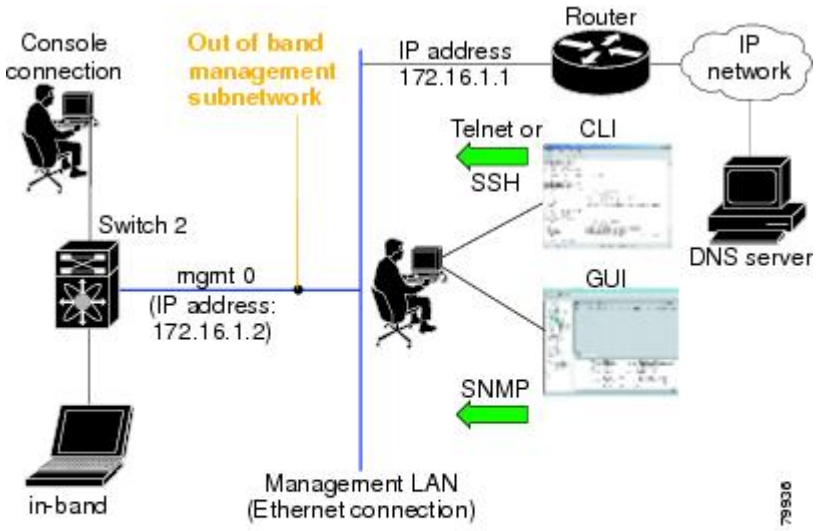
This chapter includes the following sections:


- Traffic Management Services, on page 190
- Management Interface Configuration, on page 190
- Default Gateway, on page 192
- IPv4 Default Network Configuration, on page 193
- IP over Fibre Channel, on page 194
- IPv4 Static Routes, on page 198
- Overlay VSANs, on page 200
- DNS Configuration, on page 204
- Displaying DNS Host Information, on page 206
- Default Settings for DNS Features, on page 206

# Traffic Management Services

In-band options are compliant with and use the RFC 2625 standards. An NMS host running the IP protocol over an Fibre Channel interface can access the switch using the IPFC functionality. If the NMS does not have a Fibre Channel HBA, in-band management can still be performed using one of the switches as an access point to the fabric as shown in the following figure.

***Figure 45: Management Access to Switches***




# Management Interface Configuration

The management interface on the switch allows multiple simultaneous Telnet or SNMP sessions. You can remotely configure the switch through the management interface, but first you must configure IP version 4 (IPv4) parameters (IP address, subnet mask) or an IP version 6 (IPv6) address and prefix length so that the switch is reachable. For information on configuring IPv6 addresses, see Configuring IPv4 for Gigabit Ethernet Interfaces , on page 229.

On director class switches, a single IP address is used to manage the switch. The active supervisor module's management (mgmt0) interface uses this IP address. The mgmt0 interface on the standby supervisor module remains in an inactive state and cannot be accessed until a switchover happens. After a switchover, the mgmt0 interface on the standby supervisor module becomes active and assumes the same IP address as the previously active supervisor module.

**Note** The port on the Ethernet switch to which the MDS management interface is connected should be configured as a host port (also known as access port) instead of a switch port. Spanning tree configuration for that port (on the Ethernet switch) should disabled. This helps avoid the delay in the MDS management port coming up due to delay from Ethernet spanning tree processing that the Ethernet switch would run if enabled. For Cisco Ethernet switches, use either the **switchport host** command in Cisco IOS is or the **set port host command** in the Catalyst OS. Refer to the configuration guide for your Ethernet switch.

**Note** Before you begin to configure the management interface manually, obtain the switch's IP address and IP subnet mask. Also make sure the console cable is connected to the console port.

This section contains the following topics:

# Configuring the mgmt0 Ethernet interface for IPv4

To configure the mgmt0 Ethernet interface for IPv4, follow these steps:

**Procedure**

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | switch# **config terminal** | Enters configuration mode. |
| **Step 2** | switch(config)# **interface mgmt0** | Enters the interface configuration mode on the management Ethernet interface (mgmt0). |
| **Step 3** | switch(config-if)# **ip address 10.1.1.1 255.255.255.0** | Enters the IPv4 address (10.1.1.1) and IPv4 subnet mask (255.255.255.0) for the management interface. |
| **Step 4** | switch(config-if)# **no shutdown** | Enables the interface. |

# Configuring the mgmt0 Ethernet interface for IPv6

To configure the mgmt0 Ethernet interface for IPv6, follow these steps:

**Procedure**

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | switch# **config terminal** | Enters configuration mode. |
| **Step 2** | switch(config)# **interface mgmt0** | Enters the interface configuration mode on the management Ethernet interface (mgmt0). |

| | Command or Action | Purpose |
|---|---|---|
| **Step 3** | switch(config-if)# **ipv6 address 2001:0db8:800:200c::417a/64** | Enters the IPv6 address (2001:0DB8:800:200C::417A) and IPv6 prefix length (/64) for the management interface and enables IPv6 processing on the interface. |
| **Step 4** | switch(config-if)# **ipv6 enable** | Automatically configures a link-local IPv6 address on the interface and enables IPv6 processing on the interface. |
| **Step 5** | switch(config-if)# **no shutdown** | Enables the interface. |

# Default Gateway

You can configure a default gateway IPv4 address on your Cisco MDS 9000 Family switch.

The default gateway IPv4 address should be configured along with the IPv4 static routing attributes (IP default network, destination prefix, and destination mask, and next hop address). If you configure the static route IP forwarding and the default-network details, these IPv4 addresses will be used regardless of the default-gateway being enabled or disabled.

The default gateway IPv4 address should be configured along with the IPv4 static routing attributes commands (IP default network, destination prefix, and destination mask, and next hop address).



**Tip** If you configure the static route IP forwarding and the default-network details, these IPv4 addresses will be used regardless of the default-gateway being enabled or disabled. If these IP addresses are configured but not available, the switch will fall back to using the default gateway IP address, if you have configured it. Be sure to configure IP addresses for all entries in the switch.

Use the **ip default-gateway** command to configure the IP address for a switch's default gateway and the **show ip route** command to verify that the IPv4 address for the default gateway is configured.

This section includes the following topics:

## Configuring the Default Gateway

To configure the default gateway, follow these steps:

**Procedure**

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | switch# **configure terminal** | Enters configuration mode. |
| **Step 2** | switch(config)# **ip default- gateway 1.12.11.1** | Configures the IPv4 address for the default gateway. |

## Verifying the Default Gateway Configuration

Use the **show ip route** command to verify the default gateway configuration.

```
switch# show ip route
Codes: C - connected, S - static

Gateway of last resort is 1.12.11.1

S 5.5.5.0/24 via 1.1.1.1, GigabitEthernet1/1
C 1.12.11.0/24 is directly connected, mgmt0
C 1.1.1.0/24 is directly connected, GigabitEthernet1/1
C 3.3.3.0/24 is directly connected, GigabitEthernet1/6
C 3.3.3.0/24 is directly connected, GigabitEthernet1/5
S 3.3.3.0/24 via 1.1.1.1, GigabitEthernet1/1
```

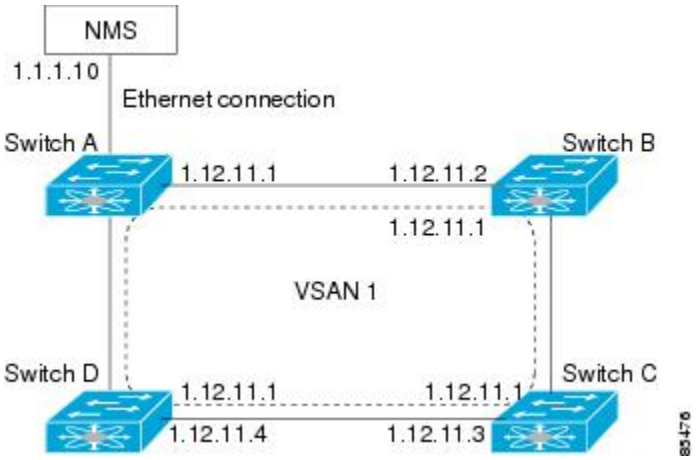
# IPv4 Default Network Configuration

If you assign the IPv4 default network address, the switch considers routes to that network as the last resort. If the IPv4 default network address is not available, the switch uses the IPv4 default gateway address. For every network configured with the IPv4 default network address, the switch flags that route as a candidate default route, if the route is available.

If you configure the static route IP forwarding and the default network details, these IPv4 addresses will be used regardless of the default gateway being enabled or disabled. If these IPv4 addresses are configured and not available, the switch will fall back to using the default gateway IPv4 address, if you have configured it. Be sure to configure IPv4 addresses for all entries in the switch if you are using IPv4.

When the Ethernet interface is configured, the switch should point to the gateway router for the IP network. The host accesses the gateway using a gateway switch. This gateway switch is configured as the default gateway. The other switches in the fabric that are connected to the same VSAN as the gateway switch can also be connected through the gateway switch. Every interface connected to this VSAN should be configured with the VSAN IPv4 address of the gateway switch as shown in the following figure.

***Figure 46: Overlay VSAN Functionality***




In the above figure, switch A has the IPv4 address 1.12.11.1, switch B has the IPv4 address 1.12.11.2, switch C has the IPv4 address 1.12.11.3, and switch D has the IPv4 address 1.12.11.4. Switch A is the gateway switch with the Ethernet connection. The NMS uses the IPv4 address 1.1.1.10 to connect to the gateway switch. Frames forwarded to any switch in the overlaid VSAN 1 are routed through the gateway switch. Configuring the gateway switch's IPv4 address (1.12.11.1) in the other switches enable the gateway switch to forward the frame to the intended destination. Similarly, if a non-gateway switch in the VSAN forwards a frame to the Ethernet, the frame is routed through the gateway switch.

When forwarding is disabled (default), IP frames are not sent from one interface to another. In these cases, the software performs local IP routing between two switches using the in-band option for Fibre Channel traffic and the mgmt0 option for Ethernet traffic.

When a VSAN is created, a VSAN interface is not created automatically. You need to specifically create the interface.

To configure default networks using IPv4 addresses, follow these steps:

**Procedure**

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | switch# **configure terminal** | Enters configuration mode. |
| **Step 2** | switch(config)# **ip default- network 190.10.1.0** | Configures the IPv4 address for the default network (190.10.1.0). |
| **Step 3** | switch(config)# **ip route 10.0.0.0 255.0.0.0 131.108.3.4** | Defines the ip route to network. |
| **Step 4** | switch(config)# **ip default-network 10.0.0.0** | Defines a static route to network 10.0.0.0 as the static default route. |

# IP over Fibre Channel

IP over Fibre Channel (IPFC) provides IP forwarding on in-band switch management over a Fibre Channel interface (rather than out-of-band using the Gigabit Ethernet mgmt 0 interface). You can be use IPFC to specify that IP frames can be transported over Fibre Channel using encapsulation techniques. IP frames are encapsulated into Fibre Channel frames so NMS information can cross the Fibre Channel network without using an overlay Ethernet network.

Once the VSAN interface is created, you can specify the IP address for that VSAN. You can assign an IPv4 address or an IPv6 address.

See the Configuring IPv4 for Gigabit Ethernet Interfaces , on page 229 for information about configuring IPv6 on the Cisco MDS 9000 Family switches.

This topic includes the following sections:

## IPFC Configuration

Follow this procedure to configure IPFC:

1. Create the VSAN to use for in-band management, if necessary.
2. Configure an IPv4 address and subnet mask for the VSAN interface.
3. Enable IPv4 routing.
4. Verify connectivity.

# Configuring an IPv4 Address in a VSAN

To create a VSAN interface and configure an IPv4 address for that interface, follow these steps:

**Procedure**

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | switch# **configure terminal** | Enters configuration mode. |
| **Step 2** | switch(config)# **interface vsan 10** | Configures the interface for the specified VSAN (10). |
| **Step 3** | switch(config-if)# **ip address 10.0.0.12 255.255.255.0** | Configures the IPv4 address and netmask for the selected interface. |
| **Step 4** | switch(config-if)# **no shutdown** | Enables the interface. |

# Verifying the VSAN Interface Configuration

Use the **show interface vsan** command to verify the configuration of the VSAN interface.

**Note** You can see the output for this command only if you have previously configured a VSAN interface.

```
switch# show interface vsan 1
vsan1 is down (Administratively down)
WWPN is 10:00:00:0c:85:90:3e:85, FCID not assigned
Internet address is 10.0.0.12/24
MTU 1500 bytes, BW 1000000 Kbit
0 packets input, 0 bytes, 0 errors, 0 multicast
0 packets output, 0 bytes, 0 errors, 0 dropped
```

# Enabling IPv4 Routing

By default, the IPv4 routing feature is disabled in all switches.

To enable the IPv4 routing feature, follow these steps:

**Procedure**

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | switch# **config terminal** | Enters configuration mode. |
| **Step 2** | switch(config)# **ip routing** | Enables IPv4 routing (disabled by default). |
| **Step 3** | switch(config)# **no ip routing** | Disables IPv4 routing and reverts to the factory settings. |

**What to do next**

**Verifying the IPv4 Routing Configuration**

Use the **show ip routing** command to verify the IPv4 routing configuration.

```
switch(config)# show ip routing
ip routing is enabled
```

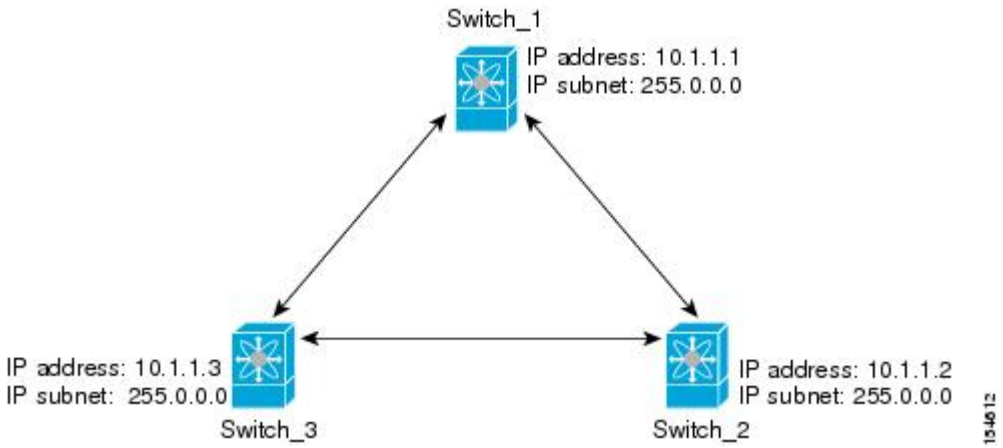
# IPFC Configuration Example

This section describe an example configuration for IPFC. The below figure shows an example network.

The example network has the following links:

- Switch_1 is connected to the main network by the mgmt 0 interface and to the fabric by an ISL.
- Switch_2 and Switch_3 are connected to the fabric by an ISL but are not connected to the main network.

***Figure 47: IPFC Example Network***




### Procedure

**Step 1** To configure Switch_1 in the example network, perform the following steps:

a) Create the VSAN interface and enter interface configuration submode:

```
switch_1# config t
switch_1(config)# interface vsan 1
switch_1(config-if)#
```

b) Configure the IP address and subnet mask:

```
switch_1(config-if)# ip address 10.1.1.1 255.0.0.0
```

c) Enable the VSAN interface and exit interface configuration submode:

```
switch_1(config-if)# no shutdown
switch_1(config-if)# exit
switch_1(config)#
```

d) Enable IPv4 routing:

```
switch_1(config)# ip routing
switch_1(config)# exit
switch_1#
```

e) Display the routes:

```
switch_1# show ip route

Codes: C - connected, S - static

C 172.16.1.0/23 is directly connect, mgmt0
C 10.0.0.0./8 is directly connected, vsan1
```

**Step 2** To configure Switch_2 in the example network, perform the following steps:

a) Enable the mgmt 0 interface:

**Note**
Configure this switch using the console connection.

```
switch_2# config terminal
switch_2(config)# interface mgmt 0
switch_2(config-if)# no shutdown
switch_2(config-if)# exit
switch_2(config)#
```

b) Create the VSAN interface and enter interface configuration:

```
switch_2# config terminal
switch_2(config)# interface vsan 1
switch_2(config-if)#
```

c) Configure the IP address and subnet mask:

```
switch_2(config-if)# ip address 10.1.1.2 255.0.0.0
```

d) Enable the VSAN interface and exit interface configuration submode:

```
switch_2(config-if)# no shutdown
switch_2(config-if)# exit
switch_2(config)#
```

e) Enable IPv4 routing:

```
switch_2(config)# ip routing
switch_2(config)# exit
switch_2#
```

f) Display the routes:

```
switch_2# show ip route

Codes: C - connected, S - static

C 10.0.0.0./8 is directly connected, vsan1
```

g) Verify the connectivity to Switch_1:

```
switch_2# ping 10.1.1.1
PING 10.1.1.1 (10.1.1.1) 56(84) bytes of data.
64 bytes from 10.1.1.1: icmp_seq=1 ttl=64 time=0.618 ms
64 bytes from 10.1.1.1: icmp_seq=2 ttl=64 time=0.528 ms
64 bytes from 10.1.1.1: icmp_seq=3 ttl=64 time=0.567 ms

--- 10.1.1.1 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 4998 ms
rtt min/avg/max/mdev = 0.528/0.570/0.618/0.057 ms
```

**Step 3** To configure Switch_3 in the example network, perform the following steps:

a) Enable the mgmt 0 interface:

**Note**
Configure this switch using the console connection.

```
switch_3# config terminal
switch_3(config)# interface mgmt 0
switch_3(config-if)# no shutdown
switch_3(config-if)# exit
switch_3(config)#

switch_3# config terminal
switch_3(config)# interface vsan 1
switch_3(config-if)#
```

b) Configure the IP address and subnet mask:

```
switch_3(config-if)# ip address 10.1.1.3 255.0.0.0
```

c) Enable the VSAN interface and exit interface configuration submode:

```
switch_3(config-if)# no shutdown
switch_3(config-if)# exit
switch_3(config)#
```

d) Enable IPv4 routing:

```
switch_3(config)# ip routing
switch_3(config)# exit
switch_3#
```

e) Display the routes:

```
switch_3# show ip route

Codes: C - connected, S - static

C 10.0.0.0./8 is directly connected, vsan1
```

f) Verify the connectivity to Switch_1:

```
switch_3# ping 10.1.1.1
PING 10.1.1.1 (10.1.1.1) 56(84) bytes of data.
64 bytes from 10.1.1.1: icmp_seq=1 ttl=64 time=1.19 ms
64 bytes from 10.1.1.1: icmp_seq=2 ttl=64 time=0.510 ms
64 bytes from 10.1.1.1: icmp_seq=3 ttl=64 time=0.653 ms

--- 10.1.1.1 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2008 ms
rtt min/avg/max/mdev = 0.510/0.787/1.199/0.297 ms
```

# IPv4 Static Routes

Static routing is a mechanism to configure IPv4 routes on the switch. You can configure more than one static route.

If a VSAN has multiple exit points, configure static routes to direct traffic to the appropriate gateway switch. IPv4 routing is disabled by default on any gateway switch between the out-of-band management interface and the default VSAN, or between directly connected VSANs.

If your network configuration does not need an external router, you can configure IPv4 static routing on your MDS switch.

For information about IPv6 static routing, see the Configuring IPv4 for Gigabit Ethernet Interfaces , on page 229

This section includes the following topics:

# Configuring IPv4 Static Routes

To configure an IPv4 static route, follow these steps:

**Procedure**

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | switch# **config terminal** | Enters configuration mode. |
| **Step 2** | switch(config)# **ip route network IP address netmask next hop IPv4 address distance number interface vsan number**<br>**Example:**<br>`switch(config)# ip route 10.0.0.0 255.0.0.0 20.20.20.10 distance 10 interface vsan 1`<br>`switch(config)#` | Configures the static route for the specified IPv4 address, subnet mask, next hop, distance, and interface. |

# Verifying IPv4 Static Route Information

Use the **show ip route** command to verifying the IPv4 static route configuration:

```
switch# show ip route configured
Destination Gateway Mask Metric Interface

default 172.22.95.1 0.0.0.0 0 mgmt0
10.1.1.0 0.0.0.0 255.255.255.0 0 vsan1
172.22.95.0 0.0.0.0 255.255.255.0 0 mgmt0
```

Use the **show ip route** command to verifying the active and connected IPv4 static route:

```
switch# show ip route

Codes: C - connected, S - static

Default gateway is 172.22.95.1

C 172.22.95.0/24 is directly connected, mgmt0
C 10.1.1.0/24 is directly connected, vsan1
```

### Displaying the IP Routing Status

```
switch# show ip routing
ip routing is disabled
```

## Displaying and Clearing ARPs

Address Resolution Protocol (ARP) entries in Cisco MDS 9000 Family switches can be displayed, deleted, or cleared. The ARP feature is enabled on all switches.

Use the **show arp** command to display the ARP table.

```
switch# show arp
Protocol Address Age (min) Hardware Addr Type Interface
Internet 171.1.1.1 0 0006.5bec.699c ARPA mgmt0
Internet 172.2.0.1 4 0000.0c07.ac01 ARPA mgmt0
```

Use the **no arp** command in configuration mode to remove an ARP entry from the ARP table.

```
switch(config)# no arp 172.2.0.1
```

Use the **clear arp** command to delete all entries from the ARP table. The ARP table is empty by default:

```
switch# clear arp-cache
```

# Overlay VSANs

VSANs enable deployment of larger SANs by overlaying multiple logical SANs, each running its own instance of fabric services, on a single large physical network. This partitioning of fabric services reduces network instability by containing fabric reconfiguration and error conditions within an individual VSAN. VSANs also provide the same isolation between individual VSANs as physically separated SANs. Traffic cannot cross VSAN boundaries and devices may not reside in more than one VSAN. Because each VSAN runs separate instances of fabric services, each VSAN has its own zone server and can be zoned in exactly the same way as SANs without VSAN capability.

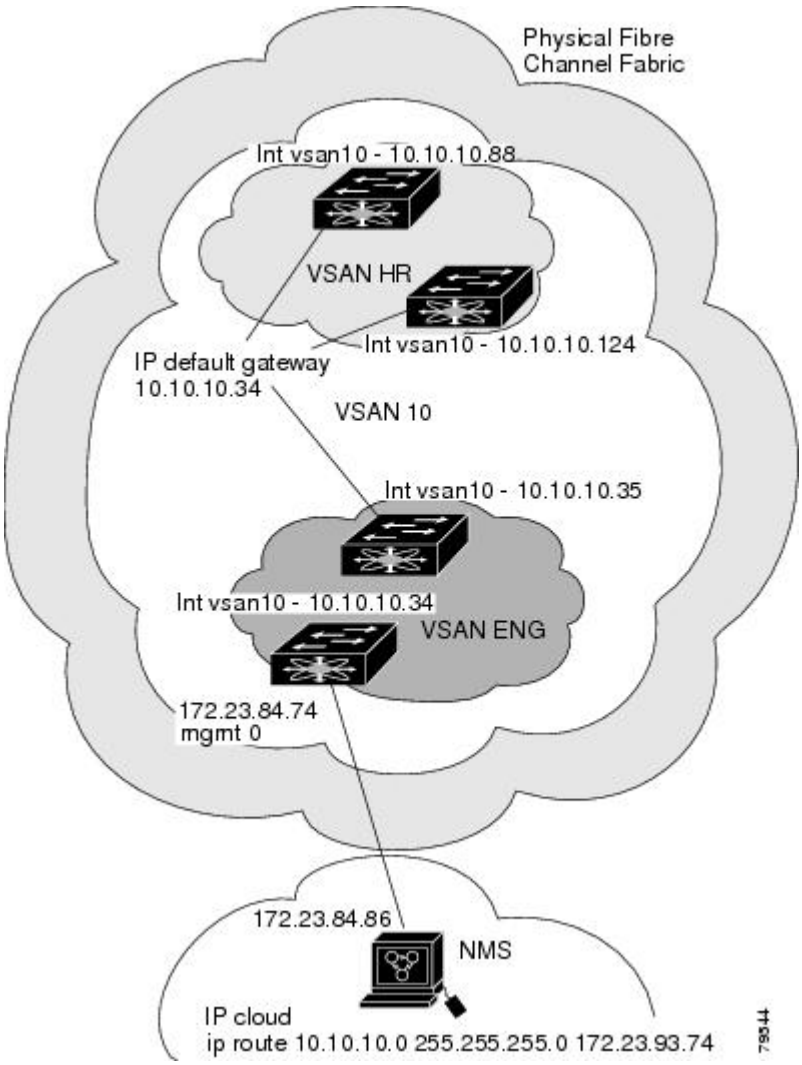
This section includes the following topics:

## Configuring Overlay VSANs

To configure an overlay VSAN, follow these steps:

1. Add the VSAN to the VSAN database on all switches in the fabric.
2. Create a VSAN interface for the VSAN on all switches in the fabric. Any VSAN interface belonging to the VSAN has an IP address in the same subnet. Create a route to the IPFC cloud on the IP side.
3. Configure a default route on every switch in the Fibre Channel fabric pointing to the switch that provides NMS access.
4. Configure the default gateway (route) and the IPv4 address on switches that point to the NMS as shown in the following figure.

*Figure 48: Overlay VSAN Configuration Example*




**Note** To configure the management interface displayed in above figure, set the default gateway to an IPv4 address on the Ethernet network.

The following procedure configures an overlay VSAN in one switch. This procedure must be repeated for each switch in the fabric.

To configure an overlay VSAN in one switch (using the example in the above figure), follow these steps:

### Procedure

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | switch# **config terminal** | Enters configuration mode. |

| | Command or Action | Purpose |
|---|---|---|
| **Step 2** | switch(config)# **vsan database** | Configures the VSAN database. |
| **Step 3** | switch--config-vsan-db# **vsan 10 name MGMT_VSAN** | Defines the VSAN in the VSAN database on all of the switches in the Fibre Channel fabric. |
| **Step 4** | switch--config-vsan-db# **exit** | Exits the VSAN database mode. |
| **Step 5** | switch(config)# **interface vsan 10** | Creates a VSAN interface (VSAN 10). |
| **Step 6** | switch(config-if)# **ip address 10.10.10.0 netmask 255.255.255.0** | Assigns an IPv4 address and subnet mask for this switch. |
| **Step 7** | switch(config-if)# **no shutdown** | Enables the configured interface. |
| **Step 8** | switch(config-if)# **end** | Exits to EXEC mode. |
| **Step 9** | switch# **exit** | Exits the switch and returns to the NMS. In this example the NMS is assumed to be on the same subnet of the Ethernet management interface of the edge that provides access to the Fibre Channel fabric. |

### What to do next

To configure the NMS station displayed in the above figure, use the following command:

| Command or Action | Purpose |
|---|---|
| `nms# route ADD 10.10.10.0 MASK 255.255.255.0 172.22.93.74` | Defines a static route on the NMS pointing to the management interface of the edge switch that provides access to the Fibre Channel fabric. |

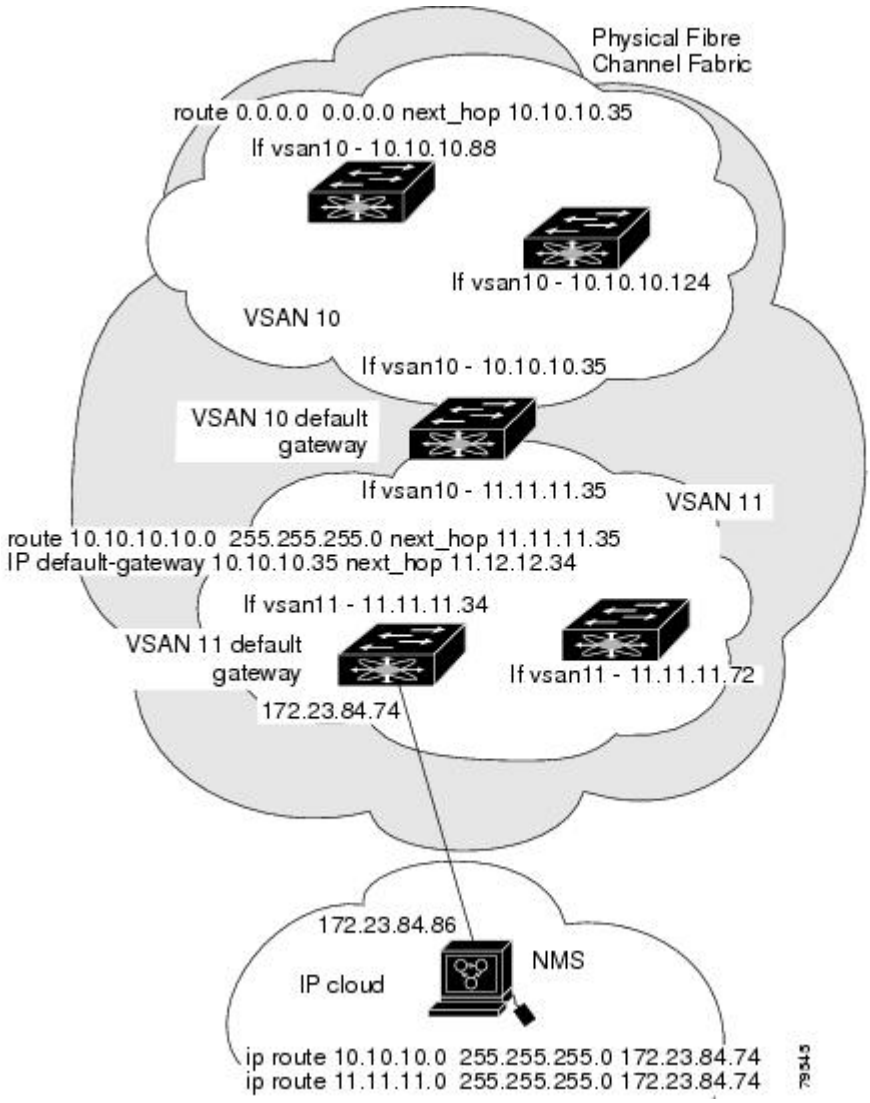
# Configuring Multiple VSANs

More than one VSAN can be used to segment the management network in multiple subnets. An active interface must be present on the switch for the VSAN interface to be enabled.

To configure multiple VSANs, follow these steps:

1. Add the VSAN to the VSAN database on any switch in the fabric.
2. Create a VSAN interface for the appropriate VSAN on any switch in the fabric.
3. Assign an IP address on every VSAN interface on the same subnet as the corresponding VSAN.
4. Define the multiple static routes on the Fibre Channel switches and the IP cloud as shown in the following figure.

*Figure 49: Multiple VSAN Configuration Example*




To configure an overlay VSAN (using the example in the previous figure), follow these steps:

**Procedure**

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | switch# **config terminal** | Enters configuration mode. |
| **Step 2** | switch(config)# **vsan database** | Configures the VSAN database. |
| **Step 3** | switch-config-vsan-db# **vsan 10 name MGMT_VSAN_10** | Defines the VSAN in the VSAN database on all of the switches in VSAN 10. |
| **Step 4** | witch-config-vsan-db# **exit** | Exits the VSAN database configuration submode. |

| | Command or Action | Purpose |
|---|---|---|
| **Step 5** | switch-config-vsan-db# **vsan 11 name MGMT_VSAN_11** | Defines the VSAN in the VSAN database on all of the switches in VSAN 11. |
| **Step 6** | switch-config-vsan-db# **exit** | Exits the VSAN database configuration submode. |
| **Step 7** | switch(config)# **interface vsan 10** | Enters the interface configuration submode for VSAN 10. |
| **Step 8** | switch(config-if)# **ip address 10.10.10.0 netmask 255.255.255.0** | Assigns an IPv4 address and subnet mask for this interface. |
| **Step 9** | switch(config-if)# **no shutdown** | Enables the configured interface for VSAN 10. |
| **Step 10** | switch(config-if)# **exit** | Exits the VSAN 10 interface mode. |
| **Step 11** | switch(config)# **interface vsan 11** | Enters the interface configuration submode for VSAN 11. |
| **Step 12** | switch(config-if)# **ip address 11.11.11.0 netmask 255.255.255.0** | Assigns an IPv4 address and subnet mask for this interface. |
| **Step 13** | switch(config-if)# **no shutdown** | Enables the configured interface for VSAN 11. |
| **Step 14** | switch(config-if)# **end** | Exits to EXEC mode. |
| **Step 15** | switch# **exit** | Exits the switch and returns to the NMS. In this example the NMS is assumed to be on the same subnet of the Ethernet management interface of the edge that provides access to the Fibre Channel fabric. |
| **Step 16** | NMS# **route ADD 10.10.10.0 MASK 255.255.255.0 172.22.93.74** | Defines a static route on the NMS pointing to the management interface of the edge switch that provides access to the IPv4 cloud. |
| **Step 17** | NMS# **route ADD 11.11.11.0 MASK 255.255.255.0 172.22.93.74** | Defines a static route for VSAN 11 on the NMS pointing to the management interface of the edge switch that provides access to the Fibre Channel fabric. |
| **Step 18** | switch# **route 10.10.10.0 255.255.255.0 next_hop 11.11.11.35** | Defines the route to reach subnet 10 from subnet 11. |

# DNS Configuration

The DNS client on the switch communicates with the DNS server to perform the IP address-name server correspondence.

The DNS server may be dropped after two attempts because of one of the following reasons:

- The IP address or the switch name is wrongly configured.
- The DNS server is not reachable because external reasons (reasons beyond our control).

**Note** When accessing a Telnet host, if the DNS server is not reachable (for any reason) the switch login prompt may take a longer time to appear. If so, verify that the DNS server is accurately configured and reachable.

To configure a DNS server, follow these steps:

### Procedure

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | switch# **config terminal** | Enters configuration mode. |
| **Step 2** | switch(config)# **ip domain-lookup** | Enables the IP Domain Naming System (DNS)-based host name-to-address translation. |
| **Step 3** | switch(config)# **no ip domain-lookup** | Disables (default) the IP DNS-based host name-to-address translation and reverts to the factory default. |
| **Step 4** | switch(config)# **ip domain-name cisco.com** | Enables the default domain name feature used to complete unqualified host names. Any IP host name that does not contain a domain name (that is, any name without a dot) will have the dot and cisco.com appended to it before being added to the host table. |
| **Step 5** | switch(config)# **no ip domain-name cisco.com** | Disables (default) the domain name. |
| **Step 6** | switch(config)# **ip domain-list harvard.edu** or<br>switch(config)# **ip domain-list stanford.edu** or<br>switch(config)# **ip domain-list yale.edu** | Defines a filter of default domain names to complete unqualified host names by using the ip domain-list global configuration command. You can define up to 10 domain names in this filter. To delete a name from a filter, use the no form of this command.<br>**Note**<br>If you have not configured a domain list, the domain name that you specified with the ip domain-name global configuration command is used. If you configured a domain list, the default domain name is not used. The ip domain-list command is similar to the ip domain-name command, except that with the ip domain-list command you can define a list of domains, each to be tried in turn. |
| **Step 7** | switch(config)# **no ip domain-list** | Deletes the defined filter and reverts to factory default. No domains are configured by default. |
| **Step 8** | switch(config)# **ip name-server 15.1.0.1 2001:0db8:800:200c::417a** | Specifies the first address (15.1.0.1) as the primary server and the second address (2001:0db8:800:200c::417a) as the secondary server. You can configure a maximum of six servers.<br>**Note**<br>Alternatively, you can configure the DNS entry using the switch names (instead of IP addresses). The configured |

| | Command or Action | Purpose |
|---|---|---|
| | | switch name automatically looks up the corresponding IP address. |
| **Step 9** | switch(config)# **no ip name-server** | Deletes the configured server(s) and reverts to factory default. No server is configured by default. |

# Displaying DNS Host Information

Use the **show hosts** command to display the DNS configuration.

### Displaying Configured Host Details

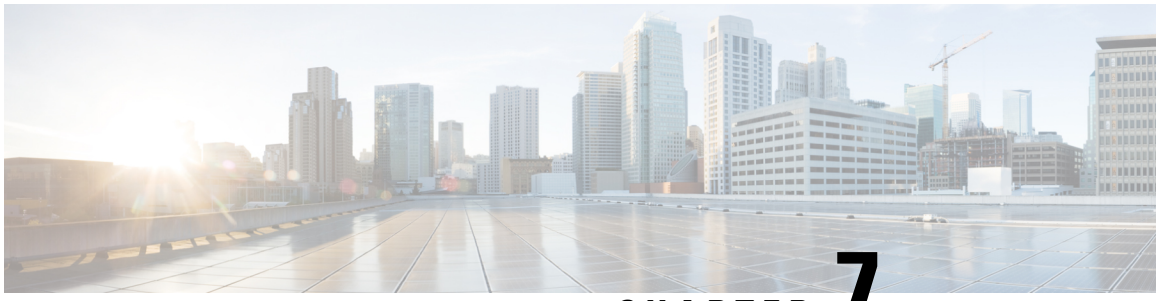
```
switch# show hosts
Default domain is cisco.com
Domain list: ucsc.edu harvard.edu yale.edu stanford.edu
Name/address lookup uses domain service
```

# Default Settings for DNS Features

The below table lists the default settings for DNS features.

*Table 10: Default DNS Settings*

| Parameters | Default |
|---|---|
| Domain lookup | Disabled |
| Domain name | Disabled |
| Domains | None |
| Domain server | None |
| Maximum domain servers | 6 |



CHAPTER 7

# Configuring IP Storage Services

Cisco MDS 9000 Series IPStorage (IPS) services extend the reach of Fibre Channel SANs by using open-standard, IP-based technology. The switch connects separated SAN islands using Fibre Channel over IP (FCIP) and it allows IP hosts to access Fibre Channel storage.

**Note** The FCIP feature is available in the modules with IPS ports and available in Cisco MDS 9250i, MDS 9220i, and Cisco MDS 9700 Director switches.

The Cisco MDS 24/10 Port SAN Extension Module for MDS 9700 series switch allows you to use Fibre Channel and FCIP features. The Cisco MDS 24/10 port SAN Extension Module can be used in any of the Cisco MDS 9700 series switches.

This chapter includes the following sections:


- Feature Information, on page 208
- Supported Hardware, on page 208
- Upgrading FCIP and iSCSI Interfaces, on page 208
- Configuring IPS Ports on Cisco MDS 9220i Switch, on page 208
- Configuring 40 Gbps Speed on Cisco MDS 24/10 port SAN Extension Module, on page 209
- Configuring IPStorage Interfaces for IPv4, on page 210
- Basic IPStorage Configuration, on page 210
- Fibre Channel Module with IPS Ports Core Dumps, on page 211
- About VLANs for IPStorage, on page 212
- Interface Subnet Requirements, on page 213
- Verifying IPStorage Connectivity, on page 213
- IPStorage IPv4-ACL Guidelines, on page 214
- Configuring IPStorage High Availability, on page 214
- Configuring CDP, on page 217
- Changing Link Speed on IPStorage Interfaces, on page 217
- Displaying Statistics, on page 223
- Displaying Ethernet MAC Statistics, on page 224
- Displaying TCP Statistics, on page 225
- Default Settings for IPStorage Services Parameters, on page 226

# Feature Information

This section briefly describes the new and updated features for releases, starting from Cisco MDS NX-OS Release 9.x.

***Table 11: Feature Information Table***

| Feature | Release | Description |
|---|---|---|
| Configuring 40 Gbps Speed on Cisco MDS 24/10 port SAN Extension Module, on page 209 | 9.3(1) | This feature allows you to configure 40-Gbps speed on the IPS ports of Cisco MDS 24/10 port SAN Extension Module. |

# Supported Hardware

You can configure the FCIP and iSCSI features on the following types of hardware:

- Cisco MDS 24/10 port SAN Extension Module in Cisco MDS 9700 Series Director Switches and Cisco MDS 9220i Fabric Switches support FCIP.
- Cisco MDS 9250i Multiservice Fabric Switches support iSCSI.

# Upgrading FCIP and iSCSI Interfaces

ISSU/D is disruptive for FCIP and iSCSI interfaces along with the update of the underlying IPS and GigabitEthernet interfaces supporting these upper layer interfaces. Fibre Channel interfaces (including FC interfaces on the same module or switch as the IPS and GigabitEthernet interfaces), FCoE Ethernet interfaces, and supervisor modules are all updated nondisruptively.

Within Cisco MDS 9700 Directors, the modules with IPS ports implement a 5 minute delay before the next module with IPS ports is upgraded.

# Configuring IPS Ports on Cisco MDS 9220i Switch

Starting from Cisco MDS NX-OS Release 8.5(1), the Cisco MDS 9220i Fabric Switch is supported. Cisco MDS 9220i switch supports six IPS ports. Ports 1 through 4 support 1- and 10-Gbps speed and port 6 supports 40-Gbps speed. By default, the first two ports are operating at a 1-Gbps speed mode and are licensed. To enable the other ports and different speed modes, you must obtain a license. You cannot move the default license from the first two ports to any other ports. For information on port licensing, see the *Cisco MDS 9000 Series Licensing Guide*.

Cisco MDS 9220i switch supports three 1 or 10 Gbps IPS ports, one 1, 10, 25 Gbps IPS port, one 25 Gbps IPS port and one 40 Gbps IPS port. All these ports are handled by a single Service Engine. The below table provides the different speed combinations that can be configured on the IPS ports of Cisco MDS 9220i switch.

The IPStorage interfaces operating in 25G-speed-mode require FEC RS-IEEE. The Ethernet interface of the connected switch must also be configured to use FEC RS-IEEE. If an IPStorage interface remains stuck in the init state, configure the Ethernet interface on the connected switch to use FEC RS-IEEE.

*Table 12: IPS Port Speed Configuration Matrix for Cisco MDS 9220i Switch*

| Speed mode | Port 1 | Port 2 | Port 3 | Port 4 | Port 5 | Port 6 |
|---|---|---|---|---|---|---|
| 1 Gbps | 1 Gbps | 1 Gbps | 1 Gbps | 1 Gbps | Out-of-service | Out-of-service |
| 10 Gbps | 10 Gbps | 10 Gbps | 10 Gbps | 10 Gbps | Out-of-service | Out-of-service |
| 25 Gbps | Out-of-service | Out-of-service | Out-of-service | 25 Gbps | 25 Gbps | Out-of-service |
| 40 Gbps | Out-of-service | Out-of-service | Out-of-service | Out-of-service | Out-of-service | 40 Gbps |

**Note** 25-Gbps speed is supported from Cisco MDS NX-OS Release 9.3(1).

# Configuring 40 Gbps Speed on Cisco MDS 24/10 port SAN Extension Module

From Cisco MDS NX-OS Release 8.5(1), you can configure 40 Gbps on the IPS ports 9 and 10 of Cisco MDS 24/10 port SAN Extension Module. Cisco MDS 24/10 port SAN Extension Module supports eight 1 or 10 Gbps IPS ports and two 40 Gbps IPS ports. Ports 1 through 4 and port 9 and handled by Service Engine 1 and Ports 5 through 8 and port 10 and handled by Service Engine 2. Each Service Engine has the capability of 40 Gbps. You can only configure 40 Gbps speed on ports 9 and 10. When you configure 40 Gbps speed on port 9, ports 1 through 4 are moved to out-of-service because the Service Engine capability is 40 Gbps. However, ports 5 through 8 can be configured for 1 or 10 Gbps speed provided port 10 is not configured for 40 Gbps speed. Similarly, when you configure 40 Gbps speed on port 10, ports 5 through 8 are moved to out-of-service. However, ports 1through 4 can be configured for 1 or 10 Gbps speed provided port 9 is not configured for 40 Gbps speed. The following table provides the different speed combinations that can be configured on the IPS ports of Cisco MDS 24/10 port SAN Extension Module.

The following are the recommended configurations for achieving maximum throughput on the 40 Gbps IPS port:

- The maximum bandwidth of FCIP tunnel is 10 Gbps. Therefore, configure four FCIP tunnels on the 40 Gbps IPS port. To create the FCIP tunnels, create four VLAN sub interfaces on the 40 Gbps IPS port. On each VLAN create an FCIP tunnel.
- Configure the number of TCP connections to 5.

*Table 13: IPS Port Speed Configuration Matrix for Cisco MDS 24/10 port SAN Extension Module*

| Service Engine 1 | | Service Engine 2 | |
|---|---|---|---|
| Ports 1 through 4 | Port 9 | Ports 5 through 8 | Port 10 |
| 1 Gbps | Out-of-service | 1 Gbps | Out-of-service |
| 10 Gbps | Out-of-service | 10 Gbps | Out-of-service |
| 1 Gbps | Out-of-service | 10 Gbps | Out-of-service |

| Service Engine 1 | | Service Engine 2 | |
|---|---|---|---|
| 10 Gbps | Out-of-service | 1 Gbps | Out-of-service |
| Out-of-service | 40 Gbps | 1 Gbps | Out-of-service |
| Out-of-service | 40 Gbps | 10 Gbps | Out-of-service |
| 1 Gbps | Out-of-service | Out-of-service | 40 Gbps |
| 10 Gbps | Out-of-service | Out-of-service | 40 Gbps |
| Out-of-service | 40 Gbps | Out-of-service | 40 Gbps |

# Configuring IPStorage Interfaces for IPv4

Both FCIP and iSCSI rely on TCP/IP for network connectivity. On each Fibre Channel module with IPS ports, connectivity is provided in the form of IP storage ports on Cisco MDS 9250i switches and Cisco MDS 9700 series switches with 24/10 port SAN Extension modules that are appropriately configured. This section covers the steps required to configure IP for subsequent use by FCIP and iSCSI.

A new port mode, called IPS, is defined for IP storage ports on each Fibre Channel module with IPS ports. IP storage ports are implicitly set to IPS mode, so it can only be used to perform iSCSI and FCIP storage functions. IP storage ports do not bridge Ethernet frames or route other IP packets.

Each IPS port represents a single virtual Fibre Channel host in the Fibre Channel SAN. All the iSCSI hosts connected to this IPS port are merged and multiplexed through the single Fibre Channel host.

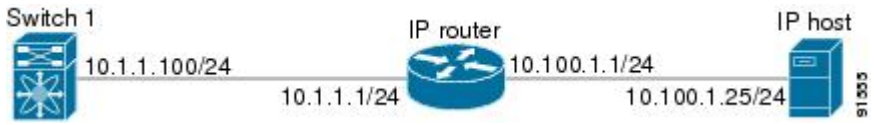
In large scale iSCSI deployments where the Fibre Channel storage subsystems require explicit LUN access control for every host device, use of proxy-initiator mode simplifies the configuration.

**Tip** IPStorage ports on any Fibre Channel module with IPS ports should not be configured in the same IPS broadcast domain as the management IPS port—they should be configured in a different broadcast domain, either by using separate standalone hubs or switches or by using separate VLANs.

# Basic IPStorage Configuration

The following figure shows an example of a basic IPStorage IP version 4 (IPv4) configuration.

***Figure 50: IPStorage IPv4 Configuration Example***



**Note** The port on the Ethernet switch to which the IPStorage interface is connected should be configured as a host port (also known as access port) instead of a switch port. Spanning tree configuration for that port (on the Ethernet switch) should disabled. This helps avoid the delay in the management port coming up due to delay from Ethernet spanning tree processing that the Ethernet switch would run if enabled. For Cisco Ethernet switches, use either the **switchport host** command in Cisco IOS or the **set port host command** in Catalyst OS.

# Fibre Channel Module with IPS Ports Core Dumps

IPS core dumps are different from the system's kernel core dumps for other modules. When the Fibre Channel module with IPS port's operating system (OS) unexpectedly resets, it is useful to obtain a copy of the memory image (called a IPS core dump) to identify the cause of the reset. Under that condition, the Fibre Channel module with IPS ports sends the core dump to the supervisor module for storage. Cisco MDS switches have two levels of IPS core dumps:

- Partial core dumps (default)—Each partial core dump consists of four parts (four files). All four files are saved in the active supervisor module.

  In Cisco MDS 9700 Series Switches with 24/10 port SAN Extension Modules, each partial core dump consists of five parts (five files). All five files are saved in the active supervisor module.

  Use the **show cores** command to list these files.

- Full core dumps—Each full core dump of Cisco MDS 9250i Switches and SSN-16 modules consists of 64 parts (64 files), and each full core dump of Cisco MDS 9700 Series Switches with 24/10 port SAN Extension Modules consists of 67 parts (67 files). The IPS core dump for MSM-18/4 modules consists of 32 parts. This dump cannot be saved on the supervisor module because of its large space requirement. They are copied directly to an external TFTP server.

  Use the **system cores tftp:** command to configure an external TFTP server to copy the IPS core dump (and other core dumps).

To configure IPS core dumps on the Fibre Channel module with IPS ports, follow these steps:

**Procedure**

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | switch# **config terminal** | Enters configuration mode. |
| **Step 2** | switch(config)# **ips core dump full** | Configures a dump of the full core generation for all Fibre Channel module with IPS ports in the switch.<br>`ips core dump full' successfully set for module 9` |
| **Step 3** | switch(config)# **no ips core dump full** | Configures a dump of the partial core (default) generation for the Fibre Channel module with IPS ports in slot 9.<br>`ips core dump partial' successfully set for module 9` |

### What to do next

#### Configuring Autonegotiation

By default, autonegotiation is enabled on all the IPStorage interface. You can enable or disable autonegotiation for a specified IPStorage interface. When autonegotiation is enabled, the port automatically detects the speed or pause method, and duplex of incoming signals based on the link partner. You can also detect link up conditions using the autonegotiation feature.

#### Configuring the MTU Frame Size

You can configure the interfaces on a switch to transfer large (or jumbo) frames on a port. The default IP maximum transmission unit (MTU) frame size is 1500 bytes for all Ethernet ports. By configuring jumbo frames on a port, the MTU size can be increased up to 9000 bytes.

**Note** The minimum MTU size is 576 bytes.

**Tip** MTU changes are disruptive, all FCIP links and iSCSI sessions flap when the software detects a change in the MTU size.

#### Configuring Promiscuous Mode

You can enable or disable promiscuous mode on a specific IPStorage interface. By enabling the promiscuous mode, the IPStorage interface receives all the packets and the software then filters and discards the packets that are not destined for that IPStorage interface.

# About VLANs for IPStorage

Virtual LANs (VLANs) create multiple virtual Layer 2 networks over a physical LAN network. VLANs provide traffic isolation, security, and broadcast control.

IPStorage ports automatically recognize Ethernet frames with IEEE 802.1Q VLAN encapsulation. If you need to have traffic from multiple VLANs terminated on one IPStorage port, configure subinterfaces—one for each VLAN.

If the Fibre Channel module with IPS ports or MPS-14/2 module is connected to a Cisco Ethernet switch, and you need to have traffic from multiple VLANs coming to one IPS port, verify the following requirements on the Ethernet switch:

- The encapsulation is set to 802.1Q and not ISL, which is the default.
- Use the VLAN ID as a subscription to the IPStorage interface name to create the subinterface name:

```
 <slot-number> / <port-number>
<VLAN-ID>
```

# Interface Subnet Requirements

IPStorage interfaces (major), subinterfaces (VLAN ID), and management interfaces (mgmt 0) can be configured in the same or different subnet depending on the configuration (see the following table).

*Table 14: Subnet Requirements for Ethernet Interfaces Within the Same Switch*

| Interface 1 | Interface 2 | Same Subnet Allowed | Notes |
|---|---|---|---|
| IPStorage 1/1 | IPStorage 1/2 | Yes | Two major interfaces can be configured in the same or different subnets. |
| IPStorage 1/1.100 | IPStorage 1/2.100 | Yes [1] | Two subinterfaces with the same VLAN ID can be configured in the same or different subnets. |
| IPStorage 1/1.100 | IPStorage 1/2.200 | No | Two subinterfaces with different VLAN IDs cannot be configured in the same subnet. |
| IPStorage 1/1 | IPStorage 1/1.100 | No | A subinterface cannot be configured on the same subnet as the major interface. |
| mgmt0 | IPStorage 1/1.100 | No | The mgmt0 interface cannot be configured in the same subnet as the IPStorage interfaces or subinterfaces. |
| mgmt0 | IPStorage 1/1 | No | |

[1] Except IPSec enabled interfaces, which should not be configured in the same IP subnet.

**Note**

- The configuration requirements in the above table applies to Ethernet PortChannels also.
- If there are multiple IPStorage interfaces configured with the same IP address or network-mask in the same IP subnet, packets may be sent out on incorrect IPS interface.

# Verifying IPStorage Connectivity

After the IPStorage interfaces are connected with valid IP addresses, verify the interface connectivity on each switch. Ping the IP host using the IP address of the host to verify that the static IP route is configured correctly.

**Note** If the connection fails, verify the following, and ping the IP host again:

- The IP address for the destination (IP host) is correctly configured.
- The host is active (powered on).
- The IP route is configured correctly.
- The IP host has a route to get to the IPStorage interface subnet.
- The IPStorage interface is in the `up` state.

# IPStorage IPv4-ACL Guidelines

**Tip** If IPv4-ACLs are already configured in a IPStorage interface, you cannot add this interface to an Ethernet PortChannel group.

Follow these guidelines when configuring IPv4-ACLs for IPStorage interfaces:

- Only use Transmission Control Protocol (TCP) or Internet Control Message Protocol (ICMP).

**Note** Other protocols such as User Datagram Protocol (UDP) and HTTP are not supported in IPStorage interfaces. Applying an ACL that contains rules for these protocols to a IPStorage interface is allowed but those rules have no effect.

- Apply IPv4-ACLs to the interface before you enable an interface. This ensures that the filters are in place before traffic starts flowing.
- Be aware of the following conditions:
  - If you use the **log-deny** option, a maximum of 50 messages are logged per second.
  - The **established** , **precedence** , and **fragments** options are ignored when you apply IPv4-ACLs (containing these options) to IPStorage interfaces.
  - If an IPv4-ACL rule applies to a preexisting TCP connection, that rule is ignored. For example if there is an existing TCP connection between A and B, and an IPv4-ACL specifies dropping all packets whose source is A and destination is B is subsequently applied, it will have no effect.

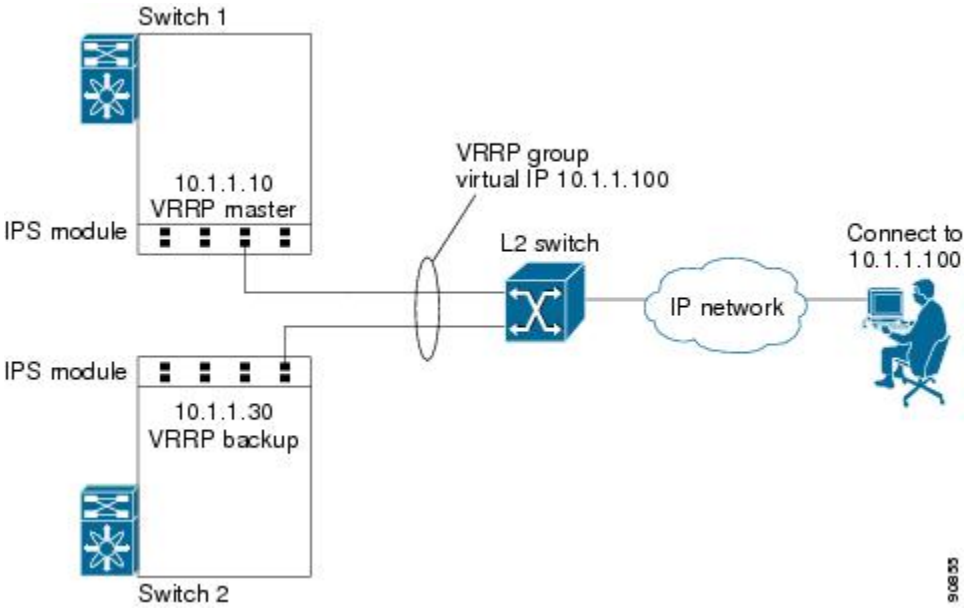
# Configuring IPStorage High Availability

Virtual Router Redundancy Protocol (VRRP) and Ethernet PortChannels are two IPStorage features that provide high availability for iSCSI and FCIP services.

# VRRP for iSCSI and FCIP Services

VRRP provides a redundant alternate path to the IPStorage port for iSCSI and FCIP services. VRRP provides IP address failover protection to an alternate IPStorage interface so the IP address is always available (see the below figure).

***Figure 51: VRRP Scenario***




In the above figure, all members of the VRRP group must be IPStorage ports. VRRP group members can be one or more of the following interfaces:

- One or more interfaces in the same Fibre Channel module with IPS ports or MSM-18/4 module
- Interfaces across Fibre Channel module with IPS ports or MSM-18/4 modules in one switch
- Interfaces across Fibre Channel module with IPS ports or MSM-18/4 modules in different switches
- IPStorage subinterfaces
- Ethernet PortChannels and PortChannel subinterfaces

You can configure no more than seven VRRP groups, both IPv4 and IPv6, on a IPStorage interface, including the main interface and all subinterfaces.

# Configuring VRRP for IPStorage Interfaces

To configure VRRP for IPStorage interfaces using IPv4 or IPv6, follow these steps:

**Procedure**

**Step 1** switch1# **config terminal**

Enters configuration mode.

**Step 2** switch(config)# **interface IPStorage 2/2**

Enters the interface configuration mode on the IPStorage interface (slot 2, port 2).

**Step 3** To configure VRRP for IPStorage interfaces using IPv4:

a) switch(config-if)# **ip address 10.1.1.10 255.255.255.0**

Assigns the IPv4 address (10.1.1.10) and subnet mask (255.255.255.0) for the IPStorage interface.

b) switch(config-if)# **no shutdown**

Enables the selected interface.

c) switch(config-if)# **vrrp 100**

Creates VR ID 100.

d) switch(config-if-vrrp) # **address 10.1.1.100**

Configures the virtual IPv4 address (10.1.1.100) for the selected VRRP group (identified by the VR ID).

**Note**
The virtual IPv4 address must be in the same subnet as the IPv4 address of the IPStorage interface. All members of the VRRP group must configure the same virtual IPv4 address.

e) switch(config-if-vrrp) # **priority 10**

Configures the priority for the selected interface within this VRRP group.

**Note**
The interface with the highest priority is selected as the master.

f) switch(config-if-vrrp) # **no shutdown**

Enables the VRRP protocol on the selected interface.

**Step 4** To configure VRRP for IPStorage interfaces using IPv6:

a) switch(config-if)# **ipv6 address 2001:0db8:800:200c::417a/64**

Assigns the IPv6 address for the IPStorage interface.

b) switch(config-if)# **no shutdown**

Enables the selected interface.

c) switch(config-if)# **vrrp ipv6 100**

Creates VR ID 100.

d) switch(config-if-vrrp-ipv6) # **address 2001:0db8:800:200c::417a**

Assigns single primary link-local IPv6 address or one of the multiple secondary IPv6 addresses.

**Note**
If this IPv6 address is the same as the physical IPv6 address, this switch is automatically the owner of this IPv6 address.

e) switch(config-if-vrrp-ipv6) # **priority 10**

Configures the priority for the selected interface within this VRRP group.

**Note**
The interface with the highest priority is selected as the master.

f) switch(config-if-vrrp-ipv6) # **no shutdown**

Enables the VRRP protocol on the selected interface.

**Note**
The VRRP **preempt** option is not supported on IPS interfaces. However, if the virtual IPv4 IP address is also the IPv4 IP address for the interface, then preemption is implicitly applied.

# Configuring CDP

The Cisco Discovery Protocol (CDP) is supported on the management Ethernet interface on the supervisor module and the IPStorage interfaces on the Fibre Channel module with IPS ports or MSM-18/4 module.

For information about configuring CDP, refer to the *Cisco MDS 9000 Series NX-OS Fundamentals Configuration Guide*.

# Changing Link Speed on IPStorage Interfaces

This sections includes the following topics:

## Changing Link Speed on Cisco MDS 9250i Multiservice Fabric Switch

The Cisco MDS 9250i Multiservice Fabric Switch has two IPStorage interfaces that support 1 Gbps and 10 Gbps link speeds. By default, IPStorage interfaces are configured at 10 Gbps link speed.

Switching between different link speeds is supported on Cisco 10 Gbps IPStorage platforms starting from Cisco MDS NX-OS Release 6.2(13). An ISSD to a release earlier than Cisco MDS NX-OS Release 6.2(13) when any of the IPStorage ports are configured at 1 Gbps, is disallowed. Reconfigure such ports back to the default link speed of 10 Gbps before attempting such a downgrade.

### Configuring 1 Gbps Link Speed on an IPStorage Interface

To configure 1 Gbps link speed on an IPStorage interface, follow these steps:

**Procedure**

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | switch1# **config terminal** | Enters configuration mode. |
| **Step 2** | switch(config)# **interface IPStorage slot-number/port-number-range** | Enters IPStorage interface configuration mode. |

| | Command or Action | Purpose |
|---|---|---|
| **Step 3** | switch(config-if)# **shutdown** | Administratively disables the interface and stops traffic through the interface. |
| **Step 4** | switch(config-if)# **switchport speed 1000**[2]<br>**Example:**<br>`This speed change will disrupt FCIP/iSCSI traffic for 5 mins on all IPStorage ports. If FCIP tunnels are configured please make sure max-bw <= 1000 Mbps and tcp-connections set to 2. Do you want to continue(y/n) ? [n]` | Sets the link speed of the interface and all subinterfaces to 1000 Mbps (1 Gbps).<br>**Note**<br>This command causes all IPStorage ports on the selected FCIP engine to be reset. This may cause traffic disruption for up to 5 minutes. By default, *n* is selected. Press Enter to abort the command. Enter *y* and press Enter to continue.<br>[3] |
| **Step 5** | switch(config-if) # **no shutdown** | Administratively enables the interface. |
| **Step 6** | switch(config-if)# **end** | Exits IPStorage interface configuration mode and returns to privileged EXEC mode. |
| **Step 7** | switch# **show ips status** | Displays the operational speed of the IPStorage port. |

## Configuring 10 Gbps Link Speed on an IPStorage Interface

To configure 10 Gbps link speed on an IPStorage interface, follow these steps:

### Procedure

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | switch1# **config terminal** | Enters configuration mode. |
| **Step 2** | switch(config)# **interface IPStorage** *slot-number/port-number-range* | Enters IPStorage interface configuration mode. |
| **Step 3** | switch(config-if)# **shutdown** | Administratively disables the interface and stops traffic through the interface. |
| **Step 4** | switch(config-if)# **switchport speed 10000**[4] | Sets the link speed of the interface and all subinterfaces to 10000 Mbps (10 Gbps).<br>**Note**<br>This command causes all IPStorage ports on the selected FCIP engine to be reset. This may cause traffic disruption for up to 5 minutes. By default, *n* is selected. Press Enter to abort the command. Enter *y* and press Enter to continue.<br>[5] |
| **Step 5** | switch(config-if) # **no shutdown** | Administratively enables the interface. |
| **Step 6** | switch(config-if)# **end** | Exits IPStorage interface configuration mode and returns to privileged EXEC mode. |

| | Command or Action | Purpose |
|---|---|---|
| **Step 7** | switch# **show ips status** | Displays the operational speed of the IPStorage port. |

### What to do next

If there is a mismatch between the configured link speed and the small form-factor pluggable (SFP) speed capabilities, the port goes into an Error Disabled state and a corresponding syslog message is logged. In such a scenario, either the configured link speed or the SFP should be changed. If the link speed is changed, even if the port is already enabled, the **shutdown** and **no shutdown** commands must be explicitly issued for the change to be applied.

For more information about supported 1 Gbps SFPs for a Cisco MDS 9250i Multiservice Fabric Switch, see the Cisco MDS 9000 Family Pluggable Transceivers Data Sheet .

For information about configuring FCIP tunnels with IPStorage interfaces at 1 Gbps speed, see Configuring FCIP, on page 7.

# Changing Link Speed on Cisco MDS 9220i Switch

**Note** For Cisco MDS 9220i switch, delete all the FCIP related configurations, such as profiles, tunnels, and so on, before switching from 1 or 10 Gbps speed to 40 Gbps speed or vice versa.

## Configuring 1 Gbps Link Speed on an IPStorage Interface

To configure 1 Gbps link speed on an IPStorage interface, follow these steps:

### Procedure

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | switch# **config terminal** | Enters configuration mode. |
| **Step 2** | switch(config)# **interface IPStorage 1/1-6** | Enters IPStorage interface configuration mode. |
| **Step 3** | switch(config-if)# **1G-speed-mode**<br>**Example:**<br>`This speed change will disrupt FCIP/iSCSI traffic for 60 seconds on selected IPStorage ports.`<br>`If FCIP tunnels are configured please make sure max-bw <= 1000 Mbps and tcp-connections set to 2.`<br>`Do you wish to continue(y/n)? [n]` | Sets the link speed of the interface IPS1/1-4 and all subinterfaces to 1000 Mbps (1 Gbps) and administratively enables the interface.<br>Sets IPS1/5-6 out-of-service. |
| **Step 4** | switch(config-if)# **end** | Exits IPStorage interface configuration mode and returns to privileged EXEC mode. |

## Configuring 10 Gbps Link Speed on an IPStorage Interface

To configure 10 Gbps link speed on an IPStorage interface, follow these steps:

**Procedure**

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | switch# **config terminal** | Enters configuration mode. |
| **Step 2** | switch(config)# **interface IPStorage 1/1-6** | Enters IPStorage interface configuration mode. |
| **Step 3** | switch(config-if)# **10G-speed-mode**<br>**Example:**<br>`This speed change will disrupt FCIP/iSCSI traffic`<br>`for 60 seconds on select IPStorage ports.`<br>`Do you wish to continue(y/n)? [n]` | Sets the link speed of the interface IPS1/1-4 and all subinterfaces to 10000 Mbps (10 Gbps) and administratively enables the interface.<br>Sets IPS1/5-6 out-of-service. |
| **Step 4** | switch(config-if)# **end** | Exits IPStorage interface configuration mode and returns to privileged EXEC mode. |

## Configuring 25 Gbps Link Speed on an IPStorage Interface

To configure 25 Gbps link speed on an IPStorage interface, follow these steps:

**Procedure**

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | switch# **config terminal** | Enters configuration mode. |
| **Step 2** | switch(config)# **interface IPStorage 1/1-6** | Enters IPStorage interface configuration mode.<br>**Note**<br>Configure the number of TCP connections to 5 for attaining maximum throughput. |
| **Step 3** | switch(config-if)# **25G-speed-mode** | Sets the link speed of the interface IPS1/4-5 to 25000 Mbps (25 Gbps) and administratively enables the interface IPS1/4-5.<br>Sets IPS1/1-3 and IPS1/6 to out-of-service.<br>**Note**<br>Configure the connected Ethernet interface to use FEC RS-IEEE when using IPStorage ports in 25g mode. |
| **Step 4** | switch(config-if)# **end** | Exits IPStorage interface configuration mode and returns to privileged EXEC mode. |

## Configuring 40 Gbps Link Speed on an IPStorage Interface

To configure 40 Gbps link speed on an IPStorage interface, follow these steps:

**Procedure**

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | switch# **config terminal** | Enters configuration mode. |
| **Step 2** | switch(config)# **interface IPStorage 1/1-6** | Enters IPStorage interface configuration mode.<br>**Note**<br>Configure the number of TCP connections to 5 for attaining maximum throughput. |
| **Step 3** | switch(config-if)# **40G-speed-mode** | Sets the link speed of the interface IPS1/6 to 40000 Mbps (40 Gbps) and administratively enables the interface.<br>Sets IPS1/1-5 to out-of-service. |
| **Step 4** | switch(config-if)# **end** | Exits IPStorage interface configuration mode and returns to privileged EXEC mode. |

# Changing Link Speed on Cisco MDS 24/10 port SAN Extension Module



**Note** For Cisco MDS 24/10 port SAN Extension Module, delete all the FCIP related configurations, such as profiles, tunnels, and so on, before switching from 1 or 10 Gbps speed to 40 Gbps speed or vice versa.

## Configuring 1 Gbps Link Speed on an IPStorage Interface

To configure 1 Gbps link speed on an IPStorage interface, follow these steps:

**Procedure**

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | switch# **config terminal** | Enters configuration mode. |
| **Step 2** | switch(config)# **interface IPStorage** *slot-number/port-number-range*, **IPStorage** *slot-number/port-number* | Enters IPStorage interface configuration mode.<br>**Note**<br>The values for *port-number-range* can be 1-4 or 5-8. The value for port number can be 9 if *port-number-range* is 1-4 and the value for port number can be 10 if *port-number-range* is 5-8. |

| | Command or Action | Purpose |
|---|---|---|
| **Step 3** | switch(config-if)# **1G-speed-mode**<br>**Example:**<br>`This speed change will disrupt FCIP/iSCSI traffic for 60 seconds on selected IPStorage ports.`<br>`If FCIP tunnels are configured please make sure max-bw <= 1000 Mbps and tcp-connections set to 2.`<br>`Do you wish to continue(y/n)? [n]` | Sets the link speed of the interface and all subinterfaces to 1000 Mbps (1 Gbps) and administratively enables the interface. |
| **Step 4** | switch(config-if)# **end** | Exits IPStorage interface configuration mode and returns to privileged EXEC mode. |

## Configuring 10 Gbps Link Speed on an IPStorage Interface

To configure 10 Gbps link speed on an IPStorage interface, follow these steps:

**Procedure**

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | switch# **config terminal** | Enters configuration mode. |
| **Step 2** | switch(config)# **interface IPStorage***slot-number/port-number-range*, **IPStorage** *slot-number/port-number* | Enters IPStorage interface configuration mode.<br>**Note**<br>The values for *port-number-range* can be 1-4 or 5-8. The value for port number can be 9 if *port-number-range* is 1-4 and the value for port number can be 10 if *port-number-range* is 5-8. |
| **Step 3** | switch(config-if)# **10G-speed-mode**<br>**Example:**<br>`This speed change will disrupt FCIP/iSCSI traffic for 60 seconds on selected IPStorage ports.`<br>`Do you wish to continue(y/n)? [n]` | Sets the link speed of the interface and all subinterfaces to 10000 Mbps (10 Gbps) and administratively enables the interface. |
| **Step 4** | switch(config-if)# **end** | Exits IPStorage interface configuration mode and returns to privileged EXEC mode. |

## Configuring 40 Gbps Link Speed on an IPStorage Interface

To configure 40 Gbps link speed on an IPStorage interface, follow these steps:

**Procedure**

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | switch# **config terminal** | Enters configuration mode. |
| **Step 2** | switch(config)# **interface IPStorage***slot-number/port-number-range*, **IPStorage** *slot-number/port-number* | Enters IPStorage interface configuration mode.<br>**Note**<br>The values for *port-number-range* can be 1-4 or 5-8. The value for port number can be 9 if *port-number-range* is 1-4 and the value for port number can be 10 if *port-number-range* is 5-8. |
| **Step 3** | switch(config-if)# **40G-speed-mode** | Sets the link speed of the interface to 40 Gbps and interfaces in the range to out-of-service. |
| **Step 4** | switch(config-if)# **end** | Exits IPStorage interface configuration mode and returns to privileged EXEC mode. |

# Displaying Statistics

This section provides examples to verify IPStorage interfaces and TCP/IP statistics on the IPStorage ports.

Use the **show interface IPStorage** command on each switch to verify that the interfaces are up and functioning as desired.

### Displaying the IPStorage Interface

```
switch# show interface IPStorage 4/1
IPStorage4/1 is up
Hardware is IPStorage, address is 008e.7339.39e7
Internet address is 10.197.141.81/24
MTU 2500 bytes
Port mode is IPS
Speed is 10 Gbps
Beacon is turned off
Auto-Negotiation is turned on
5 minutes input rate 77012744 bits/sec, 9626593 bytes/sec, 112755 frames/sec
5 minutes output rate 2762915176 bits/sec, 345364397 bytes/sec, 175258 frames/sec
71187036 packets input, 6078261484 bytes
0 multicast frames, 0 compressed
0 input errors, 0 frame, 0 overrun 0 fifo
110617842 packets output, 217860230652 bytes, 0 underruns
0 output errors, 0 collisions, 0 fifo
0 carrier errors


switch# show interface IPStorage 5/1-10 brief

--------------------------------------------------------------------------------
Interface Status IP Address Speed MTU
--------------------------------------------------------------------------------
IPStorage5/1 up 1.1.1.1/24 10 Gbps 2500
IPStorage5/2 up 2.2.2.2/24 10 Gbps 2500
```

```
IPStorage5/3 up 3.3.3.3/24 10 Gbps 2500
IPStorage5/4 up 4.4.4.4/24 10 Gbps 2500
IPStorage5/5 up 6811::3456/64 1 Gbps 2300
IPStorage5/6 up 9.9.9.1/24 1 Gbps 2500
IPStorage5/7 up 7.7.7.1/24 1 Gbps 2500
IPStorage5/8 up 8.8.8.1/24 1 Gbps 2500
IPStorage5/9 outOfServc -- auto 1500
IPStorage5/10 outOfServc -- auto 1500
```

# Displaying Ethernet MAC Statistics

The **show ips stats mac interface ips** command takes the IPStorage interface as a parameter and returns the statistics for that interface.

**Note** Use the physical interface, not the subinterface, to display Ethernet MAC statistics.

### Displaying IPStorage Interface MAC Statistics

```
switch# show ips stats mac interface ips 4/1

DPP HW IPStorage port 4 (octeon port 19) statistics

Rx stats :
  dropped          : 0          octs,   0          pkts
  l2 red pkts      : 0          mcast,  0          bcasts
  l3 red pkts      : 0          mcast,  0          bcasts
  oversize         : 0          pkts,   0          crcpkts
  runt             : 0          pkts,   0          crcpkts
  inband           : 0          octs,   0          pkts,      0      err
  pci raw          : 0          pkts
  fcs_align_err    : 0          pkts

  total            : 0          octs,   0          pkts

  length of [pkts]:-
   [64B]           : 0            [65B-127B]       : 0
   [128B-255B]     : 0            [256B-511B]      : 0
   [512B-1023B]    : 0            [1024B-1518B]    : 0
   [1519B-MAX]     : 0

Tx stats :
  total            : 0          octs,   0          pkts
  dropped          : 0          octs,   0          pkts

Hw-intf stats :
  total            : 0            Rx_octs,   0           Rx_pkts
  dropped          : 0            Rx_pkts
  total            : 0            Tx_octs,   0           Tx_pkts
  dropped          : 0            Tx_pkts
```

# Displaying TCP Statistics

Use the **show ips stats tcp interface ips** to display and verify TCP statistics. This command takes the main Ethernet interface as a parameter, and shows TCP stats along with the connection list and TCP state. The **detail** option shows all information maintained by the interface. See the following examples.

### Displaying TCP Statistics

```
switch# show ips stats tcp interface ips 4/1

TCP statistics for port IPStorage4/1
Connection Stats
0 active openings, 3 accepts
0 failed attempts, 12 reset received, 3 established
Segment stats
163 received, 355 sent, 0 retransmitted
0 bad segments received, 0 reset sent
TCP Active Connections
Local Address Remote Address State Send-Q Recv-Q
0.0.0.0:3260 0.0.0.0:0 LISTEN 0 0
```

### Displaying Detailed TCP Statistics

```
switch# show ips stats tcp interface ips 4/1 detail

TCP Statistics for port IPStorage4/1
TCP send stats
355 segments, 37760 bytes
222 data, 130 ack only packets
3 control (SYN/FIN/RST), 0 probes, 0 window updates
0 segments retransmitted, 0 bytes
0 retransmitted while on ethernet send queue, 0 packets split
0 delayed acks sent
TCP receive stats
163 segments, 114 data packets in sequence, 6512 bytes in sequence
0 predicted ack, 10 predicted data
0 bad checksum, 0 multi/broadcast, 0 bad offset
0 no memory drops, 0 short segments
0 duplicate bytes, 0 duplicate packets
0 partial duplicate bytes, 0 partial duplicate packets
0 out-of-order bytes, 1 out-of-order packets
0 packet after window, 0 bytes after window
0 packets after close
121 acks, 37764 ack bytes, 0 ack toomuch, 4 duplicate acks
0 ack packets left of snd_una, 0 non-4 byte aligned packets
8 window updates, 0 window probe
30 pcb hash miss, 0 no port, 0 bad SYN, 0 paws drops
TCP Connection Stats
0 attempts, 3 accepts, 3 established
3 closed, 2 drops, 0 conn drops
0 drop in retransmit timeout, 1 drop in keepalive timeout
0 drop in persist drops, 0 connections drained
TCP Miscellaneous Stats
115 segments timed, 121 rtt updated
0 retransmit timeout, 0 persist timeout
12 keepalive timeout, 11 keepalive probes
TCP SACK Stats
0 recovery episodes, 0 data packets, 0 data bytes
```

```
0 data packets retransmitted, 0 data bytes retransmitted
0 connections closed, 0 retransmit timeouts
TCP SYN Cache Stats
15 entries, 3 connections completed, 0 entries timed out
0 dropped due to overflow, 12 dropped due to RST
0 dropped due to ICMP unreach, 0 dropped due to bucket overflow
0 abort due to no memory, 0 duplicate SYN, 0 no-route SYN drop
0 hash collisions, 0 retransmitted
TCP Active Connections
Local Address Remote Address State Send-Q Recv-Q
0.0.0.0:3260 0.0.0.0:0 LISTEN 0 0
```

Use the **show ips stats icmp interface ips** to display and verify IP statistics. This command takes the main Ethernet interface as a parameter and returns the ICMP statistics for that interface.

### Displaying ICMP Statistics

```
switch# show ips stats icmp interface ips 4/1

ICMP Statistics for port IPStorage4/1
0 ICMP messages received
0 ICMP messages dropped due to errors
ICMP input histogram
0 destination unreachable
0 time exceeded
0 parameter problem
0 source quench
0 redirect
0 echo request
0 echo reply
0 timestamp request
0 timestamp reply
0 address mask request
0 address mask reply
ICMP output histogram
0 destination unreachable
0 time exceeded
0 parameter problem
0 source quench
0 redirect
0 echo request
0 echo reply
0 timestamp request
0 timestamp reply
0 address mask request
0 address mask reply
```

### Displaying IPStorage Ports Speed

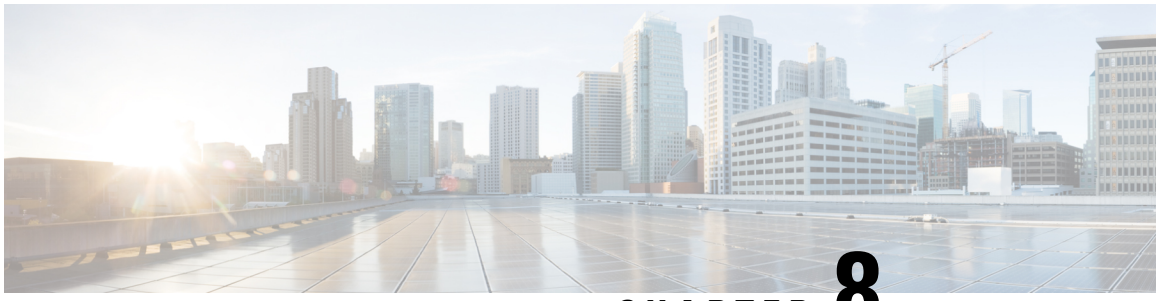
```
switch# show ips status
Port 1/1 READY 10G
Port 1/2 READY 1G
```

# Default Settings for IPStorage Services Parameters

The following table lists the default settings for IPStorage services parameters.

*Table 15: Default IPStorage Parameters*

| Parameters | Default |
|---|---|
| IPS core size | Partial |



CHAPTER 8

# Configuring IPv4 for Gigabit Ethernet Interfaces

Cisco MDS 9000 Family supports IP version 4 (IPv4) on Gigabit Ethernet interfaces. This chapter describes how to configure IPv4 addresses and other IPv4 features.

This chapter includes the following topics:


- Overview of IPv4, on page 229
- Basic Gigabit Ethernet Configuration for IPv4, on page 230
- Configuring Autonegotiation, on page 231
- Configuring the MTU Frame Size, on page 231
- Configuring Promiscuous Mode, on page 232
- Verifying Gigabit Ethernet Connectivity, on page 233
- VLANs Support in Cisco MDS NX-OS, on page 233
- Configuring Static IPv4 Routing, on page 235
- Displaying the IPv4 Route Table, on page 235
- IPv4-Access Control Lists, on page 235
- Address Resolution Protocol Cache, on page 237
- Displaying IPv4 Statistics, on page 238
- Default Settings for IPv4 Parameters, on page 238


## Overview of IPv4

Cisco MDS 9000 Family supports IP version 4 (IPv4) on Gigabit Ethernet interfaces. Both FCIP and iSCSI rely on TCP/IP for network connectivity. On each Fibre Channel module with IPS ports or MPS-14/2 module, connectivity is provided in the form of Gigabit Ethernet interfaces that are appropriately configured.

A new port mode, called IPS, is defined for Gigabit Ethernet ports on each Fibre Channel module with IPS ports or MPS-14/2 module. IP storage ports are implicitly set to IPS mode, so it can only be used to perform iSCSI and FCIP storage functions. IP storage ports do not bridge Ethernet frames or route other IP packets.

Each IPS port represents a single virtual Fibre Channel host in the Fibre Channel SAN. All the iSCSI hosts connected to this IPS port are merged and multiplexed through the single Fibre Channel host.

**Note** The Gigabit Ethernet interfaces on the MPS-14/2 module do not support EtherChannel.

Both FCIP and iSCSI rely on TCP/IP for network connectivity. On each Fibre Channel module with IPS ports or MSM-18/4 module, connectivity is provided in the form of Gigabit Ethernet interfaces that are appropriately configured. This section covers the steps required to configure IP for subsequent use by FCIP and iSCSI.

A new port mode, called IPS, is defined for Gigabit Ethernet ports on each Fibre Channel module with IPS ports or MSM-18/4 module. IP storage ports are implicitly set to IPS mode, so it can only be used to perform iSCSI and FCIP storage functions. IP storage ports do not bridge Ethernet frames or route other IP packets.

Each IPS port represents a single virtual Fibre Channel host in the Fibre Channel SAN. All the iSCSI hosts connected to this IPS port are merged and multiplexed through the single Fibre Channel host.

In large scale iSCSI deployments where the Fibre Channel storage subsystems do not require explicit LUN access control for every host device, use of proxy-initiator mode simplifies the configuration.

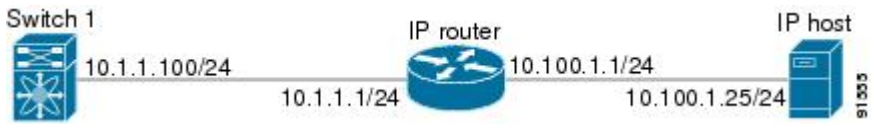
**Note** The Gigabit Ethernet interfaces on the MSM-18/4 module do not support EtherChannel.

**Tip** Gigabit Ethernet ports on any Fibre Channel module with IPS ports or MSM-18/4 module should not be configured in the same Ethernet broadcast domain as the management Ethernet port. They should be configured in a different broadcast domain, either by using separate standalone hubs or switches or by using separate VLANs.

# Basic Gigabit Ethernet Configuration for IPv4

The following figure shows an example of a basic Gigabit Ethernet IP version 4 (IPv4) configuration.

***Figure 52: Gigabit Ethernet IPv4 Configuration Example***




**Note** The port on the Ethernet switch to which the MDS Gigabit Ethernet interface is connected should be configured as a host port (also known as access port) instead of a switch port. Spanning tree configuration for that port (on the Ethernet switch) should be disabled. This helps avoid the delay in the management port coming up due to delay from Ethernet spanning tree processing that the Ethernet switch would run if enabled. For Cisco Ethernet switches, use either the **switchport host** command in Cisco IOS or the **set port host** command in the Catalyst OS.

To configure the Gigabit Ethernet interface for the example in the above image, follow these steps:

**Procedure**

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | switch# **config terminal** | Enters configuration mode. |

| | Command or Action | Purpose |
|---|---|---|
| **Step 2** | switch(config)# **interface gigabitethernet 2/2** | Enters the interface configuration mode on the Gigabit Ethernet interface (slot 2, port 2). |
| **Step 3** | switch(config-if)# **ip address 10.1.1.100 255.255.255.0** | Enters the IPv4 address (10.1.1.100) and subnet mask (255.255.255.0) for the Gigabit Ethernet interface. |
| **Step 4** | switch(config-if)# **no shutdown** | Enables the interface. |

# Configuring Autonegotiation

By default, autonegotiation is enabled all Gigabit Ethernet interface. You can enable or disable autonegotiation for a specified Gigabit Ethernet interface. When autonegotiation is enabled, the port automatically detects the speed or pause method, and duplex of incoming signals based on the link partner. You can also detect link up conditions using the autonegotiation feature.



**Note** When using DS-SFP-GE-T (copper SFPs) on Gigabit Ethernet interfaces in a DS-X9316-SSNK9 module, auto-negotiation should be disabled.

To configure autonegotiation, follow these steps:

**Procedure**

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | switch# **config terminal** | Enters configuration mode. |
| **Step 2** | switch(config)# **interface gigabitethernet 2/2** | Enters the interface configuration mode on the Gigabit Ethernet interface (slot 2, port 2). |
| **Step 3** | switch(config-if)# **switchport auto-negotiate** | Enables autonegotiation for this Gigabit Ethernet interface (default). |
| **Step 4** | switch(config-if)# **no switchport auto-negotiate** | Disables autonegotiation for this Gigabit Ethernet interface. |

# Configuring the MTU Frame Size

You can configure the interfaces on a switch to transfer large (or jumbo) frames on a port. The default IP maximum transmission unit (MTU) frame size is 1500 bytes for all Ethernet ports. By configuring jumbo frames on a port, the MTU size can be increased up to 9000 bytes.



**Note** The minimum MTU size is 576 bytes.

**Tip** MTU changes are disruptive, all FCIP links and iSCSI sessions flap when the software detects a change in the MTU size.

You do not need to explicitly issue the **shutdown** and **no shutdown** commands.

To configure the MTU frame size, follow these steps:

**Procedure**

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | switch# **config terminal** | Enters configuration mode. |
| **Step 2** | switch(config)# **interface gigabitethernet 2/2** | Enters the interface configuration mode on the Gigabit Ethernet interface (slot 2, port 2). |
| **Step 3** | switch(config-if)# **switchport mtu 3000** | Changes the MTU size to 3000 bytes. The default is 1500 bytes. |

# Configuring Promiscuous Mode

You can enable or disable promiscuous mode on a specific Gigabit Ethernet interface. By enabling the promiscuous mode, the Gigabit Ethernet interface receives all the packets and the software then filters and discards the packets that are not destined for that Gigabit Ethernet interface.

To configure the promiscuous mode, follow these steps:

**Procedure**

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | switch# **config terminal** | Enters configuration mode. |
| **Step 2** | switch(config)# **interface gigabitethernet 2/2** | Enters the interface configuration mode on the Gigabit Ethernet interface (slot 2, port 2). |
| **Step 3** | switch(config-if)# **switchport promiscuous-mode on** | Enables promiscuous mode for this Gigabit Ethernet interface. The default is **off**. |
| **Step 4** | switch(config-if)# **switchport promiscuous-mode off** | Disables (default) promiscuous mode for this Gigabit Ethernet interface. |
| **Step 5** | switch(config-if)# **no switchport promiscuous-mode** | Disables (default) the promiscuous mode for this Gigabit Ethernet interface. |

# Verifying Gigabit Ethernet Connectivity

Once the Gigabit Ethernet interfaces are connected with valid IP addresses, verify the interface connectivity on each switch. Ping the IP host using the IP address of the host to verify that the static IP route is configured correctly.

If the connection fails, verify the following, and ping the IP host again:

- The IP address for the destination (IP host) is correctly configured.
- The host is active (powered on).
- The IP route is configured correctly.
- The IP host has a route to get to the Gigabit Ethernet interface subnet.
- The Gigabit Ethernet interface is in the `up` state.

Use the **ping** command to verify the Gigabit Ethernet connectivity (see Example 7-1). The ping command sends echo request packets out to a remote device at an IP address that you specify.

Use the **show interface gigabitethernet** command to verify if the Gigabit Ethernet interface is up.

**Verifying Gigabit Ethernet Connectivity**

```
switch# ping 10.100.1.25
PING 10.100.1.25 (10.100.1.25): 56 data bytes
64 bytes from 10.100.1.25: icmp_seq=0 ttl=255 time=0.1 ms
64 bytes from 10.100.1.25: icmp_seq=1 ttl=255 time=0.1 ms
64 bytes from 10.100.1.25: icmp_seq=2 ttl=255 time=0.1 ms
--- 10.100.1.25 ping statistics ---
3 packets transmitted, 3 packets received, 0% packet loss
round-trip min/avg/max = 0.1/0.1/0.1 ms
```

# VLANs Support in Cisco MDS NX-OS

This section describes virtual LAN (VLAN) support in Cisco MDS NX-OS and includes the following topics:

## VLANs for Gigabit Ethernet

Virtual LANs (VLANs) create multiple virtual Layer 2 networks over a physical LAN network. VLANs provide traffic isolation, security, and broadcast control.

Gigabit Ethernet ports automatically recognize Ethernet frames with IEEE 802.1Q VLAN encapsulation. If you need to have traffic from multiple VLANs terminated on one Gigabit Ethernet port, configure subinterfaces—one for each VLAN.

If the Fibre Channel module with IPS ports or MSM-18/4 module is connected to a Cisco Ethernet switch, and you need to have traffic from multiple VLANs coming to one IPS port, verify the following requirements on the Ethernet switch:

**Note**

- The Ethernet switch port connected to the Fibre Channel module with IPS ports or MSM-18/4 module is configured as a trunking port.
- The encapsulation is set to 802.1Q and not ISL, which is the default.

Use the VLAN ID as a subscription to the Gigabit Ethernet interface name to create the subinterface name:

```
slot-number / port-number.VLAN-ID
```

# Configuring a VLAN Subinterface

To configure a VLAN subinterface (VLAN ID), follow these steps:

**Procedure**

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | switch# **config terminal** | Enters configuration mode. |
| **Step 2** | switch(config)# **interface gigabitethernet 2/2** | Enters the interface configuration mode on the Gigabit Ethernet interface (slot 2, port 2). |
| **Step 3** | switch(config-if)# **ip address 10.1.1.101 255.255.255.0** | Enters the IPv4 address (10.1.1.100) and subnet mask (255.255.255.0) for the Gigabit Ethernet interface. |
| **Step 4** | switch(config-if)# **shutdown** | Enables the interface. |

# Interface Subnet Requirements

Gigabit Ethernet interfaces (major), subinterfaces (VLAN ID), and management interfaces (mgmt 0) can be configured in the same or different subnet depending on the configuration (see the following table).

***Table 16: Subnet Requirements for Interfaces***

| Interface 1 | Interface 2 | Same Subnet Allowed | Notes |
|---|---|---|---|
| Gigabit Ethernet 1/1 | Gigabit Ethernet 1/2 | Yes | Two major interfaces can be configured in the same or different subnets. |
| Gigabit Ethernet 1/1.100 | Gigabit Ethernet 1/2.100 | Yes | Two subinterfaces with the same VLAN ID can be configured in the same or different subnets. |
| Gigabit Ethernet 1/1.100 | Gigabit Ethernet 1/2.200 | No | Two subinterfaces with different VLAN IDs cannot be configured in the same subnet. |
| Gigabit Ethernet 1/1 | Gigabit Ethernet 1/1.100 | No | A subinterface cannot be configured on the same subnet as the major interface. |

| Interface 1 | Interface 2 | Same Subnet Allowed | Notes |
|---|---|---|---|
| mgmt0 | Gigabit Ethernet 1/1.100 | No | The mgmt0 interface cannot be configured in the same subnet as the Gigabit Ethernet interfaces or subinterfaces. |
| mgmt0 | Gigabit Ethernet 1/1 | No | |



**Note** The configuration requirements in the above table also apply to Ethernet PortChannels.

# Configuring Static IPv4 Routing

To configure static IPv4 routing (see the above figure) through the Gigabit Ethernet interface, follow these steps:

**Procedure**

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | switch# **config terminal** | Enters configuration mode. |
| **Step 2** | switch(config)# **ip route 10.100.1.0 255.255.255.0 10.1.1.1** | Enters the IP subnet (10.100.1.0 255.255.255.0) of the IP host and configures the next hop 10.1.1.1, which is the IPv4 address of the router connected to the Gigabit Ethernet interface. |

# Displaying the IPv4 Route Table

The **ip route interface** command takes the Gigabit Ethernet interface as a parameter and returns the route table for the interface.

### Displaying an IP Route Table

```
switch# show ips ip route interface gig 8/1
Codes: C - connected, S - static
No default gateway
C 10.1.3.0/24 is directly connected, GigabitEthernet8/1
```

Connected (C) identifies the subnet in which the interface is configured (directly connected to the interface). Static (S) identifies the static routes that go through the router.

# IPv4-Access Control Lists

This section describes the guidelines for IPv4 access control lists and how to apply them to Gigabit Ethernet interfaces.

This section includes the following topics:

# Gigabit Ethernet Guidelines

Follow these guidelines when configuring for Gigabit Ethernet interfaces:

- Only use Transmission Control Protocol (TCP) or Internet Control Message Protocol (ICMP).

**Note** Other protocols such as User Datagram Protocol (UDP) and HTTP are not supported in Gigabit Ethernet interfaces. Applying an ACL that contains rules for these protocols to a Gigabit Ethernet interface is allowed but those rules have no effect.

- Apply to the interface before you enable an interface. This ensures that the filters are in place before traffic starts flowing.
- Be aware of the following conditions:
  - If you use the **log-deny** option, a maximum of 50 messages are logged per second.
  - The **established** option is ignored when you apply containing this option to Gigabit Ethernet interfaces.
  - If a rule applies to a pre-existing TCP connection, that rule is ignored. For example if there is an existing TCP connection between A and B and an IPv4-ACL which specifies dropping all packets whose source is A and destination is B is subsequently applied, it will have no effect.

**Note** If IPv4-ACLs are already configured in a Gigabit Ethernet interface, you cannot add this interface to an Ethernet PortChannel group. For information on configuring IPv4-ACLs, see the Cisco Fabric Manager Security Configuration Guide and Cisco MDS 9000 Family NX-OS Security Configuration Guide.

# Applying on Gigabit Ethernet Interfaces

To apply on a Gigabit Ethernet interface, follow these steps:

**Procedure**

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | switch# **config terminal** | Enters configuration mode. |
| **Step 2** | switch(config)# **interface gigabitethernet 3/1** | Configures a Gigabit Ethernet interface (3/1). |
| **Step 3** | switch(config-if)# **ip access-group SampleName** | Applies the IPv4-ACL SampleName on Gigabit Ethernet 3/1 for both ingress and egress traffic (if the association does not exist already). |

| | Command or Action | Purpose |
|---|---|---|
| **Step 4** | switch(config-if)# **ip access-group SampleName1 in** | Applies the IPv4-ACL SampleName on Gigabit Ethernet 3/1 for ingress traffic. |
| **Step 5** | switch(config-if)# **ip access-group SampleName2 out** | Applies the IPv4-ACL SampleName on Gigabit Ethernet 3/1 for egress traffic (if the association does not exist already). |

# Address Resolution Protocol Cache

Cisco MDS NX-OS supports Address Resolution Protocol (ARP) cache for Gigabit Ethernet interface configured for IPv4. This section includes the following topics:

## Displaying ARP Cache

You can display the ARP cache on Gigabit Ethernet interfaces.

Use the physical interface, not the subinterface, for all ARP cache commands.

Use the **show ips arp interface gigabitethernet** command to display the ARP cache on the Gigabit Ethernet interfaces. This command takes the Ethernet interface as a parameter and returns the ARP cache for that interface.

### Displaying ARP Cache

```
switch# show ips arp interface gigabitethernet 7/1
Protocol Address Age (min) Hardware Addr Type Interface
Internet 20.1.1.5 3 0005.3000.9db6 ARPA GigabitEthernet7/1
Internet 20.1.1.10 7 0004.76eb.2ff5 ARPA GigabitEthernet7/1
Internet 20.1.1.11 16 0003.47ad.21c4 ARPA GigabitEthernet7/1
Internet 20.1.1.12 6 0003.4723.c4a6 ARPA GigabitEthernet7/1
Internet 20.1.1.13 13 0004.76f0.ef81 ARPA GigabitEthernet7/1
Internet 20.1.1.14 0 0004.76e0.2f68 ARPA GigabitEthernet7/1
Internet 20.1.1.15 6 0003.47b2.494b ARPA GigabitEthernet7/1
Internet 20.1.1.17 2 0003.479a.b7a3 ARPA GigabitEthernet7/1
.
.
.
```

## Clearing ARP Cache

The ARP cache can be cleared in two ways: clearing just one entry or clearing all entries in the ARP cache.

Use the **clear ips arp** command to clear the ARP cache.

### Clearing an ARP Cache Entry

```
switch# clear ips arp address 10.2.2.2 interface gigabitethernet 8/7
arp clear successful
```

### Clearing All ARP Cache Entries

```
switch# clear ips arp interface gigabitethernet 8/7
arp clear successful
```

# Displaying IPv4 Statistics

Use the **show ips stats ip interface gigabitethernet** to display and verify IP v4 statistics. This command takes the main Ethernet interface as a parameter and returns the IPv4 statistics for that interface.

Use the physical interface, not the subinterface, to display IPv4 statistics.

### Displaying IPv4 Statistics

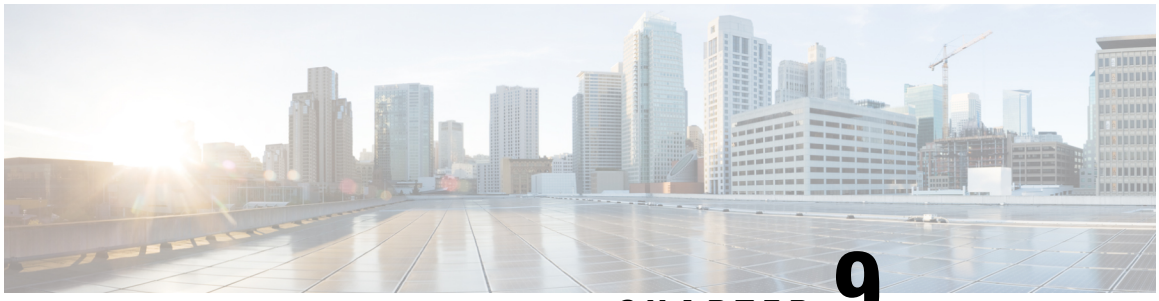
```
switch# show ips stats ip interface gigabitethernet 4/1
Internet Protocol Statistics for port GigabitEthernet4/1
168 total received, 168 good, 0 error
0 reassembly required, 0 reassembled ok, 0 dropped after timeout
371 packets sent, 0 outgoing dropped, 0 dropped no route
0 fragments created, 0 cannot fragment
```

# Default Settings for IPv4 Parameters

The below table lists the default settings for IPv4 parameters.

*Table 17: Default IPv4 Parameters*

| Parameters | Default |
|---|---|
| IPv4 MTU frame size | 1500 bytes for all Ethernet ports |
| Autonegotiation | Enabled |
| Promiscuous mode | Disabled |



CHAPTER 9

# Configuring IPv6

Cisco MDS 9000 Family supports IP version 6 This chapter describes how to configure IPv6 addresses and other IPv6 features.

This chapter includes the following topics:


- Overview of IPv6, on page 239
- Configuring Basic Connectivity for IPv6, on page 248
- Verifying Neighbor Discovery Parameter Configuration, on page 252
- Duplicate Address Detection Attempts, on page 252
- Reachability Time, on page 252
- Retransmission Time, on page 253
- Verifying Neighbor Discovery Parameter Configuration, on page 253
- IPv6 Static Routes, on page 254
- Gigabit Ethernet IPv6-ACL Guidelines, on page 255
- Transitioning from IPv4 to IPv6, on page 256
- Displaying IPv6, on page 256
- Default Settings, on page 257


# Overview of IPv6

IP version 6 (IPv6) provides extended addressing capability beyond those provided in IP version 4 (IPv4) in Cisco MDS NX-OS by quadrupling the number of network address bits from 32 bits (in IPv4) to 128 bits. The architecture of IPv6 has been designed to allow existing IPv4 users to transition easily to IPv6 while providing services such as end-to-end security, quality of service (QoS), and globally unique addresses.

IPv6 provides the following enhancements over IPv4:

- Allows networks to scale and provide global reachability.
- Reduces the need for private address and network address translation (NAT).
- Provides simpler autoconfiguration of addresses.


IPv6 provides the following enhancements over IPv4:

- Allows networks to scale and provide global reachability.
- Reduces the need for private address and network address translation (NAT).

- Provides simpler autoconfiguration of addresses.

**Note** For Cisco NX-OS features that use IP addressing, refer to the chapters in this guide that describe those features for information on IPv6 addressing support.

**Note** To configure IP version 4 (IPv4) on a Gigabit Ethernet interface, see Chapter 7, Configuring IPv4 for Gigabit Ethernet Interfaces.

This section describes the IPv6 features supported by Cisco MDS NX-OS and includes the following topics:

# Extended IPv6 Address Space for Unique Addresses

IPv6 extends the address space by quadrupling the number of network address bits from 32 bits (in IPv4) to 128 bits, which provides many more globally unique IP addresses. By being globally unique, IPv6 addresses enable global reachability and end-to-end security for networked devices, functionality that is crucial to the applications and services that are driving the demand for more addresses.

# IPv6 Address Formats

IPv6 addresses are represented as a series of 16-bit hexadecimal fields separated by colons (:) in the format x:x:x:x:x:x:x:x. The following are examples of IPv6 addresses:

- 2001:0DB8:7654:3210:FEDC:BA98:7654:3210
- 2001:0DB8:0:0:8:800:200C:417A

It is common for IPv6 addresses to contain successive hexadecimal fields of zeros. To make IPv6 addresses easier to use, two colons (::) may be used to compress successive hexadecimal fields of zeros at the beginning, middle, or end of an IPv6 address (the colons represent successive hexadecimal fields of zeros). Table 8-1 lists compressed IPv6 address formats.

**Note** Two colons (::) can be used only once in an IPv6 address to represent the longest successive hexadecimal fields of zeros.

**Note** The hexadecimal letters in IPv6 addresses are not case-sensitive.

***Table 18: Compressed IPv6 Address Formats***

| IPv6 Address Type | Uncompressed Format | Compressed Format |
|---|---|---|
| Unicast | 2001:0DB8:800:200C:0:0:0:417A | 2001:0DB8:800:200C::417A |

| IPv6 Address Type | Uncompressed Format | Compressed Format |
|---|---|---|
| Multicast | FF01:0:0:0:0:0:0:101 | FF01::101 |

# IPv6 Address Prefix Format

An IPv6 address prefix, in the format ipv6-prefix/prefix-length, can be used to represent bit-wise contiguous blocks of the entire address space. The ipv6-prefix is specified in hexadecimal using 16-bit values between the colons. The prefix-length is a decimal value that indicates how many of the

high-order contiguous bits of the address comprise the prefix (the network portion of the address). For example, 2001:0DB8:8086:6502::/32 is a valid IPv6 prefix.

# IPv6 Address Type-Unicast

An IPv6 unicast address is an identifier for a single interface on a single node. A packet that is sent to a unicast address is delivered to the interface identified by that address. The Cisco MDS NX-OS supports the following IPv6 unicast address types:

## Global Addresses

Global IPv6 addresses are defined by a global routing prefix, a subnet ID, and an interface ID. Figure 8-1 shows the structure of a global address.

***Figure 53: Global Address Format***



Addresses with a prefix of 2000::/3 (001) through E000::/3 (111) are required to have 64-bit interface identifiers in the extended universal identifier (EUI)-64 format. The Internet Assigned Numbers Authority (IANA) allocates the IPv6 address space in the range of 2000::/16 to regional registries.

The aggregatable global address typically consists of a 48-bit global routing prefix and a 16-bit subnet ID or Site-Level Aggregator (SLA). In the IPv6 aggregatable global unicast address format document (RFC 2374), the global routing prefix included two other hierarchically structured fields named Top-Level Aggregator (TLA) and Next-Level Aggregator (NLA).The IETF decided to remove the TLS and NLA fields from the RFCs because these fields are policy-based. Some existing IPv6 networks deployed before the change might still be using networks based on the older architecture.

A 16-bit subnet field called the subnet ID could be used by individual organizations to create their own local addressing hierarchy and to identify subnets. A subnet ID is similar to a subnet in IPv4, except that an organization with an IPv6 subnet ID can support up to 65,535 individual subnets.

An interface ID is used to identify interfaces on a link. The interface ID must be unique to the link. They may also be unique over a broader scope. In many cases, an interface ID will be the same as, or based on, the link-layer address of an interface, which results in a globally unique interface ID. Interface IDs used in aggregatable global unicast and other IPv6 address types must be 64 bits long and constructed in the modified EUI-64 format.

Cisco MDS NX-OS supports IEEE 802 interface types (for example, Gigabit Ethernet interfaces). The first three octets (24 bits) are taken from the Organizationally Unique Identifier (OUI) of the 48-bit link-layer address (MAC address) of the interface, the fourth and fifth octets (16 bits) are a fixed hexadecimal value of FFFE, and the last three octets (24 bits) are taken from the last three octets of the MAC address. The construction

of the interface ID is completed by setting the Universal/Local (U/L) bit—the seventh bit of the first octet—to a value of 0 or 1. A value of 0 indicates a locally administered identifier; a value of 1 indicates a globally unique IPv6 interface identifier (see Figure 8-2).

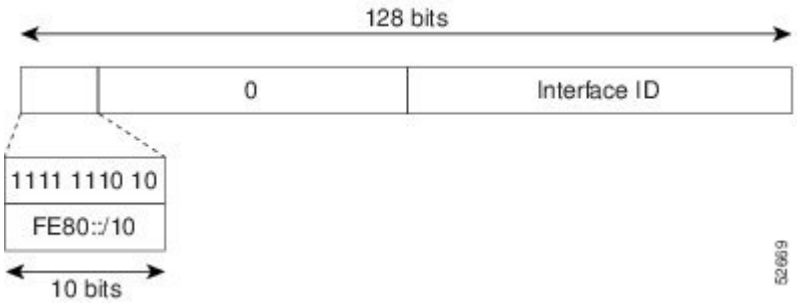
***Figure 54: Interface Identifier Format***



## Link-Local Address

A link-local address is an IPv6 unicast address that is automatically configured on an interface using the link-local prefix FE80::/10 and the interface identifier in the modified EUI-64 format. Link-local addresses are used in the neighbor discovery protocol and the stateless autoconfiguration process. Nodes on a local link can use link-local addresses to communicate. The below image shows the structure of a link-local address.

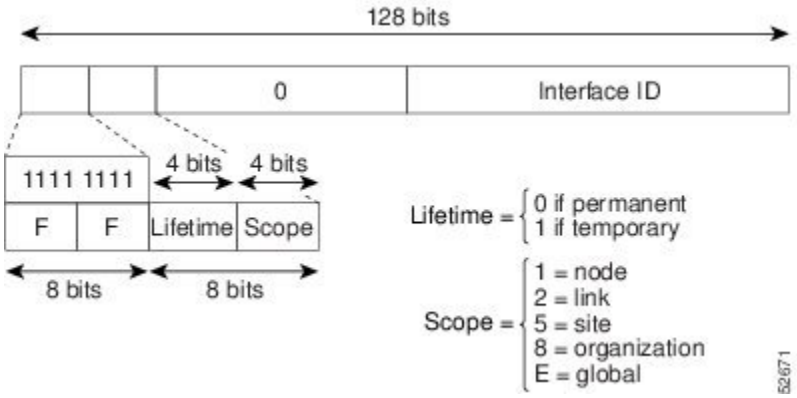
***Figure 55: Link-Local Address Format***




# IPv6 Address Type-Multicast

An IPv6 multicast address is an IPv6 address that has a prefix of FF00::/8 (1111 1111). An IPv6 multicast address is an identifier for a set of interfaces that typically belong to different nodes. A packet sent to a multicast address is delivered to all interfaces identified by the multicast address. The second octet following the prefix defines the lifetime and scope of the multicast address. A permanent multicast address has a lifetime parameter equal to 0; a temporary multicast address has a lifetime parameter equal to 1. A multicast address has the scope of a node, link, site, or organization, or a global scope has a scope parameter of 1, 2, 5, 8, or E, respectively. For example, a multicast address with the prefix FF02::/16 is a permanent multicast address with a link scope. The following figure shows the format of the IPv6 multicast address.

*Figure 56: Figure 8-4 IPv6 Multicast Address Format*

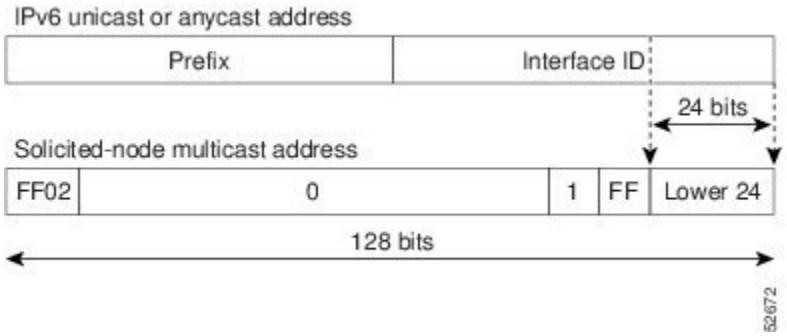



IPv6 hosts are required to join (receive packets destined for) the following multicast groups:

- All-node multicast group FF02::1.
- Solicited-node multicast group FF02:0:0:0:0:1:FF00:0000/104 concatenated with the low-order 24 bit of the unicast address.

The solicited-node multicast address is a multicast group that corresponds to an IPv6 unicast address. IPv6 nodes must join the associated solicited-node multicast group for every unicast address to which it is assigned. The IPv6 solicited-node multicast address has the prefix FF02:0:0:0:0:1:FF00:0000/104 concatenated with the 24 low-order bits of a corresponding IPv6 unicast address. (See the below figure.) For example, the solicited-node multicast address corresponding to the IPv6 address 2037::01:800:200E:8C6C is FF02::1:FF0E:8C6C. Solicited-node addresses are used in neighbor solicitation messages.

*Figure 57: IPv6 Solicited-Node Multicast Address Format*




**Note** There are no broadcast addresses in IPv6. IPv6 multicast addresses are used instead of broadcast addresses.

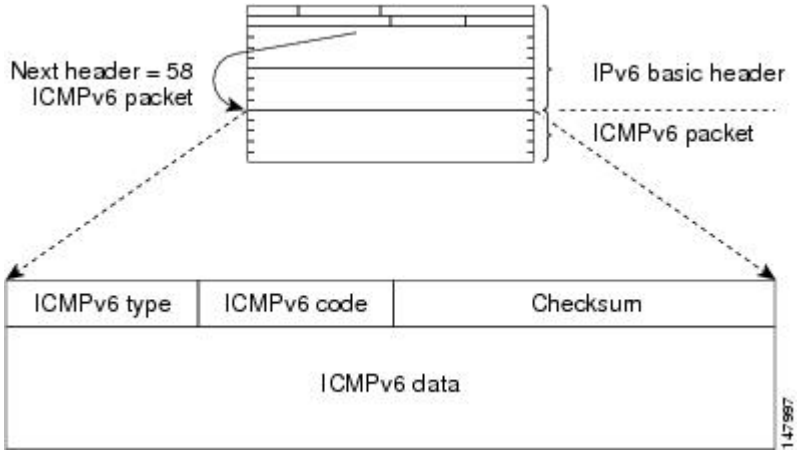
# ICMP for IPv6

Internet Control Message Protocol (ICMP) in IPv6 functions the same as ICMP in IPv4—ICMP generates error messages such as ICMP destination unreachable messages, and informational messages such as ICMP echo request and reply messages. Additionally, ICMP packets in IPv6 are used in the IPv6 neighbor discovery

process, path MTU discovery, and the Multicast Listener Discovery (MLD) protocol for IPv6. MLD is based on version 2 of the Internet Group Management Protocol (IGMP) for IPv4.

A value of 58 in the Next Header field of the basic IPv6 packet header identifies an IPv6 ICMP packet. ICMP packets in IPv6 resemble a transport-layer packet in the sense that the ICMP packet follows all the extension headers and is the last piece of information in the IPv6 packet. Within IPv6 ICMP packets, the ICMPv6 Type and ICMPv6 Code fields identify IPv6 ICMP packet specifics, such as the ICMP message type. The value in the Checksum field is derived (computed by the sender and checked by the receiver) from the fields in the IPv6 ICMP packet and the IPv6 pseudoheader. The ICMPv6 Data field contains error or diagnostic information relevant to IP packet processing. The following figure shows the IPv6 ICMP packet header format.

***Figure 58: IPv6 ICMP Packet Header Format***




# Path MTU Discovery for IPv6

As in IPv4, path MTU discovery in IPv6 allows a host to dynamically discover and adjust to differences in the MTU size of every link along a given data path. In IPv6, however, fragmentation is handled by the source of a packet when the path MTU of one link along a given data path is not large enough to accommodate the size of the packets. Having IPv6 hosts handle packet fragmentation saves IPv6 router processing resources and helps IPv6 networks run more efficiently.

**Note** In IPv4, the minimum link MTU is 68 octets, which means that the MTU size of every link along a given data path must support an MTU size of at least 68 octets.

In IPv6, the minimum link MTU is 1280 octets. We recommend using MTU value of 1500 octets for IPv6 links.

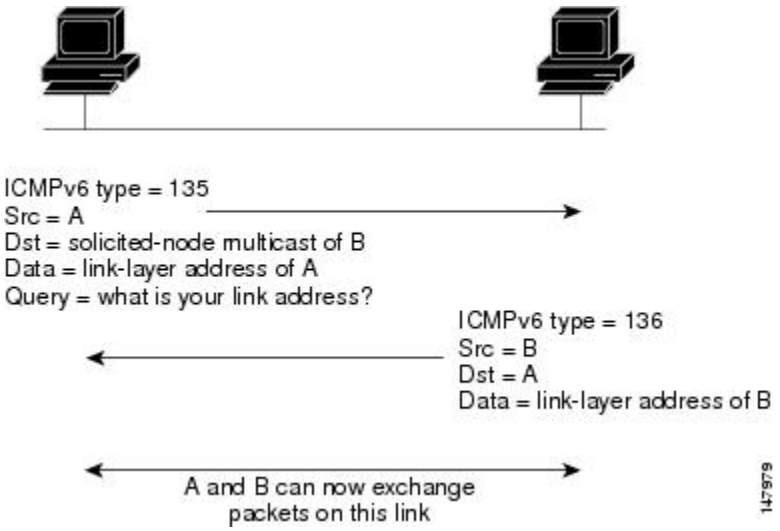
# IPv6 Neighbor Discovery

The IPv6 neighbor discovery process uses ICMP messages and solicited-node multicast addresses to determine the link-layer address of a neighbor on the same network (local link), verify the reachability of a neighbor, and keep track of neighboring routers.

## IPv6 Neighbor Solicitation and Advertisement Messages

A value of 135 in the Type field of the ICMP packet header identifies a neighbor solicitation message. Neighbor solicitation messages are sent on the local link when a node wants to determine the link-layer address of another node on the same local link. (See the following figure.) When a node wants to determine the link-layer address of another node, the source address in a neighbor solicitation message is the IPv6 address of the node sending the neighbor solicitation message. The destination address in the neighbor solicitation message is the solicited-node multicast address that corresponds to the IPv6 address of the destination node. The neighbor solicitation message also includes the link-layer address of the source node.

***Figure 59: IPv6 Neighbor Discovery—Neighbor Solicitation Message***




After receiving the neighbor solicitation message, the destination node replies by sending a neighbor advertisement message, which has a value of 136 in the Type field of the ICMP packet header, on the local link. The source address in the neighbor advertisement message is the IPv6 address of the node (more specifically, the IPv6 address of the node interface) sending the neighbor advertisement message. The destination address in the neighbor advertisement message is the IPv6 address of the node that sent the neighbor solicitation message. The data portion of the neighbor advertisement message includes the link-layer address of the node sending the neighbor advertisement message.

After the source node receives the neighbor advertisement, the source node and destination node can communicate.

Neighbor solicitation messages are also used to verify the reachability of a neighbor after the link-layer address of a neighbor is identified. When a node wants to verifying the reachability of a neighbor, the destination address in a neighbor solicitation message is the unicast address of the neighbor.

Neighbor advertisement messages are also sent when there is a change in the link-layer address of a node on a local link. When there is such a change, the destination address for the neighbor advertisement is the all-node multicast address.

Neighbor solicitation messages are also used to verify the reachability of a neighbor after the link-layer address of a neighbor is identified. Neighbor unreachability detection identifies the failure of a neighbor or the failure of the forward path to the neighbor, and is used for all paths between hosts and neighboring nodes (hosts or routers). Neighbor unreachability detection is performed for neighbors to which only unicast packets are being sent and is not performed for neighbors to which multicast packets are being sent.

A neighbor is considered reachable when the neighbor returns a positive acknowledgment indicating that it has received and processed packets previously sent to it. A positive acknowledgment could be from an upper-layer protocol such as TCP indicating that a connection is making forward progress (reaching its destination) or the receipt of a neighbor advertisement message in response to a neighbor solicitation message. If packets are reaching the peer, they are also reaching the next-hop neighbor of the source. Therefore, forward progress is also a confirmation that the next-hop neighbor is reachable.

For destinations that are not on the local link, forward progress implies that the first-hop router is reachable. When acknowledgments from an upper-layer protocol are not available, a node probes the neighbor using unicast neighbor solicitation messages to verify that the forward path is still working. The return of a solicited neighbor advertisement message from the neighbor is a positive acknowledgment that the forward path is still working (neighbor advertisement messages that have the solicited flag set to a value of 1 are sent only in response to a neighbor solicitation message). Unsolicited messages confirm only the one-way path from the source to the destination node; solicited neighbor advertisement messages indicate that a path is working in both directions.



**Note** A neighbor advertisement message that has the solicited flag set to a value of 0 must not be considered as a positive acknowledgment that the forward path is still working.

Neighbor solicitation messages are also used in the stateless autoconfiguration process to verify the uniqueness of unicast IPv6 addresses before the addresses are assigned to an interface. Duplicate address detection is performed first on a new, link-local IPv6 address before the address is assigned to an interface (the new address remains in a tentative state while duplicate address detection is performed). Specifically, a node sends a neighbor solicitation message with an unspecified source address and a tentative link-local address in the body of the message. If another node is already using that address, the node returns a neighbor advertisement message that contains the tentative link-local address. If another node is simultaneously verifying the uniqueness of the same address, that node also returns a neighbor solicitation message. If no neighbor advertisement messages are received in response to the neighbor solicitation message and no neighbor solicitation messages are received from other nodes that are attempting to verify the same tentative address, the node that sent the original neighbor solicitation message considers the tentative link-local address to be unique and assigns the address to the interface.

Every IPv6 unicast address (global or link-local) must be checked for uniqueness on the link; however, until the uniqueness of the link-local address is verified, duplicate address detection is not performed on any other IPv6 addresses associated with the link-local address.

# Router Discovery

Router discovery performs both router solicitation and router advertisement. Router solicitations are sent by hosts to all-routers multicast addresses. Router advertisements are sent by routers in response to solicitations or unsolicited and contain default router information as well as additional parameters such as the MTU and hop limit.

# IPv6 Stateless Autoconfiguration

All interfaces on IPv6 nodes must have a link-local address, which is automatically configured from the identifier for an interface and the link-local prefix FE80::/10. A link-local address enables a node to communicate with other nodes on the link and can be used to further configure the node.

Nodes can connect to a network and automatically generate site-local and global IPv6 address without the need for manual configuration or help of a server, such as a DHCP server. With IPv6, a router on the link advertises in router advertisement (RA) messages any site-local and global prefixes, and its willingness to function as a default router for the link. RA messages are sent periodically and in response to router solicitation messages, which are sent by hosts at system startup. (See the following figure.)

***Figure 60: IPv6 Stateless Autoconfiguration***

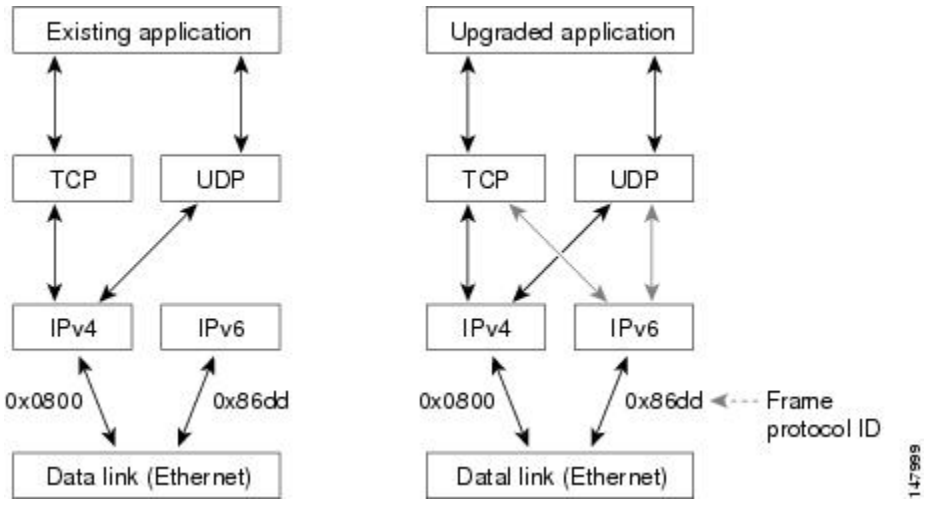



A node on the link can automatically configure site-local and global IPv6 addresses by appending its interface identifier (64 bits) to the prefixes (64 bits) included in the RA messages. The resulting 128-bit IPv6 addresses configured by the node are then subjected to duplicate address detection to ensure their uniqueness on the link. If the prefixes advertised in the RA messages are globally unique, then the IPv6 addresses configured by the node are also guaranteed to be globally unique. Router solicitation messages, which have a value of 133 in the Type field of the ICMP packet header, are sent by hosts at system startup so that the host can immediately autoconfigure without needing to wait for the next scheduled RA message.

# Dual IPv4 and IPv6 Protocol Stacks

The dual IPv4 and IPv6 protocol stack technique is one technique for a transition to IPv6. It enables gradual, one-by-one upgrades to applications running on nodes. Applications running on nodes are upgraded to make use of the IPv6 protocol stack. Applications that are not upgraded—they support only the IPv4 protocol stack—can coexist with upgraded applications on the same node. New and upgraded applications simply make use of both the IPv4 and IPv6 protocol stacks. See the below image.)

***Figure 61: Dual IPv4 and IPv6 Protocol Stack Technique***

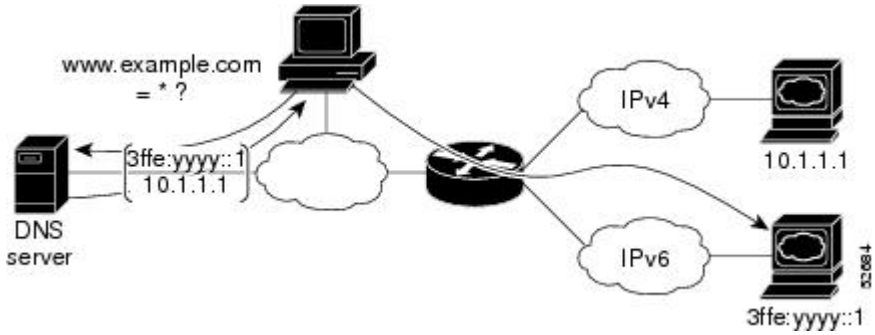



A new API has been defined to support both IPv4 and IPv6 addresses and DNS requests. An application can be upgraded to the new API and still use only the IPv4 protocol stack. The Cisco MDS NX-OS supports the

dual IPv4 and IPv6 protocol stack technique. When an interface is configured with both an IPv4 and an IPv6 address, the interface will accept and process both IPv4 and IPv6 traffic.

In the above figure, an application that supports dual IPv4 and IPv6 protocol stacks requests all available addresses for the destination host name www.a.com from a DNS server. The DNS server replies with all available addresses (both IPv4 and IPv6 addresses) for www.a.com. The application chooses an address—in most cases, IPv6 addresses are the default choice—and connects the source node to the destination using the IPv6 protocol stack.

***Figure 62: Dual IPv4 and IPv6 Protocol Stack Applications***




# Configuring Basic Connectivity for IPv6

The tasks in this section explain how to implement IPv6 basic connectivity. Each task in the list is identified as either required or optional. This section includes the following topics:

## Configuring IPv6 Addressing and Enabling IPv6 Routing

### About Configuring IPv6 Addressing and Enabling IPv6 Routing

IPv6 addresses are represented as a series of 16-bit hexadecimal fields separated by colons (:) in the format x:x:x:x:x:x:x:x. It is common for IPv6 addresses to contain successive hexadecimal fields of zeros. To make IPv6 addresses easier to use, two colons (::) may be used to compress successive hexadecimal fields of zeros at the beginning, middle, or end of an IPv6 address (the colons represent successive hexadecimal fields of zeros). By default, IPv6 addresses are not configured, and IPv6 processing is disabled. You can configure IPv6 addresses on the following interface types:

- Gigabit Ethernet
- Management
- VLAN (Gigabit Ethernet subinterface)
- VSAN

Configuring a global IPv6 address on an interface automatically configures a link-local address and activates IPv6 for that interface. Additionally, the configured interface automatically joins the following required multicast groups for that link:

- Solicited-node multicast group FF02:0:0:0:0:1:FF00::/104 for each unicast address assigned to the interface

- All-node link-local multicast group FF02::1

This task explains how to assign IPv6 addresses to individual router interfaces and enable the processing of IPv6 traffic. By default, IPv6 addresses are not configured and IPv6 processing is disabled.

You can configure IPv6 addresses on the following interface types:

- Gigabit Ethernet
- Management
- VLAN (Gigabit Ethernet subinterface)
- VSAN



**Note** The IPv6 address *ipv6-address* argument in the **ipv6 address** command must be in the form documented in RFC 2373, where the address is specified in hexadecimal using 16-bit values between colons

The IPv6 prefix *ipv6-prefix* argument in the **ipv6 address** command must be in the form documented in RFC 2373, where the address is specified in hexadecimal using 16-bit values between colons.

The IPv6 prefix length *prefix-length* argument in the **ipv6 address** command is a decimal value that indicates how many of the high-order contiguous bits of the address comprise the prefix (the network portion of the address). A slash mark must precede the decimal value.

Configuring a global IPv6 address on an interface automatically configures a link-local address and activates IPv6 for that interface. Additionally, the configured interface automatically joins the following required multicast groups for that link:

- Solicited-node multicast group FF02:0:0:0:0:1:FF00::/104 for each unicast address assigned to the interface
- All-node link-local multicast group FF02::1



**Note** The solicited-node multicast address is used in the neighbor discovery process.

The maximum number of IPv6 addresses (static and autoconfigured) allowed on an interface is eight, except on the management (mgmt 0) interface where only one static IPv6 address can be configured.

# Configuring IPv4 and IPv6 Protocol Addresses

When an interface in a Cisco networking device is configured with both an IPv4 and an IPv6 address, the interface can send and receive data on both IPv4 and IPv6 networks.

To configure an interface in a Cisco networking device to support both the IPv4 and IPv6 protocol stacks, follow these steps:

### Procedure

**Step 1** Enters configuration mode.

```
switch# config t
switch(config)#
```

**Step 2** Specify the interface, and enter interface configuration submode.

```
switch(config)# interface gigabitethernet 1/1
switch(config-if)#
```

**Step 3** Specify a primary or secondary IPv4 address for an interface.

switch(config-if)# **ip address 192.168.99.1 255.255.255.0**

**Step 4** Specify the IPv6 network assigned to the interface and enable IPv6 processing on the interface.

```
switch(config-if)# ipv6 address 2001:0DB8:c18:1::3/64
```

**Step 5** Enable the interface.

```
switch(config-if)# no shutdown
```

**Step 6** Exit interface configuration submode, and return to configuration mode.

```
switch(config-if)# exit
switch(config)
```

**Step 7** Enable the processing of IPv6 unicast datagrams.

```
switch(config)# ipv6 routing
```

# Verifying Basic IPv6 Connectivity Configuration and Operation

You can display information to verify the configuration and operation of basic IPv6 connectivity.This section provides the following **show ipv6** command output examples:

- Example Output for the show ipv6 interface Command
- Example Output for the show ipv6 neighbours Command
- Example Output for the show ipv6 traffic Command

### Example Output for the show ipv6 interface Command

In the following example, the **show ipv6 interface** command is used to verify that IPv6 addresses are configured correctly for the Gigabit Ethernet 6/1 interface:

```
switch# show ipv6 interface mgmt 0
mgmt0 is up
IPv6 is enabled
Global address(es):
2172:22::180/64
Link-local address(es):
fe80::b8db:adff:feba:d074
```

```
ND DAD is disabled
ND reachable time is 30000 milliseconds
ND retransmission time is 1000 milliseconds
Stateless autoconfig for addresses disabled
MTU is 1500 bytes
```

### Example Output for the show ipv6 neighbours Command

In the following example, the **show ipv6 neighbours** command displays IPv6 neighbor discovery cache information for all interfaces:

```
switch# show ipv6 neighbours
R - Reachable, I - Incomplete, S - Stale, F - Failed, P - Probe, D - Delay
IPv6 Address Age State Link-layer Addr Interface
fe80::211:5dff:fe53:500a 0 S 0011.5d53.500a GigE6/1
fe80::211:5dff:fe53:500a 0 S 0011.5d53.500a GigE6/2
5000:1::250 0 S 0011.5d53.500a po 4
fe80::211:5dff:fe53:500a 0 S 0011.5d53.500a po 4
fe80::211:5dff:fe53:500a 0 S 0011.5d53.500a po 4
fe80::2d0:3ff:fe61:4800 184 S 00d0.0361.4800 mgmt0
```

In the following example, the **show ipv6 neighbours interface** command displays IPv6 neighbor discovery cache information for the Gigabit Ethernet 6/1 interface:

```
switch# show ipv6 neighbours interface gigabitethernet 6/1
R - Reachable, I - Incomplete, S - Stale, F - Failed, P - Probe, D - Delay
IPv6 Address Age State Link-layer Addr Interface
fe80::211:5dff:fe53:500a 0 S 0011.5d53.500a GigE6/1
```

### Example Output for the show ipv6 traffic Command

The **show ipv6 traffic** command displays IPv6 and ICMP statistics:

```
switch# show ipv6 traffic
IPv6 Statistics:
Rcvd: 100 total, 0 local destination
0 errors, 0 truncated, 0 too big
0 unknown protocol, 0 dropped
0 fragments, 0 reassembled
0 couldn't reassemble, 0 reassembly timeouts
Sent: 0 generated, 0 forwarded 0 dropped
0 fragmented, 0 fragments created, 0 couldn't fragment

ICMPv6 Statistics:
Rcvd: 100 total, 0 errors, 0 unreachables, 0 time exceeded
0 too big, 0 param probs, 0 admin prohibs
0 echos, 0 echo reply, 0 redirects
0 group query, 0 group report, 0 group reduce
0 router solicit, 69 router advert
0 neighbor solicit, 31 neighbor advert
Sent: 55 total, 0 errors, 0 unreachables, 0 time exceeded
0 too big, 0 param probs, 0 admin prohibs
0 echos, 0 echo reply, 0 redirects
0 group query, 20 group report, 2 group reduce
0 router solicit, 0 router advert
0 neighbor solicit, 33 neighbor advert
```

### Clearing IPv6 Neighbor Discovery Cache

You can clear the IPv6 neighbor discovery cache using the **clear ipv6 neighbor** command in EXEC mode:

```
switch# clear ipv6 neighbor
```

# Verifying Neighbor Discovery Parameter Configuration

The **show ipv6 interface** command displays the configuration of the neighbor discovery parameters:

```
switch# show ipv6 interface mgmt 0
mgmt0 is up
IPv6 is enabled
Global address(es):
2003::1/64
Link-local address(es):
fe80::205:30ff:fe00:533e
ND DAD is enabled, number of DAD attempts: 5
ND reachable time is 50000 milliseconds
ND retransmission time is 3000 milliseconds
Stateless autoconfig for addresses disabled
```

# Duplicate Address Detection Attempts

To configure the number of duplicate address detection attempts, follow these steps:

**Procedure**

**Step 1** Enter configuration mode.

```
switch# config t
switch(config)#
```

**Step 2** Specify an interface and enter the interface configuration submode

```
switch(config)# interface gigabitethernet 3/1
switch(config-if)#
```

**Step 3** Set the duplicate address detection attempts count to 3. The range is 0 to 15.

```
switch(config-if)# ipv6 nd dad attempts 3
```

**Step 4** Revert to the default value (0).

```
switch(config-if)# no ipv6 nd dad attempts
```

# Reachability Time

To configure the reachability time, follow these steps:

**Procedure**

**Step 1** Enter configuration mode.

```
switch# config t
switch(config)#
```

**Step 2** Specify an interface and enter the interface configuration submode.

```
switch(config)# interface gigabitethernet 3/1
switch(config-if)#
```

**Step 3** Set the retransmission time to 10000 milliseconds. The range is 1000 to 3600000 millisecond.

```
switch(config-if)# ipv6 nd reachability-time 10000
```

**Step 4** Revert to the default value (30000 milliseconds).

```
switch(config-if)# no ipv6 nd reachability-time
```

# Retransmission Time

To configure the retransmission time, follow these steps:

### Procedure

**Step 1** Enter configuration mode.

```
switch# config t
switch(config)#
```

**Step 2** Specify an interface and enter the interface configuration submode.

```
switch(config)# interface gigabitethernet 3/1
switch(config-if)#
```

**Step 3** Set the retransmission time to 20000 milliseconds. The range is 1000 to 3600000 millisecond.

```
switch(config-if)# ipv6 nd retransmission-timer 20000
```

**Step 4** Revert to the default value (1000 milliseconds).

```
switch(config-if)# no ipv6 nd retransmission-timer
```

# Verifying Neighbor Discovery Parameter Configuration

The **show ipv6 interface** command displays the configuration of the neighbor discovery parameters:

```
switch# show ipv6 interface mgmt 0
mgmt0 is up
IPv6 is enabled
Global address(es):
2003::1/64
Link-local address(es):
fe80::205:30ff:fe00:533e
ND DAD is enabled, number of DAD attempts: 5
```

```
ND reachable time is 50000 milliseconds
ND retransmission time is 3000 milliseconds
Stateless autoconfig for addresses disabled
```

# IPv6 Static Routes

Cisco MDS NX-OS supports static routes for IPv6. This section includes the following topics:

- Configuring an IPv6 Static Route
- Verifying IPv6 Static Route Configuration and Operation

## Configuring an IPv6 Static Route

To configure a IPv6 static route, follow these steps:

### Before you begin

You must manually configure IPv6 static routes and define an explicit path between two networking devices. IPv6 static routes are not automatically updated and must be reconfigured manually if the network topology changes.

You must manually configure IPv6 static routes and define an explicit path between two networking devices. IPv6 static routes are not automatically updated and must be manually reconfigured if the network topology changes.

### Procedure

**Step 1** Enter configuration mode.

```
switch# config t
switch(config)#
```

**Step 2** Configure a static default IPv6 route on a Gigabit Ethernet interface.

```
switch(config)# ipv6 route ::/0 gigabitethernet 3/1
```

**Step 3** Configure a fully specified IPv6 static route on a Gigabit Ethernet interface.

switch(config)# **ipv6 route 2001:0DB8::/32 gigabitethernet 3/2**

## Verifying IPv6 Static Route Configuration and Operation

The **show ipv6 route** command displays the IPv6 route table for the switch:

```
switch# show ipv6 route
IPv6 Routing Table
Codes: C - Connected, L - Local, S - Static G - Gateway
G ::/0
via fe80::211:5dff:fe53:500a, GigabitEthernet6/1, distance 2
```

```
G ::/0
via fe80::2d0:3ff:fe61:4800, mgmt0, distance 2
C 2000::/64
via ::, mgmt0
C 2172:22::/64
via ::, mgmt0, distance 2
C 3000:3::/64
via fe80::205:30ff:fe01:7ed6, GigabitEthernet4/1
C 3000:4::/64
via fe80::205:30ff:fe01:7ed6, GigabitEthernet4/1.250
C 3000:5::/64
via fe80::213:1aff:fee5:e69b, GigabitEthernet5/4
C 3000:6::/64
via fe80::213:1aff:fee5:e69b, GigabitEthernet5/4.250
C 3000:7::/64
via fe80::205:30ff:fe01:7ed7, GigabitEthernet4/2
C 3000:8::/64
via fe80::205:30ff:fe01:7ed7, GigabitEthernet4/2.250
C 3000:9::/64
via fe80::213:1aff:fee5:e69e, port-channel 3
C 3000:10::/64
via fe80::213:1aff:fee5:e69e, port-channel 3.250
C 5000:1::/64
via fe80::205:30ff:fe01:3917, GigabitEthernet6/2
C 5000:1::/64
via fe80::205:30ff:fe01:3918, port-channel 4
C 6000:1:1:1::/64
via fe80::205:30ff:fe01:3916, GigabitEthernet6/1
C 7000:1::/64
via fe80::205:30ff:fe01:3917, GigabitEthernet6/2.250
C 7000:1::/64
via fe80::205:30ff:fe01:3918, port-channel 4.250
C 7000:1:1:1::/64
via fe80::205:30ff:fe01:3917, GigabitEthernet6/2, distance 2
L fe80::/10
via ::
L ff00::/8
via ::
```

# Gigabit Ethernet IPv6-ACL Guidelines

Follow these guidelines when configuring IPv6-ACLs for Gigabit Ethernet interfaces:

- Only use Transmission Control Protocol (TCP) or Internet Control Message Protocol (ICMP).



**Note** Other protocols such as User Datagram Protocol (UDP) and HTTP are not supported in Gigabit Ethernet interfaces. Applying an ACL that contains rules for these protocols to a Gigabit Ethernet interface is allowed but those rules have no effect.

- Apply IPv6-ACLs to the interface before you enable an interface. This ensures that the filters are in place before traffic starts flowing.
- Be aware of the following conditions:
  - If you use the **log-deny** option, a maximum of 50 messages are logged per second.

- The **established** option is ignored when you apply IPv6-ACLs containing this option to Gigabit Ethernet interfaces.
- If an IPv6-ACL rule applies to a preexisting TCP connection, that rule is ignored. For example, if there is an existing TCP connection between A and B and an IPv6-ACL that specifies dropping all packets whose source is A and destination is B is subsequently applied, it will have no effect.



**Note** If IPv6-ACLs are already configured in a Gigabit Ethernet interface, you cannot add this interface to a Ethernet PortChannel group. See the Cisco Fabric Manager Security Configuration GuideCisco MDS 9000 Family NX-OS Security Configuration Guide for information on configuring IPv6-ACLs.

See the Cisco Fabric Manager Security Configuration GuideCisco MDS 9000 Family NX-OS Security Configuration Guide for information on applying IPv6-ACLs to an interface.

# Transitioning from IPv4 to IPv6

Cisco MDS NX-OS does not support any transitioning mechanisms from IPv4 to IPv6. However, you can use the transitioning schemes in the Cisco router products for this purpose. For information on configuring Cisco routers to transition your network, refer to the Implementing Tunneling for IPv6 chapter.

# Displaying IPv6

Use the **show ips ipv6 neighbours interface** command for information about IPv6 neighbors for an interface:

```
switch# show ips ipv6 neighbours interface gigabitethernet 6/1
IPv6 Address Age (min) Link-layer Addr State Interface
fe80::211:5dff:fe53:500a 0 0011.5d53.500a S Gigabi tEthernet6/1
```

Use the **show ips ipv6 prefix-list interface** command for information about IPv6 prefixes for an interface:

```
switch# show ips ipv6 prefix-list interface gigabitethernet 6/1
Prefix Prefix-len Addr
Valid Preferred
6000:1:1:1:: 64 ::
2592000 604800
```

Use the **show ips ipv6 interface** command for information about the IPv6 routes for an interface:

```
switch# show ips ipv6 route interface gigabitethernet 6/1
IPv6 Routing Table - 4 entries
Codes: C - Connected, L - Local, S - Static, G - Gateway, M - Multicast
C 6000:1:1:1::/64 is directly connected, GigabitEthernet6/1
C 6000:1:1:1::/64 is directly connected, GigabitEthernet6/1
C fe80::/64 is directly connected, GigabitEthernet6/1
M ff02::/32 is multicast, GigabitEthernet6/1
G ::/0 via fe80::211:5dff:fe53:500a, GigabitEthernet6/1
```

Use the **show ips ipv6 routers interface** command for information about IPv6 routers for an interface:

```
switch# show ips ipv6 routers interface gigabitethernet 6/1
Addr Lifetime Expire
fe80::211:5dff:fe53:500a 1800 1781
```

Use the **show ips ipv6 traffic interface** command for information about IPv6 traffic statistics for an interface:

```
switch# show ips ipv6 traffic interface gigabitethernet 6/1
IPv6 statistics:
Rcvd: 5094 total
0 bad header, 0 unknown option, 0 unknown protocol
0 fragments, 0 total reassembled
0 reassembly timeouts, 0 reassembly failures
Sent: 13625 generated
0 fragmented into 0 fragments, 0 failed
2 no route
ICMP statistics:
Rcvd: 1264 input, 0 checksum errors, 0 too short
0 unknown info type, 0 unknown error type
unreach: 0 routing, 0 admin, 0 neighbor, 0 address, 0 port
parameter: 0 error, 0 header, 0 option
0 hopcount expired, 0 reassembly timeout,0 too big
0 echo request, 0 echo reply
734 group query, 0 group report, 0 group reduce
0 router solicit, 528 router advert, 0 redirects
0 neighbor solicit, 2 neighbor advert
Sent: 6045 output, 0 rate-limited
unreach: 0 routing, 0 admin, 0 neighbor, 1160 address, 0 port
parameter: 0 error, 0 header, 0 option
0 hopcount expired, 0 reassembly timeout,0 too big
0 echo request, 0 echo reply
0 group query, 1466 group report, 0 group reduce
1 router solicit, 0 router advert, 0 redirects
3412 neighbor solicit, 6 neighbor advert
```

# Default Settings

The following table lists the default settings for IPv6 parameters.

***Table 19: Default IPv6 Parameters***

| Parameters | Default |
|---|---|
| IPv6 processing | Disabled |
| Duplicate address detection attempts | 0 (neighbor discovery disabled) |
| Reachability time | 1000 milliseconds |
| Retransmission time | 30000 milliseconds |
| IPv6-ACLs | None |